Decile_Gender Composition of High School to University links

This notebook will make two types of matrices of counts for each year available. One with student school decile and school gender (coed vs single sex) by Tertiary_discipline (e.g., university_engineering, polytec_bio), one with just student gender and decile by tertiary discipline It will iterate through each year for which data is available and save independant matrices for each year. These matrices can be aggregated to all years if need be. To test code, remove the "for(year in YEARS[[1]]){" and the final "}", then replace "year" with any year string from 2000 to 2016. Delete the "#" in front of instances of "`{R}" and "`". You will then be able to run each chunk and view the data at each stage. The csvs can be put into a single excel spreadsheet when submitting.

The final matrices of counts that are produced have rr3 for unweighted counts involving education data applied after line 300.

First copy in function for random rounding to base 3

```
#Developed by: Chris Hansen
#Date: 30/05/16

#' Randomly round integer input to base n.
#'
#' @param x Vector or data frame to be rounded.
#' @param n Base.
#' @details A vector (which includes a 'scalar') will be rounded only if every
#' value could be considered an integer.  Similarly, rounding a data frame
#' results in a copy where only columns which can be considered integer-valued
#' are rounded.
#' @return A copy of x which has been randomly rounded to base n.
#' @examples
#' x <- data.frame(a=rpois(10,50), b=rpois(10,5), c=rnorm(10,0,1))
#' rrn(x, 3)

rrn <- function(x, n=3, seed)
{
   if (!missing(seed)) set.seed(seed)

   rr <- function(x, n){
      if (is.na(x)) return(0)
      if ((x%%n)==0) return(x)
      res <- abs(x)
      lo <- (res%/%n) * n
      if ((runif(1) * n) <= res%%n) res <- lo + n
      else res <- lo
      return(ifelse(x<0, (-1)*res, res))
   }

   isint <- function(x){
      x <- x[!is.na(x)]
      sum(as.integer(x)==x)==length(x)
   }

if (class(x) %in% c("numeric", "integer")){
```

```
   if(isint(x)) return(sapply(x, rr, n))
   else return(x)
}

   for (i in 1:ncol(x))
   {
      if (class(x[,i]) %in% c("numeric", "integer") & isint(x[,i])) x[,i] <- sapply(x[,i], rr, n)
   }
   x
}
```

ImportData

First, install libraries.

```
# install.packages("Rcpp")
# install.packages("dplyr")
library(RODBC)
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:plyr':
##
##     compact
```

```
library(tidyr)
library(stringr)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

Tertiary

Get info about students who went on to tertiary level studies Get years for which we have tertiary data (2000 - 2016)

```
con=odbcDriverConnect("Driver=ODBC Driver 11 for SQL Server; Trusted_Connection=YES;Server=WPRDSQL36.sta

YEARS <- (sqlQuery(con,"SELECT distinct([IDI_Clean].[moe_clean].[course].[moe_crs_year_nbr]) FROM [IDI_C

#close connection
odbcClose(con)
YEARS <- list(YEARS$moe_crs_year_nbr)

for(year in YEARS[[1]]){print(as.character(year))}
```

```
## [1] "2000"
## [1] "2001"
## [1] "2002"
## [1] "2003"
## [1] "2004"
## [1] "2005"
## [1] "2006"
## [1] "2007"
## [1] "2008"
## [1] "2009"
## [1] "2010"
## [1] "2011"
## [1] "2012"
## [1] "2013"
## [1] "2014"
## [1] "2015"
## [1] "2016"
```

```
#We will iterate through the years where data is available
for(year in YEARS[[1]]){
  #Time it
  start_time <- proc.time()
```

```
  #connect to SQL
  con=odbcDriverConnect("Driver=ODBC Driver 11 for SQL Server; Trusted_Connection=YES;
                        Server=WPRDSQL36.stats.govt.nz,49530;Database=IDI_Clean")

  #Run query to get dataframe of individual info from MoE Tertiary ([course]).
  #This table does include some info on HS (school, decile, qual).

  #Variable defs:
        # [moe_crs_category_code] #MoE Tertiary category (e.g., Arts, Compsci etc.)
        # [moe_crs_country_code] #Where is the student a citizen
        # [moe_crs_course_class_code] #MoE Tertiary class (e.g., Arts, Compsci etc.)
        # [moe_crs_complete_code] #Completed course successfully or not
        # [moe_crs_last_school_decile_code] #High school decile
        # [moe_crs_isced_level_code] #CERTIFICATE, NATIONAL CERTIFICATE LEVELS 1-3, MASTERS etc
        # [moe_crs_provider_location_code] #Tertiary institution
        # [moe_crs_course_nzsced_code] #Tertiary subject of course
  #- e.g., Mathematics, Statistics...
        # [moe_crs_qual_nzsced_code] #Tertiary subject of qualification
        # [moe_crs_prior_activity_code] # student background before tertairy
  #- Secondary school student, Non-employed or beneficiary etc...
        # [moe_crs_qacc_code] #qualification level
        # [moe_crs_course_level_code] #Course level
        # [moe_crs_qual_level_code] #HS Qual
        # [moe_crs_comp_territorial] #Territory where course completed

  #This query gets student's qualification, and the year in which they took
  #their first course in that qualification (Course_Start_Year).
  #We then specificy the year we want to sample from
  #(We will use paste0 to copy in the year of the current iteration).
  #We then join this table to a table with the rest of the
  #student's tertiary info based on their ID, Course_Year_Number and qual_nzsced_code.
  #This query allows us to count all instances of a student
  #starting a qualification per specified year
  #(some students start multiple qualifications in a year,
  #and we want all unique quals per year per student)
  #We will end up with a dataframe of ID, the year of
  #the current iteration, qual_nzsced_code
  #(The subject discipline of the students qualification),
  #student gender, student ethnicity, student tertiary institution,
  #student high school.

   Tertiary_q <- paste0("SELECT a.snz_uid,
a.Course_Start_Year,
a.moe_crs_qual_nzsced_code,
b.moe_crs_sex_snz_code,
b.moe_crs_ethnic1_snz_code,
b.moe_crs_provider_code,
b.moe_crs_last_school_nbr
  FROM (SELECT t.*
        FROM (
         SELECT [IDI_Clean].[moe_clean].[course].[snz_uid]

               ,MIN([IDI_Clean].[moe_clean].[course].[moe_crs_year_nbr]) Course_Start_Year
```

```
              ,[IDI_Clean].[moe_clean].[course].[moe_crs_qual_nzsced_code]

       FROM  [IDI_Clean].[moe_clean].[course]

             GROUP BY snz_uid, [moe_crs_qual_nzsced_code]
            ) t
 WHERE t.Course_Start_Year =", as.character(year), "
      ) a

JOIN (SELECT distinct([IDI_Clean].[moe_clean].[course].[snz_uid])
                     ,[IDI_Clean].[moe_clean].[course].[moe_crs_sex_snz_code]
                     ,[IDI_Clean].[moe_clean].[course].[moe_crs_ethnic1_snz_code]
                    ,[IDI_Clean].[moe_clean].[course].[moe_crs_year_nbr]
                     ,[IDI_Clean].[moe_clean].[course].[moe_crs_provider_code]

                    ,[IDI_Clean].[moe_clean].[course].[moe_crs_qual_nzsced_code]

                     ,[IDI_Clean].[moe_clean].[course].[moe_crs_last_school_nbr]


     FROM  [IDI_Clean].[moe_clean].[course]) b

 ON  a.snz_uid = b.snz_uid
   AND a.Course_Start_Year = b.moe_crs_year_nbr
   AND a.moe_crs_qual_nzsced_code = b.moe_crs_qual_nzsced_code


ORDER BY a.snz_uid
  ")
#Get info and put into data.table called df_tertiary
df_tertiary <- as.data.table(sqlQuery(con,Tertiary_q, as.is = T))


#close connection
odbcClose(con)
print(paste0("MainSQL for ", year, " complete. "))
proc.time() - start_time

#View Data
#glimpse(df_tertiary)
#```


#Get Metadata for Tertiary
#```{R}

#Connect to SQL
con=odbcDriverConnect("Driver=ODBC Driver 11 for SQL Server; Trusted_Connection=YES;
                      Server=WPRDSQL36.stats.govt.nz,49530;Database=IDI_Clean")

#get Provider (i.e., tertiary institution) Metadata
df_Ter_crs_provider <- sqlQuery(con, "SELECT [provider_code],
                                [provider_name],
```

```
                                [Provider_type]
                                FROM [IDI_Sandpit].[clean_read_MOE].[moe_provider_lookup_table]",
                                as.is = T)

#Convert provider code to chr before joining
df_Ter_crs_provider$provider_code <- as.character(as.integer(df_Ter_crs_provider$provider_code))

#join metadata to df_tertiary
df_tertiary <- left_join(df_tertiary, df_Ter_crs_provider,
                         by = c("moe_crs_provider_code" = "provider_code"))


#get NZSCED Metadata
df_Ter_crs_qual_nzsced <-  sqlQuery(con, "SELECT *
                                    FROM [IDI_Sandpit].[clean_read_MOE].[moe_NZSCED_code]",
                                    as.is = T)
#join metadata to df_tertiary
df_tertiary <- left_join(df_tertiary, df_Ter_crs_qual_nzsced,
                         by = c("moe_crs_qual_nzsced_code"= "MOE_NZSCEDq_code"))


#Close connection
odbcClose(con)

#```

#```{R}
#Replace student gender from 1 and 2 to Male and Female respectively
df_tertiary$moe_crs_sex_snz_code <- str_replace(as.character(df_tertiary$moe_crs_sex_snz_code),
                                                pattern = "1", replacement = "Male")
df_tertiary$moe_crs_sex_snz_code <- str_replace(as.character(df_tertiary$moe_crs_sex_snz_code),
                                                pattern = "2", replacement = "Female")
#```

#```{R}
#Mutate new column that groups NZSCED by discipline. NZSCED codes can be checked online


df_tertiary<- df_tertiary %>%
  mutate(Tertiary_Discipline =
 ifelse(str_detect(moe_crs_qual_nzsced_code, "^0109"), "Biology",
         ifelse(str_detect(moe_crs_qual_nzsced_code, "^01"), "PhysicalScience", #all other codes begi
                ifelse(str_detect(moe_crs_qual_nzsced_code, "^0201"), "Compsci",
                       ifelse(str_detect(moe_crs_qual_nzsced_code, "^03"), "Engineering",
                              ifelse(str_detect(moe_crs_qual_nzsced_code, "^06"), "Health",
                                     ifelse(str_detect(moe_crs_qual_nzsced_code, "^^090[137][139]"), "

#```
print(paste0("Tertiary Metadata for ", year, " complete. "))
proc.time() - start_time

#Since studying each subject discipline can be considered quite different depending on tertiary insti
#we will combine provider type and tertiary discipline.
```

```
#(e.g, instead of University and Engineering, it will be University_Engineering)
#```{R}
#Make new column that is the same as provider_type.
#We will then unite it with discipline to give new variable
df_Study <- df_tertiary %>% mutate(Provider_type1 = Provider_type)
df_Study  <- df_Study %>% unite(Tertiary_Discipline,
                                c("Provider_type1", "Tertiary_Discipline"),
                                sep = "_")
df_Study$Tertiary_Discipline <- as.factor(df_Study$Tertiary_Discipline)

#head(df_Study)
#```

#get high school profile info - decile and gender
#```{R}
#Connect
con=odbcDriverConnect("Driver=ODBC Driver 11 for SQL Server; Trusted_Connection=YES;
                      Server=WPRDSQL36.stats.govt.nz,49530;Database=IDI_Clean")

df_HS_Profile <- sqlQuery(con, "SELECT [SchoolNumber]
       ,[DecileID]
       ,[SchoolGender2]
 FROM [IDI_Metadata].[clean_read_CLASSIFICATIONS].[moe_school_profile]", as.is = T)

odbcClose(con)

#```
#school gender metadata obtained from MOE Schools classification excel spreadsheet
#```{R}
SchoolGender <- data.frame(SchoolGenderCode = c(1,2,3,4,7), SchoolGender = c("Co-ed",
                                            "Single sex (Boys school)",
                                            "Single sex (Girls school)",
                                            "Senior co-ed, junior boys",
                                            "Primary co-ed, secondary girls"))
df_HS_Profile <- left_join(df_HS_Profile, SchoolGender,
                           by = c("SchoolGender2" = "SchoolGenderCode"))
#```

#Join HS profile to original df by last school number
#```{R}
df_Study <-  left_join(df_Study, df_HS_Profile,
                       by = c("moe_crs_last_school_nbr" = "SchoolNumber"))
#```

#Data manipulation so df has correct variables
#```{R}

df_Study_clean <- df_Study %>%
  select(snz_uid,
         moe_crs_sex_snz_code,
         moe_crs_ethnic1_snz_code,
         DecileID, SchoolGender,
         Tertiary_Discipline,
```

```
         Course_Start_Year) %>% #Select relevant variables
  unique() %>% #Make sure all rows are unique.
  select(-snz_uid) %>% #remove snz_uid
  as.data.table()
#head(df_Study_clean)

#```
print(paste0("High School Metadata for ", year, " complete. "))
proc.time() - start_time

#Get matrix of School Decile and school Gender by TertiaryDiscipline with counts
#```{R}

df_Study_clean %>%
  select(DecileID, SchoolGender, Tertiary_Discipline) %>%
  #select relevant columns. The school decile, the school gender (single sex, coed etc),
  #and the tertiary discipline
  unite("School_DecileGender", DecileID, SchoolGender) %>%
  #Combine school decile and gender to one variable
  group_by(School_DecileGender, Tertiary_Discipline) %>%
  #group by school decile_gender and tertiary discipline
  summarise(Counts = n()) -> SchoolDecileGender_TertiaryDiscipline
#Get count of rows for each group.
#This count represents number of students in that each group
#(School_decilegender -> tertiary_discipline).

#Recast dataframe of counts as a matrix.
SchoolDecileGender_TertiaryDiscipline_matrix <- acast(SchoolDecileGender_TertiaryDiscipline,
                                                      School_DecileGender~Tertiary_Discipline,
                                                      value.var = "Counts")

print(paste0("M1 for ", year, " complete. "))
proc.time() - start_time


#Random rounding to base 3 for cell values following rule 1b for unweighted counts for education data
SchoolDecileGender_TertiaryDiscipline_matrix <- rrn(SchoolDecileGender_TertiaryDiscipline_matrix, 3)

#surpress any cell values under 5
SchoolDecileGender_TertiaryDiscipline_matrix <- ifelse(SchoolDecileGender_TertiaryDiscipline_matrix<5
                                                       !is.na(SchoolDecileGender_TertiaryDiscipline_
                                                      "S",SchoolDecileGender_TertiaryDiscipline_matri
print(paste0("RR3 M1 for ", year, " complete. "))
proc.time() - start_time


#Save matrix as csv. We will use the Year of the current iteration in the name of the file.
#e.g., if the loop is on year 2002, the file name will be
# 2002SchoolDecileGender_TertiaryDiscipline_matrix.csv
#We will end up with one file for each year
write.csv(SchoolDecileGender_TertiaryDiscipline_matrix,
          file = paste0(as.character(year), "SchoolDecileGender_TertiaryDiscipline_matrix.csv"))
```

```
  #```
  #Get matrix of student gender by tertiary discipline
  #```{R}
  df_Study_clean %>%
    select(moe_crs_sex_snz_code, DecileID, Tertiary_Discipline) %>%
    #Select relevant columns
    unite("SchoolDecile_StudentGender", DecileID, moe_crs_sex_snz_code) %>%
    #combine Decile and student gender into one variable
    group_by(SchoolDecile_StudentGender, Tertiary_Discipline) %>%
    #group by each decile_studentgender category and tertiary_discipline
    summarise(Counts = n()) -> SchoolDecileStudentGender_TertiaryDiscipline
   #Get counts for each group


  SchoolDecileStudentGender_TertiaryDiscipline_matrix <- acast(SchoolDecileStudentGender_TertiaryDiscipl
                                                               SchoolDecile_StudentGender~Tertiary_Disci
                                                               value.var = "Counts")



  #Round values using random rounding to base 3 following rules for unweighted counts with education da
  SchoolDecileStudentGender_TertiaryDiscipline_matrix <- rrn(SchoolDecileStudentGender_TertiaryDisciplir

  print(paste0("RR3 M1 for ", year, " complete. "))
  proc.time() - start_time


  #All values less than 5 (but not NA) will be assigned the value 5 to ensure confidentiality following
  #rule 1b for unweighted counts.
  SchoolDecileStudentGender_TertiaryDiscipline_matrix <- ifelse(SchoolDecileStudentGender_TertiaryDisci
                                                                & !is.na(SchoolDecileStudentGender_Terti
                                                                "S",SchoolDecileStudentGender_TertiaryDi

  print(paste0("M2 for ", year, " complete. "))
  proc.time() - start_time
  #save matrix as csv. We will use the Year of the current iteration in the name of the file.
  #e.g., if the loop is on year 2002, the file name will be
  # 2002SchoolDecileStudentGender_TertiaryDiscipline_matrix.csv
  #We will end up with one file for each year
  write.csv(SchoolDecileStudentGender_TertiaryDiscipline_matrix,
            file = paste0(as.character(year), "SchoolDecileStudentGender_TertiaryDiscipline_matrix.csv"

  #time of loop

  print(paste0("Year ", year, " loop finished", "\n"))
  proc.time()- start_time
}
```

```
## [1] "MainSQL for 2000 complete. "
## [1] "Tertiary Metadata for 2000 complete. "
## [1] "High School Metadata for 2000 complete. "
## [1] "M1 for 2000 complete. "
## [1] "RR3 M1 for 2000 complete. "
## [1] "RR3 M1 for 2000 complete. "
```

```
## [1] "M2 for 2000 complete. "
## [1] "Year 2000 loop finished\n"
## [1] "MainSQL for 2001 complete. "
## [1] "Tertiary Metadata for 2001 complete. "
## [1] "High School Metadata for 2001 complete. "
## [1] "M1 for 2001 complete. "
## [1] "RR3 M1 for 2001 complete. "
## [1] "RR3 M1 for 2001 complete. "
## [1] "M2 for 2001 complete. "
## [1] "Year 2001 loop finished\n"
## [1] "MainSQL for 2002 complete. "
## [1] "Tertiary Metadata for 2002 complete. "
## [1] "High School Metadata for 2002 complete. "
## [1] "M1 for 2002 complete. "
## [1] "RR3 M1 for 2002 complete. "
## [1] "RR3 M1 for 2002 complete. "
## [1] "M2 for 2002 complete. "
## [1] "Year 2002 loop finished\n"
## [1] "MainSQL for 2003 complete. "
## [1] "Tertiary Metadata for 2003 complete. "
## [1] "High School Metadata for 2003 complete. "
## [1] "M1 for 2003 complete. "
## [1] "RR3 M1 for 2003 complete. "
## [1] "RR3 M1 for 2003 complete. "
## [1] "M2 for 2003 complete. "
## [1] "Year 2003 loop finished\n"
## [1] "MainSQL for 2004 complete. "
## [1] "Tertiary Metadata for 2004 complete. "
## [1] "High School Metadata for 2004 complete. "
## [1] "M1 for 2004 complete. "
## [1] "RR3 M1 for 2004 complete. "
## [1] "RR3 M1 for 2004 complete. "
## [1] "M2 for 2004 complete. "
## [1] "Year 2004 loop finished\n"
## [1] "MainSQL for 2005 complete. "
## [1] "Tertiary Metadata for 2005 complete. "
## [1] "High School Metadata for 2005 complete. "
## [1] "M1 for 2005 complete. "
## [1] "RR3 M1 for 2005 complete. "
## [1] "RR3 M1 for 2005 complete. "
## [1] "M2 for 2005 complete. "
## [1] "Year 2005 loop finished\n"
## [1] "MainSQL for 2006 complete. "
## [1] "Tertiary Metadata for 2006 complete. "
## [1] "High School Metadata for 2006 complete. "
## [1] "M1 for 2006 complete. "
## [1] "RR3 M1 for 2006 complete. "
## [1] "RR3 M1 for 2006 complete. "
## [1] "M2 for 2006 complete. "
## [1] "Year 2006 loop finished\n"
## [1] "MainSQL for 2007 complete. "
## [1] "Tertiary Metadata for 2007 complete. "
## [1] "High School Metadata for 2007 complete. "
## [1] "M1 for 2007 complete. "
```

```
## [1] "RR3 M1 for 2007 complete. "
## [1] "RR3 M1 for 2007 complete. "
## [1] "M2 for 2007 complete. "
## [1] "Year 2007 loop finished\n"
## [1] "MainSQL for 2008 complete. "
## [1] "Tertiary Metadata for 2008 complete. "
## [1] "High School Metadata for 2008 complete. "
## [1] "M1 for 2008 complete. "
## [1] "RR3 M1 for 2008 complete. "
## [1] "RR3 M1 for 2008 complete. "
## [1] "M2 for 2008 complete. "
## [1] "Year 2008 loop finished\n"
## [1] "MainSQL for 2009 complete. "
## [1] "Tertiary Metadata for 2009 complete. "
## [1] "High School Metadata for 2009 complete. "
## [1] "M1 for 2009 complete. "
## [1] "RR3 M1 for 2009 complete. "
## [1] "RR3 M1 for 2009 complete. "
## [1] "M2 for 2009 complete. "
## [1] "Year 2009 loop finished\n"
## [1] "MainSQL for 2010 complete. "
## [1] "Tertiary Metadata for 2010 complete. "
## [1] "High School Metadata for 2010 complete. "
## [1] "M1 for 2010 complete. "
## [1] "RR3 M1 for 2010 complete. "
## [1] "RR3 M1 for 2010 complete. "
## [1] "M2 for 2010 complete. "
## [1] "Year 2010 loop finished\n"
## [1] "MainSQL for 2011 complete. "
## [1] "Tertiary Metadata for 2011 complete. "
## [1] "High School Metadata for 2011 complete. "
## [1] "M1 for 2011 complete. "
## [1] "RR3 M1 for 2011 complete. "
## [1] "RR3 M1 for 2011 complete. "
## [1] "M2 for 2011 complete. "
## [1] "Year 2011 loop finished\n"
## [1] "MainSQL for 2012 complete. "
## [1] "Tertiary Metadata for 2012 complete. "
## [1] "High School Metadata for 2012 complete. "
## [1] "M1 for 2012 complete. "
## [1] "RR3 M1 for 2012 complete. "
## [1] "RR3 M1 for 2012 complete. "
## [1] "M2 for 2012 complete. "
## [1] "Year 2012 loop finished\n"
## [1] "MainSQL for 2013 complete. "
## [1] "Tertiary Metadata for 2013 complete. "
## [1] "High School Metadata for 2013 complete. "
## [1] "M1 for 2013 complete. "
## [1] "RR3 M1 for 2013 complete. "
## [1] "RR3 M1 for 2013 complete. "
## [1] "M2 for 2013 complete. "
## [1] "Year 2013 loop finished\n"
## [1] "MainSQL for 2014 complete. "
## [1] "Tertiary Metadata for 2014 complete. "
```

```
## [1] "High School Metadata for 2014 complete. "
## [1] "M1 for 2014 complete. "
## [1] "RR3 M1 for 2014 complete. "
## [1] "RR3 M1 for 2014 complete. "
## [1] "M2 for 2014 complete. "
## [1] "Year 2014 loop finished\n"
## [1] "MainSQL for 2015 complete. "
## [1] "Tertiary Metadata for 2015 complete. "
## [1] "High School Metadata for 2015 complete. "
## [1] "M1 for 2015 complete. "
## [1] "RR3 M1 for 2015 complete. "
## [1] "RR3 M1 for 2015 complete. "
## [1] "M2 for 2015 complete. "
## [1] "Year 2015 loop finished\n"
## [1] "MainSQL for 2016 complete. "
## [1] "Tertiary Metadata for 2016 complete. "
## [1] "High School Metadata for 2016 complete. "
## [1] "M1 for 2016 complete. "
## [1] "RR3 M1 for 2016 complete. "
## [1] "RR3 M1 for 2016 complete. "
## [1] "M2 for 2016 complete. "
## [1] "Year 2016 loop finished\n"
```