

Data Processing - Visits & QR

John F. Adamek

2020-11-03

Contents

Chapter 1

Data Processing - QR Data

Downloading Raw files from Qualtrics

1. Downloaded the raw data files from Qualtrics:
 - Projects -> “Group Class Check-In” -> Data & Analysis -> Export & Import -> Export Data -> CSV + Download all fields + Use Choice text
 - Saved as: Qualtrics_raw_without_other_option.csv
 - Projects -> “Group Class Check-In – Updated with “other” option 7/31/17” -> Data & Analysis -> Export & Import -> Export Data -> CSV Download all fields + Use Choice text
 - Saved as: Qualtrics_raw_with_other_option.csv
 - Both files were combined into one file separate sheets
 - Saved as: QR_raw_combined_data_separate_sheets.xlsx
 - Files were then combined into one sheet to be used for data cleaning.
 - A “Run” column was created of running numbers from 1 to keep track of each individual row and to merge data on in R in later steps.
 - Saved as: Qualtrics_combined_rawdata.xlsx
2. Merged Class column and Class-other option column into 1 in excel
 - Filtered “Class” column by other. Replaced other with the class indicated.
 - N = 1067 total instances
 - Saved as: Qualtrics_cleaning.xlsx

3. Cleaned extreme outliers in “Today’sDate” that were errors (e.g., 5/28/1017).
 - Start, End, Recorded Date were used as a basis.
 - All month and days were consistent across the three columns. Example:
 - Saved in: Qualtrics_cleaning.xlsx

Data Prep in R

4. **QR_step1.R:** Loaded Qualtrics_cleaning.xlsx in R for data cleaning:
 - Separated time from date into its own column
 - Identified all instances where “recorded date” was before “todaysdate” and replaced todays date with the recorded date (n=135)
 - LOGIC: Since it is highly unlikely people were logging in their class attendance for future dates, its expected that this was a typing error when manually typing “todays date” thus recorded date was determined to be the actual date of class attendance.
 - Identified all instances where “recorded date” was after “todaysdate”. (N=845)
 - Saved as: rafter.csv
 - Two issues to consider:
 - 1) If MONTH and DAY are the same for start date, end date, recorded date and todays date but YEAR is different in todays date = year was changed to reflect recorded year n = 228
 - 2) Participants were allowed to report PAST class attendances resulting in instances where there are multiple (as much as 8) logs on a single day (illustrated via start,end,recorded date) with the actual date being manually reported in todays date. All instances that aren’t satisfied via the first issue above were individually checked and corrected.
 - * Saved as: rafter_clean.csv
 - * See <- recording_ofme_cleaning_rafter.pptx for more info
 - * Because occurrences part of issue 2 would result in TodaysDate remaining the same. No changes were made here
 - Returned back to R to pick up where we left off before bullet point c.
 - **** See r file: QR_step1.R
 - Saved as: QR_cleaned_07312020.csv

5. Opened QR_cleaned_07312020.csv file outside of R

- In this excel file – created a new column “CorrectDate”
- Then condensed Today’s Date, r_before, and r_after over to “CorrectDate”
 - Filtered “r_after” column by valid dates only(no NAs) -> checked one last time accuracy and then merged to “CorrectDate” column.
 - Filtered “r_before” column by valid dates only (no NAs) -> checked one last time accuracy and then merged to “CorrectDate” column.
 - Filtered “TodaysDate” column by valid dates only -> checked one last time accuracy and then merged to “CorrectDate” column.
 - Filtered “CorrectDate” column by BLANKS -> checked to make sure r_after and r_before columns were all “NA” and then merged “TodaysDate” to “CorrectDate”
 - No issues with the above.
 - This process was checked at the end to ensure no overlap between r_before and r_after columns with CorrectDate
- Saved as: QR_CorrectDate.csv

```
##### Pulling in the data #####
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/QR_Cleaning_Processing_Files")
data <- read_excel("Qualtrics_cleaning.xlsx")
library(tidyr)
##### Formatting and Preparing the data #####
# Separates The column by data and time
data <- separate(data, StartDate, into=c("sdate","stime"), sep = " ")
data <- separate(data, EndDate, into=c("Edate","Etime"), sep = " ")
data <- separate(data, RecordedDate, into=c("Rdate","Rtime"), sep = " ")

# Removes the time columns
data <- data[-c(5,7,9)]

# Converts class type from chr to date
data$sdate <- gsub("-", "", data$sdate)
data$Edate <- gsub("-", "", data$Edate)
data$Rdate <- gsub("-", "", data$Rdate)
data$TodaysDate <- gsub("-", "", data$TodaysDate)
data$sdate <- as.Date(data$sdate, format="%Y%m%d")
data$Edate <- as.Date(data$Edate, format="%Y%m%d")
data$Rdate <- as.Date(data$Rdate, format="%Y%m%d")
data$TodaysDate <- as.Date(data$TodaysDate, format="%Y%m%d")

##### Cleaning the Data for instances where Today's Date and Recorded Date does not match #####

# Identifies which Recorded dates are before Todays dates and changes Todays Date to Recorded Date
R_before <- with(data, data[Rdate < TodaysDate, ])
R_before$TodaysDate <- R_before$Rdate
```

```

# Identifies which Recorded dates are before Todays dates
R_after <- with(data, data[Rdate > TodaysDate, ])
# Saves this file to csv. to manually check for whether these were errors in typing the
write.csv(R_after, "rafter.csv")
# Imports the cleaned rafter file
R_after_cl <- read.csv("rafter_clean.csv")

# This matches the data on RUN and merges data from R_before and R_after_cl into main
data$r_before <- R_before$TodaysDate[match(data$Run, R_before$Run)]
data$r_after <- R_after_cl$TodaysDate[match(data$Run, R_after_cl$Run)]

# Writes the data to file
write.csv(data, "QR_cleaned_07312020.csv")

##### End of Step 1: Notes to be done before opening QR_step2.R #####
# After this process I opened the new csv file to ensure accuracy by checking the column
# The changed dates were then merged into a final CorrectDate column.
# When this is completed I continued data cleaning in QR_ID_cleaning

```

6. **QR_step2.R:** This csv file (QR_CorrectDate.csv) was then loaded into R:

- Dates were separated by month, day, and year and then tallied for all 12 months
- Afterwards the IDs were assigned to their appropriate wave
- After running this function from the total dataset (obs = 7276) each wave had:
 - Wave 1 = 1841 obs
 - Wave 2 = 1704 obs -> 1705 after correcting (see below)
 - Wave 3 = 1389 obs
 - Wave 4 = 518 obs
 - Wave 5 = 811 obs -> 812 after correcting (see below)
 - Wave 6 = 1003 obs -> 1004 after correcting (see below)
 - This totals 7266obs
 - The missing 10 observations were:
 1. ID 593 = 3x *IPaddress not found
 2. ID 365 = 2x *IPaddress universal
 3. ID 352 = 1x *IPaddress universal
 4. ID 346 = 1x *Determined to be ID342 after looking up IPaddress
 5. ID 286 = 1x *Determined to be ID296 after looking up IPaddress
 6. ID 270 = 1x *Determined to be ID170 after looking up IPaddress
 7. ID 203 = 1x *IPaddress not found
- Each wave (with combined IDs) was then saved into their folder (e.g., Wave1) as a csv (e.g., wave1_qr.csv)

- Each Individual ID was also saved into their Wave folder (e.g ID259.csv)
7. There were instances where participants kept logging their class attendance after the 12-month frame (e.g., wave 1, ID101 logged class attended on 3/4/2018 – but wave1 ended 1/31/2018).
- Each wave’s combined file was opened (e.g., wave1_qr.csv) and all data AFTER the 12-month dataframe was removed.
 - This was then saved with “clean” at the end (e.g., wave1_qr_clean.csv) so that the original file was not modified which contained the data outside of the 12-month time frame
 - Wave 1 = 40 instances
 - ID’s: 101, 128, 132, 134, 141, ,143, 158
 - Wave 2 = 5 instances (ID204 was determined to be ID294[wave5] based on date, IPAddress, and class type)
 - ID’s 156, 172, 183
 - Wave 3 = 11 instances (ID232 was changed from 11/7/2017 to 11/7/2018)
 - ID’s: 215, 217, 221, 223, 243, 244
 - Wave 4 = 5 instances
 - ID’s: 259, 288
 - Wave 5 = 3 instances (ID 306 was changed from 3/23/2018 to 3/26/2019)
 - ID: 296
 - Wave 6 = 85 instances (April&May removed due to refinery being closed)
 - *Due to COVID-19 many participants logged classes they attended at home, yard work, neighborhood walking, etc.
 - ID326 changed from 6/3/2018 to 6/5/2019
 - ID342 changed from 9/19/2018 to 9/20/2019
 - *BEFORE ANY REMOVING, DATA WAS VERIFIED WITH RAW DATAFILE (QR_raw_combined_data_separate_sheets.xlsx)

```
##### Libraries #####
library(tidyr)
library(haven)
library(dplyr)

##### Pulling in the data ####
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/QR_Cleaning_Processing_Files")
data <- read.csv("QR_CorrectDate.csv")
```

```

data <- data[c(1,2,3,10)]

##### Formatting and Preparing the data #####

# This uses the separate function to separate the date by month, day, and year
new = separate(data, col = CorrectDate, into = c("Month", "Day", "Year"), sep = "/")

# This tallys their attendance (via class attendance) per month
new$feb <- ifelse(new$Month == 2, 1, 0)
new$mar <- ifelse(new$Month == 3, 1, 0)
new$apr <- ifelse(new$Month == 4, 1, 0)
new$may <- ifelse(new$Month == 5, 1, 0)
new$june <- ifelse(new$Month == 6, 1, 0)
new$july <- ifelse(new$Month == 7, 1, 0)
new$aug <- ifelse(new$Month == 8, 1, 0)
new$sep <- ifelse(new$Month == 9, 1, 0)
new$oct <- ifelse(new$Month == 10, 1, 0)
new$nov <- ifelse(new$Month == 11, 1, 0)
new$dec <- ifelse(new$Month == 12, 1, 0)
new$jan <- ifelse(new$Month == 1, 1, 0)

# ID's filtered by Waves
wave1 <- c(100, 101, 102, 103, 104, 105, 106, 107, 108, 109,
          124, 125, 126, 127, 128, 129, 131, 132, 134, 135,
          157, 158, 161, 162)
wave2 <- c(133, 138, 144, 149, 154, 156, 159, 160, 167, 168, 169, 170, 171,
          183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
          207, 208, 209, 210, 211, 212, 213, 215, 216, 217, 218, 219, 220,
          232, 233, 234, 237, 238, 239, 240, 241, 242, 243, 244, 245, 247,
          258)
wave4 <- c(214, 259, 260, 261, 262, 263, 264, 265, 266, 267,
          284, 285, 287, 288, 289, 291)
wave5 <- c(292, 293, 294, 295, 296, 297, 298, 299, 300, 301,
          316, 317, 318, 319, 321, 322, 323, 324, 325, 326,
          342, 343, 344, 345, 347, 348, 349, 350, 351, 353, 354)

# Separating ID's into their appropriate waves and saves them as raw files by waves
wave1_qr <- new[new$ID %in% wave1,]
wave2_qr <- new[new$ID %in% wave2,]
wave3_qr <- new[new$ID %in% wave3,]
wave4_qr <- new[new$ID %in% wave4,]
wave5_qr <- new[new$ID %in% wave5,]
wave6_qr <- new[new$ID %in% wave6,]

write.csv(wave1_qr, "wave1_qr.csv")

```

```

write.csv(wave2_qr, "wave2_qr.csv")
write.csv(wave3_qr, "wave3_qr.csv")
write.csv(wave4_qr, "wave4_qr.csv")
write.csv(wave5_qr, "wave5_qr.csv")
write.csv(wave6_qr, "wave6_qr.csv")

##### Creating separate files for each ID #####

#ID100 <- subset(data, ID == 100)
ID101 <- subset(new, ID == 101)
ID102 <- subset(new, ID == 102)
ID103 <- subset(new, ID == 103)
ID104 <- subset(new, ID == 104)
ID105 <- subset(new, ID == 105)
ID106 <- subset(new, ID == 106)
ID107 <- subset(new, ID == 107)
ID108 <- subset(new, ID == 108)
ID109 <- subset(new, ID == 109)
ID110 <- subset(new, ID == 110)
ID111 <- subset(new, ID == 111)
ID112 <- subset(new, ID == 112)
ID113 <- subset(new, ID == 113)
ID114 <- subset(new, ID == 114)
ID115 <- subset(new, ID == 115)
ID116 <- subset(new, ID == 116)
ID117 <- subset(new, ID == 117)
ID118 <- subset(new, ID == 118)

ID120 <- subset(new, ID == 120)
ID121 <- subset(new, ID == 121)
ID122 <- subset(new, ID == 122)
ID123 <- subset(new, ID == 123)
ID124 <- subset(new, ID == 124)
ID125 <- subset(new, ID == 125)
ID126 <- subset(new, ID == 126)
ID127 <- subset(new, ID == 127)
ID128 <- subset(new, ID == 128)
ID129 <- subset(new, ID == 129)

ID131 <- subset(new, ID == 131)
ID132 <- subset(new, ID == 132)
ID133 <- subset(new, ID == 133)
ID134 <- subset(new, ID == 134)
ID135 <- subset(new, ID == 135)
ID136 <- subset(new, ID == 136)

```

```
ID137 <- subset(new, ID == 137)
#ID138 <- subset(data, ID == 138)
ID139 <- subset(new, ID == 139)

ID141 <- subset(new, ID == 141)
ID142 <- subset(new, ID == 142)
ID143 <- subset(new, ID == 143)
ID144 <- subset(new, ID == 144)

ID146 <- subset(new, ID == 146)
ID147 <- subset(new, ID == 147)

ID149 <- subset(new, ID == 149)
#ID150 <- subset(data, ID == 150)
ID151 <- subset(new, ID == 151)
ID152 <- subset(new, ID == 152)
ID153 <- subset(new, ID == 153)
ID154 <- subset(new, ID == 154)
ID155 <- subset(new, ID == 155)
ID156 <- subset(new, ID == 156)
ID157 <- subset(new, ID == 157)
ID158 <- subset(new, ID == 158)
ID159 <- subset(new, ID == 159)
#ID160 <- subset(data, ID == 160)
ID161 <- subset(new, ID == 161)
ID162 <- subset(new, ID == 162)

ID167 <- subset(new, ID == 167)
#ID168 <- subset(new, ID == 168)
ID169 <- subset(new, ID == 169)
ID170 <- subset(new, ID == 170)
#ID171 <- subset(new, ID == 171)
ID172 <- subset(new, ID == 172)
ID173 <- subset(new, ID == 173)
ID174 <- subset(new, ID == 174)
ID175 <- subset(new, ID == 175)
ID176 <- subset(new, ID == 176)
ID177 <- subset(new, ID == 177)
ID178 <- subset(new, ID == 178)
ID179 <- subset(new, ID == 179)

ID181 <- subset(new, ID == 181)
ID182 <- subset(new, ID == 182)
ID183 <- subset(new, ID == 183)
```

```

#ID184 <- subset(new, ID == 184)
ID185 <- subset(new, ID == 185)
ID186 <- subset(new, ID == 186)
ID187 <- subset(new, ID == 187)
ID188 <- subset(new, ID == 188)
ID189 <- subset(new, ID == 189)
ID190 <- subset(new, ID == 190)
ID191 <- subset(new, ID == 191)
ID192 <- subset(new, ID == 192)
ID193 <- subset(new, ID == 193)
ID194 <- subset(new, ID == 194)
ID195 <- subset(new, ID == 195)
ID196 <- subset(new, ID == 196)
ID197 <- subset(new, ID == 197)
ID198 <- subset(new, ID == 198)
ID199 <- subset(new, ID == 199)
ID200 <- subset(new, ID == 200)
ID201 <- subset(new, ID == 201)
ID202 <- subset(new, ID == 202)

ID204 <- subset(new, ID == 204)
ID205 <- subset(new, ID == 205)
ID206 <- subset(new, ID == 206)
ID207 <- subset(new, ID == 207)
#ID208 <- subset(new, ID == 208)
ID209 <- subset(new, ID == 209)
ID210 <- subset(new, ID == 210)
#ID211 <- subset(new, ID == 211)
ID212 <- subset(new, ID == 212)
#ID213 <- subset(new, ID == 213)
#ID214 <- subset(new, ID == 214)
ID215 <- subset(new, ID == 215)
ID216 <- subset(new, ID == 216)
ID217 <- subset(new, ID == 217)
ID218 <- subset(new, ID == 218)
ID219 <- subset(new, ID == 219)
#ID220 <- subset(new, ID == 220)
ID221 <- subset(new, ID == 221)
#ID222 <- subset(new, ID == 222)
ID223 <- subset(new, ID == 223)
ID224 <- subset(new, ID == 224)
ID225 <- subset(new, ID == 225)
ID226 <- subset(new, ID == 226)
ID227 <- subset(new, ID == 227)

```

```
ID229 <- subset(new, ID == 229)
ID230 <- subset(new, ID == 230)
ID231 <- subset(new, ID == 231)
ID232 <- subset(new, ID == 232)
ID233 <- subset(new, ID == 233)
ID234 <- subset(new, ID == 234)

ID237 <- subset(new, ID == 237)
#ID238 <- subset(data, ID == 238)
ID239 <- subset(new, ID == 239)
ID240 <- subset(new, ID == 240)
ID241 <- subset(new, ID == 241)
ID242 <- subset(new, ID == 242)
ID243 <- subset(new, ID == 243)
ID244 <- subset(new, ID == 244)
ID245 <- subset(new, ID == 245)

ID247 <- subset(new, ID == 247)
ID248 <- subset(new, ID == 248)
#ID249 <- subset(data, ID == 249)
ID250 <- subset(new, ID == 250)
#ID251 <- subset(data, ID == 251)
ID252 <- subset(new, ID == 252)
ID253 <- subset(new, ID == 253)
ID254 <- subset(new, ID == 254)
ID255 <- subset(new, ID == 255)
#ID256 <- subset(new, ID == 256)
ID257 <- subset(new, ID == 257)
ID258 <- subset(new, ID == 258)
ID259 <- subset(new, ID == 259)
ID260 <- subset(new, ID == 260)
#ID261 <- subset(new, ID == 261)
ID262 <- subset(new, ID == 262)
ID263 <- subset(new, ID == 263)
#ID264 <- subset(new, ID == 264)
ID265 <- subset(new, ID == 265)
#ID266 <- subset(new, ID == 266)
#ID267 <- subset(new, ID == 267)
ID268 <- subset(new, ID == 268)
ID269 <- subset(new, ID == 269)

ID271 <- subset(new, ID == 271)

ID273 <- subset(new, ID == 273)
#ID274 <- subset(new, ID == 274)
```

```

ID275 <- subset(new, ID == 275)

ID277 <- subset(new, ID == 277)
ID278 <- subset(new, ID == 278)
ID279 <- subset(new, ID == 279)
ID280 <- subset(new, ID == 280)
#ID281 <- subset(new, ID == 281)
#ID282 <- subset(new, ID == 282)
ID283 <- subset(new, ID == 283)
ID284 <- subset(new, ID == 284)
ID285 <- subset(new, ID == 285)

ID287 <- subset(new, ID == 287)
ID288 <- subset(new, ID == 288)
ID289 <- subset(new, ID == 289)

ID291 <- subset(new, ID == 291)
#ID292 <- subset(new, ID == 292)
ID293 <- subset(new, ID == 293)
ID294 <- subset(new, ID == 294)
ID295 <- subset(new, ID == 295)
ID296 <- subset(new, ID == 296)
ID297 <- subset(new, ID == 297)
ID298 <- subset(new, ID == 298)
ID299 <- subset(new, ID == 299)
ID300 <- subset(new, ID == 300)
ID301 <- subset(new, ID == 301)
ID302 <- subset(new, ID == 302)
ID303 <- subset(new, ID == 303)
ID304 <- subset(new, ID == 304)
ID305 <- subset(new, ID == 305)
ID306 <- subset(new, ID == 306)
ID307 <- subset(new, ID == 307)
#ID308 <- subset(new, ID == 308)
#ID309 <- subset(new, ID == 309)
ID310 <- subset(new, ID == 310)
ID311 <- subset(new, ID == 311)
#ID312 <- subset(new, ID == 312)
#ID313 <- subset(new, ID == 313)
#ID314 <- subset(new, ID == 314)

ID316 <- subset(new, ID == 316)
ID317 <- subset(new, ID == 317)
#ID318 <- subset(new, ID == 318)
ID319 <- subset(new, ID == 319)

```

```

ID321 <- subset(new, ID == 321)
ID322 <- subset(new, ID == 322)
#ID323 <- subset(new, ID == 323)
ID324 <- subset(new, ID == 324)
ID325 <- subset(new, ID == 325)
ID326 <- subset(new, ID == 326)
ID327 <- subset(new, ID == 327)
ID328 <- subset(new, ID == 328)
ID329 <- subset(new, ID == 329)
ID330 <- subset(new, ID == 330)
ID331 <- subset(new, ID == 331)
ID332 <- subset(new, ID == 332)
ID333 <- subset(new, ID == 333)
#ID334 <- subset(new, ID == 334)
#ID335 <- subset(new, ID == 335)
#ID336 <- subset(new, ID == 336)

ID338 <- subset(new, ID == 338)
#ID339 <- subset(new, ID == 339)

ID341 <- subset(new, ID == 341)
ID342 <- subset(new, ID == 342)
ID343 <- subset(new, ID == 343)
ID344 <- subset(new, ID == 344)
#ID345 <- subset(new, ID == 345)

#ID347 <- subset(new, ID == 347)
ID348 <- subset(new, ID == 348)
ID349 <- subset(new, ID == 349)
ID350 <- subset(new, ID == 350)
#ID351 <- subset(new, ID == 351)

ID353 <- subset(new, ID == 353)
ID354 <- subset(new, ID == 354)
ID355 <- subset(new, ID == 355)
ID356 <- subset(new, ID == 356)
ID357 <- subset(new, ID == 357)
#ID358 <- subset(new, ID == 358)

##### Saving individual ID's to csv file AND folder #####
write.csv(ID101, "ID101.csv")
write.csv(ID102, "ID102.csv")
write.csv(ID103, "ID103.csv")
write.csv(ID104, "ID104.csv")
write.csv(ID105, "ID105.csv")

```



```
write.csv(ID106, "ID106.csv")
write.csv(ID107, "ID107.csv")
write.csv(ID108, "ID108.csv")
write.csv(ID109, "ID109.csv")
write.csv(ID110, "ID110.csv")
write.csv(ID111, "ID111.csv")
write.csv(ID112, "ID112.csv")
write.csv(ID113, "ID113.csv")
write.csv(ID114, "ID114.csv")
write.csv(ID115, "ID115.csv")
write.csv(ID116, "ID116.csv")
write.csv(ID117, "ID117.csv")
write.csv(ID118, "ID118.csv")
write.csv(ID120, "ID120.csv")
write.csv(ID121, "ID121.csv")
write.csv(ID122, "ID122.csv")
write.csv(ID123, "ID123.csv")
write.csv(ID124, "ID124.csv")
write.csv(ID125, "ID125.csv")
write.csv(ID126, "ID126.csv")
write.csv(ID127, "ID127.csv")
write.csv(ID128, "ID128.csv")
write.csv(ID129, "ID129.csv")
write.csv(ID131, "ID131.csv")
write.csv(ID132, "ID132.csv")
write.csv(ID133, "ID133.csv")
write.csv(ID134, "ID134.csv")
write.csv(ID135, "ID135.csv")
write.csv(ID136, "ID136.csv")
write.csv(ID137, "ID137.csv")
write.csv(ID139, "ID139.csv")
write.csv(ID141, "ID141.csv")
write.csv(ID142, "ID142.csv")
write.csv(ID143, "ID143.csv")
write.csv(ID144, "ID144.csv")
write.csv(ID146, "ID146.csv")
write.csv(ID147, "ID147.csv")
write.csv(ID149, "ID149.csv")
write.csv(ID151, "ID151.csv")
write.csv(ID152, "ID152.csv")
write.csv(ID154, "ID153.csv")
write.csv(ID154, "ID154.csv")
write.csv(ID155, "ID155.csv")
write.csv(ID156, "ID156.csv")
write.csv(ID157, "ID157.csv")
```

```
write.csv(ID158, "ID158.csv")
write.csv(ID159, "ID159.csv")
write.csv(ID161, "ID161.csv")
write.csv(ID162, "ID162.csv")
write.csv(ID167, "ID167.csv")
write.csv(ID169, "ID169.csv")
write.csv(ID170, "ID170.csv")
write.csv(ID172, "ID172.csv")
write.csv(ID173, "ID173.csv")
write.csv(ID174, "ID174.csv")
write.csv(ID175, "ID175.csv")
write.csv(ID176, "ID176.csv")
write.csv(ID177, "ID177.csv")
write.csv(ID178, "ID178.csv")
write.csv(ID179, "ID179.csv")
write.csv(ID181, "ID181.csv")
write.csv(ID182, "ID182.csv")
write.csv(ID183, "ID183.csv")

write.csv(ID185, "ID185.csv")
write.csv(ID186, "ID186.csv")
write.csv(ID187, "ID187.csv")
write.csv(ID188, "ID188.csv")
write.csv(ID189, "ID189.csv")
write.csv(ID190, "ID190.csv")
write.csv(ID191, "ID191.csv")
write.csv(ID192, "ID192.csv")
write.csv(ID193, "ID193.csv")
write.csv(ID194, "ID194.csv")
write.csv(ID195, "ID195.csv")
write.csv(ID196, "ID196.csv")
write.csv(ID197, "ID197.csv")
write.csv(ID198, "ID198.csv")
write.csv(ID199, "ID199.csv")
write.csv(ID200, "ID200.csv")
write.csv(ID201, "ID201.csv")
write.csv(ID202, "ID202.csv")
write.csv(ID204, "ID204.csv")
write.csv(ID205, "ID205.csv")
write.csv(ID206, "ID206.csv")
write.csv(ID207, "ID207.csv")
write.csv(ID209, "ID209.csv")
write.csv(ID210, "ID210.csv")
write.csv(ID212, "ID212.csv")
write.csv(ID215, "ID215.csv")
```

```
write.csv(ID216, "ID216.csv")
write.csv(ID217, "ID217.csv")
write.csv(ID218, "ID218.csv")
write.csv(ID219, "ID219.csv")
write.csv(ID221, "ID221.csv")
write.csv(ID223, "ID223.csv")
write.csv(ID224, "ID224.csv")
write.csv(ID225, "ID225.csv")
write.csv(ID226, "ID226.csv")
write.csv(ID227, "ID227.csv")
write.csv(ID229, "ID229.csv")
write.csv(ID230, "ID230.csv")
write.csv(ID231, "ID231.csv")
write.csv(ID232, "ID232.csv")
write.csv(ID233, "ID233.csv")
write.csv(ID234, "ID234.csv")
write.csv(ID237, "ID237.csv")
write.csv(ID239, "ID239.csv")
write.csv(ID240, "ID240.csv")
write.csv(ID241, "ID241.csv")
write.csv(ID242, "ID242.csv")
write.csv(ID243, "ID243.csv")
write.csv(ID244, "ID244.csv")
write.csv(ID245, "ID245.csv")
write.csv(ID247, "ID247.csv")
write.csv(ID248, "ID248.csv")
write.csv(ID250, "ID250.csv")
write.csv(ID252, "ID252.csv")
write.csv(ID253, "ID253.csv")
write.csv(ID254, "ID254.csv")
write.csv(ID255, "ID255.csv")
write.csv(ID257, "ID257.csv")
write.csv(ID258, "ID258.csv")
write.csv(ID259, "ID259.csv")
write.csv(ID260, "ID260.csv")
write.csv(ID262, "ID262.csv")
write.csv(ID263, "ID263.csv")
write.csv(ID265, "ID265.csv")
write.csv(ID268, "ID268.csv")
write.csv(ID269, "ID269.csv")
write.csv(ID271, "ID271.csv")
write.csv(ID273, "ID273.csv")
write.csv(ID275, "ID275.csv")
write.csv(ID277, "ID277.csv")
write.csv(ID278, "ID278.csv")
```

```
write.csv(ID279, "ID279.csv")
write.csv(ID280, "ID280.csv")
write.csv(ID283, "ID283.csv")
write.csv(ID284, "ID284.csv")
write.csv(ID285, "ID285.csv")
write.csv(ID287, "ID287.csv")
write.csv(ID288, "ID288.csv")
write.csv(ID289, "ID289.csv")
write.csv(ID291, "ID291.csv")
write.csv(ID293, "ID293.csv")
write.csv(ID294, "ID294.csv")
write.csv(ID295, "ID295.csv")
write.csv(ID296, "ID296.csv")
write.csv(ID297, "ID297.csv")
write.csv(ID298, "ID298.csv")
write.csv(ID299, "ID299.csv")
write.csv(ID300, "ID300.csv")
write.csv(ID301, "ID301.csv")
write.csv(ID302, "ID302.csv")
write.csv(ID303, "ID303.csv")
write.csv(ID304, "ID304.csv")
write.csv(ID305, "ID305.csv")
write.csv(ID306, "ID306.csv")
write.csv(ID307, "ID307.csv")
write.csv(ID310, "ID310.csv")
write.csv(ID311, "ID311.csv")
write.csv(ID316, "ID316.csv")
write.csv(ID317, "ID317.csv")
write.csv(ID319, "ID319.csv")
write.csv(ID321, "ID321.csv")
write.csv(ID322, "ID322.csv")
write.csv(ID324, "ID324.csv")
write.csv(ID325, "ID325.csv")
write.csv(ID326, "ID326.csv")
write.csv(ID327, "ID327.csv")
write.csv(ID328, "ID328.csv")
write.csv(ID329, "ID329.csv")
write.csv(ID330, "ID330.csv")
write.csv(ID331, "ID331.csv")
write.csv(ID332, "ID332.csv")
write.csv(ID333, "ID333.csv")
write.csv(ID338, "ID338.csv")
write.csv(ID341, "ID341.csv")
write.csv(ID342, "ID342.csv")
write.csv(ID343, "ID343.csv")
```

```

write.csv(ID344, "ID344.csv")
write.csv(ID348, "ID348.csv")
write.csv(ID349, "ID349.csv")
write.csv(ID350, "ID350.csv")
write.csv(ID353, "ID353.csv")
write.csv(ID354, "ID354.csv")
write.csv(ID355, "ID355.csv")
write.csv(ID356, "ID356.csv")
write.csv(ID357, "ID357.csv")

```

End of Step 2: Notes to be done before opening QR_step3.R

*# Next step: Opened each ID's csv file to check for accuracy and ensure data is within the 12-months
going to the gym after the studies 12-months and future dates needed to be removed*

8. **QR_step3.R:** Each waves clean csv file (e.g., wave1_qr_clean.csv) was loaded into r

- Data was aggregated and summed so that the resulting file included a single row of each ID with each months class attendance summed for that particular ID
- A Total class attendance column was created at the end
- “Wave” column was added
- Names of month was changed from jan, feb ... into month 1 month 2 ...
 - This was different for even and odd months – obviously
 - Order of months/columns were reordered
- Results were then saved for each individual wave as: w1_final.csv
- In this file some “visualization” was done for plotting – though this is insignificant for the data processing steps.

```

# Aggregating and formating data #
##### Pulling in the data #####
# Get wave 1 data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave1")
w1_data <- read.csv("wave1_qr_clean.csv")
# Get wave 2 data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave2")
w2_data <- read.csv("wave2_qr_clean.csv")
# Get wave 3 data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave3")
w3_data <- read.csv("wave3_qr_clean.csv")
# Get wave 4 data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave4")

```

```

w4_data <- read.csv("wave4_qr_clean.csv")
# Get wave 5 data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave5")
w5_data <- read.csv("wave5_qr_clean.csv")
# Get wave 6 data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave6")
w6_data <- read.csv("wave6_qr_clean.csv")

##### Preparing and Processing the data to aggregate #####

# The Below codes aggregates each months class attendance by ID. Final variable includ
w1_final <- w1_data[-c(1,3,4,5,6)] # Makes dataframe of just
w1_final <- aggregate(~ID, w1_final, sum) # aggregates total for each m
w1_final$Total_YTD <- rowSums(w1_final[-1]) # This [-1] is to make sure the

w2_final <- w2_data[-c(1,3,4,5,6)]
w2_final <- aggregate(~ID, w2_final, sum)
w2_final$Total_YTD <- rowSums(w2_final[-1])

w3_final <- w3_data[-c(1,3,4,5,6)]
w3_final <- aggregate(~ID, w3_final, sum)
w3_final$Total_YTD <- rowSums(w3_final[-1])

w4_final <- w4_data[-c(1,3,4,5,6)]
w4_final <- aggregate(~ID, w4_final, sum)
w4_final$Total_YTD <- rowSums(w4_final[-1])

w5_final <- w5_data[-c(1,3,4,5,6)]
w5_final <- aggregate(~ID, w5_final, sum)
w5_final$Total_YTD <- rowSums(w5_final[-1])

w6_final <- w6_data[-c(1,3,4,5,6)]
w6_final <- aggregate(~ID, w6_final, sum)
w6_final$Total_YTD <- rowSums(w6_final[-1])

# Adds column indicating wave
w1_final$wave <- rep(1, nrow(w1_final))
w2_final$wave <- rep(2, nrow(w2_final))
w3_final$wave <- rep(3, nrow(w3_final))
w4_final$wave <- rep(4, nrow(w4_final))
w5_final$wave <- rep(5, nrow(w5_final))
w6_final$wave <- rep(6, nrow(w6_final))

# Labels Month 1, month 2 ... for each wave
colnames(w1_final) <- c("ID", "Month1", "Month2", "Month3", "Month4", "Month5", "Month6")

```

```

colnames(w3_final) <- c("ID", "Month1", "Month2", "Month3", "Month4", "Month5", "Month6", "Month7", "Month8", "Month9", "Month10", "Month11", "Month12", "Month13", "Month14")
colnames(w5_final) <- c("ID", "Month1", "Month2", "Month3", "Month4", "Month5", "Month6", "Month7", "Month8", "Month9", "Month10", "Month11", "Month12", "Month13", "Month14")
colnames(w2_final) <- c("ID", "Month9", "Month10", "Month11", "Month12", "Month1", "Month2", "Month3", "Month4", "Month5", "Month6", "Month7", "Month8", "Month9", "Month10", "Month11", "Month12", "Month13", "Month14")
colnames(w4_final) <- c("ID", "Month9", "Month10", "Month11", "Month12", "Month1", "Month2", "Month3", "Month4", "Month5", "Month6", "Month7", "Month8", "Month9", "Month10", "Month11", "Month12", "Month13", "Month14")
colnames(w6_final) <- c("ID", "Month9", "Month10", "Month11", "Month12", "Month1", "Month2", "Month3", "Month4", "Month5", "Month6", "Month7", "Month8", "Month9", "Month10", "Month11", "Month12", "Month13", "Month14")

# Puts Waves 2,4,6 columns in correct order ("month1, month2...") and puts "wave" into the 2nd column
w1_final <- w1_final[c(1,15,2,3,4,5,6,7,8,9,10,11,12,13,14)]
w2_final <- w2_final[c(1,15,6,7,8,9,10,11,12,13,2,3,4,5,14)]
w3_final <- w3_final[c(1,15,2,3,4,5,6,7,8,9,10,11,12,13,14)]
w4_final <- w4_final[c(1,15,6,7,8,9,10,11,12,13,2,3,4,5,14)]
w5_final <- w5_final[c(1,15,2,3,4,5,6,7,8,9,10,11,12,13,14)]
w6_final <- w6_final[c(1,15,6,7,8,9,10,11,12,13,2,3,4,5,14)]

# Creates a master file and saves it
master <- rbind(w1_final, w2_final, w3_final, w4_final, w5_final, w6_final)
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data")
write.csv(master, "QR_master.csv")

# Writes final data as a csv.
write.csv(w1_final, "w1_final.csv")
write.csv(w2_final, "w2_final.csv")
write.csv(w3_final, "w3_final.csv")
write.csv(w4_final, "w4_final.csv")
write.csv(w5_final, "w5_final.csv")
write.csv(w6_final, "w6_final.csv")

##### Visualization #####

data <- master
# Taking the mean of each month by wave
m1 <- rbind(mean(data$Month1[data$wave == 1]), mean(data$Month1[data$wave == 2]), mean(data$Month1[data$wave == 3]), mean(data$Month1[data$wave == 4]), mean(data$Month1[data$wave == 5]), mean(data$Month1[data$wave == 6]), mean(data$Month1[data$wave == 7]), mean(data$Month1[data$wave == 8]), mean(data$Month1[data$wave == 9]), mean(data$Month1[data$wave == 10]), mean(data$Month1[data$wave == 11]), mean(data$Month1[data$wave == 12]), mean(data$Month1[data$wave == 13]), mean(data$Month1[data$wave == 14]))
m2 <- rbind(mean(data$Month2[data$wave == 1]), mean(data$Month2[data$wave == 2]), mean(data$Month2[data$wave == 3]), mean(data$Month2[data$wave == 4]), mean(data$Month2[data$wave == 5]), mean(data$Month2[data$wave == 6]), mean(data$Month2[data$wave == 7]), mean(data$Month2[data$wave == 8]), mean(data$Month2[data$wave == 9]), mean(data$Month2[data$wave == 10]), mean(data$Month2[data$wave == 11]), mean(data$Month2[data$wave == 12]), mean(data$Month2[data$wave == 13]), mean(data$Month2[data$wave == 14]))
m3 <- rbind(mean(data$Month3[data$wave == 1]), mean(data$Month3[data$wave == 2]), mean(data$Month3[data$wave == 3]), mean(data$Month3[data$wave == 4]), mean(data$Month3[data$wave == 5]), mean(data$Month3[data$wave == 6]), mean(data$Month3[data$wave == 7]), mean(data$Month3[data$wave == 8]), mean(data$Month3[data$wave == 9]), mean(data$Month3[data$wave == 10]), mean(data$Month3[data$wave == 11]), mean(data$Month3[data$wave == 12]), mean(data$Month3[data$wave == 13]), mean(data$Month3[data$wave == 14]))
m4 <- rbind(mean(data$Month4[data$wave == 1]), mean(data$Month4[data$wave == 2]), mean(data$Month4[data$wave == 3]), mean(data$Month4[data$wave == 4]), mean(data$Month4[data$wave == 5]), mean(data$Month4[data$wave == 6]), mean(data$Month4[data$wave == 7]), mean(data$Month4[data$wave == 8]), mean(data$Month4[data$wave == 9]), mean(data$Month4[data$wave == 10]), mean(data$Month4[data$wave == 11]), mean(data$Month4[data$wave == 12]), mean(data$Month4[data$wave == 13]), mean(data$Month4[data$wave == 14]))
m5 <- rbind(mean(data$Month5[data$wave == 1]), mean(data$Month5[data$wave == 2]), mean(data$Month5[data$wave == 3]), mean(data$Month5[data$wave == 4]), mean(data$Month5[data$wave == 5]), mean(data$Month5[data$wave == 6]), mean(data$Month5[data$wave == 7]), mean(data$Month5[data$wave == 8]), mean(data$Month5[data$wave == 9]), mean(data$Month5[data$wave == 10]), mean(data$Month5[data$wave == 11]), mean(data$Month5[data$wave == 12]), mean(data$Month5[data$wave == 13]), mean(data$Month5[data$wave == 14]))

```

```

m6 <- rbind(mean(data$Month6[data$wave == 1]), mean(data$Month6[data$wave == 2]), mean
            mean(data$Month6[data$wave == 4]), mean(data$Month6[data$wave == 5]), mean
m7 <- rbind(mean(data$Month7[data$wave == 1]), mean(data$Month7[data$wave == 2]), mean
            mean(data$Month7[data$wave == 4]), mean(data$Month7[data$wave == 5]), mean
m8 <- rbind(mean(data$Month8[data$wave == 1]), mean(data$Month8[data$wave == 2]), mean
            mean(data$Month8[data$wave == 4]), mean(data$Month8[data$wave == 5]), mean
m9 <- rbind(mean(data$Month9[data$wave == 1]), mean(data$Month9[data$wave == 2]), mean
            mean(data$Month9[data$wave == 4]), mean(data$Month9[data$wave == 5]), mean
m10 <- rbind(mean(data$Month10[data$wave == 1]), mean(data$Month10[data$wave == 2]), m
            mean(data$Month10[data$wave == 4]), mean(data$Month10[data$wave == 5]), m
m11 <- rbind(mean(data$Month11[data$wave == 1]), mean(data$Month11[data$wave == 2]), m
            mean(data$Month11[data$wave == 4]), mean(data$Month11[data$wave == 5]), m
m12 <- rbind(mean(data$Month12[data$wave == 1]), mean(data$Month12[data$wave == 2]), m
            mean(data$Month12[data$wave == 4]), mean(data$Month12[data$wave == 5]), m

# Prepare the variables to be plotted
wave <- cbind(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12)
colnames(wave) <- paste("Month", 1:12)
rownames(wave) <- paste("Wave", 1:6)
master <- master[, -c(1,2,15)]
monthAVGc <- colMeans(master, na.rm = TRUE)

# Plotting the main master plot
plot(monthAVGc,
     main = "Average CLASSES per Month",
     xlab = "Months",
     ylab = "Average vists",
     sub = "Across all waves",
     pch = 15,
     col = "blue")
legend("topright", legend=c("Average across waves"),
     text.col=c("black"), pch=c(15), col=c("blue"))

# Plotting the data
par(mfrow=c(2,3))
plot(wave[1,],
     main = "Wave 1: Average CLASSES per Month",
     xlab = "Months",
     ylab = "Average CLASSES",
     pch = 15,
     col = "black")
legend("topright", legend=c("January = M12"),
     text.col=c("blue"), pch=c(17), col=c("black"))
axis(1, seq(1,12,1))
plot(wave[3,],

```



```

    main = "Wave 3: Average CLASSES per Month",
    xlab = "Months",
    ylab = "Average CLASSES",
    pch = 15,
    col = "black")
legend("topright", legend=c("January = M12"),
      text.col=c("blue"), pch=c(17), col=c("black"))
axis(1, seq(1,12,1))
plot(wave[5,],
     main = "Wave 5: Average CLASSES per Month",
     xlab = "Months",
     ylab = "Average CLASSES",
     pch = 15,
     col = "black")
legend("topright", legend=c("January = M12"),
      text.col=c("blue"), pch=c(17), col=c("black"))
axis(1, seq(1,12,1))
plot(wave[2,],
     main = "Wave 2: Average CLASSES per Month",
     xlab = "Months",
     ylab = "Average CLASSES",
     pch = 15,
     col = "red")
legend("topright", legend=c("January = M8"),
      text.col=c("blue"), pch=c(17), col=c("red"))
axis(1, seq(1,12,1))
plot(wave[4,],
     main = "Wave 4: Average CLASSES per Month",
     xlab = "Months",
     ylab = "Average CLASSES",
     pch = 15,
     col = "red")
legend("topright", legend=c("January = M8"),
      text.col=c("blue"), pch=c(17), col=c("red"))
axis(1, seq(1,12,1))
plot(wave[6,],
     main = "Wave 6: Average CLASSES per Month",
     xlab = "Months",
     ylab = "Average CLASSES",
     pch = 15,
     col = "red")
legend("topright", legend=c("Covid: M11 & M12"),
      text.col=c("blue"), pch=c(17), col=c("red"))
axis(1, seq(1,12,1))

```

9. **QR_classes.R:** Each waves clean csv file (e.g., wave1_qr_clean.csv) was loaded into R
- “wave” column was added and names of month was changed -> see c. and d. above
 - SIDE NOTE: The reason for repeating these two steps again is that it made it easier to aggregate the data first, then performed c. and d. in QR_step3.R
 - In this step, we don’t want to work with the aggregated data so we’re pulling in the non-modified csv file (e.g., wave1_qr_clean.csv) and thus have to repeat those 2 data cleaning steps.
 - Waves were then bind together for a master file containing all 6 waves.
 - Saved as: QR_master_class.csv
 - Each individual wave was also saved as w1_class.csv
 - QR_master_class.csv: File was open externally and classes were then cleaned for the “other” responses using rules per SPM:
 - For those with “other” response – if they did NOT have a keyswipe that day give them a tallied variable that will be used in the future.
 - Neighborhood walking does not count UNLESS it was during COVID time
 - the unique responses for other may have some commonalities. So you can probably break them down
 - COVID time online classes should only list the class or online class.
 - This was all cleaned manually in excel. In total n =357 removed
 - **SEE “List of classes removed & recoded.xlsx” file to see what was removed and what was recoded
 - Saved as: QR_master_class_cleaned
 - QR_Classes.R: the class cleaned filed above was loaded back into R
 - Two files created: Class count by wave and class count overall
 - Saved as: Class_frequency_bywave.csv
 - Saved as: Class_frequency_overall.csv

```
##### Libraries #####
library(dplyr)

##### Side Task: Prepare separate file containing only classes #####
# Creates a file containing ID, month, day, year, class for each wave
## Pulls in each waves data
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave1")
w1_class <- read.csv("wave1_qr_clean.csv")
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave2")
w2_class <- read.csv("wave2_qr_clean.csv")
```

```

setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave3")
w3_class <- read.csv("wave3_qr_clean.csv")
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave4")
w4_class <- read.csv("wave4_qr_clean.csv")
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave5")
w5_class <- read.csv("wave5_qr_clean.csv")
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data/Wave6")
w6_class <- read.csv("wave6_qr_clean.csv")

# Adds column indicating wave
w1_class$wave <- rep(1, nrow(w1_class))
w2_class$wave <- rep(2, nrow(w2_class))
w3_class$wave <- rep(3, nrow(w3_class))
w4_class$wave <- rep(4, nrow(w4_class))
w5_class$wave <- rep(5, nrow(w5_class))
w6_class$wave <- rep(6, nrow(w6_class))

# Puts Waves 2,4,6 columns in correct order ("month1, month2...") and puts "wave" into the 2nd col
w1_class <- w1_class[c(1,2,19,3,4,5,6,7,8,9,10,11,12,13,14,18,16,17,18)]
w2_class <- w2_class[c(1,2,19,3,4,5,6,11,12,13,14,15,16,17,18,7,8,9,10)]
w3_class <- w3_class[c(1,2,19,3,4,5,6,7,8,9,10,11,12,13,14,18,16,17,18)]
w4_class <- w4_class[c(1,2,19,3,4,5,6,11,12,13,14,15,16,17,18,7,8,9,10)]
w5_class <- w5_class[c(1,2,19,3,4,5,6,7,8,9,10,11,12,13,14,18,16,17,18)]
w6_class <- w6_class[c(1,2,19,3,4,5,6,11,12,13,14,15,16,17,18,7,8,9,10)]

# Labels Month 1, month 2 ... for each wave
colnames(w1_class) <- c("run", "ID", "wave", "Class", "Month", "day", "year", "Month1", "Month2",
colnames(w3_class) <- c("run", "ID", "wave", "Class", "Month", "day", "year", "Month1", "Month2",
colnames(w5_class) <- c("run", "ID", "wave", "Class", "Month", "day", "year", "Month1", "Month2",
colnames(w2_class) <- c("run", "ID", "wave", "Class", "Month", "day", "year", "Month1", "Month2",
colnames(w4_class) <- c("run", "ID", "wave", "Class", "Month", "day", "year", "Month1", "Month2",
colnames(w6_class) <- c("run", "ID", "wave", "Class", "Month", "day", "year", "Month1", "Month2",

# Creates master dataframe of all waves and tallys them
masterclass <- rbind(w1_class, w2_class, w3_class, w4_class, w5_class, w6_class)
QR_count_bywave <- masterclass %>% count(wave, Class)

## Saves file to directory #####
setwd("C:/Users/jadamek2/Desktop/LAB_NUC/QR_data")
write.csv(w1_class, "w1_class.csv")
write.csv(w2_class, "w2_class.csv")
write.csv(w3_class, "w3_class.csv")
write.csv(w4_class, "w4_class.csv")

```

```

write.csv(w5_class, "w5_class.csv")
write.csv(w6_class, "w6_class.csv")
write.csv(masterclass, "QR_master_class.csv")

##### Cleaning the "other" class responses as per SPM rules (see document) #####
## File QR_master_class.csv was open externally and classes were cleaned and saved as:
## See QR_Class_Data_Processing.docx and List of classes removed & recoded.xlsx

##### Importing cleaned class file back in #####
setwd("S:/LAB_Documents/Visits_QR_data/QR_data/QR_Cleaned_Processed_Files")
class_clean <- read.csv("QR_master_class_cleaned.csv")

QR_count_bywave <- class_clean %>% count(wave, Class)
QR_count_overall <- class_clean %>% count(Class)
write.csv(QR_count_bywave, "Class_frequency_bywave.csv")
write.csv(QR_count_overall, "Class_frequency_overall.csv")

john <- data.frame(unique(class_clean$Class))
colnames(john) <- "Classes"
write.csv(john, "class_Descr.csv")

```

Sean's Note (7/28/20 FLAM): Sean "Don't worry about classes 4-hour splits"
 – its invalid with this data due to manual typing in. "The 4-hour rule is meant
 for the visits data"