

Appendix A: R Markdown for Investigation into Chronic Disease and Medicine Adherence in the Soifua Manuia (Good Health) Study

Samantha Pettersen

April 28, 2022, 17:27

Contents

1	Loading Needed Libraries and Data	1
1.1	Creation of Subset of Chronic Disease	2
1.2	Merge diagnostic data with health interview data.	3
1.3	Check flags.	4
1.4	Loop to pull diagnoses of participants	6
2	Adherence	8
2.1	Generate Diagnosis-Specific Non/Compliant Data Frames	8
2.2	Fully compliant vs. partially compliant vs. non-compliant.	10
3	Demographics of chronic disease sample.	15
3.1	Sex	15
3.2	Reasons for Stopping Medication	17
3.3	Age	18
3.4	Education	20
3.5	Occupation	21
3.6	Years since Diagnosis	22
3.7	Census Region	23
3.8	Self reported health	24
3.9	Use of Traditional Methods for Disease Treatment	25
4	Session Info	25

This markdown outlines the work for “Investigation of Medicine Adherence in the Samoan Population” (MPH Essay, Department of Human Genetics, Graduate School of Public Health, University of Pittsburgh, 2022). The focus of this work is describing trends in the data as they relate to demographics of Soifua Manuia research participants who have one or more of the following chronic diseases– heart disease, hypertension, and high cholesterol–and their use of medicine to treat their diagnosed disease(s). (Non)adherence/(non)compliance used interchangeably throughout to indicate current use of medication for doc dx of corresponding disease.

```
# Wrap long lines of code.  
  
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

1 Loading Needed Libraries and Data

Demographic and Health Interview data was used in this project.

```

# uploading libraries
library(readxl)
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

# Path to main directories:
# /home/shared_data/samoa/2018_Samoa_Phenotypes/2017-2019_Adiposity_Study_Data_Files

# Demographic data: /Demographic_Data/Soifua Manuia
# Demographic Data December 21 2019.xlsx Health interview
# data: /Health_Interview/Soifua Manuia Health Interview
# Data February 14 2020.xlsx

demographicdata <- read_xlsx(path = "/home/shared_data/samoa/2018_Samoa_Phenotypes/2017-2019_Adiposity_Study_Data_Files/Demographic_Data/Soifua Manuia Demographic Data December 21 2019.xlsx",
                             col_names = TRUE)

Helint <- read_xlsx(path = "/home/shared_data/samoa/2018_Samoa_Phenotypes/2017-2019_Adiposity_Study_Data_Files/Health_Interview/Soifua Manuia Health Interview Data February 14 2020.xlsx",
                   col_names = TRUE)

```

1.1 Creation of Subset of Chronic Disease

```

# Identify which diagnoses have NAs.

table(is.na(Helint$chol_diag_doc))

##
## FALSE TRUE
## 517 2

table(is.na(Helint$htn_diag_doc))

##
## FALSE
## 519

table(is.na(Helint$heartdisease_doc_diag))

##
## FALSE TRUE
## 517 2

# Cholesterol and heart disease each have 2 participants
# who are NA for doctor dx.
Helint[which(is.na(Helint$chol_diag_doc)), 1]

## # A tibble: 2 x 1

```

```

##   IDNumber
##     <dbl>
## 1     310
## 2    3210

# Participants 310 and 3210 are NA for cholesterol dx.
Helint[which(is.na(Helint$heartdisease_doc_diag)), 1]

## # A tibble: 2 x 1
##   IDNumber
##     <dbl>
## 1     310
## 2    1797

# Participants 310 and 1797 are NA for heart disease dx.

# Create values to prevent NA being introduced into
# dataframes usin %in% instead of 'df$col=='.

yes <- 1
no <- 0

# Diagnostics dataframe (create df that has participants
# with chronic disease of interest.)

diag <- Helint[Helint$chol_diag_doc %in% yes | Helint$htn_diag_doc %in%
  yes | Helint$heartdisease_doc_diag %in% yes, ]
range(diag$IDNumber, na.rm = F)

## [1] 194 3471

# double check the dimensions
dim(diag)

## [1] 103 79

# There are 103 participants who have one or more of htn,
# high cholesterol, or heart disease.

```

1.2 Merge diagnostic data with health interview data.

```

diag_demo <- merge(x = diag, y = demographicdata, by.x = "IDNumber",
  by.y = "IDNumber", all.x = TRUE, all.y = FALSE)

# Check the dataset dimensions.
dim(diag_demo)

## [1] 103 94

colnames(diag_demo)

## [1] "IDNumber"                "htn_diag_doc"
## [3] "HTN_diagnosis_flag"      "years_htn_diagnosed"
## [5] "current_meds_htn"       "meds_list_htn"
## [7] "htn_med_flag"           "years_htn_meds"
## [9] "past_htn_med"           "list_past_htn_meds"

```

```

## [11] "years_past_htn_meds"           "stop_htn_meds_reason"
## [13] "stop_htn_meds_otherreason"     "trad_healer_htn"
## [15] "current_trad_meds_htn"         "trad_meds_htn_name"
## [17] "severe_heartpain"             "severe_heartpain_flag"
## [19] "heart_attack"                  "heart_attack_flag"
## [21] "history_stroke"                "history_stroke_flag"
## [23] "heartdisease_doc_diag"         "heartdisease_diagnosis_flag"
## [25] "years_diag_heartdisease"       "current_meds_heartdisease"
## [27] "list_meds_heartdisease"        "years_meds_heartdisease"
## [29] "past_meds_heartdisease"        "list_past_meds_heartdis"
## [31] "years_pastmeds_heartdisease"   "stop_meds_heartdisease_reason"
## [33] "stop_meds_hd_otherreason"      "trad_healer_heartdisease"
## [35] "current_tradmeds_heartdisease" "list_trad_med_heartdisease"
## [37] "dm_diag_doc"                   "diabetes_diagnosis_flag"
## [39] "years_dm_diag"                 "help_diabetes"
## [41] "dm_pill_type"                  "Insulin_type"
## [43] "list_dm_treatment"            "past_meds_dm"
## [45] "past_dm_meds_type"            "years_past_dm_meds"
## [47] "stop_meds_dm_reason"          "stop_meds_dm_otherreason"
## [49] "trad_healer_dm"               "curr_trad_meds_dm"
## [51] "list_trad_meds_dm"            "chol_diag_doc"
## [53] "years_diag_chol"              "curr_chol_meds"
## [55] "list_chol_meds"               "years_chol_meds"
## [57] "past_chol_meds"               "trad_healer_chol"
## [59] "curr_trad_chol_meds"          "list_trad_chol_meds"
## [61] "cancer_diagnosis"             "dialysis"
## [63] "contra_pacemaker"             "contra_metal_implant"
## [65] "contra_metal_implant_type"    "contra_procedure"
## [67] "contra_procedure_type"        "contra_xray_this_year"
## [69] "diag_liver_disease"           "diag_asthma"
## [71] "diag_gout"                     "other_conditions_text"
## [73] "other_conditions_flag"        "missed_work"
## [75] "missed_work_days"             "self_reported_health"
## [77] "self_reported_weight"         "men_number_children_biological"
## [79] "men_number_children_adopted"  "ID_Flag"
## [81] "Village_Code"                 "Census_Region"
## [83] "DecAge"                        "Age_Group"
## [85] "Sex"                           "Marital_Status"
## [87] "MaritalStatus_Flag"           "Ed_Group"
## [89] "Ed_Years"                      "Ed_Flag"
## [91] "workplace"                     "job_title"
## [93] "work_full_time"                "occupation_code"

```

1.3 Check flags.

```
table(diag_demo$ID_Flag, useNA = "always")
```

```
##
##    0 <NA>
## 103    0
```

```
# Hypertension (HTN_diagnosis_flag) DX Flag definitions
# from code book: 0 = No flag 1 = Participant responded
```

```

# "no" to having ever been diagnosed with hypertension in
# 2017-19 but reported a diagnosis in 2010 2 = Participant
# did not report a hypertension diagnosis in 2010, but
# 2017-19 response to years since diagnosis is prior to
# 2010 For this variable, flag =0 or =2 not issue (diseases
# can progress), but =1 may be problematic.

```

```
table(diag_demo$HTN_diagnosis_flag, useNA = "always")
```

```
##
##    0    1    2 <NA>
##   84    1   18    0
```

```
diag_demo[which(diag_demo$HTN_diagnosis_flag == 1), 1] # Participant 2779 has flag for HTN dx.
```

```
## [1] 2779
```

```

# Meds (htn_med_flag) Ignore this flag. Participants in
# this group report doctor dx of htn and this work not
# looking at DM.

```

```

# Heart disease DX (heartdisease_diagnosis_flag) 0 = No
# flag 1 = Participant responded "no" to having ever been
# diagnosed with hypertension in 2017-19 but reported a
# diagnosis in 2010 2 = Participant did not report a
# hypertension diagnosis in 2010, but 2017-19 response to
# years since diagnosis is prior to 2010 3 = Participant
# reports diagnosis prior to 2010 & medication reported not
# used to treat heart disease For this variable, flag =0,
# =2, or =3 is not issue (diseases can progress), but =1
# may be problematic.

```

```
table(diag_demo$heartdisease_diagnosis_flag, useNA = "always")
```

```
##
##    0    1    2    3 <NA>
##   98    1    2    1    1
```

```
diag_demo[which(diag_demo$heartdisease_diagnosis_flag == 1),
1] # Participant 399 has flag for heart disease dx.
```

```
## [1] 399
```

```
# Cholesterol has no flag variables.
```

```

# Participants 2779 and 399 have other doctor dx (confirm
# below) than what they have flags for. Retain these
# participants in analyses and make note in discussion re:
# lack of consistency/recall bias for doctor dx of
# diseases.

```

```

# doc_dx columns htn: 2, $htn_diag_doc heart disease: 23,
# $heartdisease_doc_diag cholesterol: 52, $chol_diag_doc

```

```
diag_demo[which(diag_demo$IDNumber == 2779), c(1, 2, 23, 52)]
```

```
##      IDNumber htn_diag_doc heartdisease_doc_diag chol_diag_doc
```

```
## 90      2779          0          1          0
diag_demo[which(diag_demo$IDNumber == 399), c(1, 2, 23, 52)]

##   IDNumber htn_diag_doc heartdisease_doc_diag chol_diag_doc
## 9         399          1          0          0
```

1.4 Loop to pull diagnoses of participants

This will determine how many people have how many diseases and what diseases people have.

```
# Jon Chernus helped in development of this code.

diag_demo <- as.data.frame(diag_demo)

dim(diag_demo)

## [1] 103 94

# 103 rows in this df indicating 103 unique people with at
# least 1 diagnosis (of hypertension, heart disease, and
# high cholesterol). This is the 'chronic disease sample'.

# First make a vector of the column names we want, sorted
# alphabetically. This is just to avoid having to keep
# typing the column names.

dx_cols <- sort(grep(pattern = "doc", x = names(diag_demo), value = TRUE))
# This will pull dx names, the 3 we're interested in as
# well as DM.

dx_cols

## [1] "chol_diag_doc"          "dm_diag_doc"          "heartdisease_doc_diag"
## [4] "htn_diag_doc"

# Drop DM from list.
dx_cols <- dx_cols[!dx_cols %in% c("dm_diag_doc")]

# Confirm that worked.
dx_cols

## [1] "chol_diag_doc"          "heartdisease_doc_diag" "htn_diag_doc"

# First let's break down 'diag_demo' according to number of
# diagnoses. Make a column that counts up the number of
# diagnoses (0, 1, 2, or 3) No one should have 0 diagnoses,
# but will be good to check. Make a table that shows how
# many people have 0, 1, 2, or 3 diagnoses. Note that
# since our initial filtering to create diag might've
# captured people with NAs in their diagnosis variables,
# it's possible we picked up some people with NA in one or
# more (or even all 4) categories The NAs get counted as
# '0' in the sum here (i.e, the sum of the number of
# diagnoses) (# This should be participants 310,1797, and
# 3210.)
```

```

# Create column that indicates n dx for each participant.
diag_demo$n_diag <- apply(diag_demo[, dx_cols], 1, function(x) {
  sum(x, na.rm = TRUE)
})
table(diag_demo$n_diag, useNA = "always")

##
##    1    2    3 <NA>
##   91   11    1    0

# Abbreviate the dx names.
dx_cols

## [1] "chol_diag_doc"          "heartdisease_doc_diag" "htn_diag_doc"
dxs_short <- c("CL", "HD", "HT")

# It would be nice to create a single column that
# summarizes exactly which diagnoses an individual has.
# This will be easier to summarize than having the data
# spread over 4 columns it currently is. Let's create a
# function to do that... First, make some abbreviations
# (HD=heart disease, HT=hypertension, CL=high cholesterol)
# Define a function paste_dx, which will be applied to each
# row (person) in diag_demo

# This is how you set up any function. The argument, x,
# will represent a row from diag_demo (i.e., a horizontal
# 'slice' of the data from one person)
paste_dx <- function(x) {

  # Pull out the diagnosis (0 or 1) for each of the 4
  # conditions Store the 0 or 1 for the person's status
  # for each disease in a variable
  dx_cl <- x[dx_cols[1]]
  dx_hd <- x[dx_cols[2]]
  dx_ht <- x[dx_cols[3]]

  # Put the 0s/1s together in a vector, treating 0 as
  # FALSE and 1 as TRUE Notice that the order matches the
  # order in the vector of abbreviations we made earlier,
  # dxs_short Replace NA with 0 here
  diagnoses_indices <- as.logical(c(dx_cl, dx_hd, dx_ht))
  diagnoses_indices[is.na(diagnoses_indices)] <- FALSE

  # For the diagnoses the person has, take their
  # abbreviations and stitch them together into one a
  # single 'word'
  diagnosis_short <- paste0(dxs_short[diagnoses_indices], collapse = "_")

  # So for a person with chol_diag_doc==1,
  # htn_diag_doc==0, heartdisease_doc_diag==0, and
  # dm_diag_doc==1, The function will return the string
  # 'DM_CL', meaning they had 2 diagnoses, namely

```

```

    # diabetes and high cholesterol
    return(diagnosis_short)
} # This is the end of the function

# Use the 'apply' function to apply the paste_dx function
# to each row (MARGIN=1) of the data (diag_demo) Store the
# result (each person's diagnosis string) as a new column
diag_demo$diagnoses <- apply(X = diag_demo[, dx_cols], MARGIN = 1,
    FUN = paste_dx)

# Now you can look at what diagnoses actually 'go together'
# (i.e. which participants have which dxs).
table(diag_demo$n_diag, diag_demo$diagnoses, useNA = "always")

```

```

##
##      CL CL_HD CL_HD_HT CL_HT HD HD_HT HT <NA>
##  1     3     0         0   0 14     0 74     0
##  2     0     1         0   4  0     6  0     0
##  3     0     0         1   0  0     0  0     0
## <NA>  0     0         0   0  0     0  0     0

```

```

# For people with exactly 2 diagnoses, what are those
# diagnoses?
table(diag_demo$diagnoses[diag_demo$n_diag == 1], useNA = "always")

```

```

##
##   CL   HD   HT <NA>
##   3  14  74    0

```

```

# For people with exactly 2 diagnoses, what are those
# diagnoses?
table(diag_demo$diagnoses[diag_demo$n_diag == 2], useNA = "always")

```

```

##
## CL_HD CL_HT HD_HT <NA>
##    1    4    6     0

```

```

# For people with exactly 3 diagnoses, what are those
# diagnoses?
table(diag_demo$diagnoses[diag_demo$n_diag == 3], useNA = "always")

```

```

##
## CL_HD_HT <NA>
##      1     0

```

2 Adherence

2.1 Generate Diagnosis-Specific Non/Compliant Data Frames

```

# For each diagnosis, generate a dataframe that includes
# all individuals who have dx but are not currently taking
# medication.

```

```

# Hypertension

```



```
# Non-compliant
htn.nc <- diag_demo[(diag_demo$htn_diag_doc %in% yes & diag_demo$current_meds_htn %in%
  no), ]
dim(htn.nc)
```

```
## [1] 19 96
```

```
# Compliant
htn.C <- diag_demo[(diag_demo$htn_diag_doc %in% yes & diag_demo$current_meds_htn %in%
  yes), ]
dim(htn.C)
```

```
## [1] 66 96
```

```
# Safety check
```

```
nrow(subset(diag_demo, diag_demo$htn_diag_doc %in% yes)) - (nrow(htn.C) +
  nrow(htn.nc))
```

```
## [1] 0
```

```
# Heart Disease
```

```
# Non-compliant
```

```
hd.nc <- diag_demo[(diag_demo$heartdisease_doc_diag %in% yes &
  diag_demo$current_meds_heartdisease %in% no), ]
dim(hd.nc)
```

```
## [1] 5 96
```

```
# Compliant
```

```
hd.C <- diag_demo[(diag_demo$heartdisease_doc_diag %in% yes &
  diag_demo$current_meds_heartdisease %in% yes), ]
dim(hd.C)
```

```
## [1] 17 96
```

```
# Safety check
```

```
nrow(subset(diag_demo, diag_demo$heartdisease_doc_diag %in% yes)) -
  (nrow(hd.C) + nrow(hd.nc))
```

```
## [1] 0
```

```
# Cholesterol
```

```
# Non-compliant
```

```
chol.nc <- diag_demo[(diag_demo$chol_diag_doc %in% yes & diag_demo$curr_chol_meds %in%
  no), ]
dim(chol.nc)
```

```
## [1] 3 96
```

```
# Compliant
```

```
chol.C <- diag_demo[(diag_demo$chol_diag_doc %in% yes & diag_demo$curr_chol_meds %in%
```

```

    yes), ]
dim(chol.C)

## [1] 6 96
# Safety check Safety check

nrow(subset(diag_demo, diag_demo$chol_diag_doc %in% yes)) - (nrow(chol.C) +
  nrow(chol.nc))

## [1] 0
# Rate of non-compliance in each disease.

# Heart Disease
nrow(hd.nc)

## [1] 5
nrow(hd.C)

## [1] 17
nrow(hd.nc)/sum(nrow(hd.nc) + nrow(hd.C))

## [1] 0.2272727
# Hypertension
nrow(htn.nc)

## [1] 19
nrow(htn.C)

## [1] 66
nrow(htn.nc)/sum(nrow(htn.nc) + nrow(htn.C))

## [1] 0.2235294
# Cholesterol
nrow(chol.nc)

## [1] 3
nrow(chol.C)

## [1] 6
nrow(chol.nc)/sum(nrow(chol.nc) + nrow(chol.C))

## [1] 0.3333333

```

2.2 Fully compliant vs. partially compliant vs. non-compliant.

```

# Disease specific dfs above are helpful, but people with
# multiple diagnoses may be taking medication for one, but
# not another. Need to do distinguish beyond
# compliant/non-compliant for thoses individuals who have 2
# or more diagnoses.

```

```

table(diag_demo$n_diag, useNA = "always")

##
##      1      2      3 <NA>
##    91     11      1      0

# ID individuals who have > 1 dx (n_diag > 1)
mult.dx <- diag_demo[(which(diag_demo$n_diag > 1)), 1]
sing.dx <- diag_demo[(which(diag_demo$n_diag == 1)), 1]

length(diag_demo$IDNumber) - (length(sing.dx) + length(mult.dx))

## [1] 0

# For individuals with single dx, if they are
# non-compliant, they are fully non-compliant. For
# individuals with multiple dx, will depend on n dx.

# Safety check: max n_diag should be 3
range(diag_demo$n_diag, na.rm = FALSE)

## [1] 1 3

table(diag_demo$diagnoses, diag_demo$n_diag, useNA = "always")

##
##           1  2  3 <NA>
##    CL           3  0  0  0
##    CL_HD        0  1  0  0
##    CL_HD_HT     0  0  1  0
##    CL_HT        0  4  0  0
##    HD           14  0  0  0
##    HD_HT        0  6  0  0
##    HT           74  0  0  0
##    <NA>         0  0  0  0

# Single dx individuals will be easy, so let's start there.

# Hypertension only. (55/74 individuals compliant)
htn.only <- subset(diag_demo, diag_demo$diagnoses == "HT")
dim(htn.only)

## [1] 74 96

htn.only$compliance <- ifelse(htn.only$current_meds_htn %in%
  yes, "comp", "noncomp")
table(htn.only$compliance, useNA = "always")

##
##      comp noncomp  <NA>
##     55      19      0

# Cholesterol only (1/3 individuals compliant)
chol.only <- subset(diag_demo, diag_demo$diagnoses == "CL")
dim(chol.only)

## [1] 3 96

```

```
chol.only$compliance <- ifelse(chol.only$curr_chol_meds %in%
  yes, "comp", "noncomp")
table(chol.only$compliance, useNA = "always")
```

```
##
##   comp noncomp   <NA>
##    1      2      0
```

```
# Heart disease only (10/14 individuals compliant)
```

```
hd.only <- subset(diag_demo, diag_demo$diagnoses == "HD")
dim(hd.only)
```

```
## [1] 14 96
```

```
hd.only$compliance <- ifelse(hd.only$current_meds_heartdisease %in%
  yes, "comp", "noncomp")
table(hd.only$compliance, useNA = "always")
```

```
##
##   comp noncomp   <NA>
##    10      4      0
```

```
# Merge these dfs.
```

```
sing.dx.comp <- rbind(hd.only, chol.only)
sing.dx.comp <- rbind(sing.dx.comp, htn.only)
dim(sing.dx.comp)
```

```
## [1] 91 97
```

```
prop.table(table(sing.dx.comp$compliance))
```

```
##
##   comp   noncomp
## 0.7252747 0.2747253
```

```
# Next do the single individual who has 3 dxs. (100%
# compliance...)
```

```
trip.dx.comp <- subset(diag_demo, diag_demo$n_diag == 3)
trip.dx.comp$curr_chol_meds # compliant for chol
```

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

```
trip.dx.comp$current_meds_htn # compliant for htn
```

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

```
trip.dx.comp$current_meds_heartdisease # compliant for heart disease
```

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

```
trip.dx.comp$compliance <- "comp"
```

```
# Make new df that will have ALL the compliance info thus
# far.
```

```
all.compliance <- rbind(trip.dx.comp, sing.dx.comp)
```

```
# Now for the tricky ones (CL_HD, CL_HT, HD_HT)
```

```

cl_hd.only <- subset(diag_demo, diag_demo$diagnoses == "CL_HD")
dim(cl_hd.only)

## [1] 1 96
cl_hd.only$curr_chol_meds # compliant for cholesterol

## [1] 1
cl_hd.only$current_meds_heartdisease # compliant for heart disease

## [1] 1
cl_hd.only$compliance <- "comp"

cl_ht.only <- subset(diag_demo, diag_demo$diagnoses == "CL_HT")
dim(cl_ht.only)

## [1] 4 96
table(cl_ht.only$curr_chol_meds, cl_ht.only$current_meds_htn,
      useNA = "always")

##
##      1 <NA>
##  0    1    0
##  1    3    0
## <NA> 0    0
# 1 person not adherent for cholesterol, but for htn.

cl_ht.only$compliance <- ifelse(cl_ht.only$curr_chol_med %in%
  yes, "comp", "part")

# Create dataframe for compliance for individuals with 2
# dxs.
dbl.dx.comp <- rbind(cl_hd.only, cl_ht.only)
dim(dbl.dx.comp)

## [1] 5 97
hd_ht.only <- subset(diag_demo, diag_demo$diagnoses == "HD_HT")
dim(hd_ht.only)

## [1] 6 96
table(hd_ht.only$current_meds_htn, hd_ht.only$current_meds_heartdisease,
      useNA = "always")

##
##      0 1 <NA>
##  1    1 5    0
## <NA> 0 0    0
# One person is non compliant for heart disease, but not
# htn.
hd_ht.only$compliance <- ifelse(hd_ht.only$current_meds_heartdisease %in%
  yes, "comp", "part")

```

```

dbl.dx.comp <- rbind(dbl.dx.comp, hd_ht.only)
dim(dbl.dx.comp)

## [1] 11 97
# Merge all compliance with and double dx compliance dfs to
# have compliance for all individuals in one spot.

all.compliance <- rbind(all.compliance, dbl.dx.comp)
dim(all.compliance)

## [1] 103 97
table(all.compliance$compliance, useNA = "always")

##
##      comp noncomp      part      <NA>
##      76      25         2         0

prop.table(table(all.compliance$compliance, useNA = "always"))

##
##      comp      noncomp      part      <NA>
## 0.73786408 0.24271845 0.01941748 0.00000000

table(all.compliance$compliance, all.compliance$n_diag, useNA = "always")

##
##      1 2 3 <NA>
## comp  66 9 1  0
## noncomp 25 0 0  0
## part   0 2 0  0
## <NA>   0 0 0  0

# All fully noncompliant individuals have only one dx

# Make one more column that collapses partially compliant
# and compliant individuals

all.compliance$bin.comp <- ifelse(all.compliance$compliance ==
  "noncomp", 0, 1)

table(all.compliance$bin.comp)

##
## 0 1
## 25 78

prop.table(table(all.compliance$bin.comp))

##
##      0      1
## 0.2427184 0.7572816

# Make df for individuals who are fully compliant and
# non-compliant (based on binary definition).

comp <- subset(all.compliance, all.compliance$bin.comp %in% yes)

```

```
noncomp <- subset(all.compliance, all.compliance$bin.comp %in%
  no)
```

```
# Overall non-adherence in chronic disease sample (24.2%)
nrow(noncomp)/nrow(all.compliance)
```

```
## [1] 0.2427184
```

3 Demographics of chronic disease sample.

3.1 Sex

```
# 0 = Male
```

```
# 1 = Female
```

```
# Sex breakdown across whole disease sample (n=103).
```

```
table(diag_demo$Sex)
```

```
##
```

```
## 0 1
```

```
## 41 62
```

```
# Hypertension
```

```
table(diag_demo$Sex, diag_demo$htn_diag_doc, useNA = "always")
```

```
##
```

```
##      0 1 <NA>
```

```
## 0    10 31  0
```

```
## 1     8 54  0
```

```
## <NA>  0  0  0
```

```
table(diag_demo$current_meds_htn, diag_demo$Sex, useNA = "always")
```

```
##
```

```
##      0 1 <NA>
```

```
## 0     9 10  0
```

```
## 1    22 44  0
```

```
## <NA> 10  8  0
```

```
# Percent female hypertension individuals.
```

```
length(diag_demo[which(diag_demo$htn_diag_doc %in% yes & diag_demo$Sex ==
  1), 1])/length(diag_demo[which(diag_demo$htn_diag_doc %in%
  yes), 1])
```

```
## [1] 0.6352941
```

```
# Heart Disease
```

```
table(diag_demo$Sex, diag_demo$heartdisease_doc_diag, useNA = "always")
```

```
##
```

```
##      0 1 <NA>
```

```
## 0    32  9  0
```

```
## 1    48 13  1
```

```
## <NA>  0  0  0
```

```

table(diag_demo$current_meds_heartdisease, diag_demo$Sex, useNA = "always")

##
##      0  1 <NA>
##  0    2  3    0
##  1    7 10    0
## <NA> 32 49    0

# Percent female heart disease individuals.
length(diag_demo[which(diag_demo$heartdisease_doc_diag %in% yes &
  diag_demo$Sex == 1), 1])/length(diag_demo[which(diag_demo$heartdisease_doc_diag %in%
  yes), 1])

## [1] 0.5909091

# Hypercholesterolemia
table(diag_demo$Sex, diag_demo$chol_diag_doc, useNA = "always")

##
##      0  1 <NA>
##  0   36  5    0
##  1   58  4    0
## <NA>  0  0    0

table(diag_demo$curr_chol_meds, diag_demo$Sex, useNA = "always")

##
##      0  1 <NA>
##  0    2  1    0
##  1    3  3    0
## <NA> 36 58    0

# Percent female hypercholesterolemia individuals.
length(diag_demo[which(diag_demo$chol_diag_doc %in% yes & diag_demo$Sex ==
  1), 1])/length(diag_demo[which(diag_demo$chol_diag_doc %in%
  yes), 1])

## [1] 0.4444444

# Compliance by sex overall
table(noncomp$Sex, useNA = "always")

##
##  0  1 <NA>
## 11 14  0

prop.table(table(noncomp$Sex, useNA = "always"))

##
##  0  1 <NA>
## 0.44 0.56 0.00

# Disease-specific non-compliance by sex

# Heart Disease
table(hd.nc$Sex, useNA = "always")

```



```
##
##    0    1 <NA>
##    2    3    0
prop.table(table(hd.nc$Sex, useNA = "always"))
```

```
##
##    0    1 <NA>
## 0.4 0.6 0.0
```

```
# Hypertension
```

```
table(htn.nc$Sex, useNA = "always")
```

```
##
##    0    1 <NA>
##    9   10    0
```

```
prop.table(table(htn.nc$Sex, useNA = "always"))
```

```
##
##          0          1          <NA>
## 0.4736842 0.5263158 0.0000000
```

```
# Cholesterol
```

```
table(chol.nc$Sex, useNA = "always")
```

```
##
##    0    1 <NA>
##    2    1    0
```

```
prop.table(table(chol.nc$Sex, useNA = "always"))
```

```
##
##          0          1          <NA>
## 0.6666667 0.3333333 0.0000000
```

3.2 Reasons for Stopping Medication

```
# Cholesterolemia does not have any data (survey question
# not included) regarding why stopping medication. Note in
# limitations and when discussing reasons for stopping
# meds.
```

```
table(diag_demo$Sex, diag_demo$stop_htn_meds_reason)
```

```
##
##    0 4 5 10 11 12 13
##    0 1 1 0 0 1 1 0
##    1 0 0 1 1 0 1 1
```

```
table(diag_demo$Sex, diag_demo$stop_meds_hd_otherreason)
```

```
##
##      The disease cure from the medication
##    0                                0
##    1                                1
```

```
##
## The doctor ordered her to stop and continue her check up every month.
## 0 0
## 1 1
```

```
table(noncomp$stop_htn_meds_otherreason)
```

```
##
## Allergic and feel t.
## The doctor told her not to take it as usual
## The doctor told her to stop
## The doctor told me to stop
## Tired of waiting for the doctor, and stay for a long time. Had a Zumba program with my cousin and ea
##
```

```
table(noncomp$stop_meds_hd_otherreason)
```

```
##
## The disease cure from the medication 1
## The doctor ordered her to stop and continue her check up every month. 1
##
```

3.3 Age

```
# Age data for all of Soifua Manuia (n=519), not just
# chronic disease set.
```

```
mean(demographicdata$DecAge)
```

```
## [1] 52.16139
```

```
range(demographicdata$DecAge)
```

```
## [1] 30.68036 72.69541
```

```
median(demographicdata$DecAge)
```

```
## [1] 51.85216
```

```
# This is age data for the chronic disease group (n=103)
```

```
mean(diag_demo$DecAge, trim = 0, na.rm = TRUE)
```

```
## [1] 56.79699
```

```
range(diag_demo$DecAge, na.rm = FALSE, finite = TRUE)
```

```
## [1] 35.53457 71.13484
```

```
median(diag_demo$DecAge, na.rm = TRUE)
```

```
## [1] 57.45927
```

```
# Plot age distribution for chronic disease sample.
```

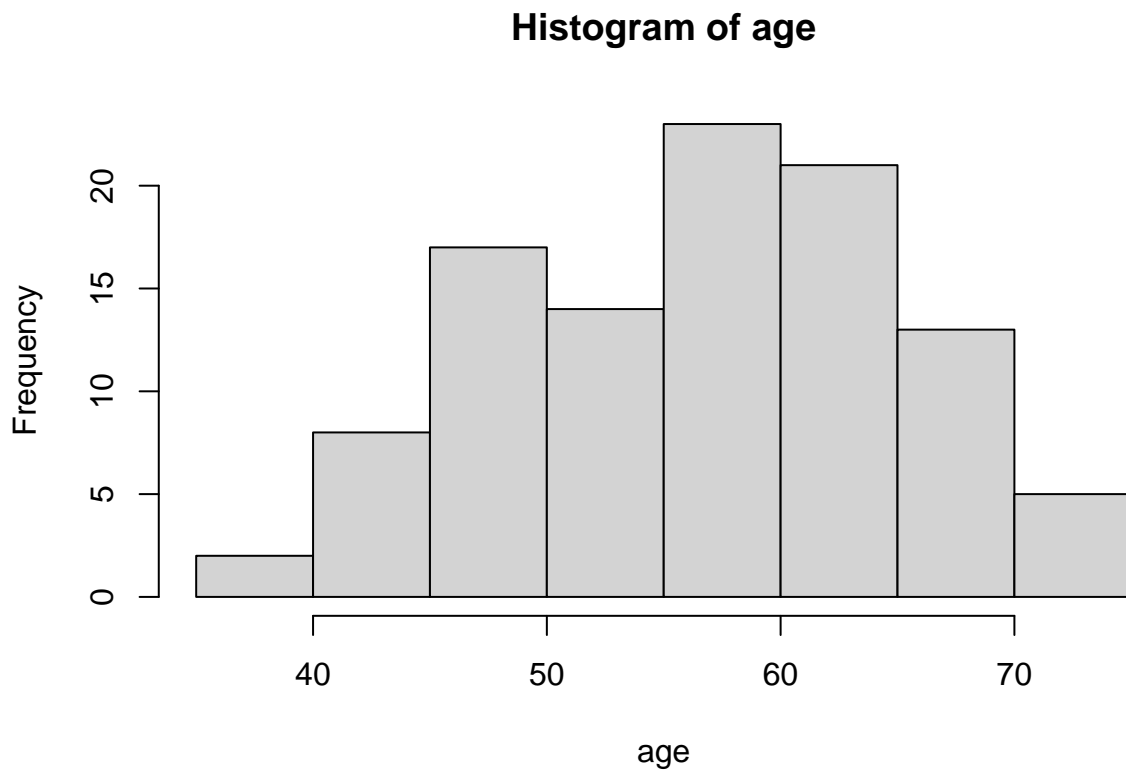
```
diag_demo <- as.data.frame(diag_demo)
```

```
dim(diag_demo)
```

```
## [1] 103 96
```

```
# First, make simple distribution plot of ages.
```

```
age <- diag_demo$DecAge  
hist(age)
```

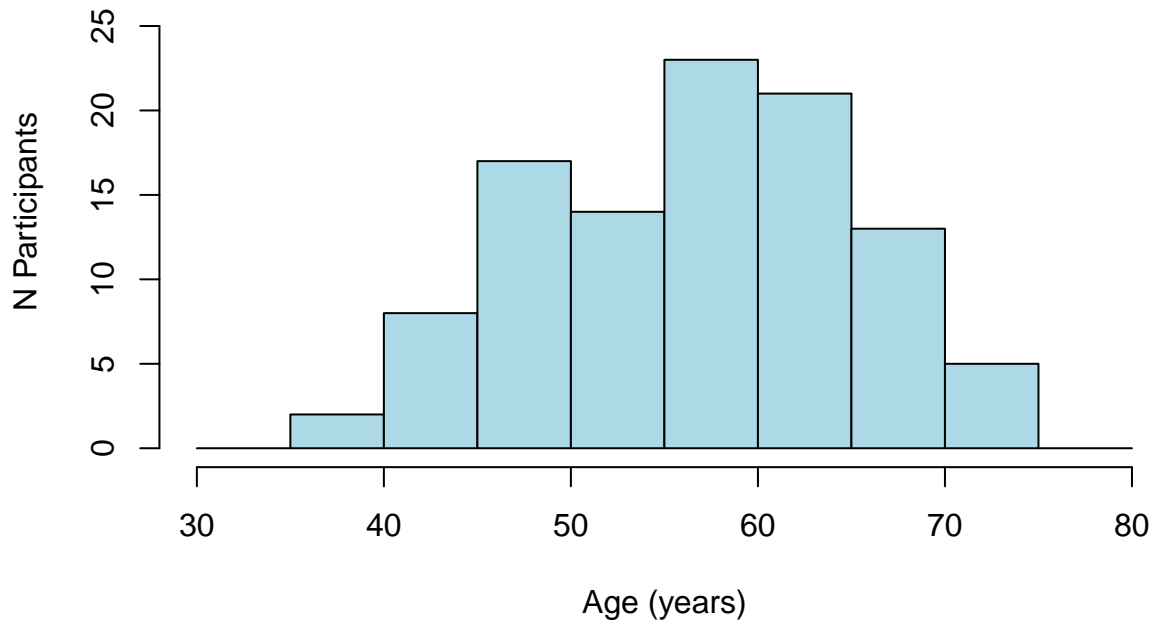


```
range(age)
```

```
## [1] 35.53457 71.13484
```

```
# Make it look nice.
```

```
hist(age, freq = TRUE, col = "lightblue", breaks = c(30, 35,  
40, 45, 50, 55, 60, 65, 70, 75, 80), main = "", xlab = "Age (years)",  
ylab = "N Participants", ylim = c(0, 28))
```



```
# Age for non-compliant group (This is where I want a
# subgroup of the complete non compliance to compare to the
# larger data groups.)
```

```
mean(noncomp$DecAge, trim = 0, na.rm = TRUE)
```

```
## [1] 53.5609
```

```
range(noncomp$DecAge, na.rm = FALSE, finite = TRUE)
```

```
## [1] 35.53457 69.55510
```

```
median(noncomp$DecAge, na.rm = FALSE)
```

```
## [1] 53.23203
```

```
# Age for compliant
```

```
mean(comp$DecAge, trim = 0, na.rm = TRUE)
```

```
## [1] 57.8342
```

```
range(comp$DecAge, na.rm = FALSE, finite = TRUE)
```

```
## [1] 42.03970 71.13484
```

```
median(comp$DecAge, na.rm = FALSE)
```

```
## [1] 58.33949
```

3.4 Education

```
# Entire Sample (n=519)
```

```
mean(demographicdata$Ed_Years, trim = 0, na.rm = TRUE)
```

```
## [1] 11.24324
```

```

median(demographicdata$Ed_Years)

## [1] NA
range(demographicdata$Ed_Years)

## [1] NA NA
# Chronic disease sample (n=103, including 2 NA)
mean(diag_demo$Ed_Years, trim = 0, na.rm = TRUE)

## [1] 10.9902
range(diag_demo$Ed_Years, na.rm = FALSE, finite = TRUE)

## [1] 4 19
median(diag_demo$Ed_Years, na.rm = TRUE)

## [1] 12
# Noncompliant group (n=25)
mean(noncomp$Ed_Years, trim = 0, na.rm = FALSE)

## [1] 11.52
range(noncomp$Ed_Years, na.rm = FALSE, finite = TRUE)

## [1] 7 19
median(noncomp$Ed_Years, na.rm = TRUE)

## [1] 12

```

3.5 Occupation

```

# Occupation codes:
# 0 = Unemployed
# 1 = Skilled labor
# 2 = Semi-skilled labor
# 3 = Plantation work
# 4 = Menial/Custodial
# 5 = Clerical
# 6 = Management
# 7 = Salesperson
# 8 = Educator

```

```

# 9 = Health worker

# 10 = Public Service

# 11 = Student

# 12 = Clergy/Mayor

# 13 = Retired

# 14 = Small Business Owner

# Jobs of individuals with chronic disease.
table(diag_demo$occupation_code, useNA = "always")

##
##    0    1    2    3    4    5    6    8    9   12   14 <NA>
##   61    4    1   19    3    3    4    1    1    3    2    1

# For individuals who are medicine non-adherent.

table(noncomp$occupation_code, useNA = "always")

##
##    0    1    3    8   12   14 <NA>
##   14    1    7    1    1    1    0

```

3.6 Years since Diagnosis

```

# Noncompliant Heart Disease

mean(hd.nc$years_diag_heartdisease)

## [1] 11.4

median(hd.nc$years_diag_heartdisease)

## [1] 10

# Recall hypertension has a few NAs
mean(htn.nc$years_htn_diagnosed, trim = 0, na.rm = TRUE)

## [1] 4.444444

median(htn.nc$years_htn_diagnosed, trim = 0, na.rm = TRUE)

## [1] 3

# Cholesterol

mean(chol.nc$years_diag_chol)

## [1] 1.333333

median(chol.nc$years_diag_chol)

## [1] 2

```

```

# Compliant Heart Disease
mean(hd.C$years_diag_heartdisease)

## [1] 3.529412
median(hd.C$years_diag_heartdisease)

## [1] 4
# Hypertension
mean(htn.C$years_htn_diagnosed, trim = 0, na.rm = TRUE)

## [1] 5.382969
median(htn.C$years_htn_diagnosed, trim = 0, na.rm = TRUE)

## [1] 3
# Cholesterol
mean(chol.C$years_diag_chol)

## [1] 1.333333
median(chol.C$years_diag_chol)

## [1] 1
# Specifically for heart disease the non-compliant people
# have a higher average of years since diagnosis- according
# to my research this is absolutely not surprising.

```

3.7 Census Region

```

# Census region codes:
# 1 = Apia Urban Area (AUA)
# 2 = Northwest Upolu (NWU)
# 3 = Rest of Upolu (ROU)
# Chronic disease sample
table(diag_demo$Census_Region)

##
## 1 2 3
## 31 39 33
prop.table(table(diag_demo$Census_Region))

##
##      1      2      3
## 0.3009709 0.3786408 0.3203883
# Noncompliant participants
table(noncomp$Census_Region)

##

```

```

## 1 2 3
## 7 9 9
prop.table(table(noncomp$Census_Region))

##
##      1      2      3
## 0.28 0.36 0.36
# Hypercholesterolemia

table(diag_demo$curr_chol_meds, diag_demo$Census_Region)

##
##      1 2 3
## 0 1 1 1
## 1 2 3 1
# Heart disease

table(diag_demo$current_meds_heartdisease, diag_demo$Census_Region)

##
##      1 2 3
## 0 2 2 1
## 1 7 7 3
# Hypertension

table(diag_demo$current_meds_htn, diag_demo$Census_Region)

##
##      1 2 3
## 0 5 7 7
## 1 19 25 22

```

3.8 Self reported health

```

# Scale for self-reported health: 0 = Excellent 1 = Very
# good 2 = Good 3 = Poor 4 = Very poor

# Table self-reported health by binary compliance

table(all.compliance$bin.comp, all.compliance$self_reported_health,
      useNA = "always")

##
##      0 1 2 3 4 <NA>
## 0      2 8 15 0 0 0
## 1      6 17 48 6 1 0
## <NA> 0 0 0 0 0 0
# Wildly, people compliant with medication are the only
# ones that report poor health.

```


3.9 Use of Traditional Methods for Disease Treatment

```
# Make vector of IDs who have ever visited traditional
# healer for htn, hd, or chol.
trad.healer <- all.compliance[which(all.compliance$trad_healer_htn ==
  1 | all.compliance$trad_healer_heartdisease == 1 | all.compliance$trad_healer_htn ==
  1), 1]
length(trad.healer)

## [1] 6

# Make vector of IDs who have ever used traditional meds
trad.meds <- all.compliance[which(all.compliance$current_trad_meds_htn ==
  1 | all.compliance$current_tradmeds_heartdisease == 1 | all.compliance$current_trad_meds_htn ==
  1), 1]
length(trad.meds)

## [1] 12

all.compliance$trad.meds <- ifelse(all.compliance$IDNumber %in%
  trad.meds, 1, 0)
all.compliance$trad.heal <- ifelse(all.compliance$IDNumber %in%
  trad.healer, 1, 0)

table(all.compliance$trad.heal)

##
## 0 1
## 97 6

table(all.compliance$trad.meds)

##
## 0 1
## 91 12

all.compliance$trad.both <- all.compliance$trad.meds + all.compliance$trad.heal

table(all.compliance$bin.comp, all.compliance$trad.both, useNA = "always")

##
##      0  1  2 <NA>
## 0    24  1  0    0
## 1    64 11  3    0
## <NA>  0  0  0    0
```

4 Session Info

```
sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
```

```

## BLAS: /usr/local/lib64/R/lib/libRblas.so
## LAPACK: /usr/local/lib64/R/lib/libRlapack.so
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8 LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8 LC_NAME=C
## [9] LC_ADDRESS=C LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## other attached packages:
## [1] dplyr_1.0.7 readxl_1.3.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.7 rstudioapi_0.13 knitr_1.36 magrittr_2.0.1
## [5] tidyselect_1.1.1 R6_2.5.1 rlang_0.4.12 fastmap_1.1.0
## [9] fansi_0.5.0 highr_0.9 stringr_1.4.0 tools_4.1.2
## [13] xfun_0.28 utf8_1.2.2 cli_3.1.0 DBI_1.1.1
## [17] htmltools_0.5.2 ellipsis_0.3.2 assertthat_0.2.1 yaml_2.2.1
## [21] digest_0.6.28 tibble_3.1.6 lifecycle_1.0.1 crayon_1.4.2
## [25] purrr_0.3.4 formatR_1.11 vctrs_0.3.8 glue_1.5.0
## [29] evaluate_0.14 rmarkdown_2.11 stringi_1.7.5 compiler_4.1.2
## [33] pillar_1.6.4 cellranger_1.1.0 generics_0.1.1 pkgconfig_2.0.3

```