

# Bios 6301: Homework 7

Ashley Spirrison

45/40 GREAT!

*Due Thursday, 02 November, 1:00 PM*

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

40 points total.

Submit a single knitr file (named `homework7.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 `homework7.rmd` or include author name may result in 5 points taken off.

## Question 1

### 21 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)
```

*#2. For each `id`, determining if there is more than a one year gap in between observations.  
#Adding a new row at the one year mark (two for two years, etc.), with the `a1c`  
#value set to missing.*

```
addMissingRows <- function(data) {
  result <- vector("list", length = nrow(data))
  k <- 1
```

```

data <- data[order(data$id, data$dt), ] #1. Ordering the data set by `id` and `dt`.

for (i in unique(data$id)) {
  temp_data <- data[data$id == i, ]
  temp_data$dt <- as.POSIXct(temp_data$dt)
  temp_data <- temp_data[order(temp_data$dt), ]

  for (j in 1:(nrow(temp_data) - 1)) {
    time_diff <- as.numeric(difftime(temp_data$dt[j + 1], temp_data$dt[j], units = "days"))
    if (!is.na(time_diff) && time_diff >= 365) {
      num_missing_years <- floor(time_diff / 365)
      for (m in 1:(num_missing_years - 1)) {
        if (m == 1) {
          new_dt <- as.POSIXct(temp_data$dt[j] + as.difftime(365 * m, units = "days"))
          new_row <- data.frame(id = i, dt = new_dt, a1c = NA)
          temp_data <- rbind(temp_data[1:j, ], new_row, temp_data[(j + 1):nrow(temp_data), ])
          j <- j + 1
        }
      }
    }
  }
  result[[k]] <- temp_data
  k <- k + 1
}

result <- do.call(rbind, result)
return(result)
}

x <- addMissingRows(x)

#3. Creating a new column `visit`. For each `id`, adding the visit number from
#1 to n depending on the number of observations for each individual. Observations
#created with missing a1c values are included.
x$visit <- ave(x$id, x$id, FUN = seq_along)

#4. For each `id`, determining and replacing missing values with the
#mean `a1c` value for each individual.
mean_a1c <- aggregate(a1c ~ id, data = x, FUN = function(x) mean(x, na.rm = TRUE))

for (i in 1:nrow(mean_a1c)) {
  id_val <- mean_a1c[i, "id"]
  mean_val <- mean_a1c[i, "a1c"]
  x$a1c[x$id == id_val & is.na(x$a1c)] <- mean_val
}

#5. Printing the mean a1c for each id.
print(mean_a1c)

```

```

##      id      a1c
## 1     1 6.654444
## 2     2 9.789132
## 3     3 6.951820

```

```
## 4 4 8.191985
## 5 5 9.429694
## 6 6 7.133443
## 7 7 7.879138
## 8 8 6.244061
## 9 9 4.420523
## 10 10 6.028370
## 11 11 4.838279
## 12 12 6.691181
## 13 13 8.504632
## 14 14 9.122968
## 15 15 6.737092
## 16 16 7.420245
## 17 17 6.546329
## 18 18 6.151311
## 19 19 8.628037
## 20 20 8.923518
## 21 21 5.444430
## 22 22 5.763931
## 23 23 6.351112
## 24 24 9.377525
## 25 25 5.058097
## 26 26 8.692078
## 27 27 7.371831
## 28 28 4.243469
## 29 29 6.345254
## 30 30 4.135795
## 31 31 8.670622
## 32 32 5.130167
## 33 33 6.528153
## 34 34 8.445030
## 35 35 3.832195
## 36 36 9.514603
## 37 37 8.612608
## 38 38 10.160773
## 39 39 8.976697
## 40 40 7.583232
## 41 41 3.804325
## 42 42 6.787170
## 43 43 5.654235
## 44 44 5.613283
## 45 45 8.876623
## 46 46 7.485824
## 47 47 4.752133
## 48 48 7.415459
## 49 49 5.562809
## 50 50 4.970288
```

```
#6.Calculating and printing the total number of visits for each id.
total_visits <- aggregate(visit ~ id, data = x, FUN = length)

print(total_visits)
```

```
## id visit
```

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

```
#7.Printing the observations for id = 15.
print(x[x$id == 15, ])
```

```
##      id          dt      a1c visit
## 300 15 2000-10-21 01:08:17 7.401322    1
## 127 15 2001-08-08 14:23:08 5.896318    2
## 165 15 2001-08-15 07:03:29 7.457722    3
## 109 15 2002-03-15 21:23:10 5.330917    4
## 319 15 2002-04-14 09:08:25 6.484003    5
## 255 15 2002-10-10 18:27:43 8.139101    6
## 224 15 2003-02-19 12:58:53 6.446557    7
## 481 15 2003-03-02 06:58:10 7.432291    8
## 425 15 2003-06-30 07:20:49 7.113792    9
## 259 15 2004-01-22 20:30:42 5.668897   10
```

## Question 2

### 16 points

Install the `lexicon` package. Load the `sw_fry_1000` vector, which contains 1,000 common words.

```
options(repos = c(CRAN = "http://cran.rstudio.com/"))
```

```
install.packages("lexicon")
```

```
## Installing package into 'C:/Users/ashle/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'lexicon' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\ashle\AppData\Local\Temp\RtmpAt0cTv\downloaded_packages
```

```
library(lexicon)
data('sw_fry_1000', package = 'lexicon')
head(sw_fry_1000)
```

```
## [1] "the" "of" "to" "and" "a" "in"
```

```
#1. Removing all non-alphabetical characters and making all characters lowercase.
#Saving the result as a.
```

```
a <- tolower(gsub("[^[:alpha:]]", "", sw_fry_1000))
save(a, file = "a.RData")
print(a)
```

```
##      [1] "the"      "of"      "to"      "and"      "a"
##      [6] "in"       "is"      "it"      "you"      "that"
##     [11] "he"       "was"     "for"     "on"       "are"
##     [16] "with"     "as"      "i"       "his"      "they"
##     [21] "be"       "at"      "one"     "have"     "this"
##     [26] "from"     "or"      "had"     "by"       "hot"
##     [31] "word"     "but"     "what"    "some"     "we"
```

##	[36]	"can"	"out"	"other"	"were"	"all"
##	[41]	"there"	"when"	"up"	"use"	"your"
##	[46]	"how"	"said"	"an"	"each"	"she"
##	[51]	"which"	"do"	"their"	"time"	"if"
##	[56]	"will"	"way"	"about"	"many"	"then"
##	[61]	"them"	"write"	"would"	"like"	"so"
##	[66]	"these"	"her"	"long"	"make"	"thing"
##	[71]	"see"	"him"	"two"	"has"	"look"
##	[76]	"more"	"day"	"could"	"go"	"come"
##	[81]	"did"	"number"	"sound"	"no"	"most"
##	[86]	"people"	"my"	"over"	"know"	"water"
##	[91]	"than"	"call"	"first"	"who"	"may"
##	[96]	"down"	"side"	"been"	"now"	"find"
##	[101]	"any"	"new"	"work"	"part"	"take"
##	[106]	"get"	"place"	"made"	"live"	"where"
##	[111]	"after"	"back"	"little"	"only"	"round"
##	[116]	"man"	"year"	"came"	"show"	"every"
##	[121]	"good"	"me"	"give"	"our"	"under"
##	[126]	"name"	"very"	"through"	"just"	"form"
##	[131]	"sentence"	"great"	"think"	"say"	"help"
##	[136]	"low"	"line"	"differ"	"turn"	"cause"
##	[141]	"much"	"mean"	"before"	"move"	"right"
##	[146]	"boy"	"old"	"too"	"same"	"tell"
##	[151]	"does"	"set"	"three"	"want"	"air"
##	[156]	"well"	"also"	"play"	"small"	"end"
##	[161]	"put"	"home"	"read"	"hand"	"port"
##	[166]	"large"	"spell"	"add"	"even"	"land"
##	[171]	"here"	"must"	"big"	"high"	"such"
##	[176]	"follow"	"act"	"why"	"ask"	"men"
##	[181]	"change"	"went"	"light"	"kind"	"off"
##	[186]	"need"	"house"	"picture"	"try"	"us"
##	[191]	"again"	"animal"	"point"	"mother"	"world"
##	[196]	"near"	"build"	"self"	"earth"	"father"
##	[201]	"head"	"stand"	"own"	"page"	"should"
##	[206]	"country"	"found"	"answer"	"school"	"grow"
##	[211]	"study"	"still"	"learn"	"plant"	"cover"
##	[216]	"food"	"sun"	"four"	"between"	"state"
##	[221]	"keep"	"eye"	"never"	"last"	"let"
##	[226]	"thought"	"city"	"tree"	"cross"	"farm"
##	[231]	"hard"	"start"	"might"	"story"	"saw"
##	[236]	"far"	"sea"	"draw"	"left"	"late"
##	[241]	"run"	"dont"	"while"	"press"	"close"
##	[246]	"night"	"real"	"life"	"few"	"north"
##	[251]	"open"	"seem"	"together"	"next"	"white"
##	[256]	"children"	"begin"	"got"	"walk"	"example"
##	[261]	"ease"	"paper"	"group"	"always"	"music"
##	[266]	"those"	"both"	"mark"	"often"	"letter"
##	[271]	"until"	"mile"	"river"	"car"	"feet"
##	[276]	"care"	"second"	"book"	"carry"	"took"
##	[281]	"science"	"eat"	"room"	"friend"	"began"
##	[286]	"idea"	"fish"	"mountain"	"stop"	"once"
##	[291]	"base"	"hear"	"horse"	"cut"	"sure"
##	[296]	"watch"	"color"	"face"	"wood"	"main"
##	[301]	"enough"	"plain"	"girl"	"usual"	"young"

##	[306]	"ready"	"above"	"ever"	"red"	"list"
##	[311]	"though"	"feel"	"talk"	"bird"	"soon"
##	[316]	"body"	"dog"	"family"	"direct"	"pose"
##	[321]	"leave"	"song"	"measure"	"door"	"product"
##	[326]	"black"	"short"	"numeral"	"class"	"wind"
##	[331]	"question"	"happen"	"complete"	"ship"	"area"
##	[336]	"half"	"rock"	"order"	"fire"	"south"
##	[341]	"problem"	"piece"	"told"	"knew"	"pass"
##	[346]	"since"	"top"	"whole"	"king"	"space"
##	[351]	"heard"	"best"	"hour"	"better"	"true"
##	[356]	"during"	"hundred"	"five"	"remember"	"step"
##	[361]	"early"	"hold"	"west"	"ground"	"interest"
##	[366]	"reach"	"fast"	"verb"	"sing"	"listen"
##	[371]	"six"	"table"	"travel"	"less"	"morning"
##	[376]	"ten"	"simple"	"several"	"vowel"	"toward"
##	[381]	"war"	"lay"	"against"	"pattern"	"slow"
##	[386]	"center"	"love"	"person"	"money"	"serve"
##	[391]	"appear"	"road"	"map"	"rain"	"rule"
##	[396]	"govern"	"pull"	"cold"	"notice"	"voice"
##	[401]	"unit"	"power"	"town"	"fine"	"certain"
##	[406]	"fly"	"fall"	"lead"	"cry"	"dark"
##	[411]	"machine"	"note"	"wait"	"plan"	"figure"
##	[416]	"star"	"box"	"noun"	"field"	"rest"
##	[421]	"correct"	"able"	"pound"	"done"	"beauty"
##	[426]	"drive"	"stood"	"contain"	"front"	"teach"
##	[431]	"week"	"final"	"gave"	"green"	"oh"
##	[436]	"quick"	"develop"	"ocean"	"warm"	"free"
##	[441]	"minute"	"strong"	"special"	"mind"	"behind"
##	[446]	"clear"	"tail"	"produce"	"fact"	"street"
##	[451]	"inch"	"multiply"	"nothing"	"course"	"stay"
##	[456]	"wheel"	"full"	"force"	"blue"	"object"
##	[461]	"decide"	"surface"	"deep"	"moon"	"island"
##	[466]	"foot"	"system"	"busy"	"test"	"record"
##	[471]	"boat"	"common"	"gold"	"possible"	"plane"
##	[476]	"stead"	"dry"	"wonder"	"laugh"	"thousand"
##	[481]	"ago"	"ran"	"check"	"game"	"shape"
##	[486]	"equate"	"hot"	"miss"	"brought"	"heat"
##	[491]	"snow"	"tire"	"bring"	"yes"	"distant"
##	[496]	"fill"	"east"	"paint"	"language"	"among"
##	[501]	"grand"	"ball"	"yet"	"wave"	"drop"
##	[506]	"heart"	"am"	"present"	"heavy"	"dance"
##	[511]	"engine"	"position"	"arm"	"wide"	"sail"
##	[516]	"material"	"size"	"vary"	"settle"	"speak"
##	[521]	"weight"	"general"	"ice"	"matter"	"circle"
##	[526]	"pair"	"include"	"divide"	"syllable"	"felt"
##	[531]	"perhaps"	"pick"	"sudden"	"count"	"square"
##	[536]	"reason"	"length"	"represent"	"art"	"subject"
##	[541]	"region"	"energy"	"hunt"	"probable"	"bed"
##	[546]	"brother"	"egg"	"ride"	"cell"	"believe"
##	[551]	"fraction"	"forest"	"sit"	"race"	"window"
##	[556]	"store"	"summer"	"train"	"sleep"	"prove"
##	[561]	"lone"	"leg"	"exercise"	"wall"	"catch"
##	[566]	"mount"	"wish"	"sky"	"board"	"joy"
##	[571]	"winter"	"sat"	"written"	"wild"	"instrument"

##	[576]	"kept"	"glass"	"grass"	"cow"	"job"
##	[581]	"edge"	"sign"	"visit"	"past"	"soft"
##	[586]	"fun"	"bright"	"gas"	"weather"	"month"
##	[591]	"million"	"bear"	"finish"	"happy"	"hope"
##	[596]	"flower"	"clothe"	"strange"	"gone"	"jump"
##	[601]	"baby"	"eight"	"village"	"meet"	"root"
##	[606]	"buy"	"raise"	"solve"	"metal"	"whether"
##	[611]	"push"	"seven"	"paragraph"	"third"	"shall"
##	[616]	"held"	"hair"	"describe"	"cook"	"floor"
##	[621]	"either"	"result"	"burn"	"hill"	"safe"
##	[626]	"cat"	"century"	"consider"	"type"	"law"
##	[631]	"bit"	"coast"	"copy"	"phrase"	"silent"
##	[636]	"tall"	"sand"	"soil"	"roll"	"temperature"
##	[641]	"finger"	"industry"	"value"	"fight"	"lie"
##	[646]	"beat"	"excite"	"natural"	"view"	"sense"
##	[651]	"ear"	"else"	"quite"	"broke"	"case"
##	[656]	"middle"	"kill"	"son"	"lake"	"moment"
##	[661]	"scale"	"loud"	"spring"	"observe"	"child"
##	[666]	"straight"	"consonant"	"nation"	"dictionary"	"milk"
##	[671]	"speed"	"method"	"organ"	"pay"	"age"
##	[676]	"section"	"dress"	"cloud"	"surprise"	"quiet"
##	[681]	"stone"	"tiny"	"climb"	"cool"	"design"
##	[686]	"poor"	"lot"	"experiment"	"bottom"	"key"
##	[691]	"iron"	"single"	"stick"	"flat"	"twenty"
##	[696]	"skin"	"smile"	"crease"	"hole"	"trade"
##	[701]	"melody"	"trip"	"office"	"receive"	"row"
##	[706]	"mouth"	"exact"	"symbol"	"die"	"least"
##	[711]	"trouble"	"shout"	"except"	"wrote"	"seed"
##	[716]	"tone"	"join"	"suggest"	"clean"	"break"
##	[721]	"lady"	"yard"	"rise"	"bad"	"blow"
##	[726]	"oil"	"blood"	"touch"	"grew"	"cent"
##	[731]	"mix"	"team"	"wire"	"cost"	"lost"
##	[736]	"brown"	"wear"	"garden"	"equal"	"sent"
##	[741]	"choose"	"fell"	"fit"	"flow"	"fair"
##	[746]	"bank"	"collect"	"save"	"control"	"decimal"
##	[751]	"sight"	"gentle"	"woman"	"captain"	"practice"
##	[756]	"separate"	"difficult"	"doctor"	"please"	"protect"
##	[761]	"noon"	"whose"	"locate"	"ring"	"character"
##	[766]	"insect"	"caught"	"period"	"indicate"	"radio"
##	[771]	"spoke"	"atom"	"human"	"history"	"effect"
##	[776]	"electric"	"expect"	"crop"	"modern"	"element"
##	[781]	"hit"	"student"	"corner"	"party"	"supply"
##	[786]	"bone"	"rail"	"imagine"	"provide"	"agree"
##	[791]	"thus"	"capital"	"wont"	"chair"	"danger"
##	[796]	"fruit"	"rich"	"thick"	"soldier"	"process"
##	[801]	"operate"	"guess"	"necessary"	"sharp"	"wing"
##	[806]	"create"	"neighbor"	"wash"	"bat"	"rather"
##	[811]	"crowd"	"corn"	"compare"	"poem"	"string"
##	[816]	"bell"	"depend"	"meat"	"rub"	"tube"
##	[821]	"famous"	"stream"	"fear"	"thin"	"triangle"
##	[826]	"planet"	"hurry"	"chief"	"colony"	"clock"
##	[831]	"mine"	"tie"	"enter"	"major"	"fresh"
##	[836]	"search"	"send"	"yellow"	"gun"	"rose"
##	[841]	"allow"	"print"	"dead"	"spot"	"desert"



## [846]	"suit"	"current"	"lift"	"continue"	"block"
## [851]	"chart"	"hat"	"sell"	"success"	"company"
## [856]	"subtract"	"event"	"particular"	"deal"	"swim"
## [861]	"term"	"opposite"	"wife"	"shoe"	"shoulder"
## [866]	"spread"	"arrange"	"camp"	"invent"	"cotton"
## [871]	"born"	"determine"	"quart"	"nine"	"truck"
## [876]	"noise"	"level"	"chance"	"gather"	"shop"
## [881]	"stretch"	"throw"	"shine"	"property"	"column"
## [886]	"molecule"	"select"	"wrong"	"gray"	"repeat"
## [891]	"require"	"broad"	"prepare"	"salt"	"nose"
## [896]	"plural"	"anger"	"claim"	"continent"	"oxygen"
## [901]	"sugar"	"death"	"pretty"	"skill"	"women"
## [906]	"season"	"solution"	"magnet"	"silver"	"thank"
## [911]	"branch"	"match"	"suffix"	"especially"	"fig"
## [916]	"afraid"	"huge"	"sister"	"steel"	"dollar"
## [921]	"discuss"	"forward"	"similar"	"guide"	"experience"
## [926]	"score"	"apple"	"bought"	"led"	"pitch"
## [931]	"coat"	"mass"	"card"	"band"	"rope"
## [936]	"slip"	"win"	"dream"	"evening"	"condition"
## [941]	"feed"	"tool"	"total"	"basic"	"smell"
## [946]	"valley"	"nor"	"double"	"seat"	"arrive"
## [951]	"master"	"track"	"parent"	"shore"	"division"
## [956]	"sheet"	"substance"	"favor"	"connect"	"post"
## [961]	"spend"	"chord"	"fat"	"glad"	"original"
## [966]	"share"	"station"	"dad"	"bread"	"charge"
## [971]	"proper"	"bar"	"offer"	"segment"	"slave"
## [976]	"duck"	"instant"	"market"	"degree"	"populate"
## [981]	"chick"	"dear"	"enemy"	"reply"	"drink"
## [986]	"occur"	"support"	"speech"	"nature"	"range"
## [991]	"steam"	"motion"	"path"	"liquid"	"log"
## [996]	"meant"	"quotient"	"teeth"	"shell"	"neck"

*#2. Determining the number of words containing the string "ar".*

```
count <- sum(grepl("ar", a))
print(count)
```

```
## [1] 64
```

*#3. Finding a six-letter word that starts with "l" and ends with "r".*

```
six_letter_word <- a[grepl("^l[a-z]{4}r$", a)]
print(six_letter_word)
```

```
## [1] "letter"
```

*#4. Returning all words starting with "col" or end with "eck".*

```
result <- a[grepl("^col|eck$", a)]
print(result)
```

```
## [1] "color" "cold" "check" "collect" "colony" "column" "neck"
```

```
#5. Finding the number of words that contain 4 or more
#consonants (with y included as a consonant).
pattern <- "(.*[bcdfghjklmnpqrstvwxyz]{4,}.*)"
count <- sum(grepl(pattern, a))
print(count)
```

```
## [1] 8
```

```
#6 Returning all words that have a "q" not followed by a "ui".
result <- a[grepl("q(?!ui)", a, perl = TRUE)]
print(result)
```

```
## [1] "question" "equate" "square" "equal" "quart" "quotient"
```

```
#7. Finding all words that contain "k" followed by another letter.
#Running table command on first character following first "k" of each word.
matches <- grep("k[A-Za-z]", a, value = TRUE)

following_chars <- regmatches(matches, regexpr("(?<=k).", matches, perl = TRUE))

char_table <- table(following_chars)

print(char_table)
```

```
## following_chars
## e i n y
## 10 5 2 1
```

```
#Removing all vowels and determining the number of character strings found once.
a_no_vowels <- gsub("[aeiou]", "", a)

unique_count <- length(which(table(a_no_vowels) == 1))

print(unique_count)
```

```
## [1] 581
```

### 3 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/couthcommander/Bios6301/raw/main/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 eval(expr, envir, enclos): object 'death' not found>
```

```
#1. Debugging the myfun function
```

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

```
## debugging in: myfun(haart_df, "death")  
## debug at <text>#1: {  
##   form <- as.formula(response ~ .)  
##   coef(summary(glm(form, data = dat, family = binomial(logit))))  
## }  
## debug at <text>#2: form <- as.formula(response ~ .)  
## debug at <text>#3: coef(summary(glm(form, data = dat, family = binomial(logit))))
```

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

```
#through debugging, an error message is recieved that indicates  
#that variable lengths differ.  
#I believe this is due to the as.formula(response~.) section, which indicates  
#that a variable 'response' should appear as a column in dat. The formula construction  
#uses a character string and therefore does not recognize the variable 'response'  
#as a column of dat. The 'variable lengths differ' error message is due to the  
#function not being able to find the response variable in the dataset.
```

```
#bonus. creating a working function.
```

```
url <- "https://github.com/couthcommander/Bios6301/raw/main/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
```

```

myfun <- function(dat, response) {
  form <- as.formula(paste(response, "~ ."))
  coef(summary(glm(form, data=dat, family=binomial(logit))))
}
# Creating a sample dataset
set.seed(123)
n <- 100
dat <- data.frame(
  response = sample(c(0, 1), n, replace = TRUE),
  x1 = rnorm(n),
  x2 = rnorm(n),
  x3 = rnorm(n)
)

# Testing the function
result <- myfun(dat, "response")
print(result)

```

```

##              Estimate Std. Error    z value Pr(>|z|)
## (Intercept) -0.2999208  0.2058900 -1.45670420 0.1451981
## x1          -0.0165579  0.2119459 -0.07812323 0.9377300
## x2          -0.2360786  0.2257802 -1.04561263 0.2957399
## x3           0.1263709  0.2077793  0.60819758 0.5430564

```