Bios 6301: Assignment 5

Yudong Cao

Grade 51/50

Note: Great job. Also, it's worth learning how Cole solved Question 2 using laaply and tapply. In the future, for packages that might not be installed by collaborators, you can use the following to check for and install a package:

```
if("lubridate" %in% rownames(installed.packages()) == FALSE) {
  install.packages("lubridate",repos="http://cran.rstudio.com/")
}
```

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)

```
haart<-read.csv('haart.csv')
```

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

```
haart[,'last.visit'] <- as.POSIXct(haart[,'last.visit'], format="%m/%d/%y")
haart[,'init.date'] <- as.POSIXct(haart[,'init.date'], format="%m/%d/%y")
haart[,'date.death'] <- as.POSIXct(haart[,'date.death'], format="%m/%d/%y")
init.date.year<-sub("([0-9]{4})-([0-9]{2})-([0-9]{2})","\\1",haart[,'init.date'])
table(init.date.year)
```

```
## init.date.year
## 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?

```
time.diff<-haart[,'date.death']-haart[,'init.date']
haart[,'death.within.1year']<-ifelse(time.diff<=365,1,0)
table(haart[,'death.within.1year'])
```

```
##
##  0  1
## 25 92
```

92 observations died in year 1.

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.

```
fut.visit<-difftime(haart[,'last.visit'], haart[,'init.date'], units="days")
fut.visit<-ifelse(fut.visit>365,365,fut.visit)
fut.death<-difftime(haart[,'date.death'], haart[,'init.date'], units="days")
fut.death<-ifelse(fut.death>365,365,fut.death)
attach(haart)
followup.time<-0
for (i in 1:nrow(haart)) {
  if (is.na(fut.death[i])) {
    followup.time[i]=fut.visit[i]
  } else {
    followup.time[i]=min(fut.visit[i],fut.death[i])
  }
}
haart[,'fut']<-ceiling(followup.time)
quantile(followup.time,na.rm=TRUE)
```

```
##   0%  25%  50%  75% 100%
##    0  338  365  365  365
```

JC Grading -2 Close, but this code: followup.time[i]=min(fut.visit[i],fut.death[i]) should be followup.time[i]=min(fut.visit[i],fut.death[i], na.rm=TRUE). Look at these cross-tabs:

```
table(is.na(fut.death),is.na(fut.visit))
```

```
##
##         FALSE TRUE
##   FALSE   106   11
##   TRUE    883    0
```

```
table(is.na(followup.time))
```

```
##
## FALSE  TRUE
##   989    11
```

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?

```
haart[,'loss.tfu']<-ifelse(haart[,'fut']==365 & haart[,'death']==0,1,0)
table(haart[,'loss.tfu'])
```

```
##
##   0   1
## 290 710
```

710 records are lost-to-followup.

JC Grading -2

710 overcounts the answer of 173. Since follow-up time is truncated at 365, it includes many who had visits beyond the first year. You could check if anyone had a 365 visit and if not consider ltf those whose visit was less than 365.

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?

```
haart[,'init.reg']<-as.character(haart[,'init.reg'])
all.reg<-strsplit(haart[,'init.reg'],",")
all.reg<-unlist(all.reg)
table(all.reg)
```

```
## all.reg
## 3TC ABC ATV AZT D4T DDC DDI EFV FPV FTC IDV LPV NFV NVP RTV SQV T20 TDF
## 973  56   2 794 146   1  38 516   2   8  27  31   8 358  79  29   1  10
```

3TC, AZT, D4T, EFV and NVP are found over 100 times.

```
all.reg<-unique(all.reg)
row.reg<-strsplit(haart[,'init.reg'], ",")
user.reg<-sapply(all.reg,function(j) sapply(row.reg,function(i) j %in% i))
haart<-cbind(haart,+user.reg)
head(haart)
```

```
##   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
## 6    0  34    0         157    NA 54.8856         NA 3TC,AZT,NVP
##    init.date last.visit death date.death death.within.1year fut loss.tfu
## 1 2003-07-01 2007-02-26     0       <NA>                NA 365        1
## 2 2004-11-23 2008-02-22     0       <NA>                NA 365        1
## 3 2003-04-30 2005-11-21     1 2006-01-11                 0 365        0
## 4 2006-03-25 2006-05-05     1 2006-05-07                 1  41        0
## 5 2004-09-01 2007-11-13     0       <NA>                NA 365        1
## 6 2003-12-02 2008-02-28     0       <NA>                NA 365        1
##   3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   1   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   1   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   1   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

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.

```
haart2<-read.csv('haart2.csv')
haart2[,'last.visit'] <- as.POSIXct(haart2[,'last.visit'], format="%m/%d/%y")
haart2[,'init.date'] <- as.POSIXct(haart2[,'init.date'], format="%m/%d/%y")
haart2[,'date.death'] <- as.POSIXct(haart2[,'date.death'], format="%m/%d/%y")
time.diff2<-haart2[,'date.death']-haart2[,'init.date']
haart2[,'death.within.1year']<-ifelse(time.diff2<=365,1,0)
fut.visit2<-difftime(haart2[,'last.visit'], haart2[,'init.date'], units="days")
fut.visit2<-ifelse(fut.visit2>365,365,fut.visit2)
attach(haart2)
```

```
## The following objects are masked from haart:
##
##     age, aids, cd4baseline, date.death, death, death.within.1year,
##     hemoglobin, init.date, init.reg, last.visit, logvl, male,
##     weight
```

```
followup.time2<-0
for (i in 1:nrow(haart2)) {
    followup.time2[i]=fut.visit2[i]
}
haart2[,'fut']<-ceiling(followup.time2)
haart2[,'loss.tfu']<-ifelse(haart2[,'fut']==365 & haart2[,'death']==0,1,0)
haart2[,'init.reg']<-as.character(haart2[,'init.reg'])
row.reg2<-strsplit(haart2[,'init.reg'], ",")
user.reg2<-sapply(all.reg,function(j) sapply(row.reg2,function(i) j %in% i))
haart2<-cbind(haart2,+user.reg2)
haart<-rbind(haart,haart2)
head(haart,n=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 death.within.1year fut loss.tfu
## 1 2003-07-01 2007-02-26     0       <NA>                NA 365        1
## 2 2004-11-23 2008-02-22     0       <NA>                NA 365        1
## 3 2003-04-30 2005-11-21     1 2006-01-11                 0 365        0
## 4 2006-03-25 2006-05-05     1 2006-05-07                 1  41        0
## 5 2004-09-01 2007-11-13     0       <NA>                NA 365        1
##   3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC NFV T20 ATV FPV
## 1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   1   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   1   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
tail(haart,n=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 death.within.1year
## 1000 3TC,D4T,NVP 2003-07-03 2008-02-29     0       <NA>                NA
## 1001 3TC,AZT,NVP 2003-12-01 2004-01-05     0       <NA>                NA
## 1002 3TC,AZT,NVP 2002-09-26 2004-03-29     0       <NA>                NA
## 1003 3TC,DDI,EFV 2007-01-31 2007-04-16     0       <NA>                NA
## 1004 3TC,D4T,NVP 2003-12-03 2007-10-11     0       <NA>                NA
##      fut loss.tfu 3TC AZT EFV NVP D4T ABC DDI IDV LPV RTV SQV FTC TDF DDC
## 1000 365        1   1   0   0   1   1   0   0   0   0   0   0   0   0   0
## 1001  35        0   1   1   0   1   0   0   0   0   0   0   0   0   0   0
## 1002 365        1   1   1   0   1   0   0   0   0   0   0   0   0   0   0
```

```
## 1003  75        0   1   0   1   0   0   0   1   0   0   0   0   0   0   0
## 1004 365        1   1   0   0   1   1   0   0   0   0   0   0   0   0   0
##      NFV T20 ATV FPV
## 1000   0   0   0   0
## 1001   0   0   0   0
## 1002   0   0   0   0
## 1003   0   0   0   0
## 1004   0   0   0   0
```

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']),]
```

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
```

```
d<-0
for (i in 1:499) {
  if (x[,'id'][i]==x[,'id'][i+1]) {
    d[i]<-floor(difftime(x[,'dt'][i+1],x[,'dt'][i])/365)
```

```
  }
}
a<-data.frame(x$id[which(d==1)],x$dt[which(d==1)]+years(1),NA)
names(a)<-c('id','dt','a1c')
b<-data.frame(x$id[which(d==2)],x$dt[which(d==2)]+years(1),NA)
names(b)<-c('id','dt','a1c')
c<-data.frame(x$id[which(d==2)],x$dt[which(d==2)]+years(2),NA)
names(c)<-c('id','dt','a1c')
x<-rbind(x,a,b,c)
x<-x[order(x$id,x$dt),]
```

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.

```
visit<-1
for (i in 1:555) {
  if (x$id[i]==x$id[i+1]) {
    visit[i+1]<-visit[i]+1
  } else {
    visit[i+1]<-1
  }
}
x[,'visit']<-visit
```

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`.

```
attach(x)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     visit
```

```
m<-matrix(0,nrow=2,ncol=50)
for (i in 1:50) {
  m[1,i]<-unique(id)[i]
  m[2,i]<-mean(x[,'a1c'][id==i])
}
row.names(m)=c('id','mean a1c')
print(m)
```

```
##              [,1]     [,2]    [,3]     [,4]     [,5]     [,6]     [,7]
## id       1.000000 2.000000 3.00000 4.000000 5.000000 6.000000 7.000000
## mean a1c 4.063372 7.544643 6.75764 3.892127 9.512311 7.555965 9.161686
##              [,8]     [,9]     [,10]     [,11]     [,12]     [,13]
## id       8.000000 9.000000 10.000000 11.000000 12.000000 13.000000
## mean a1c 7.189064 9.283873  7.975217  6.917562  7.034021  9.145282
##              [,14]     [,15]     [,16]     [,17]     [,18]    [,19]
## id       14.000000 15.000000 16.000000 17.000000 18.000000 19.00000
```

```
## mean a1c  6.623756  8.012406  4.222158  3.996034  9.164873  5.50721
##              [,20]     [,21]     [,22]     [,23]     [,24]     [,25]
## id       20.000000 21.000000 22.000000 23.000000 24.000000 25.000000
## mean a1c  3.726675  8.140939  5.637501  7.366889  7.439316  6.877135
##              [,26]     [,27]     [,28]     [,29]     [,30]     [,31]
## id       26.000000 27.000000 28.000000 29.000000 30.000000 31.000000
## mean a1c  6.556759  4.926457  7.433917  4.508086  6.045577  7.116586
##              [,32]     [,33]     [,34]   [,35]    [,36]     [,37]
## id       32.000000 33.000000 34.000000 35.0000 36.00000 37.000000
## mean a1c  6.568791  6.494069  6.768615  8.4767  9.60441  9.606253
##              [,38]     [,39]     [,40]     [,41]    [,42]     [,43]
## id       38.000000 39.000000 40.000000 41.000000 42.00000 43.000000
## mean a1c  5.355979  6.917013  9.530136  9.802424  3.89177  6.095849
##              [,44]     [,45]     [,46]     [,47]   [,48]     [,49]
## id       44.00000 45.000000 46.000000 47.000000 48.0000 49.000000
## mean a1c  9.09167  6.737204  9.621763  9.231489  6.4046  6.096076
##              [,50]
## id       50.000000
## mean a1c  8.962319
```

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

```
attach(x)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     visit
```

```
## The following objects are masked from x (pos = 3):
##
##     a1c, dt, id, visit
```

```
v<-matrix(0,nrow=2,ncol=50)
for (i in 1:50) {
  v[1,i]<-unique(id)[i]
  v[2,i]<-max(visit[id==i])
}
row.names(v)=c('id','max visit')
print(v)
```

```
##           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## id           1    2    3    4    5    6    7    8    9    10    11    12
## max visit   11   20   14   12   14   10    9   12   11    12    10    10
##           [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## id           13    14    15    16    17    18    19    20    21    22
## max visit     8    12     8     9    12    10    10     9    10     8
##           [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32]
## id           23    24    25    26    27    28    29    30    31    32
## max visit     8    15    12    14    11    14    10     7    11     5
##           [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40] [,41] [,42]
## id           33    34    35    36    37    38    39    40    41    42
## max visit     8    12    11     9    17    15     8     7    17    14
##           [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
## id           43    44    45    46    47    48    49    50
## max visit    11    11    14     9    12    11    12    10
```

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

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

```
##      id                  dt      a1c visit
## 11   15 2000-04-30 00:34:50 7.527105     1
## 406  15 2001-01-17 21:11:02 5.898371     2
## 306  15 2001-04-25 06:23:05 8.566593     3
## 1101 15 2002-04-25 06:23:05 8.012406     4
## 1102 15 2003-04-25 06:23:05 8.012406     5
## 484  15 2003-06-06 14:06:00 9.133769     6
## 1810 15 2004-06-06 14:06:00 8.012406     7
## 263  15 2004-08-20 17:47:11 8.936190     8
```

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.

```
addr<-read.table('addr.txt',header=F,sep='\t',colClasses=c('character'))
u<-unlist(strsplit(addr[,1],split="  "))
u<-gsub("^\\s+|\\s+$","",u)
u<-u[u!=""]
x<-matrix(u,ncol=6,byrow=T)
y<-data.frame(streetno<-sub("^(\\w+)\\s?(.*)$","\\1",x[,3]),streetname<-sub("^(\\w+)\\s?(.*)$","\\2",x[
z<-as.data.frame(cbind(y,x)[-5])
colnames(z)<-(c( 'streetno', 'streetname','lastname', 'firstname', 'city', 'state', 'zip'))
z<-z[,c(3,4,1,2,5,6,7)]
print(z)
```

```
##       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.

```
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:

```
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.

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

```
## <simpleError in model.frame.default(formula = form, data = dat, drop.unused.levels = TRUE): variable
```

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

The error is that death itself is not a character object variable that needs to be in the as.formula function. To fix it, we need to paste it with "~ .".

5 bonus points

Create a working function.

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

This new function with paste, deparse and substitute to process the formula produces exactly the same results as before.

JC Grading +5