EMIT\_Quarantine\_Main\_Cleaning.R

jbueno

Sat Apr 6 12:35:12 2019

# EMIT Main Quarantine Paper Data Curation - cleaning raw data to produce cleaned spreadsheets  
# Program Objective: Take the datasets identified as critical, clean them, and merge to form curated one or more curated datasets  
# Author: Jacob Bueno de Mesquita  
# Date: September 14 - October 25, 2018; December 2018; February, 2019  
# Summary: Overall, this script cleans data from 15 raw datafiles (including some workbooks) and produces a total of 11 cleaned spreadsheets that can then be bound together into an authoritative dataset for this analysis The 11 cleaned datasets are stored in:  
# "/Users/jbueno/Box Sync/EMIT/EMIT\_Data\_Analysis\_Jake/EMIT\_Quarantine/Curated Data/Cleaned Data  
  
#### Load required packages and set working directory ####  
library(tidyverse)  
library(RcppRoll)  
library(readxl)  
library(knitr)  
library(data.table)  
library(lubridate)  
  
setwd("/Users/jbueno/Box Sync/EMIT/EMIT\_Data\_Analysis\_Jake/EMIT\_Quarantine")  
  
sessionInfo()

## R version 3.5.1 (2018-07-02)  
## Platform: x86\_64-apple-darwin15.6.0 (64-bit)  
## Running under: macOS 10.14.4  
##   
## Matrix products: default  
## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/libBLAS.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] bindrcpp\_0.2.2 lubridate\_1.7.4 data.table\_1.11.8  
## [4] knitr\_1.20 readxl\_1.1.0 RcppRoll\_0.3.0   
## [7] forcats\_0.3.0 stringr\_1.3.1 dplyr\_0.7.7   
## [10] purrr\_0.2.5 readr\_1.1.1 tidyr\_0.8.2   
## [13] tibble\_1.4.2 ggplot2\_3.1.0 tidyverse\_1.2.1   
## [16] htmlTable\_1.12 rmarkdown\_1.10 markdown\_0.8   
##   
## loaded via a namespace (and not attached):  
## [1] tidyselect\_0.2.5 haven\_1.1.2 lattice\_0.20-38 colorspace\_1.3-2  
## [5] htmltools\_0.3.6 yaml\_2.2.0 utf8\_1.1.4 rlang\_0.3.0.1   
## [9] pillar\_1.3.0 glue\_1.3.0 withr\_2.1.2 modelr\_0.1.2   
## [13] bindr\_0.1.1 plyr\_1.8.4 munsell\_0.5.0 gtable\_0.2.0   
## [17] cellranger\_1.1.0 rvest\_0.3.2 htmlwidgets\_1.3 evaluate\_0.12   
## [21] fansi\_0.4.0 broom\_0.5.0 Rcpp\_0.12.19 scales\_1.0.0   
## [25] backports\_1.1.2 checkmate\_1.8.5 jsonlite\_1.5 hms\_0.4.2   
## [29] digest\_0.6.18 stringi\_1.2.4 grid\_3.5.1 rprojroot\_1.3-2   
## [33] cli\_1.0.1 tools\_3.5.1 magrittr\_1.5 lazyeval\_0.2.1   
## [37] crayon\_1.3.4 pkgconfig\_2.0.2 xml2\_1.2.0 assertthat\_0.2.0  
## [41] httr\_1.3.1 rstudioapi\_0.8 R6\_2.3.0 nlme\_3.1-137   
## [45] compiler\_3.5.1

#### Cleaning the 1st dataset (Final PCR Emit\_Results\_by\_cdc\_id - raw data file received 19-Aug-2014 with checksums.xlsx) ####  
# Storing as CDCpcr.csv in UK Quarantine Study Data and Notes/Curated Data/Cleaned Data  
  
#a <- read\_xlsx("Nottingham Files from Emails/CDC results/Final PCR Emit\_Results\_by\_cdc\_id - raw data file received 19-Aug-2014 with checksums.xlsx")  
#a <- a %>%  
 # select(-CDC\_ID\_NUMBER, -Subject\_ID, -Received\_Box, -Location, -`InfA\_Ct checksums`) %>%  
 # mutate(Study\_Day = parse\_number(Study\_Day)) %>%  
 # arrange(Randomisation\_number, Study\_Day, desc(Result\_Reported)) %>%  
 # group\_by(Randomisation\_number, Study\_Day) %>%  
 # filter(row\_number()==1) %>%  
 # ungroup() %>%   
 # separate(Randomisation\_number, into = c("Randomization", "SubjectID"), sep = c(1), remove = TRUE, convert = TRUE) %>%  
 # select(SubjectID, everything()) %>%  
 # rename(StudyDay = Study\_Day)  
  
#write.csv(a, "Curated Data/Cleaned Data/CDCpcr.csv")  
  
# The above snippet of code was, upon further examination, found to have some oddities that don't make sense with the current version of table1.  
# Upon additional review it looks like there is another PCR datafile from CDC which reports only InfA\_ct values but appears to make a selection of the original assay or the repeat instances where a sample was re-assayed with rt-qPCR. The CDC collaborators appear to have sorted through these.   
# Repeat assays one by one and made decisions based on all of the knowledge available to them at the time.   
# Replicating their decision criteria in code may be possible but may not match this final "September" version of the InfA data in the end.  
# Thus we will read in a new dataset for PCR and repeat cleaning and merge steps.  
# File to use is: "Emit\_Results\_By\_CDC\_ID sorted plus SDC 3-Sep-2014 with checksums.xls"  
  
a <- read\_xls("Quarantine Raw Data/Nottingham Files from Emails/CDC results/Emit\_Results\_By\_CDC\_ID sorted plus SDC 3-Sep-2014 with checksums.xls")  
a <- a %>%  
 select(Randomisation\_number, Q, Study\_Day, Day, InfA\_Ct) %>%  
 mutate(Study\_Day = parse\_number(Study\_Day)) %>%  
 filter(!is.na(Randomisation\_number) & Randomisation\_number != "missing" & Randomisation\_number != "Q1" & Randomisation\_number != "Q2" & Randomisation\_number != "Q3") %>%  
 mutate(Day = as.numeric(Day)) %>%  
 mutate(StudyDay = coalesce(Study\_Day, Day)) %>%  
 select(-Study\_Day, -Day) %>%  
 select(Randomisation\_number, Q, StudyDay, InfA\_Ct) %>%  
 separate(Randomisation\_number, into = c("Randomization", "SubjectID"), sep = c(1), remove = TRUE, convert = TRUE) %>%  
 select(SubjectID, everything())

## Warning: 5 parsing failures.  
## row # A tibble: 5 x 4 col row col expected actual expected <int> <int> <chr> <chr> actual 1 582 NA a number missing row 2 583 NA a number missing col 3 584 NA a number missing expected 4 757 NA a number missing actual 5 879 NA a number missing

write.csv(a, "Curated Data/Cleaned Data/CDCpcr.csv")  
  
#### Cleaning the 2nd dataset (CDC Serology EMIT FINAL REPORT 06252014.xls) ####  
# Storing as CDCserology.csv in UK Quarantine Study Data and Notes/Curated Data/Cleaned Data  
  
b <- read\_xls("Quarantine Raw Data/Nottingham Workspace Files/Study Data/CDC Serology EMIT FINAL REPORT 06252014.xls")  
  
# In looking at the dates, it looks like there was a typo on 228's date of screening. It is listed as March 27, 2014, but there were no measurements from 2014 and this should really have been March 27, 2013. Go ahead and make the change  
b$DRAWDATE[173] = as\_date("2013-03-27")  
  
# Now clean up the dataframe  
b <- b %>%   
 filter(`D/R ID` != "Q1 Donors" & `D/R ID` != "Q2 Donors" & `D/R ID` != "Q3 Donors" & `D/R ID` != "Q1 Recipients" & `D/R ID` != "Q2 Recipients" & `D/R ID` != "Q3 Recipients" & `D/R ID` != "End of data") %>%  
 group\_by(`D/R ID`) %>%  
 arrange(`D/R ID`, DRAWDATE) %>%  
 mutate(`MicroneutTiter\*4` = `Microneutralization Titer to A/Wisconsin/67/2005`\*4)  
  
# To classify the seroconversion status during the quarantine...   
# Compare the day 28 with day -2 (Q baseline) microneutralization assay titers  
# And compare the day -2 (Q baseline) with the screening microneutralization assay titers  
b$Seroconversion <- 0  
nrow(b)

## [1] 380

for (i in 2:nrow(b)) {  
 if(b$`D/R ID`[i] == b$`D/R ID`[i-1]) {  
 if(b[i, 5] > b[i-1, 6]) {  
 b$Seroconversion[i] <- b$Seroconversion[i]+1  
 }  
 }  
}  
  
# Checking the output from the above loop  
b\_screening <- b %>%  
 filter(Visit == "Screening") %>%  
 filter(Seroconversion == 1)  
b\_baseline <- b %>%  
 filter(Visit == "Q baseline") %>%  
 filter(Seroconversion == 1)  
  
# Now prepare the data for the clean output file  
b <- b %>%  
 ungroup() %>%  
 select(-`MicroneutTiter\*4`) %>%  
 rename(SubjectID = `D/R ID`, DrawDate = DRAWDATE, VisitType = Visit, QuarantineNumber = LOCAL\_ID\_1) %>%  
 separate(SubjectID, into = c("SubjectID", "RecipientIorC"), sep = "\\s\*\\-\\s\*", convert = TRUE) %>%  
 separate(SubjectID, into = c("Randomization", "SubjectID"), sep = c(1), remove = TRUE, convert = TRUE) %>%  
 select(SubjectID, everything()) %>%  
 unite(RecipientIorC\_Randomization, RecipientIorC, Randomization, sep = "", remove = FALSE) %>%  
 mutate(RecipientIorC\_Randomization = str\_remove(RecipientIorC\_Randomization, "NA")) %>%  
 select(-Randomization, -RecipientIorC) %>%  
 rename(Randomization = RecipientIorC\_Randomization) %>%  
 mutate(QuarantineNumber = str\_remove(QuarantineNumber, "Q"))

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 156 rows [1,  
## 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

write.csv(b, "Curated Data/Cleaned Data/CDCserology.csv")  
  
#### Cleaning the 3rd-5th datasets (3 tabs from Glasgow\_EMIT\_raw data.xlsx)####  
## Critical note: there is PCR data here and I can extract it and compare it with the CDC data...  
## But since the UK team has already done a thorough analysis of this, I will stick with the CDC data for PCR...  
## And use this Glasgow data soley for the purpose of HAI serology  
## There are three tabs with data that is useful here: "EMIT1 NPS," "EMIT2 NPS," "EMIT3 NPS,":  
## The EMIT1 TS," "EMIT2 TS," "EMIT3 TS," tabs have the same serology so, if we ignore PCR, then the "TS" tabs contain the same useful HAI data as the "NPS" tabs  
  
# First cleaning and then combining all three dfs, then storing as GLASGOWserology.csv in UK Quarantine Study Data and Notes/Curated Data/Cleaned Data  
  
# Import and clean the HAI data from Q1  
c <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/Glasgow\_EMIT\_raw data.xlsx", sheet = "EMIT1 NPS", na = c('','NA'), skip=2, col\_names=TRUE, trim\_ws=TRUE, col\_types="text")  
c <- c %>%  
 select(-`1`, -`2`, -`3`, -`4`, -`5`, -`6`, -`7`, -`8`, -`9`, -`10`) %>%  
 mutate(`HAI-2recodeNDA` = recode(`HAI -2`, NDA = "5"),   
 `HAI28recodeNDA` = recode(`HaI 28`, NDA = "5"),   
 `HAI-2recodeNDA` = as.numeric(`HAI-2recodeNDA`),   
 `HAI28recodeNDA` = as.numeric(`HAI28recodeNDA`),   
 `HAI-2recodeNDA\*4` = `HAI-2recodeNDA`\*4) %>%  
 rename(Preliminary.HAI.Classification = HAI, HAI28 = `HaI 28`, `HAI-2` = `HAI -2`) %>%  
 mutate(SeroconversionHAI = if\_else(`HAI-2recodeNDA\*4`-`HAI28recodeNDA`<= 0, 1, 0)) %>%  
 select(-Preliminary.HAI.Classification, everything())  
c$QuarantineNumber <- 1  
  
# Import and clean the HAI data from Q2  
d <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/Glasgow\_EMIT\_raw data.xlsx", sheet = "EMIT2 NPS", na = c('','NA'), skip=2, col\_names=TRUE, trim\_ws=TRUE, col\_types="text")  
d <- d %>%  
 select(-`1`, -`2`, -`3`, -`4`, -`5`, -`6`, -`7`, -`8`, -`9`, -`10`) %>%  
 mutate(`HAI-2recodeNDA` = recode(`HAI -2`, NDA = "5"),   
 `HAI28recodeNDA` = recode(`HAI 28`, NDA = "5"),   
 `HAI-2recodeNDA` = as.numeric(`HAI-2recodeNDA`),   
 `HAI28recodeNDA` = as.numeric(`HAI28recodeNDA`),   
 `HAI-2recodeNDA\*4` = `HAI-2recodeNDA`\*4) %>%  
 rename(Preliminary.HAI.Classification = Serology, HAI28 = `HAI 28`, `HAI-2` = `HAI -2`) %>%  
 mutate(SeroconversionHAI = if\_else(`HAI-2recodeNDA\*4`-`HAI28recodeNDA`<= 0, 1, 0)) %>%  
 select(-Preliminary.HAI.Classification, everything())  
d$QuarantineNumber <- 2  
  
# Import and clean the HAI data from Q3  
e <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/Glasgow\_EMIT\_raw data.xlsx", sheet = "EMIT3 NPS", na = c('','NA'), skip=2, col\_names=TRUE, trim\_ws=TRUE, col\_types="text")  
e <- e %>%  
 select(-`1`, -`2`, -`3`, -`4`, -`5`, -`6`, -`7`, -`8`, -`9`, -`10`) %>%  
 mutate(`HAI-2recodeNDA` = recode(`HAI -2`, NDA = "5"),   
 `HAI28recodeNDA` = recode(`HAI 28`, NDA = "5"),   
 `HAI-2recodeNDA` = as.numeric(`HAI-2recodeNDA`),   
 `HAI28recodeNDA` = as.numeric(`HAI28recodeNDA`),   
 `HAI-2recodeNDA\*4` = `HAI-2recodeNDA`\*4) %>%  
 rename(Preliminary.HAI.Classification = Serology, HAI28 = `HAI 28`, `HAI-2` = `HAI -2`) %>%  
 mutate(SeroconversionHAI = if\_else(`HAI-2recodeNDA\*4`-`HAI28recodeNDA`<= 0, 1, 0)) %>%  
 select(-Preliminary.HAI.Classification, everything())  
e$QuarantineNumber <- 3  
  
cde <- rbind(c, d, e)  
cde <- cde %>%   
 separate(ID, into = c("Randomization", "SubjectID"), sep = c(1), remove = TRUE, convert = TRUE) %>%  
 select(SubjectID, everything())  
  
write.csv(cde, "Curated Data/Cleaned Data/GLASGOWserology.csv")  
  
#### Cleaning the 6th-7th datasets (2 tabs from EMIT Donors, RIs and RCs.xlsx) ####  
## Note the Donors are in Sheet 1 and the Recipients are in Sheet 2  
## Will grab each of these sheets as their own dataframes  
  
# Storing a combined dataframe (Donors and Recipients)...  
# As DonorRecipientSubjectIDlist.csv in UK Quarantine Study Data and Notes/Curated Data/Cleaned Data  
  
# Manipulating the donors to get in format (subjectID, randomizationCode, quarantineNumber)  
f <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Donors, RIs and RCs.xlsx", sheet = "Donors", na = c('','NA'), skip=0, col\_names=TRUE, trim\_ws=TRUE, col\_types="text")  
f <- f %>%  
 select(-X\_\_1, -X\_\_2, -X\_\_3, -X\_\_4) %>%  
 gather(key, value, Q1, Q2, Q3, na.rm = TRUE, convert = TRUE) %>%  
 select(value, everything()) %>%  
 rename(subjectID = value, quarantineNumber = key) %>%  
 mutate(quarantineNumber = str\_remove(quarantineNumber, "Q"))  
f$randomizationCode <- "D"  
f <- f %>%  
 select(-quarantineNumber, everything())  
  
# Manipulating the recipients to get in format (subjectID, randomizationCode, quarantineNumber)  
g <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Donors, RIs and RCs.xlsx", sheet = "Recips", na = c('','NA'), skip = 0, col\_names = TRUE, trim\_ws = TRUE, col\_types = "text")  
g <- g %>%  
 select(-X\_\_2, -X\_\_3, -X\_\_5, -X\_\_6)  
  
g1 <- g[ , c(1:2)]  
g1 <- g1 %>%  
 rename(randomizationCode = X\_\_1, subjectID = Q1)  
g1$quarantineNumber <- 1  
  
g2 <- g[ , c(3:4)]  
g2 <- g2 %>%  
 rename(randomizationCode = X\_\_4, subjectID = Q2)  
g2$quarantineNumber <- 2  
  
g3 <- g[ , c(5:6)]  
g3 <- g3 %>%  
 rename(randomizationCode = X\_\_7, subjectID = Q3)  
g3$quarantineNumber <- 3  
  
g <- rbind(g1, g2, g3)  
g <- g %>%   
 filter(!is.na(subjectID))  
  
# Merging the donors (dataframe f) with the recipients (dataframe g)  
fg <- rbind(f, g)  
  
# Cleaning up the names of the variable values (removing text from subject id codes, etc.)  
fg <- fg %>%  
 mutate(randomizationCode = recode(randomizationCode, control = "CR"),   
 randomizationCode = recode(randomizationCode, intervention = "IR"),  
 subjectID = str\_remove(subjectID, "D"),   
 subjectID = str\_remove(subjectID, "R")) %>%  
 rename(SubjectID = subjectID, Randomization = randomizationCode, QuarantineNumber = quarantineNumber)  
  
write.csv(fg, "Curated Data/Cleaned Data/DonorRecipientSubjectIDlist.csv")  
  
#### Cleaning the 8th-10th datasets (EMIT Q1-Q3 Symptom scores and temperature data .xlsx files) ####  
  
# Once cleaned, stored as "Symptoms for all volunteers Q1-Q3.csv" in "Curated Data/Cleaned Data/"  
  
# Q1 Symptoms...  
# Extracting and binding together all the sheets (one sheet per volunteer) from the Q1 symptom workbook  
# (Collaboration with CJ)  
Q1Sx <- "Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Q1 Symptom scores and temperature data\_march2019.xlsx"  
Q1SxSheets <- excel\_sheets(path = Q1Sx) #"Sheet13" (the 42nd sheet) is empty so we'll ignore it and use sheets = 41  
h1 = c()  
Q1SxVarTables = c()  
for(i in 1:41){  
 temp = read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Q1 Symptom scores and temperature data\_march2019.xlsx", sheet = i,  
 col\_types =   
 c("text", "numeric", "text", "text", "text", "text", "text", "numeric",   
 "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric",  
 "numeric", "numeric", "numeric"))  
 Q1SxVarTables = rbind(Q1SxVarTables, c(i, names(temp)))  
 if(i==1){  
 tempName = names(temp)  
 }else{  
 names(temp) = tempName  
 }  
 h1 = rbind(h1, temp)  
}  
# Note: the generation of Q1SxVarTables is important to check that the columns match up for each sheet  
# We see that some of the variable names are slightly different (e.g., "muscleAches" and "Muscle aches")  
# Also note: it is important to specify the column (variable) types  
# Otherwise errors arise from the POSIXct type of data from the date and time columns  
# Could probably do a loop within a loop to extract from sheets from the respective 3 quarantines  
# But the different row numbers between Q1-Q3 was causing errors in experimentation with this code  
# Because there are only 3 excel workbooks for Sx, it's not onerous (and more clear) to extract and bind sheets individually  
  
# Also note: it appears that subject 203, represented in sheet 203, actually lists symtpoms scores for each of the study days for subjectID 202. I'm not sure if 203 never had any Sx scores reported and the repeated sheet is in error or if it is possible that the Sx scores for 203 have been lost somehow.   
# We should make a note of this in the paper because it throws off the numbers in the table 3 for reporting symptoms  
# There is data on serology/other things for 203, just that the Sx report is a copy of the report for SubjectID 202.  
# Update to this: 203 had not symptoms for any of the self-reported symptoms. The data was missing and cannot be retransfered from HVIVO without a long process due to some data sharing agreements and contracting.   
  
# Q2 Symptoms...  
# Extracting and binding together all the sheets (one sheet per volunteer) from the Q2 symptom workbook  
Q2Sx <- "Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Q2 Symptom scores and temperature data Sept 2018.xlsx"  
Q2SxSheets <- excel\_sheets(path = Q2Sx) #"Sheet13" (the 33rd sheet) is empty so we'll ignore it and use sheets = 32  
# Later discovered that there was an erroneous, hidden sheet ("D105") - deleting this gives sheets = 31  
h2 = c()  
Q2SxVarTables = c()  
for(i in 1:31){  
 temp = read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Q2 Symptom scores and temperature data Sept 2018.xlsx", sheet = i,  
 col\_types =   
 c("text", "numeric", "text", "text", "text", "text", "text", "numeric",   
 "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric",  
 "numeric", "numeric", "numeric"))  
 Q2SxVarTables = rbind(Q2SxVarTables, c(i, names(temp)))  
 if(i == 1){  
 tempName = names(temp)  
 }else{  
 names(temp) = tempName  
 }  
 h2 = rbind(h2, temp)  
}  
# Note: the original file "EMIT Q2 Symptom scores and temperature data.xlsx"  
# Had a hidden sheet with data for D105. This caused an error in the binding  
# A new version of the .xlsx file was created by deleting this sheet (and using the Sept 2018 version)  
# The data from D105 was already accounted for in the Q1 worksheet  
# The 12th sheet (for "D132") had extra columns, which were similarly deleted for the Sept 2018 version  
  
# Q3 Symptoms...  
# Extracting and binding together all the sheets (one sheet per volunteer) from the Q3 symptom workbook  
Q3Sx <- "Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Q3 Symptom scores and temperature data Sept 2018.xlsx"  
Q3SxSheets <- excel\_sheets(path = Q3Sx) #"Sheet13" (the 57th sheet) is empty so we'll ignore it and use sheets = 56  
# Later discovered that there was an erroneous, hidden sheet ("D105") - deleting this gives sheets = 55  
h3 = c()  
Q3SxVarTables = c()  
for(i in 1:55){  
 temp = read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT Q3 Symptom scores and temperature data Sept 2018.xlsx", sheet = i,  
 col\_types =   
 c("text", "numeric", "text", "text", "text", "text", "text", "numeric",   
 "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric",  
 "numeric", "numeric", "numeric"))  
 Q3SxVarTables = rbind(Q3SxVarTables, c(i, names(temp)))  
 if(i == 1){  
 tempName = names(temp)  
 }else{  
 names(temp) = tempName  
 }  
 h3 = rbind(h3, temp)  
}  
# Note: the original file "EMIT Q3 Symptom scores and temperature data .xlsx"  
# Had a hidden sheet with data for D105. This caused an error in the binding  
# A new version of the .xlsx file was created by deleting this sheet (Sept 2018 version)  
# The data from D105 was already accounted for in the Q1 worksheet  
# The 12th sheet (for "D150") had extra columns, which were similarly deleted for the Sept 2018 version  
  
# Bind together (row bind) symptoms from Q1-Q3 (stored in R as dfs h1-h3)  
# First check to ensure the variable names allign (like we did previously with the VarTables)  
VarTables <- rbind(Q1SxVarTables, Q2SxVarTables, Q3SxVarTables)  
  
# Next bind together, but first...  
# Change the name of h1 "Tympanic temp degrees C" to match that of h2 and h3 "Tympanic temp (degrees C)"  
h1 <- h1 %>%   
 rename("Tympanic temp (degrees C)" = `Tympanic temp degrees C`)  
h <- bind\_rows(h1, h2, h3)  
  
# Now clean and prepare curated symptom data  
h <- h %>%  
 select(-X\_\_1) %>% # Contains no data  
 filter(!is.na(ID)) %>%  
 select(ID, everything(), -`Tympanic temp (degrees C)`, everything()) %>% # Move temp variable to end and ID to front  
 select(-Day) # "Symptom Diary Card gives moore precise informatiion"  
h$SDC\_date <- as.numeric(h$SDC\_date) # Getting date numeric to convert to date format

## Warning: NAs introduced by coercion

h$SDC\_date <- as.Date(h$SDC\_date, origin = "1899-12-30") # Converting to date format  
h$SDC\_time <- as.numeric(h$SDC\_time) # Getting time in numeric to convert to date format

## Warning: NAs introduced by coercion

h <- h %>%   
 mutate(SDC\_time = SDC\_time\*86400) # Getting time to format where it can be translated to date/time  
h$SDC\_time <- as\_datetime(h$SDC\_time, origin = "1970-01-01") # Converting to date format  
h$SDC\_time <- round\_date(h$SDC\_time, unit = "minutes") # Rounding to minutes (the conversion earlier left some times off by one second)  
h$SDC\_time <- format(h$SDC\_time, "%H:%M") # Taking time as just hour:minute (ignore seconds)  
h <- h %>%  
 separate(ID, into = c("Randomization", "SubjectID"), sep = c(1), remove = TRUE, convert = TRUE) %>%  
 select(SubjectID, everything())  
  
write.csv (h, "Curated Data/Cleaned Data/Symptoms for all volunteers Q1-Q3.csv")  
  
#### Cleaning the 11th dataset (EMIT quarantine days and dates.xlsx) ####  
  
# Once cleaned, stored as "QuarantineCalendar.csv" in "Curated Data/Cleaned Data/"  
  
i <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT quarantine days and dates.xlsx", col\_names = FALSE)  
  
# The Q1-Q3 days and dates are listed in the same excel sheet but in an odd format  
# Process here will be to select out the rows/columns with the days and dates of each Q1-Q3  
# Then merge together with a full\_join  
# Note: "day" refers to the day number with respect to quarantine entry (number days before/after)  
# Note: "date" refers to the date corresponding to each day of the quarantines  
  
# Deal with Q1 days and dates  
i1 <- i[4:5, 1:15]  
i1 <- as.data.frame(t(i1))  
row.names(i1) <- c(1:15)  
i1 <- i1 %>%  
 rename(Q1days = V1, Q1dates = V2)  
i1$Q1dates <- as.character(i1$Q1dates)  
i1$Q1dates <- as.numeric(i1$Q1dates)  
i1$Q1dates <- as.Date(i1$Q1dates, origin = "1899-12-30")  
i1$Q1days <- as.character(i1$Q1days)  
  
# Deal with Q2 days and dates  
i2 <- i[9:10, 1:15]  
i2 <- as.data.frame(t(i2))  
row.names(i2) <- c(1:15)  
i2 <- i2 %>%  
 rename(Q2days = V1, Q2dates = V2)  
i2$Q2dates <- as.character(i2$Q2dates)  
i2$Q2dates <- as.numeric(i2$Q2dates)  
i2$Q2dates <- as.Date(i2$Q2dates, origin = "1899-12-30")  
i2$Q2days <- as.character(i2$Q2days)  
i2[2, "Q2days"] <- "D-1a"  
i2[4, "Q2days"] <- "D0repeat"  
  
# Deal with Q3 days and dates  
i3 <- i[14:15, 1:14]  
i3 <- as.data.frame(t(i3))  
row.names(i3) <- c(1:14)  
i3 <- i3 %>%  
 rename(Q3days = V1, Q3dates = V2)  
i3$Q3dates <- as.character(i3$Q3dates)  
i3$Q3dates <- as.numeric(i3$Q3dates)  
i3$Q3dates <- as.Date(i3$Q3dates, origin = "1899-12-30")  
i3$Q3days <- as.character(i3$Q3days)  
i3[2, "Q3days"] <- "D-1a"  
  
# prepare for full join -- need a common column name on which to join  
i1 <- i1 %>%   
 rename(Qdays = Q1days)  
i2 <- i2 %>%   
 rename(Qdays = Q2days)  
i3 <- i3 %>%   
 rename(Qdays = Q3days)  
  
i <- full\_join(i1, i2, by = "Qdays") %>%  
 full\_join(i3, by = "Qdays")  
  
write.csv(i, "Curated Data/Cleaned Data/QuarantineCalendar.csv")  
  
#### Cleaning the 12th dataset (EMIT volunteers - demographic data.xlsx) ####  
  
# Once cleaned, stored as "QuarantineDemographics.csv" in "Curated Data/Cleaned Data/"  
  
j <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/EMIT volunteers - demographic data.xlsx")  
j <- j %>%  
 rename(SubjectID = DMRNDNUM, Age = DDAge) %>%  
 filter(!is.na(SubjectID)) %>%  
 arrange(SubjectID)  
j <- j[-c(53:58),] # getting rid of "junk" rows  
j <- j %>%  
 separate(SubjectID, into = c("Randomization", "SubjectID"), sep = c(1), convert = TRUE) %>%  
 select(SubjectID, everything())  
  
write.csv(j, "Curated Data/Cleaned Data/QuarantineDemographics.csv")  
  
#### Cleaning the 13th dataset (Q1-3 Directed physical exam data (final).xlsx) ####  
  
# Once cleaned, stored as "PhysicalExam.csv" in "Curated Data/Cleaned Data/"  
   
k <- read\_xlsx("Quarantine Raw Data/Nottingham Workspace Files/Study Data/Q1-3 Directed physical exam data (final).xlsx")  
  
k <- k %>%  
 filter(!is.na(DMRNDNUM)) %>%  
 arrange(DMRNDNUM) %>%  
 filter(DMRNDNUM != "End of data" & DMRNDNUM != "Recipients") %>% # eliminate these filler rows  
 separate(DMRNDNUM, into = c("Randomization", "SubjectID"), sep = c(1), remove = TRUE, convert = TRUE) %>%  
 select(SubjectID, everything()) %>%  
 rename(VisitName = VISITNAM)  
  
write.csv(k, "Curated Data/Cleaned Data/PhysicalExam.csv")  
  
#### Cleaning the 14th dataset (G2 PCR copy number data from UMD lab) ####  
##  
# UKg2 <- readRDS("Quarantine Raw Data/Aerosol\_PCR/UK fine and coarse PCR results.RDS")  
#   
# UKg2mean <- UKg2 %>%  
# group\_by(subject.id, sample.day, sample.type) %>%  
# summarize(avg.copies = mean(final.copies, na.rm = TRUE), n.avg = sum(!is.na(final.copies)))  
#   
# UKg2mean <- UKg2mean %>%  
# ungroup() %>%  
# rename(G2.pcr.copies.sample.type = sample.type, G2.pcr.avg.copies = avg.copies, G2.pcr.n.pos.in.avg = n.avg)  
#   
# UKg2mean$subject.id <- as.numeric(UKg2mean$subject.id)  
# UKg2mean$sample.day <- as.numeric(UKg2mean$sample.day)  
# UKg2mean$G2.pcr.copies.sample.type <- as.factor(UKg2mean$G2.pcr.copies.sample.type)  
## The above is commented out because we are moving this piece to the analysis script, because the purpose of this piece of code was really to get out a GM and GSD for the aerosols - we can do this in the anlaysis script and meanwhile, keep the integrity of the main, authoritative, script.   
  
# The above code used an older version of the PCR data that had not been processed to address the situation when we have one replicate that is detected as positive and one replicate that is negative.   
# Jing ran some code and replaced the NA replicates (where the other replicate was positive) with 1414, which comes from ((1/sqrt(2))\*2000)  
  
# Let's replicate this here,   
  
  
# Additionally, the subject 109 was excluded from the dataset -- was this because it was considered a false positive? There is no note explaining why 109 should have what looks like a positive single replicate removed (coarse sample for study day 4 had one replicate that had 1038.66 RNA copies).   
# The new raw data file that has 2 values for each positive sampling instance.   
  
## Commenting out this piece.  
# UKg2\_b <- read.csv("Quarantine Raw Data/Aerosol\_PCR/UKg2\_negativesdel.csv")  
#   
# UKg2mean\_b <- UKg2\_b %>%  
# group\_by(subject.id, sample.day, sample.type) %>%  
# summarize(avg.copies = mean(final.copies), n.avg = sum(!is.na(final.copies))) %>%  
# ungroup()  
#   
# UKg2\_mean\_combined <- UKg2mean %>%  
# left\_join(UKg2mean\_b, by = c("subject.id" = "subject.id", "sample.day" = "sample.day", "G2.pcr.copies.sample.type" = "sample.type")) %>%  
# select(subject.id, sample.day, G2.pcr.copies.sample.type, avg.copies, n.avg) %>%  
# rename(G2.pcr.avg.copies = avg.copies, G2.pcr.n.pos.in.avg = n.avg)  
#   
# write.csv(UKg2\_mean\_combined, "Curated Data/Cleaned Data/UKg2mean.csv")  
  
## The above UKg2\_mean\_combined object is fine, however it represents a mean value of the pcr replicates whereby non-detects were computed as equal to 1414 = LOQ\*1/sqrt(2), where LOQ for fluA = 2000 RNA copies.   
  
## The above bit of code is commented out because we are moving this piece to the analysis script, because the purpose of this piece of code was really to get out a GM and GSD for the aerosols - we can do this in the anlaysis script and meanwhile, keep the integrity of the main, authoritative, script.   
  
# Regardless, both replicates should be in the dataset.  
# To be clear, what we have above are 2 objects,   
# 1) UKg2mean takes the non-detectable replicates (i.e., for samples that had one replicate positive) and treats them as not existing. Therefore, when the mean value is computed (and only mean values are displayed here for each sample - so the replicates vanish from this df), it is just computed using the single replicate where there is one replicate positive, and using 2 replicates where both replicates are positive.   
# 2) UKg2\_mean\_combined takes the non-detectable replicates (i.e, for samples that had one replicate positive) and treats them as being equal to 1414 RNA copies, which is computed from 2000\*(1/sqrt(2)) and then averages all of the pairs of replicates (and each pair of replicates will have a positive RNA copy number because the non-detects now take on the value of 1414). This, UKg2\_mean\_combined objects, however still represents a single value for each sample. The problem with this is that when we want to do a mixed model with tobits to account for the non-detect samples, we will need to have a place in the data for each replicate that is positive with RNA copy number or non-detect with no copy number. To do this we cannot have the data averaged, but rather need all of the data available. Thus, for each sampling day, for fine aerosol and for coarse aerosol, we will need to have 2 observations of pcr data, one for each replicate (these PCR assays were run in duplicate so there are 2 replicates per sample). The problem with achieving such a dataset is that I only have access to the "UK fine and coarse PCR results.RDS" data file for these pcr results and this dataset does have observations for each replicate for each sample, however it has positive pcr data for subject ID 109, which according to email correspondence with Dr. Milton and Dr. Jing Yan, these positives should be ignored and marked as negative. So we will do this and add this as another variable to the final Qdata for the eventual purpose of modeling the aerosol with respect to various covariates.   
  
UKg2 <- readRDS("Quarantine Raw Data/Aerosol\_PCR/UK fine and coarse PCR results.RDS")  
  
# UKg2\_all <- UKg2 %>%  
# select(subject.id, sample.day, sample.type, copy.num, cfactor, final.copies) %>%  
# rename(final.copies.replicate = final.copies, copy.num.replicate = copy.num)  
  
## The below is commented out, but it was originally written to eliminate the final.copy data for subject 109, however on second thought, it may be better to keep all of the data in this raw data file and then ignore it at the step of analysis. For example, the reason why 109's data may be ignored here is because 109 may not be considered an infected viral donor and thus aerosol shedding from this volunteer is ignored. If this is the case, then the analysis script will select out only the donors that had evidence of infection for the analysis and thus, having pcr positive data in this raw datafile should not disrupt any analytical output downstream.   
#   
# UKg2\_all\_without\_109 <- UKg2\_all %>%  
# filter(subject.id != 109)  
#   
# UKg2\_edit\_109 <- UKg2\_all %>%  
# filter(subject.id == 109)  
# UKg2\_edit\_109$final.copies.replicate <- NA  
#  
# Note that the copy.num.replicate will still show that there was a positive detection for one of the coarse samples for subject 109, however because this is being called as a false positive, the final.copies.replicate is being changed from the value that it was, to NA.   
#   
# UKg2\_all\_with\_edited\_109 <- UKg2\_all\_without\_109 %>%  
# bind\_rows(UKg2\_edit\_109) %>%  
# arrange(subject.id, sample.day, sample.type, final.copies.replicate)  
## End of commenting out this code block.   
  
# The qPCR calibration factor is included in this dataset and the final.copies.replicate variable is simply the product of the calibration factor times the copy.num.replicate variable. The copy.number.replicate variable looks like it is the variable that comes directly from the output of the MxPro pcr assay analysis program (converting ct to virus particle -- the assay was standardized on an EM quantified PR/8 Influenza A virus stock).  
# The RNA copy#-to-virus particle ratio for the viral standards (PR/8 and B/Lee) used in the original cleaning process was found to be 80 and 411 for flu A and B, respectively. Michael Grantham's original experiments indicate that this ratio should actually be 250 and 272 for flu A and B, respectively. However, after applying the overall EMIT project's standard curve information to these experiments these final, adjusted ratios become 80 and 411 for flu A and B, repectively. The PCR experiments that support this are found in the "EMIT\_Data\_Analysis\_Jake/EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/EMIT RNA copies per virion" directory. Thus, Jing was using 80 for this RNA copies to virus particle ratio. 25\*80 = 2,000 and this is what was used.   
  
UKg2\_all <- UKg2 %>%  
 select(subject.id, sample.day, sample.type, copies.in, copy.num, cfactor, final.copies) %>%  
 mutate(copy.num = copies.in\*2000) %>%  
 mutate(final.copies = copy.num\*cfactor) %>%  
 rename(final.copies.replicate = final.copies, copy.num.replicate = copy.num, G2.pcr.copies.sample.type = sample.type)  
  
write.csv(UKg2\_all, "Curated Data/Cleaned Data/UKg2\_all.csv")  
  
# Let's actually just use this UKg2\_all df in the creation of the authoritative, merged dataset.  
# Then, in the analysis script, we can add some script that takes the GM and GSD while accounting for LOQ\*1/sqrt(2)  
  
  
#### Cleaning the 15th dataset (cough data collected during G-II visits) ####  
  
UKcough <- read.csv("Quarantine Raw Data/EMIT UK GII log 2013 cough-temp/EMITGIILogUK\_DATA\_2013-12-13\_1607.csv",   
 na.strings = c("", "NA"))  
  
UKcough <- UKcough %>%  
 select(subject\_id, start\_dt, start\_time, reserv\_time, cough\_count, notes) %>%  
 filter(subject\_id != "test")  
UKcough$subject\_id <- as.numeric(as.character(UKcough$subject\_id))  
UKcough$start\_dt <- as.POSIXct(UKcough$start\_dt)  
UKcough$notes <- as.character(UKcough$notes)  
  
# There were 7 instances where cough\_count was not documented. Audio recordings were lost.  
# However, there are notes where there was evidence of cough or a lot of cough for these missing values.  
# We will use these notes to estimate cough numbers for these 7 individuals.  
  
# First we will duplicate the cough\_count variable and then estimate missing cough counts on this new variable.  
UKcough$cough\_count\_complete <- UKcough$cough\_count  
  
# Now we will estimate cough counts on the cough\_count\_complete variable for all 7 of the missing instances of cough count.  
# 6 instances seems to have some coughing -- we assigned a cough count of 3.  
# 1 instance was described as "A lot of coughing!" and we assigned a cough count of 30 here.   
  
# In the end we won't do this and we will remove these imputations -- commenting out the below lines.   
# UKcough$cough\_count\_complete[4] <- 3  
# UKcough$cough\_count\_complete[16] <- 3  
# UKcough$cough\_count\_complete[19] <- 3  
# UKcough$cough\_count\_complete[24] <- 3  
# UKcough$cough\_count\_complete[25] <- 30  
# UKcough$cough\_count\_complete[29] <- 3  
# UKcough$cough\_count\_complete[34] <- 3  
  
# Now we can classify coughs categorically  
# Cough count of 0 is none, up to 6 (non including 6) is low, and 6 and above is high  
UKcough <- UKcough %>%  
 mutate(cough\_categorical = cut(cough\_count\_complete, breaks = c(-Inf, 0, 6, Inf), labels = c("none", "low", "high")))  
  
# Rename the variables before saving the curated data  
UKcough <- UKcough %>%  
 rename(visit\_date = start\_dt, duration\_GII\_collection = reserv\_time)  
  
# Save the dataframe as "CoughData.csv" in "Curated Data/Cleaned Data/"  
write.csv (UKcough, "Curated Data/Cleaned Data/UKCoughData.csv")