EMIT\_UMD\_Natural\_Infection\_Cleaning.R

jbueno

Sun Feb 10 17:14:46 2019

# EMIT UMD Natural Infection Study Data Curation - cleaning raw data to produce cleaned spreadsheets  
# EMIT\_UMD\_Natural\_Infection\_Cleaning.R  
# Program Objective: Take the datasets identified as critical, clean them, and later merge to form curated one or more curated datasets  
# Author: Jacob Bueno de Mesquita using material from Jing Yan and Don Milton  
# Date: December 14, 2018 - February 2019  
# Summary:   
# This script uses bits and pieces of code from Jing Yan and Don Milton that were previously used to produce cleaned data, for the Yan et al., 2018 PNAS publication on the EMIT study, and reproduces them in logical, linear fashion, whereby the steps in the cleaning and merging process are clearly understood. The original bits of code are cited in this script, however the author goes further to enhance the readability and functionality of the code, in many places streamlining. This ensures that the script maintains a high level of readability throughout. Originally, it was uncertain which of the original scripts (there are quite a few) were required to reproduce the cleaned dataset used in the PNAS publication analysis. It was also uncertain which raw datafiles (there are quite a few) should be used. It was also uncertain the linear order in which data cleaning steps should occur to produce the desired, clean data. Upon providing answers to these three, aforementioned areas of uncertainty, the question of how to deal with what appeared to be missing code arose and were dealt with by the author's own code development. The PNAS cleaned dataset was compared with the emerging, cleaned, reproduced dataset from this script for guidance. Other issues also arose such as the evolving nature of the packages used and the default settings contained in the commands of certain packages. For example, the "distinct" command from the "dplyr" package changed it's default settings from ".keep\_all = T" to ".keep\_all = F", effectively eliminating variables from the dataframe under operation that were not named in the distinct() command. To ensure future reproducibility, scripts (especially Markdown files) are stamped with the sessionInfo() command which prints out the information about the version of R Studio and each package.   
  
# The current script goes beyond what was previously done. Rather than simply producing a single, cleaned dataset for the PNAS publication, this script produces 7 cleaned datasets:  
# 1) all\_screened.csv: all data on all screened participants  
# 2) all\_cases.csv: all data on all participants who met case criteria and were enrolled  
# 3) all\_cases\_gii\_samples.csv: data only where a G-II sample was taken for all participants who met case criteria and were enrolled  
# 4) flu\_cases.csv: all data on enrolled cases who were positive for influenza virus  
# 5) flu\_cases\_gii\_samples.csv: data only where a G-II sample was taken for all enrolled cases who were positive for influenza virus  
# 6) PNAS\_data\_full.csv: final, cleaned dataset from among enrolled cases who were positive for influenza virus and who had complete cough data, limited to visits between days 1 and 3 of illness onset, and had complete PCR data.  
# 7) There is a seventh dataset, which is identical to the PNAS\_data\_full.csv dataset. It is called finaldatasetrepeatupdate.csv and is the version of the cleaned, PNAS dataset that is created by following closely with the process laid out by the pieces of code from Jing and Dr. Milton. It is exactly replicated by the PNAS\_data\_full.csv dataset, which was produced by using an updated, more streamlined, and more logical approach.   
  
# A lucidchart diagram provides an excellent description of the cleaning and merging process that gives rise to these 7 datasets. This lucidchart diagram can be accessed here: https://www.lucidchart.com/invitations/accept/2d2c49c7-0b46-42e6-af8a-bec41ceff41f.   
# Additionally, the word file under EMIT\_Data\_Analysis\_Jake/EMIT\_UMD\_Natural\_Infection/Analysis Notes called "samples need to be removed from final anlaysis.docx" provides a summary (see the table called "Moving from the total screened df to the PNAS analytical df" in the document) of how the dataset with data from all of the screened participants, dwindles down to the final, cleaned dataset. This is also reflected in the lucidchart diagram. This table is a more highly resolved and detailed version of Table S1 from Yan et al., 2018.  
# Several key discrepancies have been raised between the resulting PNAS\_data\_full.csv and Jing's original PNAS cleaned dataset (which can be found in the EMIT\_Data\_Analysis box.com directory).  
# 1) The labelling of the type (flu A or B) or infection for participants who were coinfected with both types, in the previous version, incorrectly labeled all of the pcr results as being either flu A or B. This was updated to reflect the accurate type of virus that was being targetting in the associated pcr result variables. This resulted in the addition of a few observations to the datasets.   
# 2) There were a couple of instances where pcr files had 2 different names, but were actually the same file. This occured where the date in the name of the pcr experiment used a "0" in front of a 2-digit month abbreviation (i.e., 07 for august, as opposed to simply 7). These instances were removed in the updated dataframes, thus eliminating a few, repeated observations from the datasets.   
# 3) The RNA copy#-to-virus particle ratio used in the original cleaning process was found to be 80 and 411 for flu A and B, respectively. However Michael Grantham indicates that this ratio should actually be 250 and 272 for flu A and B, respectively. The PCR experiments that support Grantham's claim are found in the EMIT\_Data\_Analysis\_Jake/EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/EMIT RNA copies per virion directory. Thus, we updated this RNA copy#-to-virus particle ratio in the new datasets.  
# 4) It appears that the variables cur\_asthma and lung\_sym\_2pos in Jing's original dataset, cannot be reproduced from any of the available raw data. Email correspondence with Jing has yet to reveal how to reproduce these variables, or what they mean precisely. There is an "asthma" variable in the dataset that doesn't seem to correspond exactly with the "cur\_asthma" variable in Jing's original dataset.   
# 5) The fluvac\_last2y variable appears to have quite a few instances of missingness. NAs, as opposed to 0 (for no flu vaccine within the last 2 years), and 1 (for yes, flu vaccine taken within the last 2 years), were observed for a number of participants. Jing's original dataset had marked these NA's as 0, but I have decided to not do this unless otherwise directed. Thus, this represents another discrepancy. As a result, the bothyear variable (flu vaccine taken both years) is also incomplete because of these NA values that have not been forced to 0 as they appear to have been done in the original PNAS data.  
# 6) f) Although Table S1 in Yan et al., 2018 shows that there were 178 breath collection visits from 178 subjects, we were unable to replicate this finding. We only ever show that there is breath collection data for 276 visits from 178 subjects. Perhaps there were a couple of intstances where breath collection was initiated but not completed and not marked has having occured in the REDCap database.   
  
#### Load required packages and set working directory ####  
  
library(tidyverse)  
library(RcppRoll)  
library(readxl)  
library(knitr)  
library(data.table)  
library(lubridate)  
library(arsenal)  
library(lme4)  
  
setwd("/Users/jbueno/Box Sync/EMIT/EMIT\_Data\_Analysis\_Jake/EMIT\_UMD\_Natural\_Infection")  
  
sessionInfo() # for reproducibility

## R version 3.5.1 (2018-07-02)  
## Platform: x86\_64-apple-darwin15.6.0 (64-bit)  
## Running under: macOS 10.14.2  
##   
## 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 lme4\_1.1-19 Matrix\_1.2-14   
## [4] arsenal\_1.5.0 lubridate\_1.7.4 data.table\_1.11.8  
## [7] knitr\_1.20 readxl\_1.1.0 RcppRoll\_0.3.0   
## [10] forcats\_0.3.0 stringr\_1.3.1 dplyr\_0.7.7   
## [13] purrr\_0.2.5 readr\_1.1.1 tidyr\_0.8.2   
## [16] tibble\_1.4.2 ggplot2\_3.1.0 tidyverse\_1.2.1   
## [19] htmlTable\_1.12 rmarkdown\_1.10 markdown\_0.8   
##   
## loaded via a namespace (and not attached):  
## [1] tidyselect\_0.2.5 splines\_3.5.1 haven\_1.1.2 lattice\_0.20-38   
## [5] testthat\_2.0.1 colorspace\_1.3-2 htmltools\_0.3.6 yaml\_2.2.0   
## [9] utf8\_1.1.4 rlang\_0.3.0.1 nloptr\_1.2.1 pillar\_1.3.0   
## [13] glue\_1.3.0 withr\_2.1.2 modelr\_0.1.2 bindr\_0.1.1   
## [17] plyr\_1.8.4 munsell\_0.5.0 gtable\_0.2.0 cellranger\_1.1.0  
## [21] rvest\_0.3.2 htmlwidgets\_1.3 evaluate\_0.12 fansi\_0.4.0   
## [25] highr\_0.7 broom\_0.5.0 Rcpp\_0.12.19 scales\_1.0.0   
## [29] backports\_1.1.2 checkmate\_1.8.5 jsonlite\_1.5 hms\_0.4.2   
## [33] digest\_0.6.18 stringi\_1.2.4 grid\_3.5.1 rprojroot\_1.3-2   
## [37] cli\_1.0.1 tools\_3.5.1 magrittr\_1.5 lazyeval\_0.2.1   
## [41] crayon\_1.3.4 pkgconfig\_2.0.2 MASS\_7.3-51 xml2\_1.2.0   
## [45] minqa\_1.2.4 assertthat\_0.2.0 httr\_1.3.1 rstudioapi\_0.8   
## [49] R6\_2.3.0 nlme\_3.1-137 compiler\_3.5.1

# Now pasting code from Jing Yan and Don Milton that was used in previous work on the EMIT UMD data and working with it to make sure that it functions like it was intended and to reproduce the final dataset that was used in the PNAS analysis.  
# The goal here is to review their script and improve the clarity  
  
#### \*\*\*\* Using Script: Jing Yan and Dr. Milton's "Merge 1-3-update.R" \*\*\*\* ####  
  
##  
# Original file information:  
  
# From Jing Yan & Don Milton; January 20-25, 2016  
# Purpose: follow the data analysis plan in folder EMIT\_Data\_Analysis (described below)  
  
# Purpose, input and output files:   
# 1. Read in Clinical (encounter and questionnaire) data;  
# a. Count rows and columns check that there are no missing required fields (e.g. date of visit, subject\_id).   
# Print number of rows total and numbers for eachredcap\_event\_name.   
# b. Create a dataframe with visits (1,2,3) and another df with g2\_run\_1,2,3; then if date\_visit=”” and redcap\_event\_name = “g2\_run\_1”   
# then date\_visit = date\_g2\_1; next create an indicator for visit(1,2,3) called visit\_num with values (1,2, or 3) and an   
# indicator called g2\_run with values (1,2, or3) in the respective df; then merge these df on date\_visit and subj\_id.   
# The result of this merge is that we get one record per subj\_id and date. Create an enrolled indicator: if g2\_run is.na   
# then enrolled=FALSE, if g2\_run = (1, 2, or 3) then enrolled = TRUE. Print out numbers of enrolled for each of   
# visit\_num = 1, 2 and 3 (i.e. enrolled 1st, 2nd, and 3rd screening visits).  
# c. Keep just: subj\_id, date\_visit, visit\_num, g2\_num, enrolled indicator -> sum\_clinical with variable names:  
# "field\_subj\_id" "date\_visit" "visit\_name" "visit\_num" "g2\_name" "g2\_run" "enrolled" "clinical.i"   
# 2. Read in G-II\_Log:  
# a. Count rows and columns check that there are no missing required fields (e.g. date of visit, subject\_id).   
# b. Create indicator g2\_coll\_num = 1, 2 or 3 based on redcap\_event\_name and rename start\_dt to date\_visit.  
# c. Print out numbers of rows for each redcap\_event\_name   
# d. Convert date\_visit from char to date format & create indicator for g2 log record (g2lm.i)  
# d. Keep: subj\_id, date\_visit, g2\_coll\_num, and g2lm.i -> g2\_log\_min with variable names:  
# "field\_subj\_id" "date\_visit" "g2\_coll\_num" "g2lm.i""field\_subj\_id" "date\_visit" "g2\_coll\_num" "g2lm.i"   
# 3. Merge 1: merge sum\_clinical and g2\_log\_min  
# a. Merge by subj\_id and date\_visit  
# b. Print data checks (e.g. number of visits, subjects, enrolled, etc.)  
# c. Check that all records that are marked as “enrolled” have g2\_coll\_num that is not na otherwise delete that extra visit   
# (e.g. 69). Print table of number of obs by number of visits, etc. as data checks.  
# d. Output dataframe with one obs per subj\_id and date -> merge1 with variable names:   
# "field\_subj\_id" "date\_visit" "g2\_coll\_num" "enrolled" "visit\_num" "g2\_run" "clinical.i" "g2lm.i" "merge1.i"  
# 4. Read Field Db   
# a. Check for empty row etc.   
# b. Delete empty rows, create indicator for record present in Field Db (field.db1.i)  
# c. Print number of obs and number of obs by sample type  
# d. Convert date\_visit from char to date data type  
# d. Output file -> field.db1 with variable names:  
# "field\_subj\_id" "sample\_id" "date\_visit" "sample\_type" "field.db1.i"  
# 5. Merge 2: merge merge1 and field.db1 (by field\_subj\_id and date\_visit)  
# a. Use the enrollment indicator to identify & remove G-II samples that were not collected but included in Field Db.   
# b. Data checks for numbers of rows, IDs, etc.  
# c. Output dataframe with one obs per sample\_id -> merge2  
# 6. Read UMD Samples 2013 from Redcap:  
# a. Check for empty rows etc. delete empty rows with no date\_visit or subj\_id  
# b. Print number of obs and number of obs by sample type.  
# c. Separate collection\_1, assay\_1, and assay\_2   
# d. Pull out the passage and focus assays into separate dataframes (drop pcr variables)  
# d. Compute passage and focus assay results  
# e. Merge by sample\_id to get one row per sample with culture results.   
# f. Keep focus and passage variables, date\_visit, sample type, subj\_id, sample\_id, and create indicator variables.  
# g. Print number of obs and number of obs by sample type  
# h. Do data checks for problems (e.g. impactors that were cultured or culture results for samples without a sample type)  
# i. List problem samples  
# j. Output dataframe with one obs per sample -> culture\_results  
# 7. Merge 3: Use merge2 and culture\_results  
# a. By sample\_id and date\_visit.  
# b. Check that merge worked by count of rows and columns and obs by sample type and check for empty variables that   
# should have values. Check that sample types match. Print data checks  
# c. Drop obs from field db that do not have corresponding values in the Redcap sample log.   
# d. Output dataframe with naone obs per sample\_id -> samples.cc  
  
##  
  
#### READ in and work with CLINICAL DATABASE ####  
  
clinical\_in\_file <- 'UMD\_Raw\_Data/REDCAP/EMITClinicalUMD2013.csv'  
clinical\_umd <- read.csv(clinical\_in\_file)  
  
# Clean up the raw data just a little   
# clinical\_umd$date\_indx\_on <- as.Date(clinical\_umd$date\_indx\_on, format = "%m/%d/%y")  
# clinical\_umd$date\_indx\_visit <- as.Date(clinical\_umd$date\_indx\_visit, format = "%m/%d/%y")  
# clinical\_umd$date\_ref <- as.Date(clinical\_umd$date\_ref, format = "%m/%d/%y")  
# clinical\_umd$date\_enroll <- as.Date(clinical\_umd$date\_enroll, format = "%m/%d/%y")  
# clinical\_umd$date\_on\_sx <- as.Date(clinical\_umd$date\_on\_sx, format = "%m/%d/%y")  
# clinical\_umd$date\_visit <- as.Date(clinical\_umd$date\_visit, format = "%m/%d/%y")  
# clinical\_umd$date\_fever\_on <- as.Date(clinical\_umd$date\_fever\_on, format = "%m/%d/%y")  
# clinical\_umd$date\_cough <- as.Date(clinical\_umd$date\_cough, format = "%m/%d/%y")  
# clinical\_umd$date\_g2\_1 <- as.Date(clinical\_umd$date\_g2\_1, format = "%m/%d/%y")  
  
# Let's produce some summary information about this clinical\_umd df  
  
print(nrow(clinical\_umd))

## [1] 653

print(ncol(clinical\_umd))

## [1] 196

print(sum(clinical\_umd$redcap\_event\_name == 'visit\_1\_part\_a\_arm\_1'))

## [1] 355

print(sum(clinical\_umd$redcap\_event\_name == 'screen\_visit\_2\_arm\_1'))

## [1] 20

print(sum(clinical\_umd$redcap\_event\_name == 'screen\_visit\_3\_arm\_1'))

## [1] 1

print(sum(clinical\_umd$redcap\_event\_name == 'visit\_1\_part\_a\_arm\_1') +  
 sum(clinical\_umd$redcap\_event\_name == 'screen\_visit\_2\_arm\_1') +  
 sum(clinical\_umd$redcap\_event\_name == 'screen\_visit\_3\_arm\_1'))

## [1] 376

print(sum(clinical\_umd$redcap\_event\_name == 'g2\_run\_1\_arm\_1'))

## [1] 178

print(sum(clinical\_umd$redcap\_event\_name == 'g2\_run\_2\_arm\_1'))

## [1] 72

print(sum(clinical\_umd$redcap\_event\_name == 'g2\_run\_3\_arm\_1'))

## [1] 27

print(sum(clinical\_umd$redcap\_event\_name == 'g2\_run\_1\_arm\_1') +  
 sum(clinical\_umd$redcap\_event\_name == 'g2\_run\_2\_arm\_1') +  
 sum(clinical\_umd$redcap\_event\_name == 'g2\_run\_3\_arm\_1'))

## [1] 277

print(addmargins(with(clinical\_umd, table(redcap\_event\_name, exclude = c()))))

## redcap\_event\_name  
## g2\_run\_1\_arm\_1 g2\_run\_2\_arm\_1 g2\_run\_3\_arm\_1   
## 178 72 27   
## screen\_visit\_2\_arm\_1 screen\_visit\_3\_arm\_1 visit\_1\_part\_a\_arm\_1   
## 20 1 355   
## Sum   
## 653

# Note that one subject was enrolled twice!  
print(select(filter(clinical\_umd, field\_subj\_id == 47 | field\_subj\_id == 187),  
 field\_subj\_id, date\_visit, redcap\_event\_name, date\_g2\_1, rapid\_flu\_\_\_1, rapid\_flu\_\_\_2, spec\_note))

## field\_subj\_id date\_visit redcap\_event\_name date\_g2\_1 rapid\_flu\_\_\_1  
## 1 47 1/4/13 visit\_1\_part\_a\_arm\_1 0  
## 2 47 g2\_run\_1\_arm\_1 1/4/13 NA  
## 3 47 1/5/13 g2\_run\_2\_arm\_1 NA  
## 4 187 2/7/13 visit\_1\_part\_a\_arm\_1 1  
## 5 187 g2\_run\_1\_arm\_1 2/7/13 NA  
## 6 187 2/8/13 g2\_run\_2\_arm\_1 NA  
## 7 187 2/9/13 g2\_run\_3\_arm\_1 NA  
## rapid\_flu\_\_\_2  
## 1 0  
## 2 NA  
## 3 NA  
## 4 0  
## 5 NA  
## 6 NA  
## 7 NA  
## spec\_note  
## 1 Subject was tested positive for RSV and the decision was made to put her into the GII  
## 2   
## 3   
## 4 Subject was also subject 47 for a previous illness, 1 month ago.  
## 5   
## 6   
## 7

# This means that there was actually one less unique invidivual than we have unique subjects ids.  
# We note the above and treat subject IDs as person-illness-episodes, not persons.  
  
# Clinical data split into screening visits and g2 runs and remerged to get one row per encounter date  
clinical\_min <- clinical\_umd %>%   
 select(field\_subj\_id, redcap\_event\_name, date\_visit, date\_g2\_1, rapid\_flu\_\_\_3, rapid\_flu\_loc, body\_temp, date\_on\_sx)  
  
clinical\_visit <- clinical\_min %>%   
 filter(grepl('visit', redcap\_event\_name))  
  
clinical\_g2 <- clinical\_min %>%   
 filter(grepl('^g2\_run', redcap\_event\_name))  
  
clinical\_visit$visit\_num <- ifelse(clinical\_visit$redcap\_event\_name == 'visit\_1\_part\_a\_arm\_1', 1,   
 ifelse(clinical\_visit$redcap\_event\_name == 'screen\_visit\_2\_arm\_1', 2, 3))   
  
clinical\_g2$g2\_run <- ifelse(clinical\_g2$redcap\_event\_name == 'g2\_run\_1\_arm\_1', 1,   
 ifelse(clinical\_g2$redcap\_event\_name == 'g2\_run\_2\_arm\_1', 2, 3))  
  
# Find the G2 first time sample collection data  
clinical\_g2\_1 <- clinical\_g2 %>%   
 filter(grepl('^g2\_run\_1', redcap\_event\_name)) %>%  
 select(field\_subj\_id, redcap\_event\_name, date\_g2\_1, g2\_run) %>%  
 rename(date\_visit = date\_g2\_1)  
  
# Find the G2 2nd and 3rd time sample collection data  
clinical\_g2\_23 <- clinical\_g2 %>%  
 filter(!grepl('^g2\_run\_1', redcap\_event\_name)) %>%   
 select(field\_subj\_id, redcap\_event\_name, date\_visit, g2\_run)  
  
# Merge the G2 sample collection visits (1, 2, and 3) together into the clinical\_g2 df  
clinical\_g2 <- merge(clinical\_g2\_1, clinical\_g2\_23,   
 c('field\_subj\_id', 'date\_visit', 'redcap\_event\_name', 'g2\_run'), all = TRUE)  
  
sum\_clinical <- merge(select(clinical\_visit, -contains("date\_g2\_1")), clinical\_g2,   
 c('field\_subj\_id', 'date\_visit'), all = TRUE)  
  
sum\_clinical$enrolled <- ifelse(!is.na(sum\_clinical$g2\_run), TRUE, FALSE)  
  
sum\_clinical <- sum\_clinical %>%   
 rename(visit\_name = redcap\_event\_name.x, g2\_name = redcap\_event\_name.y)  
  
sum\_clinical$g2\_run <- with(sum\_clinical, ifelse(is.na(g2\_run), 0, g2\_run))   
  
sum\_clinical$visit\_num <- with(sum\_clinical, ifelse(is.na(visit\_num), 999, visit\_num))  
  
sum\_clinical$clinical.i <- TRUE #indicator for presence of record in sum\_clinical  
  
# Number of rows in summary data (total number of unique encouters)  
nrow(sum\_clinical)

## [1] 475

# Tabulations for data checks  
print(addmargins(with(sum\_clinical, table(visit\_name, visit\_num, exclude = c()))))

## visit\_num  
## visit\_name 1 2 3 999 Sum  
## g2\_run\_1\_arm\_1 0 0 0 0 0  
## g2\_run\_2\_arm\_1 0 0 0 0 0  
## g2\_run\_3\_arm\_1 0 0 0 0 0  
## screen\_visit\_2\_arm\_1 0 20 0 0 20  
## screen\_visit\_3\_arm\_1 0 0 1 0 1  
## visit\_1\_part\_a\_arm\_1 355 0 0 0 355  
## <NA> 0 0 0 99 99  
## Sum 355 20 1 99 475

print(addmargins(with(sum\_clinical, table(g2\_name, g2\_run, exclude = c()))))

## g2\_run  
## g2\_name 0 1 2 3 Sum  
## g2\_run\_1\_arm\_1 0 178 0 0 178  
## g2\_run\_2\_arm\_1 0 0 72 0 72  
## g2\_run\_3\_arm\_1 0 0 0 27 27  
## screen\_visit\_2\_arm\_1 0 0 0 0 0  
## screen\_visit\_3\_arm\_1 0 0 0 0 0  
## visit\_1\_part\_a\_arm\_1 0 0 0 0 0  
## <NA> 198 0 0 0 198  
## Sum 198 178 72 27 475

print(addmargins(with(sum\_clinical, table(enrolled, clinical.i, exclude = c()))))

## clinical.i  
## enrolled TRUE Sum  
## FALSE 198 198  
## TRUE 277 277  
## Sum 475 475

print(addmargins(with(sum\_clinical, table(visit\_num, g2\_run, exclude = c()))))

## g2\_run  
## visit\_num 0 1 2 3 Sum  
## 1 181 174 0 0 355  
## 2 17 3 0 0 20  
## 3 0 1 0 0 1  
## 999 0 0 72 27 99  
## Sum 198 178 72 27 475

# Total number of g2 runs (sum of runs 1, 2, and 3) according to initial clinical data  
sum(sum\_clinical$g2\_run > 0)

## [1] 277

# Fix the format of the date variable in sum\_clinical  
sum\_clinical$date\_visit <- as.Date(as.character(sum\_clinical$date\_visit), format = "%m/%d/%y")  
sum\_clinical$date\_on\_sx <- as.Date(as.character(sum\_clinical$date\_on\_sx), format = "%m/%d/%y")  
sum\_clinical$date\_on\_sx <- as.factor(sum\_clinical$date\_on\_sx)  
print(head(tbl\_df(sum\_clinical)))

## # A tibble: 6 x 12  
## field\_subj\_id date\_visit visit\_name rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## <int> <date> <fct> <int> <int> <dbl>  
## 1 7 2012-12-04 visit\_1\_p… 1 NA 37.7  
## 2 8 2012-12-04 visit\_1\_p… 1 NA 39.5  
## 3 9 2012-12-05 visit\_1\_p… 1 NA 37.2  
## 4 10 2012-12-05 visit\_1\_p… 1 NA 37   
## 5 12 2012-12-10 visit\_1\_p… 1 NA 37   
## 6 13 2012-12-10 visit\_1\_p… 1 NA 39.1  
## # ... with 6 more variables: date\_on\_sx <fct>, visit\_num <dbl>,  
## # g2\_name <fct>, g2\_run <dbl>, enrolled <lgl>, clinical.i <lgl>

sum\_clinical\_subjectID\_count <- sum\_clinical %>%  
 group\_by(field\_subj\_id) %>%  
 summarise(count = n())  
# This shows that this df has information on all 355 screened participants  
  
#### READ in and work with G2 LOG DATA ####  
  
g2\_in\_file <- 'UMD\_Raw\_Data/GII/EMITGIILogUMD2013.csv'  
g2\_log <- read.csv(g2\_in\_file)  
  
# Check the number of sampling instances here  
g2\_log\_sampling\_instance\_check <- g2\_log %>%  
 distinct(subject\_id, start\_dt)  
# This yields 277 but this is before the data clean.   
  
print(nrow(g2\_log))

## [1] 277

print(ncol(g2\_log))

## [1] 87

# Date Entry Error Correction  
# Subject\_id 284 g2 collection\_2\_arm\_1 was entered as 2013-03-17 but baseline was on 2013-02-16 and collection\_3 was 2013-02-18.  
# Therefore recode collection\_2 date to February from March (i.e. to 2013-02-17).  
print(select(filter(g2\_log, subject\_id == 284), subject\_id, redcap\_event\_name, start\_dt))

## subject\_id redcap\_event\_name start\_dt  
## 1 284 baseline\_and\_colle\_arm\_1 2013-02-16  
## 2 284 collection\_2\_arm\_1 2013-03-17  
## 3 284 collection\_3\_arm\_1 2013-02-18

g2\_log$start\_dt[which(g2\_log$subject\_id == 284 & g2\_log$start\_dt == '2013-03-17', arr.ind = TRUE)] <- '2013-02-17'  
  
# Number of subject(s) without a start\_dt  
print(sum((g2\_log$start\_dt) == ""))

## [1] 1

id <- as.integer(select(filter(g2\_log, start\_dt == ""), subject\_id))   
  
# G-II log for cases without start date  
print(tbl\_df(select(filter(g2\_log, subject\_id==id),   
 subject\_id,   
 redcap\_event\_name,   
 start\_dt,   
 g2\_unit,operator,  
 chiller\_t1,  
 subj\_min)))

## # A tibble: 2 x 7  
## subject\_id redcap\_event\_na… start\_… g2\_unit operator chiller\_t1 subj\_min  
## <int> <fct> <fct> <int> <int> <dbl> <int>  
## 1 81 baseline\_and\_co… 2013-0… 0 1 28.1 30  
## 2 81 collection\_2\_ar… "" NA NA NA NA

# Clinical data for cases without start date   
print(tbl\_df(filter(sum\_clinical, field\_subj\_id == id)))

## # A tibble: 1 x 12  
## field\_subj\_id date\_visit visit\_name rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## <int> <date> <fct> <int> <int> <dbl>  
## 1 81 2013-01-22 visit\_1\_p… 0 0 37  
## # ... with 6 more variables: date\_on\_sx <fct>, visit\_num <dbl>,  
## # g2\_name <fct>, g2\_run <dbl>, enrolled <lgl>, clinical.i <lgl>

# Data Entry Error Correction  
# Remove empty record identified as having no start\_dt, subject\_id:  
g2\_log$subject\_id[which(g2\_log$start\_dt == "", arr.ind = TRUE)]

## [1] 81

g2\_log <- g2\_log %>%  
 filter(!(g2\_log$start\_dt) == "")  
  
# Number of rows in g2\_log data  
print(nrow(g2\_log))

## [1] 276

# Number of cols in g2\_log data  
print(ncol(g2\_log))

## [1] 87

# Number of subjects with a coll\_arm\_1   
print(sum(g2\_log$redcap\_event\_name == 'baseline\_and\_colle\_arm\_1'))

## [1] 178

# Number of subjects with a coll\_2 arm  
print(sum(g2\_log$redcap\_event\_name == 'collection\_2\_arm\_1'))

## [1] 71

# Number of subjects with a coll\_3 arm  
print(sum(g2\_log$redcap\_event\_name == 'collection\_3\_arm\_1'))

## [1] 27

g2\_log <- g2\_log %>%  
 rename(date\_visit = start\_dt)  
  
g2\_log\_min <- g2\_log %>%   
 select(subject\_id, redcap\_event\_name, date\_visit, subj\_min, cough\_number, sneeze\_number)  
# Not sure if we need to keep the cough\_count and subject\_min data in here.  
# However, note that at this point much of the cough data is missing   
# These mising cough values can be filled in based on the audio recordings.  
# Adding in the audio recording cough data is done elsewhere, but need to find where?  
# Here we kept in the cough\_number variable but it appears to be missing some of the values that have audio and haven't yet been added into the df.  
  
g2\_log\_min$g2\_coll\_num <- ifelse(g2\_log\_min$redcap\_event\_name == 'baseline\_and\_colle\_arm\_1', 1,   
 ifelse(g2\_log\_min$redcap\_event\_name == 'collection\_2\_arm\_1', 2, 3))   
  
# Numbers of subjects by g2 collection event  
print(ftable(addmargins(with(g2\_log\_min, table(redcap\_event\_name, g2\_coll\_num, exclude = c())))))

## g2\_coll\_num 1 2 3 Sum  
## redcap\_event\_name   
## baseline\_and\_colle\_arm\_1 178 0 0 178  
## collection\_2\_arm\_1 0 71 0 71  
## collection\_3\_arm\_1 0 0 27 27  
## Sum 178 71 27 276

# This is an important print out of the number of G2 collection events by visit number.  
  
g2\_log\_min <- g2\_log\_min %>%   
 select(subject\_id, date\_visit, g2\_coll\_num, subj\_min, cough\_number, sneeze\_number) %>%  
 rename(field\_subj\_id = subject\_id)  
  
g2\_log\_min$g2lm.i <- TRUE #indicator for preseence of record in g2\_log\_min  
  
g2\_log\_min$date\_visit <- as.Date(g2\_log\_min$date\_visit)  
  
# Check the variable names in g2\_log\_min  
print(head(tbl\_df(g2\_log\_min)))

## # A tibble: 6 x 7  
## field\_subj\_id date\_visit g2\_coll\_num subj\_min cough\_number sneeze\_number  
## <int> <date> <dbl> <int> <int> <int>  
## 1 7 2012-12-04 1 30 22 NA  
## 2 8 2012-12-04 1 30 42 NA  
## 3 13 2012-12-10 1 30 1 NA  
## 4 13 2012-12-11 2 30 0 NA  
## 5 13 2012-12-12 3 30 3 NA  
## 6 17 2012-12-11 1 30 125 NA  
## # ... with 1 more variable: g2lm.i <lgl>

g2\_log\_min\_subjectID\_count <- g2\_log\_min %>%  
 group\_by(field\_subj\_id) %>%  
 summarise (count = n())  
# 178 subjects  
  
# Check the number of sampling instances here  
g2\_log\_sampling\_instance\_check <- g2\_log\_min %>%  
 distinct(field\_subj\_id, date\_visit)  
print(nrow(g2\_log\_sampling\_instance\_check))

## [1] 276

# This shows that this df has information on the 276 instances of Gii runs for the 178 enrolled participants  
  
#### MERGE CLINICAL AND G2-LOG DATA ####  
  
# Here we will merge the sum\_clinical df and the g2\_log\_min df  
# These dfs were manipulated in the previous two sections of code in this script ...  
# ... in preparation for this merging step.  
  
merge1 <- merge(sum\_clinical, g2\_log\_min, by = c('field\_subj\_id', 'date\_visit'), all=TRUE)  
  
# Merge 1 check the dimensions agains the source dfs  
print(ftable(addmargins(with(merge1, table(clinical.i, g2lm.i, exclude = c())))))

## g2lm.i TRUE NA Sum  
## clinical.i   
## TRUE 276 199 475  
## Sum 276 199 475

# Select the variables of importance and their order.   
merge1 <- merge1 %>%   
 select(field\_subj\_id,   
 date\_visit,   
 date\_on\_sx,  
 g2\_coll\_num,   
 enrolled,   
 visit\_num,   
 g2\_run,   
 clinical.i,   
 g2lm.i,   
 rapid\_flu\_\_\_3,   
 rapid\_flu\_loc,   
 body\_temp,   
 cough\_number,  
 sneeze\_number)  
  
# Do the g2\_coll\_num match with the g2\_run? They should match.  
identical(merge1$g2\_coll\_num, merge1$g2\_run)

## [1] FALSE

# However they don't match! The next lines will address the discrepancies  
  
# If there is no g2 run then the collection number (g2\_coll\_num) and run number (g2\_run) ...  
# ... are set to zero here to match the g2\_run variable.  
merge1$g2\_coll\_num[is.na(merge1$g2\_coll\_num)] <- 0  
merge1$g2\_run[is.na(merge1$g2\_run)] <- 0  
merge1$indicator <- ifelse(merge1$g2\_coll\_num == merge1$g2\_run, 1, 0)   
i <- which(merge1$indicator == 0, arr.ind = TRUE)  
  
# Since g2\_coll\_num and g2\_run don't match, find the unmatched subject, ...  
# ... and show which row and column the subject(s) located:  
i

## [1] 81

# Subject that does not match field\_subj\_id  
merge1$field\_subj\_id[i]

## [1] 69

# Initiate data editing to resolve this discrepancy.  
  
## Data Editing ##  
  
# Subject 69 has data that doesn't match in the g2\_coll\_num and g2\_run variables (which should be the same)  
# The g2\_coll\_num variable comes from the clinical database df and the g2\_run var comes from the g2\_log df.   
  
merge1$field\_subj\_id[which(merge1$indicator == 0, arr.ind = TRUE)]

## [1] 69

# Subject 69 had a second g2 visit in clinical data, but did not provide sample -   
# See REDCap comments for details - removed from final analysis data sets  
merge1 <- merge1 %>%   
 filter(indicator == 1) %>%   
 select(-indicator)  
  
# Recheck after delection: Merge 1 record sources  
print(ftable(addmargins(with(merge1, table(clinical.i, g2lm.i, exclude = c())))))

## g2lm.i TRUE NA Sum  
## clinical.i   
## TRUE 276 198 474  
## Sum 276 198 474

# Number of rows in edited merged data  
print(nrow(merge1))

## [1] 474

# Number of subjects with a first visit  
print(sum(merge1$visit\_num == 1, na.rm = TRUE))

## [1] 355

# Number of subjects with a second screen  
print(sum(merge1$visit\_num == 2, na.rm = TRUE))

## [1] 20

# Number of subjects with a third screen  
print(sum(merge1$visit\_num == 3, na.rm = TRUE))

## [1] 1

# Number of unique subjects  
print(length(unique(merge1$field\_subj\_id)))

## [1] 355

# Number of screening visits  
sum(merge1$visit\_num == 1, na.rm = TRUE) +   
 sum(merge1$visit\_num == 2, na.rm = TRUE) +   
 sum(merge1$visit\_num == 3, na.rm = TRUE)

## [1] 376

# Number of subjects with a 1st g2 run ...  
# g2\_num:  
print(sum(merge1$g2\_run == 1))

## [1] 178

# g2\_coll\_num:  
print(sum(merge1$g2\_coll\_num == 1))

## [1] 178

# Number of subjects with a 2nd g2 run ...  
# g2\_num:  
print(sum(merge1$g2\_run == 2))

## [1] 71

# g2\_coll\_num:  
print(sum(merge1$g2\_coll\_num == 2))

## [1] 71

# Number of subjects with a 3rd g2 run  
# g2\_num:  
print(sum(merge1$g2\_run == 3))

## [1] 27

# g2\_coll\_num:   
print(sum(merge1$g2\_coll\_num == 3))

## [1] 27

# Total number of g2 runs  
# based on g2\_num:  
print(sum(!merge1$g2\_run == 0))

## [1] 276

# based on g2\_coll\_num:  
print(sum(!merge1$g2\_coll\_num == 0))

## [1] 276

# Total number of screenings without a g2 run whether or not later enrolled ...  
# ... based on g2\_num  
print(sum(merge1$g2\_run == 0))

## [1] 198

# ... based on g2\_coll\_num  
print(sum(merge1$g2\_coll\_num == 0))

## [1] 198

t1 <- sum(!merge1$g2\_run == 0) + sum(merge1$g2\_run == 0)  
t2 <- sum(!merge1$g2\_coll\_num == 0) + sum(merge1$g2\_coll\_num == 0)  
  
# Total number of encounters ...   
# ... based on g2\_run:  
t1

## [1] 474

# ... based on g2\_coll\_num  
t2

## [1] 474

merge1$merge1.i <- T #indicator for record in merge1  
  
# Cross tab tables of data in merge1  
print(ftable(addmargins(with(merge1, table(visit\_num, g2\_run, g2\_coll\_num, exclude = c())))))

## g2\_coll\_num 0 1 2 3 Sum  
## visit\_num g2\_run   
## 1 0 181 0 0 0 181  
## 1 0 174 0 0 174  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## Sum 181 174 0 0 355  
## 2 0 17 0 0 0 17  
## 1 0 3 0 0 3  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## Sum 17 3 0 0 20  
## 3 0 0 0 0 0 0  
## 1 0 1 0 0 1  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## Sum 0 1 0 0 1  
## 999 0 0 0 0 0 0  
## 1 0 0 0 0 0  
## 2 0 0 71 0 71  
## 3 0 0 0 27 27  
## Sum 0 0 71 27 98  
## Sum 0 198 0 0 0 198  
## 1 0 178 0 0 178  
## 2 0 0 71 0 71  
## 3 0 0 0 27 27  
## Sum 198 178 71 27 474

# Variables in merge1  
print(head(tbl\_df(merge1)))

## # A tibble: 6 x 15  
## field\_subj\_id date\_visit date\_on\_sx g2\_coll\_num enrolled visit\_num g2\_run  
## <int> <date> <fct> <dbl> <lgl> <dbl> <dbl>  
## 1 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 2 8 2012-12-04 2012-12-02 1 TRUE 1 1  
## 3 9 2012-12-05 2012-12-03 0 FALSE 1 0  
## 4 10 2012-12-05 2012-12-02 0 FALSE 1 0  
## 5 12 2012-12-10 2012-12-07 0 FALSE 1 0  
## 6 13 2012-12-10 2012-12-08 1 TRUE 1 1  
## # ... with 8 more variables: clinical.i <lgl>, g2lm.i <lgl>,  
## # rapid\_flu\_\_\_3 <int>, rapid\_flu\_loc <int>, body\_temp <dbl>,  
## # cough\_number <int>, sneeze\_number <int>, merge1.i <lgl>

#### READ in and work with the FIELD SAMPLE DATABASE ####  
  
# Input Field Sample Data  
field\_db\_in\_file <- 'UMD\_Raw\_Data/EMIT UMD Field\_db/field\_db.csv'  
field.db <- read.csv(field\_db\_in\_file, as.is = T)  
  
print(head(tbl\_df(field.db)))

## # A tibble: 6 x 7  
## SUBJECT\_IDENTIF… SAMPLE\_ID COLLECTION\_DT ID TYPE\_NAME UNIT\_NAME  
## <int> <chr> <chr> <int> <chr> <chr>   
## 1 7 7\_1 12/4/2012 1 Nasophar… count   
## 2 7 7\_2 12/4/2012 2 Impactor… count   
## 3 7 7\_3 12/4/2012 3 GII cond… mililite…  
## 4 7 7\_4 12/4/2012 4 anterior… count   
## 5 7 7\_5 12/4/2012 5 Throat S… count   
## 6 8 8\_1 12/4/2012 1 Nasophar… count   
## # ... with 1 more variable: RAPID\_TEST <lgl>

field.db1 <- field.db %>%   
 select(SUBJECT\_IDENTIFIER, SAMPLE\_ID, COLLECTION\_DT, TYPE\_NAME) %>%  
 rename(field\_subj\_id = SUBJECT\_IDENTIFIER) %>%  
 rename(sample\_id = SAMPLE\_ID) %>%  
 rename(date\_visit = COLLECTION\_DT) %>%  
 rename(sample\_type = TYPE\_NAME)  
field.db1$field.db1.i <- TRUE #indicator that data is in field.db1  
  
# Input field sample field.db file  
field\_db\_in\_file

## [1] "UMD\_Raw\_Data/EMIT UMD Field\_db/field\_db.csv"

# Number of rows in field sample database  
print(nrow(field.db1))

## [1] 2760

# Number of columns in database (selected columns)  
ncol(field.db1)

## [1] 5

# Tabluation of number of rows by sample type in field database  
print(addmargins(with(field.db1, table(sample\_type, field.db1.i, exclude = c()))))

## field.db1.i  
## sample\_type TRUE Sum  
## 20 20  
## anterior nasal swab 386 386  
## GII condensate NO mask 479 479  
## Impactor 5 um NO mask 479 479  
## Nasopharyngeal swab 916 916  
## Throat Swab 480 480  
## Sum 2760 2760

# Number of rows that have a missing sample\_type  
print(sum(field.db1$sample\_type == ""))

## [1] 20

# Number of rows that have a missing date\_visit  
print(sum(field.db1$date\_visit == ""))

## [1] 0

# Number of rows that have a NA for date\_visit  
print(sum(is.na(field.db1$date\_visit)))

## [1] 0

# Number of subjects that have a missing sample\_id  
print(sum(field.db1$sample\_id == ""))

## [1] 0

# Number of subjects that have a NA for sample\_id  
print(sum(is.na(field.db1$sample\_id)))

## [1] 0

field.db1$date\_visit <- as.Date(field.db1$date\_visit, format = "%m/%d/%Y")  
  
## Data Editing ##  
  
# Field\_subj\_id 225 was moved to field\_subj\_id 250 in clinical database because second screening visit was ...  
# ... erroneously given a new ID number. However the samples are still shown in the field database as 225.  
# Therefore, I am recoding the subject id to 250 but leaving the sample\_id as 225\_x -- at least for now  
  
field.db1$field\_subj\_id <- with(field.db1, ifelse(field\_subj\_id == 225, 250, field\_subj\_id))  
print(tbl\_df(filter(field.db1, field\_subj\_id == 250)))

## # A tibble: 12 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 250 225\_1 2013-02-11 Nasopharyngeal swab TRUE   
## 2 250 225\_2 2013-02-11 Impactor 5 um NO mask TRUE   
## 3 250 225\_3 2013-02-11 GII condensate NO mask TRUE   
## 4 250 225\_4 2013-02-11 anterior nasal swab TRUE   
## 5 250 225\_5 2013-02-11 Throat Swab TRUE   
## 6 250 225\_6 2013-02-11 Nasopharyngeal swab TRUE   
## 7 250 250\_1 2013-02-13 Nasopharyngeal swab TRUE   
## 8 250 250\_2 2013-02-13 Impactor 5 um NO mask TRUE   
## 9 250 250\_3 2013-02-13 GII condensate NO mask TRUE   
## 10 250 250\_4 2013-02-13 anterior nasal swab TRUE   
## 11 250 250\_5 2013-02-13 Throat Swab TRUE   
## 12 250 250\_6 2013-02-13 Nasopharyngeal swab TRUE

# Recode date\_visit subj 10 sample\_id = 10\_6 to be correct date of 2012-12-05  
  
# orginal data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 10)))

## # A tibble: 3 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 10 10\_1 2012-12-05 Nasopharyngeal swab TRUE   
## 2 10 10\_5 2012-12-05 Throat Swab TRUE   
## 3 10 10\_6 2012-12-07 Nasopharyngeal swab TRUE

field.db1$date\_visit <- with(field.db1,   
 ifelse(field\_subj\_id == 10 & date\_visit == as.Date("2012-12-07"),   
 as.Date("2012-12-05"), date\_visit))  
field.db1$date\_visit <- as.Date(field.db1$date\_visit, format = "%Y-%m-%d", origin = "1970-01-01")  
  
# recoded data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 10)))

## # A tibble: 3 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 10 10\_1 2012-12-05 Nasopharyngeal swab TRUE   
## 2 10 10\_5 2012-12-05 Throat Swab TRUE   
## 3 10 10\_6 2012-12-05 Nasopharyngeal swab TRUE

# Recode date\_visit subj 12 sample\_id = 12\_6 to the correct date of 2012-12-10  
  
# orginal data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 12)))

## # A tibble: 6 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 12 12\_1 2012-12-10 Nasopharyngeal swab TRUE   
## 2 12 12\_2 2012-12-10 Impactor 5 um NO mask TRUE   
## 3 12 12\_3 2012-12-10 GII condensate NO mask TRUE   
## 4 12 12\_4 2012-12-10 anterior nasal swab TRUE   
## 5 12 12\_5 2012-12-10 Throat Swab TRUE   
## 6 12 12\_6 2013-02-08 Nasopharyngeal swab TRUE

field.db1$date\_visit <- with(field.db1,   
 ifelse(field\_subj\_id == 12 & date\_visit == as.Date("2013-02-08"), as.Date("2012-12-10"), date\_visit))  
field.db1$date\_visit <- as.Date(field.db1$date\_visit, format = "%Y-%m-%d", origin = "1970-01-01")  
  
# recoded data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 12)))

## # A tibble: 6 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 12 12\_1 2012-12-10 Nasopharyngeal swab TRUE   
## 2 12 12\_2 2012-12-10 Impactor 5 um NO mask TRUE   
## 3 12 12\_3 2012-12-10 GII condensate NO mask TRUE   
## 4 12 12\_4 2012-12-10 anterior nasal swab TRUE   
## 5 12 12\_5 2012-12-10 Throat Swab TRUE   
## 6 12 12\_6 2012-12-10 Nasopharyngeal swab TRUE

# There was no subject 11, 28, 53, 73, or 76; Samples were generated in error.  
# Delete samples for non-existant subjects  
field.db1 <- filter(field.db1, field\_subj\_id != 11)%>%  
 filter(field\_subj\_id != 28) %>%  
 filter(field\_subj\_id != 53) %>%  
 filter(field\_subj\_id != 73) %>%   
 filter(field\_subj\_id != 76)  
  
# There was no second visit for subj 30; Samples generated in error.  
# Delete samples for subject 30 on 2012-12-18.   
  
# Original data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 30)))

## # A tibble: 12 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 30 30\_1 2012-12-17 Nasopharyngeal swab TRUE   
## 2 30 30\_2 2012-12-17 Impactor 5 um NO mask TRUE   
## 3 30 30\_3 2012-12-17 GII condensate NO mask TRUE   
## 4 30 30\_4 2012-12-17 anterior nasal swab TRUE   
## 5 30 30\_5 2012-12-17 Throat Swab TRUE   
## 6 30 30\_7 2012-12-17 Nasopharyngeal swab TRUE   
## 7 30 30\_8 2012-12-18 GII condensate NO mask TRUE   
## 8 30 30\_9 2012-12-18 Nasopharyngeal swab TRUE   
## 9 30 30\_10 2012-12-18 Nasopharyngeal swab TRUE   
## 10 30 30\_11 2012-12-18 Impactor 5 um NO mask TRUE   
## 11 30 30\_12 2012-12-18 Throat Swab TRUE   
## 12 30 30\_13 2012-12-18 anterior nasal swab TRUE

# Make correction  
field.db1 <- field.db1 %>%  
 filter(!(field\_subj\_id == 30 & date\_visit == as.Date("2012-12-18")))   
  
# Corrected data subject 30  
print(tbl\_df(filter(field.db1, field\_subj\_id == 30)))

## # A tibble: 6 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 30 30\_1 2012-12-17 Nasopharyngeal swab TRUE   
## 2 30 30\_2 2012-12-17 Impactor 5 um NO mask TRUE   
## 3 30 30\_3 2012-12-17 GII condensate NO mask TRUE   
## 4 30 30\_4 2012-12-17 anterior nasal swab TRUE   
## 5 30 30\_5 2012-12-17 Throat Swab TRUE   
## 6 30 30\_7 2012-12-17 Nasopharyngeal swab TRUE

# There was no second visit for subj 120 on 2013-02-08 and sample 120\_12 is not in REDCap sample database  
  
# Original data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 120)))

## # A tibble: 12 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 120 120\_1 2013-01-31 Nasopharyngeal swab TRUE   
## 2 120 120\_2 2013-01-31 Impactor 5 um NO mask TRUE   
## 3 120 120\_3 2013-01-31 GII condensate NO mask TRUE   
## 4 120 120\_4 2013-01-31 anterior nasal swab TRUE   
## 5 120 120\_5 2013-01-31 Throat Swab TRUE   
## 6 120 120\_6 2013-01-31 Nasopharyngeal swab TRUE   
## 7 120 120\_7 2013-02-01 Nasopharyngeal swab TRUE   
## 8 120 120\_8 2013-02-01 Nasopharyngeal swab TRUE   
## 9 120 120\_9 2013-02-01 GII condensate NO mask TRUE   
## 10 120 120\_10 2013-02-01 Throat Swab TRUE   
## 11 120 120\_11 2013-02-01 Impactor 5 um NO mask TRUE   
## 12 120 120\_12 2013-02-08 Nasopharyngeal swab TRUE

# Delete sample 120\_12  
field.db1 <- filter(field.db1, sample\_id != "120\_12")  
  
# Corrected data  
print(tbl\_df(filter(field.db1, field\_subj\_id == 120)))

## # A tibble: 11 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 120 120\_1 2013-01-31 Nasopharyngeal swab TRUE   
## 2 120 120\_2 2013-01-31 Impactor 5 um NO mask TRUE   
## 3 120 120\_3 2013-01-31 GII condensate NO mask TRUE   
## 4 120 120\_4 2013-01-31 anterior nasal swab TRUE   
## 5 120 120\_5 2013-01-31 Throat Swab TRUE   
## 6 120 120\_6 2013-01-31 Nasopharyngeal swab TRUE   
## 7 120 120\_7 2013-02-01 Nasopharyngeal swab TRUE   
## 8 120 120\_8 2013-02-01 Nasopharyngeal swab TRUE   
## 9 120 120\_9 2013-02-01 GII condensate NO mask TRUE   
## 10 120 120\_10 2013-02-01 Throat Swab TRUE   
## 11 120 120\_11 2013-02-01 Impactor 5 um NO mask TRUE

# Delete samples (NP only) from erroneous second g2 visit for subject 69 (see above)  
field.db1 <- filter(field.db1, sample\_id != "69\_6" & sample\_id != "69\_7")  
  
## End of Data Editing Field Sample Database ##  
  
# Number of columns in EDITED filed sample database (selected columns)  
print(ncol(field.db1))

## [1] 5

# Variable names in field.db1  
print(head(tbl\_df(field.db1)))

## # A tibble: 6 x 5  
## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## <dbl> <chr> <date> <chr> <lgl>   
## 1 7 7\_1 2012-12-04 Nasopharyngeal swab TRUE   
## 2 7 7\_2 2012-12-04 Impactor 5 um NO mask TRUE   
## 3 7 7\_3 2012-12-04 GII condensate NO mask TRUE   
## 4 7 7\_4 2012-12-04 anterior nasal swab TRUE   
## 5 7 7\_5 2012-12-04 Throat Swab TRUE   
## 6 8 8\_1 2012-12-04 Nasopharyngeal swab TRUE

# How many subjectIDs are represented in the field.db1?  
field.db1\_subjectID\_check <- field.db1 %>%  
 group\_by(field\_subj\_id) %>%  
 count()  
print(nrow(field.db1\_subjectID\_check))

## [1] 355

# Data for 355 individuals - all samples collected - including clinic and gii  
  
#### MERGE FIELD SAMPLE DATABASE WITH COMBINED CLINICAL DATABASE & G2 LOG ####  
  
## Merge2 = merge of merge1 with field.db1 by field\_subj\_id and date\_visit ##  
merge2 <- merge(merge1, field.db1, by = c("field\_subj\_id", "date\_visit"), all = T)  
  
# Check number of subjectIDs now - should still be 355  
merge2\_subjectID\_check <- merge2 %>%  
 group\_by(field\_subj\_id) %>%  
 count()  
print(nrow(merge2\_subjectID\_check))

## [1] 355

# Yes, this seems to be correct.   
  
# Head of merge2  
print(head(tbl\_df(merge2)))

## # A tibble: 6 x 18  
## field\_subj\_id date\_visit date\_on\_sx g2\_coll\_num enrolled visit\_num g2\_run  
## <int> <date> <fct> <dbl> <lgl> <dbl> <dbl>  
## 1 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 2 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 3 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 4 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 5 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 6 8 2012-12-04 2012-12-02 1 TRUE 1 1  
## # ... with 11 more variables: clinical.i <lgl>, g2lm.i <lgl>,  
## # rapid\_flu\_\_\_3 <int>, rapid\_flu\_loc <int>, body\_temp <dbl>,  
## # cough\_number <int>, sneeze\_number <int>, merge1.i <lgl>,  
## # sample\_id <chr>, sample\_type <chr>, field.db1.i <lgl>

# Source of data in rows of merge2  
print(ftable(addmargins(with(merge2, table(merge1.i, field.db1.i, exclude = c())))))

## field.db1.i TRUE NA Sum  
## merge1.i   
## TRUE 2724 1 2725  
## Sum 2724 1 2725

# Remove all samples that were assigned in the field db but not collected from the unenrolled subjects  
merge2 <- merge2 %>%   
 filter(!(enrolled == F & sample\_type %in%   
 c("GII condensate NO mask", "Throat Swab", "Impactor 5 um NO mask", "anterior nasal swab")))  
  
# Source of data in rows of merge2 after removing extraneous samples  
print(ftable(addmargins(with(merge2, table(merge1.i, field.db1.i, exclude = c())))))

## field.db1.i TRUE NA Sum  
## merge1.i   
## TRUE 1938 1 1939  
## Sum 1938 1 1939

# Merge2: rows where merge1 was not matched by rows from field.db1  
print(filter(merge2, is.na(field.db1.i)))

## field\_subj\_id date\_visit date\_on\_sx g2\_coll\_num enrolled visit\_num  
## 1 135 2013-02-05 <NA> 0 FALSE 2  
## g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## 1 0 TRUE NA 0 0 36.5  
## cough\_number sneeze\_number merge1.i sample\_id sample\_type field.db1.i  
## 1 NA NA TRUE <NA> <NA> NA

# All rows in merge2 for subjects that have some merge1 rows not matching field.db1  
x <- distinct(select(filter(merge2, is.na(field.db1.i)), field\_subj\_id))  
print(inner\_join(merge2, x, by = "field\_subj\_id"))

## field\_subj\_id date\_visit date\_on\_sx g2\_coll\_num enrolled visit\_num  
## 1 135 2013-02-02 <NA> 0 FALSE 1  
## 2 135 2013-02-02 <NA> 0 FALSE 1  
## 3 135 2013-02-05 <NA> 0 FALSE 2  
## g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## 1 0 TRUE NA 0 0 37.0  
## 2 0 TRUE NA 0 0 37.0  
## 3 0 TRUE NA 0 0 36.5  
## cough\_number sneeze\_number merge1.i sample\_id sample\_type  
## 1 NA NA TRUE 135\_1 Nasopharyngeal swab  
## 2 NA NA TRUE 135\_6 Nasopharyngeal swab  
## 3 NA NA TRUE <NA> <NA>  
## field.db1.i  
## 1 TRUE  
## 2 TRUE  
## 3 NA

# All rows in field.db1 for subjects that had some merge1 rows not matching field.db1  
print(inner\_join(field.db1, x, by = "field\_subj\_id"))

## field\_subj\_id sample\_id date\_visit sample\_type field.db1.i  
## 1 135 135\_1 2013-02-02 Nasopharyngeal swab TRUE  
## 2 135 135\_2 2013-02-02 Impactor 5 um NO mask TRUE  
## 3 135 135\_3 2013-02-02 GII condensate NO mask TRUE  
## 4 135 135\_4 2013-02-02 anterior nasal swab TRUE  
## 5 135 135\_5 2013-02-02 Throat Swab TRUE  
## 6 135 135\_6 2013-02-02 Nasopharyngeal swab TRUE

# Subject 135 returned for a second screening visit but was never enrolled.   
# There is nothing in the REDCap clinical record to explain why no samples in the field DB were associated with the second visit.  
# There are also no samples in the REDCap sample database for subject 135 except 135\_1.  
  
# As a result, we delete field\_subj\_id == 135 & date\_visit == 2013-02-05   
merge2 <- merge2 %>%  
 filter(!(field\_subj\_id == 135 & date\_visit == "2013-02-05"))  
  
# Checking merge2 rows where field.db1 not matched by rows from merge1  
print(filter(merge2, is.na(merge1.i)))

## [1] field\_subj\_id date\_visit date\_on\_sx g2\_coll\_num enrolled   
## [6] visit\_num g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3  
## [11] rapid\_flu\_loc body\_temp cough\_number sneeze\_number merge1.i   
## [16] sample\_id sample\_type field.db1.i   
## <0 rows> (or 0-length row.names)

# Checking: merge1 rows for subjects who had some field.db1 rows not matching merge1 rows  
x <- distinct(select(filter(merge2, is.na(merge1.i)), field\_subj\_id))  
nrow(x)

## [1] 0

# Checking: Table of these rows  
print(tbl\_df(inner\_join(merge1, x, by = "field\_subj\_id")))

## # A tibble: 0 x 15  
## # ... with 15 variables: field\_subj\_id <int>, date\_visit <date>,  
## # date\_on\_sx <fct>, g2\_coll\_num <dbl>, enrolled <lgl>, visit\_num <dbl>,  
## # g2\_run <dbl>, clinical.i <lgl>, g2lm.i <lgl>, rapid\_flu\_\_\_3 <int>,  
## # rapid\_flu\_loc <int>, body\_temp <dbl>, cough\_number <int>,  
## # sneeze\_number <int>, merge1.i <lgl>

# Source of data in rows of merge2 after removing 135\_6.  
print(ftable(addmargins(with(merge2, table(merge1.i, field.db1.i, exclude = c())))))

## field.db1.i TRUE Sum  
## merge1.i   
## TRUE 1938 1938  
## Sum 1938 1938

# Giving an indicator variable to this finalized merge2 df  
merge2$merge2.i <- T  
  
# Head of merge2  
print(head(tbl\_df(merge2)))

## # A tibble: 6 x 19  
## field\_subj\_id date\_visit date\_on\_sx g2\_coll\_num enrolled visit\_num g2\_run  
## <int> <date> <fct> <dbl> <lgl> <dbl> <dbl>  
## 1 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 2 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 3 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 4 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 5 7 2012-12-04 2012-12-02 1 TRUE 1 1  
## 6 8 2012-12-04 2012-12-02 1 TRUE 1 1  
## # ... with 12 more variables: clinical.i <lgl>, g2lm.i <lgl>,  
## # rapid\_flu\_\_\_3 <int>, rapid\_flu\_loc <int>, body\_temp <dbl>,  
## # cough\_number <int>, sneeze\_number <int>, merge1.i <lgl>,  
## # sample\_id <chr>, sample\_type <chr>, field.db1.i <lgl>, merge2.i <lgl>

# Repeating from above now that data manipulation has been done: Check number of subjectIDs now - should still be 355  
merge2\_subjectID\_check <- merge2 %>%  
 group\_by(field\_subj\_id) %>%  
 count()  
print(nrow(merge2\_subjectID\_check))

## [1] 355

# Yes, this seems to be correct.   
  
#### READ in and work with the UMD SAMPLES DATABASE (REDCAP DATA) ####  
sample\_in\_file <- 'UMD\_Raw\_Data/REDCAP/EMITUMDSamples2013\_DATA.csv'  
sample\_in <- read.csv(sample\_in\_file, as.is = T)  
  
sample\_in$count\_tech <- as.factor(sample\_in$count\_tech)  
  
# Input UMD samples file (from REDCap)  
sample\_in\_file

## [1] "UMD\_Raw\_Data/REDCAP/EMITUMDSamples2013\_DATA.csv"

# Check number of subjectIDs now  
sample\_in\_subjectID\_check <- sample\_in %>%  
 group\_by(field\_subj\_id) %>%  
 filter(!is.na(field\_subj\_id)) %>%  
 summarise(count = n())  
print(nrow(sample\_in\_subjectID\_check))

## [1] 355

# Still data for all 355 screened participants.  
  
# Number of rows  
print(nrow(sample\_in))

## [1] 2657

# Number of cols  
print(ncol(sample\_in))

## [1] 42

sample\_in <- sample\_in %>%  
 rename(date\_visit = dt\_visit)  
sample\_in$date\_visit <- as.Date(sample\_in$date\_visit, format = "%m/%d/%Y")  
  
# Head of sample\_in dataframe after renaming date\_visit and setting count\_tech as factor  
print(tbl\_df(sample\_in))

## # A tibble: 2,657 x 42  
## sample\_id redcap\_event\_na… field\_subj\_id sample\_type date\_visit  
## <chr> <chr> <int> <chr> <date>   
## 1 7\_1 collection\_arm\_1 7 Nasopharyn… 2012-12-04  
## 2 7\_1 assay1\_arm\_1 NA "" NA   
## 3 7\_2 collection\_arm\_1 7 Impactor 5… 2012-12-04  
## 4 7\_2 assay1\_arm\_1 NA "" NA   
## 5 7\_3 collection\_arm\_1 7 GII conden… 2012-12-04  
## 6 7\_3 assay1\_arm\_1 NA "" NA   
## 7 7\_4 collection\_arm\_1 7 anterior n… 2012-12-04  
## 8 7\_5 collection\_arm\_1 7 Throat Swab 2012-12-04  
## 9 7\_5 assay1\_arm\_1 NA "" NA   
## 10 8\_1 collection\_arm\_1 8 Nasopharyn… 2012-12-04  
## # ... with 2,647 more rows, and 37 more variables: numb\_aliquots <lgl>,  
## # volume <lgl>, collection\_complete <int>, passage\_id <chr>,  
## # passage\_id\_problem <int>, dt\_pass <chr>, pass\_tech <int>,  
## # passage\_1 <int>, passage\_2 <int>, dt\_pass\_2 <chr>,  
## # passage\_complete <int>, focus\_id <chr>, focus\_id\_problem <int>,  
## # dt\_focus <chr>, dilution\_factor <int>, dt\_stained <chr>,  
## # dt\_count <chr>, count\_tech <fct>, plate\_type <int>, count\_meth <int>,  
## # grid\_1 <int>, grid\_2 <int>, grid\_3 <int>, grid\_4 <int>, grid\_5 <int>,  
## # grid\_6 <int>, grid\_7 <int>, grid\_8 <int>, grid\_9 <int>, grid\_10 <int>,  
## # well <int>, focus\_complete <int>, pcr\_id <chr>, pcr\_id\_problem <lgl>,  
## # assay\_number <dbl>, dt\_pcr <chr>, pcr\_complete <int>

# Number of rows that have a missing sample\_id  
print(sum(sample\_in$sample\_id == ""))

## [1] 0

# Number of rows that have a NA for sample\_id  
print(sum(is.na(sample\_in$sample\_id)))

## [1] 0

## Data Editing ##  
# Samples 69\_6 and 69\_7 collected for a g2\_run = 2 that was not performed. See above data editing section. Deleted here.  
# Also, samples 20-1 & 97-9 are typos and duplications. They are also deleted here.  
  
sample\_in <- sample\_in %>%  
 filter(!(sample\_id %in% c("69\_6", "69\_7", "20-1", "97-9")))  
sample\_in$dt\_stained <- with(sample\_in, ifelse(sample\_id == "20\_1" & dt\_count == "12/15/2012", "12/15/2012", dt\_stained))  
  
collection <- sample\_in %>%  
 filter(redcap\_event\_name == "collection\_arm\_1") %>%  
 select(sample\_id, field\_subj\_id, sample\_type, date\_visit)  
  
assay1 <- sample\_in %>%   
 filter(grepl('^assay1', redcap\_event\_name))  
  
assay2 <- sample\_in %>%   
 filter(grepl('^assay2', redcap\_event\_name))  
  
# Number of collection records  
print(nrow(collection))

## [1] 1696

# Number of assay 1 records  
print(nrow(assay1))

## [1] 862

# Number of assay 2 records  
print(nrow(assay2))

## [1] 95

# Event name and sample\_type read in from REDCap sample database  
print(ftable(addmargins(with(sample\_in, table(redcap\_event\_name, sample\_type, exclude = c())))))

## sample\_type anterior nasal swab GII condensate NO mask Impactor 5 um NO mask Nasopharyngeal swab Throat Swab Sum  
## redcap\_event\_name   
## assay1\_arm\_1 862 0 0 0 0 0 862  
## assay2\_arm\_1 95 0 0 0 0 0 95  
## collection\_arm\_1 0 186 276 276 682 276 1696  
## Sum 957 186 276 276 682 276 2653

# Head of sample\_in samples with no sample\_type  
print(head(filter(sample\_in, sample\_type == "")))

## sample\_id redcap\_event\_name field\_subj\_id sample\_type date\_visit  
## 1 7\_1 assay1\_arm\_1 NA <NA>  
## 2 7\_2 assay1\_arm\_1 NA <NA>  
## 3 7\_3 assay1\_arm\_1 NA <NA>  
## 4 7\_5 assay1\_arm\_1 NA <NA>  
## 5 8\_1 assay1\_arm\_1 NA <NA>  
## 6 8\_2 assay1\_arm\_1 NA <NA>  
## numb\_aliquots volume collection\_complete passage\_id passage\_id\_problem  
## 1 NA NA NA 7\_1 NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA 7\_5 NA  
## 5 NA NA NA 8\_1 NA  
## 6 NA NA NA NA  
## dt\_pass pass\_tech passage\_1 passage\_2 dt\_pass\_2 passage\_complete  
## 1 12/12/2012 1 1 2 12/17/2012 2  
## 2 NA NA NA 0  
## 3 NA NA NA 0  
## 4 12/12/2012 1 1 2 12/17/2012 2  
## 5 12/12/2012 1 1 2 12/17/2013 2  
## 6 NA NA NA 0  
## focus\_id focus\_id\_problem dt\_focus dilution\_factor dt\_stained dt\_count  
## 1 7\_1 NA 0 12/15/2012  
## 2 NA NA   
## 3 7\_3 NA 0 12/15/2012  
## 4 7\_5 NA 0 12/15/2012  
## 5 NA NA   
## 6 NA NA   
## count\_tech plate\_type count\_meth grid\_1 grid\_2 grid\_3 grid\_4 grid\_5  
## 1 1 2 1 0 0 0 NA NA  
## 2 <NA> NA NA NA NA NA NA NA  
## 3 1 2 1 0 0 0 NA NA  
## 4 1 2 1 0 0 0 NA NA  
## 5 <NA> NA NA NA NA NA NA NA  
## 6 <NA> NA NA NA NA NA NA NA  
## grid\_6 grid\_7 grid\_8 grid\_9 grid\_10 well focus\_complete pcr\_id  
## 1 NA NA NA NA NA NA 2 7\_1  
## 2 NA NA NA NA NA NA 0 7\_2  
## 3 NA NA NA NA NA NA 2 7\_3  
## 4 NA NA NA NA NA NA 2   
## 5 NA NA NA NA NA NA 0 8\_1  
## 6 NA NA NA NA NA NA 0 8\_2  
## pcr\_id\_problem assay\_number dt\_pcr pcr\_complete  
## 1 NA 13.088 1/17/2014 2  
## 2 NA 13.110 2/17/2014 2  
## 3 NA 13.110 2/17/2014 2  
## 4 NA NA 0  
## 5 NA 13.088 1/17/2014 2  
## 6 NA 13.110 2/17/2014 2

passage <- select(filter(sample\_in, !is.na(passage\_1)),   
 sample\_id,   
 passage\_id,   
 passage\_id\_problem,   
 dt\_pass,   
 pass\_tech,   
 passage\_1,   
 passage\_2,   
 dt\_pass\_2,   
 passage\_complete)  
  
## PROBLEM OBSERVATIONS THAT NEED EDITING ##  
  
# Sample\_id does not match Passage\_id. Refer for lab review.   
# Meanwhile, will use sample\_id as it is not duplicated.  
print(tbl\_df(filter(passage, sample\_id != passage\_id)))

## # A tibble: 1 x 9  
## sample\_id passage\_id passage\_id\_prob… dt\_pass pass\_tech passage\_1  
## <chr> <chr> <int> <chr> <int> <int>  
## 1 38\_3 38\_1 NA 12/29/… 1 1  
## # ... with 3 more variables: passage\_2 <int>, dt\_pass\_2 <chr>,  
## # passage\_complete <int>

passage$passpos <- (passage$passage\_1 == 2 | passage$passage\_2 == 2) # Passage is + if either passage is +  
passage$validp <- !is.na(passage$passpos)  
passage <- passage %>%  
 select(sample\_id, passpos, validp)  
  
# Initial look at passage assays  
# Number of passage assays  
print(nrow(passage))

## [1] 666

# Number of passage assays with missing date of passage  
print(sum(passage$dt\_pass == ""))

## [1] 0

# Number of valid passage assays  
print(sum(passage$validp))

## [1] 618

# Number of invalid passage assays  
print(sum(!passage$validp))

## [1] 48

# Number of positive passage assays  
print(sum(passage$passpos, na.rm = T))

## [1] 405

# Number of negative passage assays  
print(sum(!passage$passpos, na.rm = T))

## [1] 213

focus1 <- filter(assay1, !dt\_count == "")[ , c(1, 17:37)]  
  
focus2 <- filter(assay2, !dt\_count == "")[ , c(1, 17:37)]  
  
# Focus Assays  
  
# Samples with miss match sample\_id and focus\_id focus1  
print(sum(!(focus1$sample\_id == focus1$focus\_id)))

## [1] 0

# Samples with miss match sample\_id and focus\_id focus2  
print(sum(!(focus2$sample\_id == focus2$focus\_id)))

## [1] 0

# Number of focus1 counts  
print(nrow(focus1))

## [1] 655

# Number of focus2 counts  
print(nrow(focus2))

## [1] 85

# Number of focus1 rows with no dt\_count  
print(sum(focus1$dt\_count == ""))

## [1] 0

# Number of focus2 rows with no dt\_count  
print(sum(focus2$dt\_count == ""))

## [1] 0

#### Computation of focus assay results ####  
  
focus1$df <- 10^(ifelse(is.na(focus1$dilution\_factor), 0, focus1$dilution\_factor))  
  
focus2$df <- 10^(ifelse(is.na(focus2$dilution\_factor), 0, focus2$dilution\_factor))  
  
area.24 <- pi\*(15.4/2)^2  
area.g <- 0.64  
  
focus1$ct\_24g <- rowSums(focus1[ , c(11:20)], na.rm = T) / (10\*area.g)\*area.24\*focus1$df / 150\*1000  
focus1$ct\_24w <- focus1$well\*focus1$df / 150\*1000  
focus1$ct\_96 <- rowSums(focus1[ , c(11:13)], na.rm = T)\*focus1$df / 150\*1000  
  
focus1\_24g <- focus1 %>%  
 filter((focus1$plate\_type == 1 | is.na(focus1$plate\_type)) & focus1$count\_meth == 1) %>%   
 select(-ct\_96, -ct\_24w) %>%  
 rename(ct = ct\_24g)  
  
focus1\_24w <- focus1 %>%   
 filter((focus1$plate\_type == 1 | is.na(focus1$plate\_type)) & focus1$count\_meth == 2) %>%   
 select(-ct\_96, -ct\_24g) %>%  
 rename(ct = ct\_24w)  
  
focus1\_96 <- focus1 %>%  
 filter(focus1$plate\_type == 2) %>%   
 select(-ct\_24w, -ct\_24g) %>%  
 rename(ct = ct\_96)  
  
focus1\_c <- arrange(rbind(focus1\_96, focus1\_24w, focus1\_24g))  
  
focus2$ct\_24g <- rowSums(focus2[ , c(11:20)], na.rm = T) / (10\*area.g)\*area.24\*focus2$df / 150\*1000  
  
focus2$ct\_24w <- focus2$well\*focus2$df / 150\*1000  
  
focus2$ct\_96 <- rowSums(focus2[ , c(11:13)], na.rm = T)\*focus2$df / 150\*1000  
  
focus2\_24g <- focus2 %>%  
 filter((focus2$plate\_type == 1 | is.na(focus2$plate\_type)) & focus2$count\_meth == 1) %>%   
 select(-ct\_96, -ct\_24w) %>%  
 rename(ct = ct\_24g)  
  
focus2\_24w <- focus2 %>%  
 filter((focus2$plate\_type == 1 | is.na(focus2$plate\_type)) & focus2$count\_meth == 2) %>%   
 select(-ct\_96, -ct\_24g) %>%  
 rename(ct = ct\_24w)  
  
focus2\_96 <- focus2 %>%   
 filter(focus2$plate\_type == 2) %>%   
 select(-ct\_24w, -ct\_24g) %>%  
 rename(ct = ct\_96)  
  
focus2\_c <- arrange(rbind(focus2\_96, focus2\_24w, focus2\_24g))  
  
focus1\_c <- select(focus1\_c, sample\_id, dt\_count, count\_tech, ct)  
  
focus2\_c <- select(focus2\_c, sample\_id, dt\_count, count\_tech, ct)  
  
focus <- merge(focus1\_c, focus2\_c, by = "sample\_id", all = T)  
focus$ct <- rowMeans(cbind(focus$ct.x,focus$ct.y), na.rm = T)  
  
missing\_focus <- tbl\_df(filter(focus, is.nan(ct)))  
focus\_allv <- focus  
focus <- focus %>%  
 select(sample\_id, ct)  
  
## FOCUS ASSAY RESULTS ##  
  
# Samples listed as having a focus assay but without results  
print(missing\_focus)

## # A tibble: 1 x 8  
## sample\_id dt\_count.x count\_tech.x ct.x dt\_count.y count\_tech.y ct.y  
## <chr> <chr> <fct> <dbl> <chr> <fct> <dbl>  
## 1 153\_1 6/6/2013 1 NA <NA> <NA> NA  
## # ... with 1 more variable: ct <dbl>

summary(focus)

## sample\_id ct   
## Length:655 Min. : 0.0   
## Class :character 1st Qu.: 0.0   
## Mode :character Median : 0.0   
## Mean : 7564.3   
## 3rd Qu.: 73.3   
## Max. :514169.1   
## NA's :1

#### MERGE CULTURE RESULTS PIECE FROM FIELD SAMPLE DATABASE TO THE CUMULATIVE CLIN DB + G2 LOG + FIELD SAMPLE DB ####  
  
# How many subjectIDs in the collection df?  
collection\_subjectID\_check <- collection %>%  
 group\_by(field\_subj\_id) %>%  
 filter(!is.na(field\_subj\_id)) %>%  
 count()  
print(nrow(collection\_subjectID\_check))

## [1] 355

# All 355 are accounted for here  
  
# How many subjectIDs in the passage df?  
passage\_subjectID\_check <- passage %>%  
 mutate(subject\_id = gsub('\_[0-9]\*', '', sample\_id)) %>%  
 group\_by(subject\_id) %>%  
 filter(!is.na(subject\_id)) %>%  
 count()  
print(nrow(passage\_subjectID\_check))

## [1] 158

# 158 are included in the passage data - not sure which 158 these are  
  
# How many subjectIDs in the focus df?  
focus\_subjectID\_check <- focus %>%  
 mutate(subject\_id = gsub('\_[0-9]\*', '', sample\_id)) %>%  
 group\_by(subject\_id) %>%  
 filter(!is.na(subject\_id)) %>%  
 count()  
print(nrow(focus\_subjectID\_check))

## [1] 153

# 153 are included in the passage data - not sure which 153 these are  
  
culture\_results <- merge(collection, passage, by = "sample\_id", all = T)  
culture\_results <- merge(culture\_results, focus, by = "sample\_id", all = T)  
  
# N rows collection  
print(nrow(collection))

## [1] 1696

# N rows passage  
print(nrow(passage))

## [1] 666

# N rows focus  
print(nrow(focus))

## [1] 655

# N rows culture\_results  
print(nrow(culture\_results))

## [1] 1698

culture\_results$np <- ifelse(culture\_results$sample\_type == 'Nasopharyngeal swab', T, F)  
culture\_results$impactor <- ifelse(culture\_results$sample\_type == 'Impactor 5 um NO mask', T, F)  
culture\_results$condensate <- ifelse(culture\_results$sample\_type == 'GII condensate NO mask', T, F)  
culture\_results$antnasal <- ifelse(culture\_results$sample\_type == 'anterior nasal swab', T, F)   
culture\_results$throat <- ifelse(culture\_results$sample\_type == 'Throat Swab', T, F)   
culture\_results$focus.i <- ifelse(is.na(culture\_results$ct), F, T)  
culture\_results$passage.i <- ifelse(is.na(culture\_results$validp), F, T)  
culture\_results$cr.i <- TRUE #Indicator for record present in culture\_results  
  
# Samples with (1) and without (0) passage assays and focus assays  
print(ftable(addmargins(with(culture\_results, table(sample\_type, passage.i, focus.i, exclude = c())))))

## focus.i FALSE TRUE Sum  
## sample\_type passage.i   
## anterior nasal swab FALSE 185 1 186  
## TRUE 0 0 0  
## Sum 185 1 186  
## GII condensate NO mask FALSE 60 5 65  
## TRUE 4 207 211  
## Sum 64 212 276  
## Impactor 5 um NO mask FALSE 273 1 274  
## TRUE 0 2 2  
## Sum 273 3 276  
## Nasopharyngeal swab FALSE 453 2 455  
## TRUE 8 219 227  
## Sum 461 221 682  
## Throat Swab FALSE 51 1 52  
## TRUE 8 216 224  
## Sum 59 217 276  
## NA FALSE 0 0 0  
## TRUE 2 0 2  
## Sum 2 0 2  
## Sum FALSE 1022 10 1032  
## TRUE 22 644 666  
## Sum 1044 654 1698

## Note: All samples should have a sample type; anterior nasal swabs & impactors should not have culture assays ##  
  
## PROBLEMATIC SAMPLES THAT NEED TO BE REVIEWED IN NOTEBOOKS AND REDCAP ##  
  
# Based on review of culture results alone, may be resolved after merge with field field.db  
  
# Samples with missing sample\_type or sample\_type=NA  
print(tbl\_df(filter(culture\_results, is.na(sample\_type)|sample\_type == "")))

## # A tibble: 2 x 15  
## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct np   
## <chr> <int> <chr> <date> <lgl> <lgl> <dbl> <lgl>  
## 1 301\_3 NA <NA> NA FALSE TRUE NA NA   
## 2 301\_5 NA <NA> NA TRUE TRUE NA NA   
## # ... with 7 more variables: impactor <lgl>, condensate <lgl>,  
## # antnasal <lgl>, throat <lgl>, focus.i <lgl>, passage.i <lgl>,  
## # cr.i <lgl>

# Ant Nasal samples with either passage or focus assay results (even if neg/0)  
print(tbl\_df(culture\_results %>%   
 filter(antnasal == T, focus.i == T | passage.i == T)))

## # A tibble: 1 x 15  
## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct np   
## <chr> <int> <chr> <date> <lgl> <lgl> <dbl> <lgl>  
## 1 114\_10 114 anterior n… 2013-02-02 NA NA 0 FALSE  
## # ... with 7 more variables: impactor <lgl>, condensate <lgl>,  
## # antnasal <lgl>, throat <lgl>, focus.i <lgl>, passage.i <lgl>,  
## # cr.i <lgl>

# Impactor samples with either passage or focus assay results (even if neg/0)  
print(tbl\_df(culture\_results %>%   
 filter(impactor == T, focus.i == T | passage.i == T)))

## # A tibble: 3 x 15  
## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct np   
## <chr> <int> <chr> <date> <lgl> <lgl> <dbl> <lgl>  
## 1 187\_15 187 Impactor 5… 2013-02-09 TRUE TRUE 6.67 FALSE  
## 2 196\_16 196 Impactor 5… 2013-02-10 NA NA 0 FALSE  
## 3 47\_11 47 Impactor 5… 2013-01-05 FALSE TRUE 0 FALSE  
## # ... with 7 more variables: impactor <lgl>, condensate <lgl>,  
## # antnasal <lgl>, throat <lgl>, focus.i <lgl>, passage.i <lgl>,  
## # cr.i <lgl>

# NP swabs with a focus assay but no passage (even invalid) assay, or with a passage but no focus assay  
print(tbl\_df(culture\_results %>%   
 filter(np == T, (focus.i == T & passage.i == F) | (focus.i == F & passage.i == T))))

## # A tibble: 10 x 15  
## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct  
## <chr> <int> <chr> <date> <lgl> <lgl> <dbl>  
## 1 108\_1 108 Nasopharyn… 2013-01-29 TRUE TRUE NA   
## 2 13\_1 13 Nasopharyn… 2012-12-10 FALSE TRUE NA   
## 3 153\_1 153 Nasopharyn… 2013-02-05 TRUE TRUE NaN   
## 4 301\_1 301 Nasopharyn… 2013-02-19 TRUE TRUE NA   
## 5 303\_1 303 Nasopharyn… 2013-02-20 NA NA 3547.  
## 6 356\_9 356 Nasopharyn… 2013-03-08 NA NA 0   
## 7 38\_1 38 Nasopharyn… 2012-12-21 FALSE TRUE NA   
## 8 40\_1 40 Nasopharyn… 2012-12-28 FALSE TRUE NA   
## 9 62\_8 62 Nasopharyn… 2013-01-12 TRUE TRUE NA   
## 10 8\_1 8 Nasopharyn… 2012-12-04 TRUE TRUE NA   
## # ... with 8 more variables: np <lgl>, impactor <lgl>, condensate <lgl>,  
## # antnasal <lgl>, throat <lgl>, focus.i <lgl>, passage.i <lgl>,  
## # cr.i <lgl>

# G-II condensate with a focus assay but no passage (even invalid) assay, or with a passage but no focus assay  
print(tbl\_df(culture\_results %>%   
 filter(condensate == T, (focus.i == T & passage.i == F) | (focus.i == F & passage.i == T))))

## # A tibble: 9 x 15  
## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct np   
## <chr> <int> <chr> <date> <lgl> <lgl> <dbl> <lgl>  
## 1 108\_3 108 GII conden… 2013-01-29 NA FALSE NA FALSE  
## 2 174\_3 174 GII conden… 2013-02-06 TRUE TRUE NA FALSE  
## 3 19\_3 19 GII conden… 2012-12-11 NA NA 0 FALSE  
## 4 27\_3 27 GII conden… 2012-12-14 NA NA 0 FALSE  
## 5 303\_3 303 GII conden… 2013-02-20 NA NA 0 FALSE  
## 6 38\_3 38 GII conden… 2012-12-21 FALSE TRUE NA FALSE  
## 7 40\_3 40 GII conden… 2012-12-28 FALSE TRUE NA FALSE  
## 8 64\_10 64 GII conden… 2013-01-13 NA NA 0 FALSE  
## 9 7\_3 7 GII conden… 2012-12-04 NA NA 0 FALSE  
## # ... with 7 more variables: impactor <lgl>, condensate <lgl>,  
## # antnasal <lgl>, throat <lgl>, focus.i <lgl>, passage.i <lgl>,  
## # cr.i <lgl>

# Culture\_results  
print(head(tbl\_df((culture\_results))))

## # A tibble: 6 x 15  
## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct  
## <chr> <int> <chr> <date> <lgl> <lgl> <dbl>  
## 1 10\_1 10 Nasopharyn… 2012-12-05 NA NA NA   
## 2 100\_1 100 Nasopharyn… 2013-01-28 TRUE TRUE 16686.   
## 3 100\_2 100 Impactor 5… 2013-01-28 NA NA NA   
## 4 100\_3 100 GII conden… 2013-01-28 NA NA NA   
## 5 100\_4 100 anterior n… 2013-01-28 NA NA NA   
## 6 100\_5 100 Throat Swab 2013-01-28 TRUE TRUE 93.3  
## # ... with 8 more variables: np <lgl>, impactor <lgl>, condensate <lgl>,  
## # antnasal <lgl>, throat <lgl>, focus.i <lgl>, passage.i <lgl>,  
## # cr.i <lgl>

## Merge3 = merge of merge2 with (culture\_results from REDCap) by sample\_id (only) ##  
# x=merge2, y=culture\_results  
  
# Variables merge2  
print(names(merge2))

## [1] "field\_subj\_id" "date\_visit" "date\_on\_sx" "g2\_coll\_num"   
## [5] "enrolled" "visit\_num" "g2\_run" "clinical.i"   
## [9] "g2lm.i" "rapid\_flu\_\_\_3" "rapid\_flu\_loc" "body\_temp"   
## [13] "cough\_number" "sneeze\_number" "merge1.i" "sample\_id"   
## [17] "sample\_type" "field.db1.i" "merge2.i"

# Variables culture\_results  
print(names(culture\_results))

## [1] "sample\_id" "field\_subj\_id" "sample\_type" "date\_visit"   
## [5] "passpos" "validp" "ct" "np"   
## [9] "impactor" "condensate" "antnasal" "throat"   
## [13] "focus.i" "passage.i" "cr.i"

# How many subjectIDs in the collection df?  
culture\_results\_subjectID\_check <- culture\_results %>%  
 group\_by(field\_subj\_id) %>%  
 filter(!is.na(field\_subj\_id)) %>%  
 count()  
print(nrow(culture\_results\_subjectID\_check))

## [1] 355

# All 355 are accounted for here  
  
merge3 <- merge(merge2, culture\_results, c('sample\_id'), all = TRUE)  
  
# Number of rows in merge2  
print(nrow(merge2))

## [1] 1938

# Number of rows in culture\_results  
print(nrow(culture\_results))

## [1] 1698

# Number of rows in merge3  
print(nrow(merge3))

## [1] 1940

#Number of cols in merge3  
print(ncol(merge3))

## [1] 33

print(head(tbl\_df(merge3)))

## # A tibble: 6 x 33  
## sample\_id field\_subj\_id.x date\_visit.x date\_on\_sx g2\_coll\_num enrolled  
## <chr> <int> <date> <fct> <dbl> <lgl>   
## 1 10\_1 10 2012-12-05 2012-12-02 0 FALSE   
## 2 10\_6 10 2012-12-05 2012-12-02 0 FALSE   
## 3 100\_1 100 2013-01-28 2013-01-25 1 TRUE   
## 4 100\_2 100 2013-01-28 2013-01-25 1 TRUE   
## 5 100\_3 100 2013-01-28 2013-01-25 1 TRUE   
## 6 100\_4 100 2013-01-28 2013-01-25 1 TRUE   
## # ... with 27 more variables: visit\_num <dbl>, g2\_run <dbl>,  
## # clinical.i <lgl>, g2lm.i <lgl>, rapid\_flu\_\_\_3 <int>,  
## # rapid\_flu\_loc <int>, body\_temp <dbl>, cough\_number <int>,  
## # sneeze\_number <int>, merge1.i <lgl>, sample\_type.x <chr>,  
## # field.db1.i <lgl>, merge2.i <lgl>, field\_subj\_id.y <int>,  
## # sample\_type.y <chr>, date\_visit.y <date>, passpos <lgl>, validp <lgl>,  
## # ct <dbl>, np <lgl>, impactor <lgl>, condensate <lgl>, antnasal <lgl>,  
## # throat <lgl>, focus.i <lgl>, passage.i <lgl>, cr.i <lgl>

# Table to check source of records after merge 3  
print(ftable(addmargins(with(merge3, table(merge2.i, cr.i, exclude = c())))))

## cr.i TRUE NA Sum  
## merge2.i   
## TRUE 1696 242 1938  
## NA 2 0 2  
## Sum 1698 242 1940

# Do date\_visit match?  
d.err <- filter(merge3, date\_visit.x != date\_visit.y | is.na(date\_visit.x != date\_visit.y))  
  
# Number of samples where the date\_visit.x (merge2) not equal date\_visit.y (culture\_results)  
print(nrow(d.err))

## [1] 244

# First 10 rows with non matching date\_visit ordered by sample\_id  
print(top\_n(select(d.err,   
 sample\_id,   
 date\_visit.x,   
 date\_visit.y,   
 enrolled,   
 visit\_num,   
 sample\_type.x,   
 sample\_type.y),   
 10, sample\_id))

## sample\_id date\_visit.x date\_visit.y enrolled visit\_num  
## 1 87\_6 2013-01-24 <NA> FALSE 1  
## 2 89\_6 2013-01-24 <NA> FALSE 1  
## 3 90\_6 2013-01-24 <NA> TRUE 1  
## 4 91\_6 2013-01-25 <NA> FALSE 1  
## 5 92\_6 2013-01-25 <NA> TRUE 1  
## 6 93\_6 2013-01-25 <NA> FALSE 1  
## 7 95\_6 2013-01-25 <NA> TRUE 1  
## 8 97\_6 2013-01-28 <NA> TRUE 1  
## 9 98\_6 2013-01-28 <NA> FALSE 1  
## 10 99\_6 2013-01-28 <NA> FALSE 1  
## sample\_type.x sample\_type.y  
## 1 Nasopharyngeal swab <NA>  
## 2 Nasopharyngeal swab <NA>  
## 3 Nasopharyngeal swab <NA>  
## 4 Nasopharyngeal swab <NA>  
## 5 Nasopharyngeal swab <NA>  
## 6 Nasopharyngeal swab <NA>  
## 7 Nasopharyngeal swab <NA>  
## 8 Nasopharyngeal swab <NA>  
## 9 Nasopharyngeal swab <NA>  
## 10 Nasopharyngeal swab <NA>

# Do sample\_type match?  
t.err <- merge3 %>%  
 filter(sample\_type.x != sample\_type.y | is.na(sample\_type.x != sample\_type.y))  
  
# Number of rows where sample types don't match =  
print(nrow(t.err))

## [1] 245

# Columns show whether culture\_result sample types were missing?, Rows likewise for merge2 sample type.  
print(ftable(addmargins(with(merge3,   
 table(miss.x <- is.na(sample\_type.x), miss.y <- is.na(sample\_type.y))))))

## FALSE TRUE Sum  
##   
## FALSE 1696 242 1938  
## TRUE 0 2 2  
## Sum 1696 244 1940

# Table of sample types by data source. (x=merge2, y=culture\_results)  
print(ftable(addmargins(with(merge3,   
 table(merge2.i, cr.i, sample\_type.x, sample\_type.y, exclude = c())))))

## sample\_type.y anterior nasal swab GII condensate NO mask Impactor 5 um NO mask Nasopharyngeal swab Throat Swab NA Sum  
## merge2.i cr.i sample\_type.x   
## TRUE TRUE 0 0 0 1 0 0 1  
## anterior nasal swab 186 0 0 0 0 0 186  
## GII condensate NO mask 0 276 0 0 0 0 276  
## Impactor 5 um NO mask 0 0 276 0 0 0 276  
## Nasopharyngeal swab 0 0 0 681 0 0 681  
## Throat Swab 0 0 0 0 276 0 276  
## NA 0 0 0 0 0 0 0  
## Sum 186 276 276 682 276 0 1696  
## NA 0 0 0 0 0 19 19  
## anterior nasal swab 0 0 0 0 0 0 0  
## GII condensate NO mask 0 0 0 0 0 0 0  
## Impactor 5 um NO mask 0 0 0 0 0 0 0  
## Nasopharyngeal swab 0 0 0 0 0 223 223  
## Throat Swab 0 0 0 0 0 0 0  
## NA 0 0 0 0 0 0 0  
## Sum 0 0 0 0 0 242 242  
## Sum 0 0 0 1 0 19 20  
## anterior nasal swab 186 0 0 0 0 0 186  
## GII condensate NO mask 0 276 0 0 0 0 276  
## Impactor 5 um NO mask 0 0 276 0 0 0 276  
## Nasopharyngeal swab 0 0 0 681 0 223 904  
## Throat Swab 0 0 0 0 276 0 276  
## NA 0 0 0 0 0 0 0  
## Sum 186 276 276 682 276 242 1938  
## NA TRUE 0 0 0 0 0 0 0  
## anterior nasal swab 0 0 0 0 0 0 0  
## GII condensate NO mask 0 0 0 0 0 0 0  
## Impactor 5 um NO mask 0 0 0 0 0 0 0  
## Nasopharyngeal swab 0 0 0 0 0 0 0  
## Throat Swab 0 0 0 0 0 0 0  
## NA 0 0 0 0 0 2 2  
## Sum 0 0 0 0 0 2 2  
## NA 0 0 0 0 0 0 0  
## anterior nasal swab 0 0 0 0 0 0 0  
## GII condensate NO mask 0 0 0 0 0 0 0  
## Impactor 5 um NO mask 0 0 0 0 0 0 0  
## Nasopharyngeal swab 0 0 0 0 0 0 0  
## Throat Swab 0 0 0 0 0 0 0  
## NA 0 0 0 0 0 0 0  
## Sum 0 0 0 0 0 0 0  
## Sum 0 0 0 0 0 0 0  
## anterior nasal swab 0 0 0 0 0 0 0  
## GII condensate NO mask 0 0 0 0 0 0 0  
## Impactor 5 um NO mask 0 0 0 0 0 0 0  
## Nasopharyngeal swab 0 0 0 0 0 0 0  
## Throat Swab 0 0 0 0 0 0 0  
## NA 0 0 0 0 0 2 2  
## Sum 0 0 0 0 0 2 2  
## Sum TRUE 0 0 0 1 0 0 1  
## anterior nasal swab 186 0 0 0 0 0 186  
## GII condensate NO mask 0 276 0 0 0 0 276  
## Impactor 5 um NO mask 0 0 276 0 0 0 276  
## Nasopharyngeal swab 0 0 0 681 0 0 681  
## Throat Swab 0 0 0 0 276 0 276  
## NA 0 0 0 0 0 2 2  
## Sum 186 276 276 682 276 2 1698  
## NA 0 0 0 0 0 19 19  
## anterior nasal swab 0 0 0 0 0 0 0  
## GII condensate NO mask 0 0 0 0 0 0 0  
## Impactor 5 um NO mask 0 0 0 0 0 0 0  
## Nasopharyngeal swab 0 0 0 0 0 223 223  
## Throat Swab 0 0 0 0 0 0 0  
## NA 0 0 0 0 0 0 0  
## Sum 0 0 0 0 0 242 242  
## Sum 0 0 0 1 0 19 20  
## anterior nasal swab 186 0 0 0 0 0 186  
## GII condensate NO mask 0 276 0 0 0 0 276  
## Impactor 5 um NO mask 0 0 276 0 0 0 276  
## Nasopharyngeal swab 0 0 0 681 0 223 904  
## Throat Swab 0 0 0 0 276 0 276  
## NA 0 0 0 0 0 2 2  
## Sum 186 276 276 682 276 244 1940

# All non-matching sample types seem to be due to missing (NA) values.  
  
# All sample ids begining with 237 in merge3  
print(filter(merge3, grepl("^237", sample\_id)))

## sample\_id field\_subj\_id.x date\_visit.x date\_on\_sx g2\_coll\_num enrolled  
## 1 237\_1 237 2013-02-12 2013-02-11 1 TRUE  
## 2 237\_2 237 2013-02-12 2013-02-11 1 TRUE  
## 3 237\_3 237 2013-02-12 2013-02-11 1 TRUE  
## 4 237\_4 237 2013-02-12 2013-02-11 1 TRUE  
## 5 237\_5 237 2013-02-12 2013-02-11 1 TRUE  
## 6 237\_6 237 2013-02-12 2013-02-11 1 TRUE  
## visit\_num g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## 1 1 1 TRUE TRUE 1 NA 37.8  
## 2 1 1 TRUE TRUE 1 NA 37.8  
## 3 1 1 TRUE TRUE 1 NA 37.8  
## 4 1 1 TRUE TRUE 1 NA 37.8  
## 5 1 1 TRUE TRUE 1 NA 37.8  
## 6 1 1 TRUE TRUE 1 NA 37.8  
## cough\_number sneeze\_number merge1.i sample\_type.x field.db1.i  
## 1 14 0 TRUE Nasopharyngeal swab TRUE  
## 2 14 0 TRUE Impactor 5 um NO mask TRUE  
## 3 14 0 TRUE GII condensate NO mask TRUE  
## 4 14 0 TRUE anterior nasal swab TRUE  
## 5 14 0 TRUE Throat Swab TRUE  
## 6 14 0 TRUE TRUE  
## merge2.i field\_subj\_id.y sample\_type.y date\_visit.y passpos  
## 1 TRUE 237 Nasopharyngeal swab 2013-02-12 NA  
## 2 TRUE 237 Impactor 5 um NO mask 2013-02-12 NA  
## 3 TRUE 237 GII condensate NO mask 2013-02-12 NA  
## 4 TRUE 237 anterior nasal swab 2013-02-12 NA  
## 5 TRUE 237 Throat Swab 2013-02-12 NA  
## 6 TRUE 237 Nasopharyngeal swab 2013-02-12 NA  
## validp ct np impactor condensate antnasal throat focus.i passage.i  
## 1 NA NA TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## 2 NA NA FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## 3 NA NA FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## 4 NA NA FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## 5 NA NA FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## 6 NA NA TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## cr.i  
## 1 TRUE  
## 2 TRUE  
## 3 TRUE  
## 4 TRUE  
## 5 TRUE  
## 6 TRUE

# All sample ids begining with 237 in culture\_results  
print(filter(culture\_results, grepl("^237", sample\_id)))

## sample\_id field\_subj\_id sample\_type date\_visit passpos validp  
## 1 237\_1 237 Nasopharyngeal swab 2013-02-12 NA NA  
## 2 237\_2 237 Impactor 5 um NO mask 2013-02-12 NA NA  
## 3 237\_3 237 GII condensate NO mask 2013-02-12 NA NA  
## 4 237\_4 237 anterior nasal swab 2013-02-12 NA NA  
## 5 237\_5 237 Throat Swab 2013-02-12 NA NA  
## 6 237\_6 237 Nasopharyngeal swab 2013-02-12 NA NA  
## ct np impactor condensate antnasal throat focus.i passage.i cr.i  
## 1 NA TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## 2 NA FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE  
## 3 NA FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE  
## 4 NA FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE  
## 5 NA FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE  
## 6 NA TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE

# Sample 237\_6 is an enrolled roommate, enrolled based on fever, therefore second NP swab should be in the lab.  
  
# Samples in merge3 where culture\_results data had no match in merge2 (field data)  
x <- filter(merge3, cr.i == T, is.na(merge2.i))  
print(select(x, sample\_id))

## sample\_id  
## 1 301\_3  
## 2 301\_5

# All sample ids begining with 301 in merge3  
print(filter(merge3, grepl("^301", sample\_id)))

## sample\_id field\_subj\_id.x date\_visit.x date\_on\_sx g2\_coll\_num enrolled  
## 1 301\_1 301 2013-02-19 2013-02-18 0 FALSE  
## 2 301\_3 NA <NA> <NA> NA NA  
## 3 301\_5 NA <NA> <NA> NA NA  
## 4 301\_6 301 2013-02-19 2013-02-18 0 FALSE  
## visit\_num g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## 1 1 0 TRUE NA 0 0 36.7  
## 2 NA NA NA NA NA NA NA  
## 3 NA NA NA NA NA NA NA  
## 4 1 0 TRUE NA 0 0 36.7  
## cough\_number sneeze\_number merge1.i sample\_type.x field.db1.i  
## 1 NA NA TRUE Nasopharyngeal swab TRUE  
## 2 NA NA NA <NA> NA  
## 3 NA NA NA <NA> NA  
## 4 NA NA TRUE Nasopharyngeal swab TRUE  
## merge2.i field\_subj\_id.y sample\_type.y date\_visit.y passpos validp  
## 1 TRUE 301 Nasopharyngeal swab 2013-02-19 TRUE TRUE  
## 2 NA NA <NA> <NA> FALSE TRUE  
## 3 NA NA <NA> <NA> TRUE TRUE  
## 4 TRUE NA <NA> <NA> NA NA  
## ct np impactor condensate antnasal throat focus.i passage.i cr.i  
## 1 NA TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE  
## 2 NA NA NA NA NA NA FALSE TRUE TRUE  
## 3 NA NA NA NA NA NA FALSE TRUE TRUE  
## 4 NA NA NA NA NA NA NA NA NA

# All records in merge3 with field\_subj\_id 301 from either source dataframe  
print(filter(merge3, field\_subj\_id.x == 301 | field\_subj\_id.y == 301))

## sample\_id field\_subj\_id.x date\_visit.x date\_on\_sx g2\_coll\_num enrolled  
## 1 301\_1 301 2013-02-19 2013-02-18 0 FALSE  
## 2 301\_6 301 2013-02-19 2013-02-18 0 FALSE  
## visit\_num g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp  
## 1 1 0 TRUE NA 0 0 36.7  
## 2 1 0 TRUE NA 0 0 36.7  
## cough\_number sneeze\_number merge1.i sample\_type.x field.db1.i  
## 1 NA NA TRUE Nasopharyngeal swab TRUE  
## 2 NA NA TRUE Nasopharyngeal swab TRUE  
## merge2.i field\_subj\_id.y sample\_type.y date\_visit.y passpos validp  
## 1 TRUE 301 Nasopharyngeal swab 2013-02-19 TRUE TRUE  
## 2 TRUE NA <NA> <NA> NA NA  
## ct np impactor condensate antnasal throat focus.i passage.i cr.i  
## 1 NA TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE  
## 2 NA NA NA NA NA NA NA NA NA

# Records for samples 301\_x in culture\_results: \n")  
print(filter(culture\_results, grepl("^301", sample\_id)))

## sample\_id field\_subj\_id sample\_type date\_visit passpos validp ct  
## 1 301\_1 301 Nasopharyngeal swab 2013-02-19 TRUE TRUE NA  
## 2 301\_3 NA <NA> <NA> FALSE TRUE NA  
## 3 301\_5 NA <NA> <NA> TRUE TRUE NA  
## np impactor condensate antnasal throat focus.i passage.i cr.i  
## 1 TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE  
## 2 NA NA NA NA NA FALSE TRUE TRUE  
## 3 NA NA NA NA NA FALSE TRUE TRUE

# All samples collected on same date as subject 301  
  
x <- select(filter(merge3, field\_subj\_id.y == "301"), date\_visit.x)  
  
y <- right\_join(merge3, x, by = "date\_visit.x")  
  
# Throat and Condensate samples from enrolled subject on the same day as 301  
print(filter(y, sample\_type.x %in% c("Throat Swab", "GII condensate NO mask")))

## sample\_id field\_subj\_id.x date\_visit.x date\_on\_sx g2\_coll\_num enrolled  
## 1 288\_10 288 2013-02-19 <NA> 2 TRUE  
## 2 288\_9 288 2013-02-19 <NA> 2 TRUE  
## 3 290\_10 290 2013-02-19 <NA> 2 TRUE  
## 4 290\_9 290 2013-02-19 <NA> 2 TRUE  
## 5 292\_10 292 2013-02-19 <NA> 2 TRUE  
## 6 292\_9 292 2013-02-19 <NA> 2 TRUE  
## 7 296\_10 296 2013-02-19 <NA> 2 TRUE  
## 8 296\_9 296 2013-02-19 <NA> 2 TRUE  
## 9 297\_10 297 2013-02-19 <NA> 2 TRUE  
## 10 297\_9 297 2013-02-19 <NA> 2 TRUE  
## 11 298\_3 298 2013-02-19 2013-02-17 1 TRUE  
## 12 298\_5 298 2013-02-19 2013-02-17 1 TRUE  
## 13 299\_3 299 2013-02-19 2013-02-15 1 TRUE  
## 14 299\_5 299 2013-02-19 2013-02-15 1 TRUE  
## 15 302\_3 302 2013-02-19 2013-02-15 1 TRUE  
## 16 302\_5 302 2013-02-19 2013-02-15 1 TRUE  
## visit\_num g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc  
## 1 999 2 TRUE TRUE NA NA  
## 2 999 2 TRUE TRUE NA NA  
## 3 999 2 TRUE TRUE NA NA  
## 4 999 2 TRUE TRUE NA NA  
## 5 999 2 TRUE TRUE NA NA  
## 6 999 2 TRUE TRUE NA NA  
## 7 999 2 TRUE TRUE NA NA  
## 8 999 2 TRUE TRUE NA NA  
## 9 999 2 TRUE TRUE NA NA  
## 10 999 2 TRUE TRUE NA NA  
## 11 1 1 TRUE TRUE 1 NA  
## 12 1 1 TRUE TRUE 1 NA  
## 13 1 1 TRUE TRUE 0 1  
## 14 1 1 TRUE TRUE 0 1  
## 15 1 1 TRUE TRUE 0 0  
## 16 1 1 TRUE TRUE 0 0  
## body\_temp cough\_number sneeze\_number merge1.i sample\_type.x  
## 1 NA 28 0 TRUE Throat Swab  
## 2 NA 28 0 TRUE GII condensate NO mask  
## 3 NA 0 0 TRUE Throat Swab  
## 4 NA 0 0 TRUE GII condensate NO mask  
## 5 NA 32 0 TRUE Throat Swab  
## 6 NA 32 0 TRUE GII condensate NO mask  
## 7 NA 7 0 TRUE Throat Swab  
## 8 NA 7 0 TRUE GII condensate NO mask  
## 9 NA 30 NA TRUE Throat Swab  
## 10 NA 30 NA TRUE GII condensate NO mask  
## 11 38.1 27 NA TRUE GII condensate NO mask  
## 12 38.1 27 NA TRUE Throat Swab  
## 13 37.9 15 0 TRUE GII condensate NO mask  
## 14 37.9 15 0 TRUE Throat Swab  
## 15 37.4 32 0 TRUE GII condensate NO mask  
## 16 37.4 32 0 TRUE Throat Swab  
## field.db1.i merge2.i field\_subj\_id.y sample\_type.y  
## 1 TRUE TRUE 288 Throat Swab  
## 2 TRUE TRUE 288 GII condensate NO mask  
## 3 TRUE TRUE 290 Throat Swab  
## 4 TRUE TRUE 290 GII condensate NO mask  
## 5 TRUE TRUE 292 Throat Swab  
## 6 TRUE TRUE 292 GII condensate NO mask  
## 7 TRUE TRUE 296 Throat Swab  
## 8 TRUE TRUE 296 GII condensate NO mask  
## 9 TRUE TRUE 297 Throat Swab  
## 10 TRUE TRUE 297 GII condensate NO mask  
## 11 TRUE TRUE 298 GII condensate NO mask  
## 12 TRUE TRUE 298 Throat Swab  
## 13 TRUE TRUE 299 GII condensate NO mask  
## 14 TRUE TRUE 299 Throat Swab  
## 15 TRUE TRUE 302 GII condensate NO mask  
## 16 TRUE TRUE 302 Throat Swab  
## date\_visit.y passpos validp ct np impactor condensate  
## 1 2013-02-19 TRUE TRUE 0.000000 FALSE FALSE FALSE  
## 2 2013-02-19 TRUE TRUE 0.000000 FALSE FALSE TRUE  
## 3 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE FALSE  
## 4 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE TRUE  
## 5 2013-02-19 TRUE TRUE 6.666667 FALSE FALSE FALSE  
## 6 2013-02-19 TRUE TRUE 0.000000 FALSE FALSE TRUE  
## 7 2013-02-19 TRUE TRUE 0.000000 FALSE FALSE FALSE  
## 8 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE TRUE  
## 9 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE FALSE  
## 10 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE TRUE  
## 11 2013-02-19 TRUE TRUE 866.666667 FALSE FALSE TRUE  
## 12 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE FALSE  
## 13 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE TRUE  
## 14 2013-02-19 TRUE TRUE 0.000000 FALSE FALSE FALSE  
## 15 2013-02-19 FALSE TRUE 0.000000 FALSE FALSE TRUE  
## 16 2013-02-19 TRUE TRUE 286.666667 FALSE FALSE FALSE  
## antnasal throat focus.i passage.i cr.i  
## 1 FALSE TRUE TRUE TRUE TRUE  
## 2 FALSE FALSE TRUE TRUE TRUE  
## 3 FALSE TRUE TRUE TRUE TRUE  
## 4 FALSE FALSE TRUE TRUE TRUE  
## 5 FALSE TRUE TRUE TRUE TRUE  
## 6 FALSE FALSE TRUE TRUE TRUE  
## 7 FALSE TRUE TRUE TRUE TRUE  
## 8 FALSE FALSE TRUE TRUE TRUE  
## 9 FALSE TRUE TRUE TRUE TRUE  
## 10 FALSE FALSE TRUE TRUE TRUE  
## 11 FALSE FALSE TRUE TRUE TRUE  
## 12 FALSE TRUE TRUE TRUE TRUE  
## 13 FALSE FALSE TRUE TRUE TRUE  
## 14 FALSE TRUE TRUE TRUE TRUE  
## 15 FALSE FALSE TRUE TRUE TRUE  
## 16 FALSE TRUE TRUE TRUE TRUE

# Looks like 301\_3 and 301\_5 are erroneous duplicative entries for 302\_3 and 302\_5: Will delete extra 301 samples.  
merge3 <- merge3 %>%  
 filter(!(sample\_id %in% c("301\_3", "301\_5")))  
  
# Table to check source of records after merge 3 clean-up  
print(ftable(addmargins(with(merge3, table(merge2.i, cr.i, exclude = c())))))

## cr.i TRUE NA Sum  
## merge2.i   
## TRUE 1696 242 1938  
## Sum 1696 242 1938

merge3 <- mutate(  
 merge3,   
 subject\_id = ifelse((!is.na(field\_subj\_id.x) | field\_subj\_id.x == ""),   
 field\_subj\_id.x,   
 field\_subj\_id.y),  
 sample\_type = ifelse((sample\_type.x %in%   
 c("Nasopharyngeal swab",  
 "Impactor 5 um NO mask",   
 "GII condensate NO mask",   
 "anterior nasal swab",  
 "Throat Swab")),  
 sample\_type.x,   
 sample\_type.y),  
 date\_visit = as.Date(ifelse(!is.na(date\_visit.x), date\_visit.x, date\_visit.y), origin = "1970-01-01")  
)  
  
merge3 <- merge3 %>%  
 select(-contains(".x"), -contains(".y"))  
  
# Number of rows in merge3 after clean-up with no sample type  
print(nrow(filter(merge3, is.na(sample\_type) | sample\_type == "")))

## [1] 19

# Number of rows in merge3 after clean-up with no subject id  
print(nrow(filter(merge3, is.na(subject\_id) | subject\_id == "")))

## [1] 0

# Number of rows in merge3 after clean-up with no date visit  
print(nrow(filter(merge3, is.na(date\_visit))))

## [1] 0

# Samples without sample type  
print(filter(merge3, is.na(sample\_type)))

## sample\_id date\_on\_sx g2\_coll\_num enrolled visit\_num g2\_run clinical.i  
## 1 138\_16 <NA> 3 TRUE 999 3 TRUE  
## 2 14\_6 2012-12-08 0 FALSE 1 0 TRUE  
## 3 141\_12 <NA> 2 TRUE 999 2 TRUE  
## 4 151\_6 2013-02-02 0 FALSE 1 0 TRUE  
## 5 174\_16 <NA> 3 TRUE 999 3 TRUE  
## 6 184\_7 2013-02-05 1 TRUE 1 1 TRUE  
## 7 186\_6 2013-02-05 1 TRUE 1 1 TRUE  
## 8 189\_12 <NA> 2 TRUE 999 2 TRUE  
## 9 200\_7 2013-02-05 0 FALSE 1 0 TRUE  
## 10 220\_6 <NA> 0 FALSE 1 0 TRUE  
## 11 228\_7 2013-02-09 1 TRUE 1 1 TRUE  
## 12 31\_17 <NA> 3 TRUE 999 3 TRUE  
## 13 334\_6 2013-02-20 0 FALSE 1 0 TRUE  
## 14 34\_7 2012-12-16 0 FALSE 1 0 TRUE  
## 15 35\_7 2012-12-17 1 TRUE 1 1 TRUE  
## 16 43\_7 2012-12-31 1 TRUE 1 1 TRUE  
## 17 58\_6 2013-01-07 1 TRUE 1 1 TRUE  
## 18 67\_6 2013-01-13 0 FALSE 1 0 TRUE  
## 19 70\_12 <NA> 3 TRUE 999 3 TRUE  
## g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp cough\_number sneeze\_number  
## 1 TRUE NA NA NA 24 0  
## 2 NA 1 NA 36.9 NA NA  
## 3 TRUE NA NA NA 54 0  
## 4 NA 0 0 37.3 NA NA  
## 5 TRUE NA NA NA NA NA  
## 6 TRUE 0 1 37.7 0 2  
## 7 TRUE 0 0 36.8 0 2  
## 8 TRUE NA NA NA 13 0  
## 9 NA 0 0 36.9 NA NA  
## 10 NA 0 0 37.1 NA NA  
## 11 TRUE 0 1 38.0 44 0  
## 12 TRUE NA NA NA 0 NA  
## 13 NA 0 0 36.9 NA NA  
## 14 NA 0 0 37.4 NA NA  
## 15 TRUE 0 0 37.8 102 NA  
## 16 TRUE 0 0 37.1 4 NA  
## 17 TRUE 0 0 36.9 1 NA  
## 18 NA 0 0 36.8 NA NA  
## 19 TRUE NA NA NA 12 NA  
## merge1.i field.db1.i merge2.i passpos validp ct np impactor condensate  
## 1 TRUE TRUE TRUE NA NA NA NA NA NA  
## 2 TRUE TRUE TRUE NA NA NA NA NA NA  
## 3 TRUE TRUE TRUE NA NA NA NA NA NA  
## 4 TRUE TRUE TRUE NA NA NA NA NA NA  
## 5 TRUE TRUE TRUE NA NA NA NA NA NA  
## 6 TRUE TRUE TRUE NA NA NA NA NA NA  
## 7 TRUE TRUE TRUE NA NA NA NA NA NA  
## 8 TRUE TRUE TRUE NA NA NA NA NA NA  
## 9 TRUE TRUE TRUE NA NA NA NA NA NA  
## 10 TRUE TRUE TRUE NA NA NA NA NA NA  
## 11 TRUE TRUE TRUE NA NA NA NA NA NA  
## 12 TRUE TRUE TRUE NA NA NA NA NA NA  
## 13 TRUE TRUE TRUE NA NA NA NA NA NA  
## 14 TRUE TRUE TRUE NA NA NA NA NA NA  
## 15 TRUE TRUE TRUE NA NA NA NA NA NA  
## 16 TRUE TRUE TRUE NA NA NA NA NA NA  
## 17 TRUE TRUE TRUE NA NA NA NA NA NA  
## 18 TRUE TRUE TRUE NA NA NA NA NA NA  
## 19 TRUE TRUE TRUE NA NA NA NA NA NA  
## antnasal throat focus.i passage.i cr.i subject\_id sample\_type  
## 1 NA NA NA NA NA 138 <NA>  
## 2 NA NA NA NA NA 14 <NA>  
## 3 NA NA NA NA NA 141 <NA>  
## 4 NA NA NA NA NA 151 <NA>  
## 5 NA NA NA NA NA 174 <NA>  
## 6 NA NA NA NA NA 184 <NA>  
## 7 NA NA NA NA NA 186 <NA>  
## 8 NA NA NA NA NA 189 <NA>  
## 9 NA NA NA NA NA 200 <NA>  
## 10 NA NA NA NA NA 220 <NA>  
## 11 NA NA NA NA NA 228 <NA>  
## 12 NA NA NA NA NA 31 <NA>  
## 13 NA NA NA NA NA 334 <NA>  
## 14 NA NA NA NA NA 34 <NA>  
## 15 NA NA NA NA NA 35 <NA>  
## 16 NA NA NA NA NA 43 <NA>  
## 17 NA NA NA NA NA 58 <NA>  
## 18 NA NA NA NA NA 67 <NA>  
## 19 NA NA NA NA NA 70 <NA>  
## date\_visit  
## 1 2013-02-06  
## 2 2012-12-11  
## 3 2013-02-05  
## 4 2013-02-05  
## 5 2013-02-08  
## 6 2013-02-07  
## 7 2013-02-07  
## 8 2013-02-09  
## 9 2013-02-09  
## 10 2013-02-11  
## 11 2013-02-11  
## 12 2012-12-19  
## 13 2013-02-27  
## 14 2012-12-19  
## 15 2012-12-19  
## 16 2013-01-02  
## 17 2013-01-10  
## 18 2013-01-14  
## 19 2013-01-18

# What to do with these?  
  
## For the group not enrolled, figure out how to keep the one NP swabs that was sent to the lab and discard the other record.  
  
# Examine the first visit to see which NP samples were cultured  
  
np <- select(filter(merge3, np == T|sample\_type == "Nasopharyngeal swab"),   
 subject\_id,   
 sample\_id,   
 enrolled,   
 visit\_num,   
 validp,   
 ct)   
  
np1 <- np %>%   
 filter(visit\_num == 1)  
  
np1a <- select(mutate(np1, cultured = !is.na(validp) | !is.na(ct)),   
 sample\_id,  
 subject\_id,  
 cultured)  
  
np1a <- select(separate(np1a, sample\_id, c("id", "sample.id"), "\_"),   
 subject\_id,   
 sample.id,   
 cultured)  
  
np1a.s <- spread(np1a, sample.id, cultured)  
  
names(np1a.s)[2:length(names(np1a.s))] <- paste("sample", names(np1a.s)[2:length(names(np1a.s))], sep = "\_")  
  
print(summary(np1a.s))

## subject\_id sample\_1 sample\_6 sample\_7   
## Min. : 7.0 Mode :logical Mode :logical Mode :logical   
## 1st Qu.:100.5 FALSE:201 FALSE:318 FALSE:4   
## Median :189.0 TRUE :154 NA's :37 NA's :351   
## Mean :188.8   
## 3rd Qu.:278.5   
## Max. :367.0

# NA, means not cultured / assayed.   
# Conclude that all cultured visit 1 samples were called \"\_1\" by the lab.  
  
# Examine all visits for which NP samples were cultured  
  
# Before spread can run on all np samples, must correct for reassigning 225 to 250 to avoid duplication of sample\_#  
  
np <- np %>%  
 mutate(sample\_id =   
 ifelse(subject\_id == 250 & visit\_num == 2, paste(sample\_id, "a", sep = ""), sample\_id)  
 )  
  
np.a <- select(mutate(np, cultured = !is.na(validp)|!is.na(ct)), sample\_id, subject\_id, cultured)  
np.a <- select(separate(np.a, sample\_id, c("id", "sample.id"), "\_"), subject\_id, sample.id, cultured)  
np.a.s <- spread(np.a, sample.id, cultured)  
names(np.a.s)[2:length(names(np.a.s))] <- paste("sample", names(np.a.s)[2:length(names(np.a.s))], sep = "\_")  
print(summary(np.a.s))

## subject\_id sample\_1 sample\_11 sample\_12   
## Min. : 7.0 Mode :logical Mode :logical Mode :logical   
## 1st Qu.:100.5 FALSE:201 FALSE:2 FALSE:5   
## Median :189.0 TRUE :154 TRUE :1 TRUE :17   
## Mean :188.8 NA's :352 NA's :333   
## 3rd Qu.:278.5   
## Max. :367.0   
## sample\_13 sample\_14 sample\_16 sample\_18   
## Mode :logical Mode :logical Mode :logical Mode :logical   
## FALSE:22 FALSE:7 FALSE:1 FALSE:1   
## TRUE :5 NA's :348 NA's :354 NA's :354   
## NA's :328   
##   
##   
## sample\_1a sample\_6 sample\_6a sample\_7   
## Mode:logical Mode :logical Mode :logical Mode :logical   
## TRUE:1 FALSE:322 FALSE:1 FALSE:40   
## NA's:354 TRUE :1 NA's :354 TRUE :48   
## NA's :32 NA's :267   
##   
##   
## sample\_8 sample\_9   
## Mode :logical Mode :logical   
## FALSE:71 FALSE:3   
## TRUE :1 TRUE :1   
## NA's :283 NA's :351   
##   
##

# Subjects with sample\_6 cultured or subject\_id = 250 (after reassigning subject 225 to subject 250)  
print(filter(np.a.s, sample\_6 == T|subject\_id == 250))

## subject\_id sample\_1 sample\_11 sample\_12 sample\_13 sample\_14 sample\_16  
## 1 247 FALSE NA NA NA NA NA  
## 2 250 FALSE NA NA NA NA NA  
## sample\_18 sample\_1a sample\_6 sample\_6a sample\_7 sample\_8 sample\_9  
## 1 NA NA TRUE NA FALSE NA NA  
## 2 NA TRUE FALSE FALSE NA NA NA

# All samples for subject 247  
print(filter(merge3,   
 subject\_id == "247"))

## sample\_id date\_on\_sx g2\_coll\_num enrolled visit\_num g2\_run clinical.i  
## 1 247\_1 2013-02-11 1 TRUE 1 1 TRUE  
## 2 247\_10 <NA> 2 TRUE 999 2 TRUE  
## 3 247\_2 2013-02-11 1 TRUE 1 1 TRUE  
## 4 247\_3 2013-02-11 1 TRUE 1 1 TRUE  
## 5 247\_4 2013-02-11 1 TRUE 1 1 TRUE  
## 6 247\_5 2013-02-11 1 TRUE 1 1 TRUE  
## 7 247\_6 <NA> 2 TRUE 999 2 TRUE  
## 8 247\_7 <NA> 2 TRUE 999 2 TRUE  
## 9 247\_8 <NA> 2 TRUE 999 2 TRUE  
## 10 247\_9 <NA> 2 TRUE 999 2 TRUE  
## g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp cough\_number sneeze\_number  
## 1 TRUE 0 0 37.2 97 0  
## 2 TRUE NA NA NA 92 0  
## 3 TRUE 0 0 37.2 97 0  
## 4 TRUE 0 0 37.2 97 0  
## 5 TRUE 0 0 37.2 97 0  
## 6 TRUE 0 0 37.2 97 0  
## 7 TRUE NA NA NA 92 0  
## 8 TRUE NA NA NA 92 0  
## 9 TRUE NA NA NA 92 0  
## 10 TRUE NA NA NA 92 0  
## merge1.i field.db1.i merge2.i passpos validp ct np impactor  
## 1 TRUE TRUE TRUE NA NA NA TRUE FALSE  
## 2 TRUE TRUE TRUE NA NA NA FALSE TRUE  
## 3 TRUE TRUE TRUE NA NA NA FALSE TRUE  
## 4 TRUE TRUE TRUE NA NA NA FALSE FALSE  
## 5 TRUE TRUE TRUE NA NA NA FALSE FALSE  
## 6 TRUE TRUE TRUE NA NA NA FALSE FALSE  
## 7 TRUE TRUE TRUE TRUE TRUE 0.000000 TRUE FALSE  
## 8 TRUE TRUE TRUE NA NA NA TRUE FALSE  
## 9 TRUE TRUE TRUE FALSE TRUE 0.000000 FALSE FALSE  
## 10 TRUE TRUE TRUE TRUE TRUE 3.333333 FALSE FALSE  
## condensate antnasal throat focus.i passage.i cr.i subject\_id  
## 1 FALSE FALSE FALSE FALSE FALSE TRUE 247  
## 2 FALSE FALSE FALSE FALSE FALSE TRUE 247  
## 3 FALSE FALSE FALSE FALSE FALSE TRUE 247  
## 4 TRUE FALSE FALSE FALSE FALSE TRUE 247  
## 5 FALSE TRUE FALSE FALSE FALSE TRUE 247  
## 6 FALSE FALSE TRUE FALSE FALSE TRUE 247  
## 7 FALSE FALSE FALSE TRUE TRUE TRUE 247  
## 8 FALSE FALSE FALSE FALSE FALSE TRUE 247  
## 9 TRUE FALSE FALSE TRUE TRUE TRUE 247  
## 10 FALSE FALSE TRUE TRUE TRUE TRUE 247  
## sample\_type date\_visit  
## 1 Nasopharyngeal swab 2013-02-12  
## 2 Impactor 5 um NO mask 2013-02-13  
## 3 Impactor 5 um NO mask 2013-02-12  
## 4 GII condensate NO mask 2013-02-12  
## 5 anterior nasal swab 2013-02-12  
## 6 Throat Swab 2013-02-12  
## 7 Nasopharyngeal swab 2013-02-13  
## 8 Nasopharyngeal swab 2013-02-13  
## 9 GII condensate NO mask 2013-02-13  
## 10 Throat Swab 2013-02-13

# All samples for subje\_id == 250  
print(filter(merge3,   
 subject\_id == "250"))

## sample\_id date\_on\_sx g2\_coll\_num enrolled visit\_num g2\_run clinical.i  
## 1 225\_1 2013-02-11 0 FALSE 1 0 TRUE  
## 2 225\_6 2013-02-11 0 FALSE 1 0 TRUE  
## 3 250\_1 <NA> 1 TRUE 2 1 TRUE  
## 4 250\_2 <NA> 1 TRUE 2 1 TRUE  
## 5 250\_3 <NA> 1 TRUE 2 1 TRUE  
## 6 250\_4 <NA> 1 TRUE 2 1 TRUE  
## 7 250\_5 <NA> 1 TRUE 2 1 TRUE  
## 8 250\_6 <NA> 1 TRUE 2 1 TRUE  
## g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp cough\_number sneeze\_number  
## 1 NA 0 0 36.7 NA NA  
## 2 NA 0 0 36.7 NA NA  
## 3 TRUE 0 0 36.8 14 0  
## 4 TRUE 0 0 36.8 14 0  
## 5 TRUE 0 0 36.8 14 0  
## 6 TRUE 0 0 36.8 14 0  
## 7 TRUE 0 0 36.8 14 0  
## 8 TRUE 0 0 36.8 14 0  
## merge1.i field.db1.i merge2.i passpos validp ct np impactor  
## 1 TRUE TRUE TRUE NA NA NA NA NA  
## 2 TRUE TRUE TRUE NA NA NA NA NA  
## 3 TRUE TRUE TRUE TRUE TRUE 1893.33333 TRUE FALSE  
## 4 TRUE TRUE TRUE NA NA NA FALSE TRUE  
## 5 TRUE TRUE TRUE NA FALSE 43.33333 FALSE FALSE  
## 6 TRUE TRUE TRUE NA NA NA FALSE FALSE  
## 7 TRUE TRUE TRUE TRUE TRUE 673.33333 FALSE FALSE  
## 8 TRUE TRUE TRUE NA NA NA NA NA  
## condensate antnasal throat focus.i passage.i cr.i subject\_id  
## 1 NA NA NA NA NA NA 250  
## 2 NA NA NA NA NA NA 250  
## 3 FALSE FALSE FALSE TRUE TRUE TRUE 250  
## 4 FALSE FALSE FALSE FALSE FALSE TRUE 250  
## 5 TRUE FALSE FALSE TRUE TRUE TRUE 250  
## 6 FALSE TRUE FALSE FALSE FALSE TRUE 250  
## 7 FALSE FALSE TRUE TRUE TRUE TRUE 250  
## 8 NA NA NA NA NA NA 250  
## sample\_type date\_visit  
## 1 Nasopharyngeal swab 2013-02-11  
## 2 Nasopharyngeal swab 2013-02-11  
## 3 Nasopharyngeal swab 2013-02-13  
## 4 Impactor 5 um NO mask 2013-02-13  
## 5 GII condensate NO mask 2013-02-13  
## 6 anterior nasal swab 2013-02-13  
## 7 Throat Swab 2013-02-13  
## 8 Nasopharyngeal swab 2013-02-13

merge3 <- merge3 %>%  
 rename( subject.id = subject\_id,   
 sample.id = sample\_id,   
 focus.ct = ct,   
 sample.type = sample\_type,   
 date.visit = date\_visit,   
 g2.run = g2\_run,   
 visit.num = visit\_num)  
  
samples.cc <- merge3 %>%  
 select(subject.id,   
 date.visit,   
 date\_on\_sx,  
 sample.id,   
 sample.type,   
 g2.run,   
 visit.num,   
 passpos,   
 validp,   
 focus.ct,   
 enrolled,   
 rapid\_flu\_\_\_3,   
 rapid\_flu\_loc,   
 body\_temp,   
 cough\_number,  
 sneeze\_number)  
  
# Check the number of subjectIDs in the samples.cc df  
samples.cc\_subjectID\_check <- samples.cc %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
print(nrow(samples.cc\_subjectID\_check))

## [1] 355

# There are 355 subjectIDs here = full set of screened participants  
  
#### Write out EMIT\_samples.cc.RDS file from merge of Clin DB + G2 Log + Field Sample DB ####  
  
saveRDS(samples.cc, file = "Curated Data/Cleaned Data/EMIT\_samples.cc.RDS")  
  
#### \*\*\*\* Using Script: Jing's "interun calibration.R" \*\*\*\* ####  
  
###   
## Original File Information  
  
# Author: Jing Yan  
# Date: August 9th, 2016  
# Title: Interun calibration  
# Purpose: calibrate the Ct values based on the standard interrun calibrator   
# Input files:InputFiles\_UMD  
# InputFiles\_UMD/PCR results/2016.08.05 1st visit NP swab FluA quant.csv  
# InputFiles\_UMD/PCR results/2016.08.05 1st visit NP swab FluB quant.csv  
# InputFiles\_UMD/PCR results/2016.08.08 GII and repeat NP swab FluA part Ia.csv  
# InputFiles\_UMD/PCR results/2016.08.08 GII and repeat NP swab FluA part Ib.csv  
# InputFiles\_UMD/PCR results/2016.08.05 GII and repeat NP swab FluA Part II.csv  
# InputFiles\_UMD/PCR results/2016.08.05 GII and repeat NP swabs FluA part III.csv  
# InputFiles\_UMD/PCR results/2016.08.08 GII and repeat NP swab FluB part Ia.csv  
# InputFiles\_UMD/PCR results/2016.08.08 GII and repeat NP swab FluB part Ib.csv  
# InputFiles\_UMD/PCR results/2016.06.24 GII and repeat NP swab FluB part II.csv  
# Output files: R\_output  
  
#FluA standard curve:  
#Y = -3.143\*LOG(X) + 37.14  
#FluB standard curve:  
#Y = -3.167\*LOG(X) + 33.75  
#FluA standard curve 1st NP swab  
#Y = -3.346\*LOG(X) + 37.52  
#FluB standard curve 1st NP swab  
#Y = -3.297\*LOG(X) + 34.45  
  
###  
  
# I'll note here that the first visit swabs and the rest of the pcr samples received different calibrations, ...  
# ... in addition to the A's and B's getting different calibrations  
# All of the cleaned files are written out into the Curated Data/Cleaned Data directory  
  
low = 53.2  
high = 53200  
ctLAII = -3.143\*log10(low) + 37.14  
ctHAII = -3.143\*log10(high) + 37.14  
ctLA1np = -3.346\*log10(low) + 37.52  
ctHA1np = -3.346\*log10(high) + 37.52  
ctLBII = -3.167\*log10(low) + 33.75  
ctHBII = -3.167\*log10(high) + 33.75  
ctLB1np = -3.297\*log10(low) + 34.45  
ctHB1np = -3.297\*log10(high) + 34.45  
  
#### READ in \*"2016.08.05 1st visit NP swab FluA quant.csv"\* ####  
fluAnp <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.05 1st visit NP swab FluA quant.csv', as.is = T)  
fluAnp <- fluAnp %>%   
 filter(Ct..dRn. != 'Reference')  
fluAnp$Ct..dRn. <- as.numeric(fluAnp$Ct..dRn.)

## Warning: NAs introduced by coercion

fluAnp1 <- fluAnp %>%   
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluAnplow <- fluAnp1 %>%   
 filter(grepl('\_Low', Well.Name))  
fluAnplow$dif <- fluAnplow$Ct..dRn.-ctLA1np  
  
fluAnphigh <- fluAnp1 %>%  
 filter(grepl('\_High', Well.Name))  
fluAnphigh$dif <- fluAnphigh$Ct..dRn.-ctHA1np  
  
fluAnp2 <- rbind(fluAnplow, fluAnphigh) %>%   
 ungroup %>%  
 arrange(Experiment) %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
  
fluAnp3 <- fluAnp2 %>%  
 group\_by(date) %>%  
 mutate(avgdiff = mean(dif))  
fluAnp3$cfactor <- 10^((fluAnp3$avgdiff) / 3.346)  
fluAnp3 <- fluAnp3 %>%   
 distinct(date, cfactor, .keep\_all = TRUE)  
  
fluAnp4 <- fluAnp3 %>%   
 select(date, cfactor)  
  
saveRDS(fluAnp4, "Curated Data/Cleaned Data/fluA\_1np\_calibration.RDS")  
  
#### READ in \*"2016.08.05 1st visit NP swab FluB quant.csv"\* ####  
fluBnp <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.05 1st visit NP swab FluB quant.csv', as.is = T)  
fluBnp <- fluBnp %>%  
 filter(Ct..dRn. != 'Reference')  
fluBnp$Ct..dRn. <- as.numeric(fluBnp$Ct..dRn.)

## Warning: NAs introduced by coercion

fluBnp1 <- fluBnp %>%   
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluBnplow <- fluBnp1 %>%   
 filter(grepl('\_Low', Well.Name))  
fluBnplow$dif <- fluBnplow$Ct..dRn.-ctLB1np  
fluBnplow <- fluBnplow %>%  
 filter(!is.na(Ct..dRn.))  
  
fluBnphigh <- fluBnp1 %>%   
 filter(grepl('\_High', Well.Name))  
fluBnphigh$dif <- fluBnphigh$Ct..dRn.-ctHB1np  
fluBnphigh <- fluBnphigh %>%   
 filter(!is.na(Ct..dRn.))  
  
fluBnp2 <- rbind(fluBnplow, fluBnphigh) %>%   
 ungroup %>%  
 arrange(Experiment) %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
  
fluBnp3 <- fluBnp2 %>%   
 group\_by(date) %>%  
 mutate(avgdiff = mean(dif))  
fluBnp3$cfactor <- 10^((fluBnp3$avgdiff) / 3.297)  
fluBnp3 <- fluBnp3 %>%   
 distinct(date, cfactor, .keep\_all = TRUE)  
  
fluBnp4 <- fluBnp3 %>%  
 select(date, cfactor)  
  
saveRDS(fluBnp4, "Curated Data/Cleaned Data/fluB\_1np\_calibration.RDS")  
  
#### READ in and work with the 4 PCR raw datafiles for flu A (not including the '1st visit' file) ####  
# READ in "2016.08.08 GII and repeat NP swab FluA part Ia.csv"   
# READ in "2016.08.08 GII and repeat NP swab FluA part Ib.csv"   
# READ in "2016.08.05 GII and repeat NP swab FluA Part II.csv"   
# READ in "2016.08.05 GII and repeat NP swabs FluA part III.csv"   
  
fluAI1 <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.08 GII and repeat NP swab FluA part Ia.csv', as.is = T)  
fluAI2 <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.08 GII and repeat NP swab FluA part Ib.csv', as.is = T)  
fluAII <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.05 GII and repeat NP swab FluA Part II.csv', as.is = T)  
fluAIII <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.05 GII and repeat NP swabs FluA part III.csv', as.is = T)  
  
fluAI <- rbind(fluAI1, fluAI2) %>%   
 ungroup()  
fluAI <- fluAI %>%   
 filter(Ct..dRn. != 'Reference')  
fluAI$Ct..dRn. <- as.numeric(fluAI$Ct..dRn.)

## Warning: NAs introduced by coercion

fluAI <- fluAI %>%   
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluAII <- fluAII %>%   
 filter(Ct..dRn. != 'Reference')  
fluAII$Ct..dRn. <- as.numeric(fluAII$Ct..dRn.)

## Warning: NAs introduced by coercion

fluAII1 <- fluAII %>%   
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluAIII <- fluAIII %>%   
 filter(Ct..dRn. != 'Reference')  
fluAIII$Ct..dRn. <- as.numeric(fluAIII$Ct..dRn.)

## Warning: NAs introduced by coercion

fluAIII1 <- fluAIII %>%   
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluA1 <- rbind(fluAI, fluAII1, fluAIII1) %>%  
 ungroup()  
  
fluA1low <- fluA1 %>%  
 filter(grepl('\_Low', Well.Name))  
fluA1low$dif <- fluA1low$Ct..dRn.-ctLAII  
  
fluA1high <- fluA1 %>%  
 filter(grepl('\_High', Well.Name))  
fluA1high$dif <- fluA1high$Ct..dRn.-ctHAII  
  
fluA2 <- rbind(fluA1low, fluA1high) %>%   
 ungroup()  
fluA2 <- fluA2[order(fluA2$Experiment), ]  
fluA2 <- fluA2 %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
  
fluA3 <- fluA2 %>%   
 group\_by(date) %>%   
 mutate(avgdiff = mean(dif))  
fluA3$cfactor <- 10^((fluA3$avgdiff) / 3.143)  
fluA3 <- fluA3 %>%  
 distinct(date, cfactor, .keep\_all = TRUE)  
  
fluA4 <- fluA3 %>%  
 select(date, cfactor)  
  
saveRDS(fluA4, "Curated Data/Cleaned Data/fluA\_calibration.RDS")  
  
#### READ in and work with the 3 PCR raw datafiles for flu B (not including the '1st visit' file) ####  
# Read in: "2016.08.08 GII and repeat NP swab FluB part Ia.csv"  
# Read in: "2016.08.08 GII and repeat NP swab FluB part Ib.csv"  
# Read in: "2016.06.24 GII and repeat NP swab FluB part II.csv"  
  
fluBI1 <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.08 GII and repeat NP swab FluB part Ia.csv', as.is = T)  
fluBI2 <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.08 GII and repeat NP swab FluB part Ib.csv', as.is = T)  
fluBII <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.06.24 GII and repeat NP swab FluB part II.csv', as.is = T)  
  
fluBI <- rbind(fluBI1, fluBI2) %>%   
 ungroup  
fluBI <- fluBI %>%   
 filter(Ct..dRn. != 'Reference')  
fluBI$Ct..dRn. <- as.numeric(fluBI$Ct..dRn.)

## Warning: NAs introduced by coercion

fluBI1 <- fluBI %>%  
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluBII <- fluBII %>%  
 filter(Ct..dRn. != 'Reference')  
fluBII$Ct..dRn. <- as.numeric(fluBII$Ct..dRn.)

## Warning: NAs introduced by coercion

fluBII1 <- fluBII %>%  
 select(Experiment, Well.Name, Ct..dRn.)  
  
fluB1 <- rbind(fluBI1, fluBII1) %>%  
 ungroup  
  
fluB1low <- fluB1 %>%   
 filter(grepl('\_Low', Well.Name))  
fluB1low$dif <- fluB1low$Ct..dRn.-ctLBII  
  
fluB1high <- fluB1 %>%  
 filter(grepl('\_High', Well.Name))  
fluB1high$dif <- fluB1high$Ct..dRn.-ctHBII  
  
fluB2 <- rbind(fluB1low, fluB1high) %>%   
 ungroup  
fluB2 <- fluB2[order(fluB2$Experiment),]  
fluB2 <-fluB2 %>%   
 filter(!is.na(Ct..dRn.))  
fluB2 <- fluB2 %>%  
 mutate(date = gsub('^[0-9]\*.','',Experiment))  
  
fluB3 <- fluB2 %>%  
 group\_by(date) %>%  
 mutate(avgdiff = mean(dif))  
fluB3$cfactor <- 10^((fluB3$avgdiff) / 3.167)  
fluB3 <- fluB3 %>%  
 distinct(date, cfactor, .keep\_all = TRUE)  
  
fluB4 <- fluB3 %>%  
 select(date, cfactor)  
  
saveRDS(fluB4, "Curated Data/Cleaned Data/fluB\_calibration.RDS")  
  
#### \*\*\*\* Using Script: Jing and Dr. Milton's "np fine coarse pcr quantity.R"\*\*\*\* ####  
  
###  
# Original file information:  
  
# Author: Jing Yan& Don Milton  
# Date: September 17, 2015  
# Revision Date: Jan 24, 2016  
# Title: np fine coarse pcr quantity.R  
# Purpose: To sort the data files from the lab (Michael Grantham) PCR data for 2rd or 3rd np, fine and coarse   
  
# Input files:InputFiles\_UMD  
# InputFiles\_UMD/PCR\_1.8.2016/2016.1.4 GII and repeat NP samples FluA Part I.csv  
# InputFiles\_UMD/PCR\_1.8.2016/2016.1.4 GII and repeat NP samples FluA Part II.csv  
# InputFiles\_UMD/PCR\_1.8.2016/2016.1.8 GII samples and repeat NP swabs FluB Part I.csv  
# InputFiles\_UMD/PCR\_1.8.2016/2016.1.8 GII samples and repeat NP swabs FluB Part II.csv  
  
# Output files: R\_output  
  
# Question: How to treat the samples with multiple PCR results?   
  
###  
  
#### READ in "fluA\_calibration.RDS" ####  
fluAcali <- readRDS("Curated Data/Cleaned Data/fluA\_calibration.RDS")  
  
#### READ in "2016.08.08 GII and repeat NP swab FluA part Ia.csv" ####  
pcr\_A1a <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.08.08 GII and repeat NP swab FluA part Ia.csv', as.is = T)  
  
#### READ in "2016.08.08 GII and repeat NP swab FluA part Ib.csv" ####  
pcr\_A1b <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.08.08 GII and repeat NP swab FluA part Ib.csv', as.is = T)  
  
# Bind the two flu A files together  
pcr\_A1 <- rbind(pcr\_A1a, pcr\_A1b) %>%  
 ungroup()  
  
names(pcr\_A1)

## [1] "Experiment" "Well" "Well.Name"   
## [4] "Well.Type" "Replicate" "Threshold..dRn."   
## [7] "Ct..dRn." "Quantity..copies."

# Clean the flu A data  
pcr\_A1 <- pcr\_A1 %>%  
 filter(Ct..dRn. != 'Reference') %>%  
 filter(!grepl('\_Low', Well.Name)) %>%  
 filter(!grepl('NTC', Well.Name)) %>%  
 filter(!grepl('Standard', Well.Type)) %>%  
 filter(!grepl('High', Well.Name)) %>%  
 filter(grepl('\_', Well.Name)) %>%  
 select(-Well, -Well.Type, -Threshold..dRn., -Replicate) %>%  
 mutate(Well.Name = gsub('\_A','', Well.Name)) %>%  
 mutate(Well.Name = gsub('A\_','', Well.Name)) %>%  
 mutate(subject.id = gsub('\_[0-9]\*', '', Well.Name)) %>%  
 rename(copies.in = Quantity..copies.)  
pcr\_A1[pcr\_A1 == "No Ct"] <- ''  
pcr\_A1$copies.in <- as.numeric(pcr\_A1$copies.in)  
  
# Number of rows in part1 (gii and 2rd or 3rd NP samples) influenza A PCR data  
nrow(pcr\_A1)

## [1] 600

# Number of columns in part1 (gii and 2rd or 3rd NP samples) influenza A PCR data  
ncol(pcr\_A1)

## [1] 5

pcr\_A1 <- pcr\_A1[order(pcr\_A1$Well.Name), ]  
  
#### READ in "2016.08.05 GII and repeat NP swab FluA Part II.csv" ####  
  
pcr\_A2 <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.08.05 GII and repeat NP swab FluA Part II.csv',as.is=T)  
names(pcr\_A2)

## [1] "Experiment"   
## [2] "Well"   
## [3] "Well.Name"   
## [4] "Well.Type"   
## [5] "Replicate"   
## [6] "Threshold..dRn."   
## [7] "Ct..dRn."   
## [8] "Quantity..copies."   
## [9] "X"   
## [10] "FAM..Y....3.143.LOG.X....37.14..Eff....108.0."

pcr\_A2 <- pcr\_A2 %>%  
 filter(Ct..dRn. != 'Reference') %>%  
 filter(!grepl('\_Low', Well.Name)) %>%  
 filter(!grepl('NTC', Well.Name)) %>%  
 filter(grepl('\_', Well.Name)) %>%  
 filter(!grepl('High', Well.Name)) %>%  
 filter(!grepl('Standard', Well.Type)) %>%  
 mutate(Well.Name = gsub('\_A', '', Well.Name)) %>%  
 mutate(Well.Name = gsub('A\_', '', Well.Name))%>%  
 rename(copies.in = Quantity..copies.) %>%  
 mutate(subject.id = gsub('\_[0-9]\*', '', Well.Name)) %>%   
 select(-Well, -Well.Type, -Threshold..dRn., -Replicate, -X, -FAM..Y....3.143.LOG.X....37.14..Eff....108.0.)  
pcr\_A2[pcr\_A2 == "No Ct"] <- ''  
pcr\_A2$copies.in <- as.numeric(pcr\_A2$copies.in)  
  
# Number of rows in part2 (gii and 2rd or 3rd NP samples) influenza A PCR data  
nrow(pcr\_A2)

## [1] 142

# Number of columns in part2 (gii and 2rd or 3rd NP samples) influenza A PCR data  
ncol(pcr\_A2)

## [1] 5

pcr\_A2 <- pcr\_A2[order(pcr\_A2$Well.Name), ]  
  
#### READ in "2016.08.05 GII and repeat NP swabs FluA part III.csv" ####  
  
pcr\_A3 <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.08.05 GII and repeat NP swabs FluA part III.csv', as.is = T)  
names(pcr\_A3)

## [1] "Experiment" "Well" "Well.Name"   
## [4] "Well.Type" "Threshold..dRn." "Ct..dRn."   
## [7] "Quantity..copies."

pcr\_A3 <- pcr\_A3 %>%   
 filter(Ct..dRn. != 'Reference') %>%   
 filter(!grepl('\_Low', Well.Name)) %>%  
 filter(!grepl('NTC', Well.Name)) %>%  
 filter(grepl('\_', Well.Name)) %>%  
 filter(!grepl('Standard', Well.Type)) %>%  
 filter(!grepl('B', Well.Name)) %>%  
 filter(!grepl('High', Well.Name)) %>%  
 mutate(Well.Name = gsub('\_A', '', Well.Name)) %>%  
 mutate(Well.Name = gsub('A\_', '', Well.Name)) %>%  
 rename(copies.in = Quantity..copies.) %>%  
 select(-Well, -Well.Type, -Threshold..dRn.) %>%  
 mutate(subject.id = gsub('\_[0-9]\*', '', Well.Name))  
pcr\_A3[pcr\_A3 == "No Ct"] <- ''  
pcr\_A3$copies.in <- as.numeric(pcr\_A3$copies.in)  
  
# Number of rows in part2 (gii and 2rd or 3rd NP samples) influenza A PCR data  
nrow(pcr\_A3)

## [1] 379

# Number of columns in part2 (gii and 2rd or 3rd NP samples) influenza A PCR data  
ncol(pcr\_A3)

## [1] 5

pcr\_A3 <- pcr\_A3[order(pcr\_A3$Well.Name), ]  
  
#### Bind the flu A data ####  
  
pcr\_A <- rbind(pcr\_A1, pcr\_A2, pcr\_A3) %>%   
 ungroup()  
pcr\_A <- pcr\_A %>%   
 mutate(type = 'A') %>%   
 mutate(copy.num = copies.in\*250)  
  
# Number of rows of total (gii and 2rd or 3rd NP samples) influenza A PCR data  
nrow(pcr\_A)

## [1] 1121

# Number of columns of total (gii and 2rd or 3rd NP samples) influenza A PCR data  
ncol(pcr\_A)

## [1] 7

pcr\_A$subject.id <- as.numeric(pcr\_A$subject.id)

## Warning: NAs introduced by coercion

pcr\_A <- pcr\_A %>%  
 arrange(subject.id) %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
  
pcr\_Afinal <- pcr\_A %>%  
 left\_join(fluAcali, by = 'date')  
pcr\_Afinal$virus.copies <- pcr\_Afinal$copy.num\*pcr\_Afinal$cfactor  
  
#### READ in "fluB\_calibration.RDS" ####  
fluBcali <- readRDS("Curated Data/Cleaned Data/fluB\_calibration.RDS")  
  
#### READ in "2016.08.08 GII and repeat NP swab FluB part Ia.csv" ####  
pcr\_B1a <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.08.08 GII and repeat NP swab FluB part Ia.csv', as.is = T)  
  
#### READ in "2016.08.08 GII and repeat NP swab FluB part Ib.csv" ####  
pcr\_B1b <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.08.08 GII and repeat NP swab FluB part Ib.csv', as.is = T)  
  
#### Work with the fluB pcr data ####  
  
pcr\_B1 <- rbind(pcr\_B1a, pcr\_B1b) %>%   
 ungroup  
names(pcr\_B1)

## [1] "Experiment" "Well" "Well.Name"   
## [4] "Well.Type" "Replicate" "Threshold..dRn."   
## [7] "Ct..dRn." "Quantity..copies."

## DATA EDITING ##  
# It appears that the 20.2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR experiment is already included elsewhere as 17.2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR. We will keep the 17.2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR assay and eliminate the other.  
# Also, it appears that the 15. 2015.07.29.013.243 2012-13 samples FluA\_B and FluA PCR and 17. 2015.7.29.013.243 2012-13 samples FluA\_B and FluA PCR assays are the same with just a minor change in the name. We will use the first of these, and eliminate the other  
# Also, it appears that the 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR and the 18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR dfs are carbon copies with minor naming changes. Let's elminat the second assay here.  
pcr\_B1 <- pcr\_B1 %>%  
 filter(Experiment != "20. 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR") %>%  
 filter(Experiment != "17. 2015.7.29.013.243 2012-13 samples FluA\_B and FluA PCR") %>%  
 filter(Experiment != "18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR")  
  
pcr\_B1 <- pcr\_B1 %>%   
 filter(Ct..dRn. != 'Reference') %>%   
 filter(!grepl('\_Low', Well.Name)) %>%   
 filter(!grepl('NTC', Well.Name)) %>%   
 filter(grepl('\_', Well.Name)) %>%   
 filter(!grepl('Standard', Well.Type)) %>%   
 filter(!grepl('High', Well.Name)) %>%   
 mutate(Well.Name = gsub('\_B', '', Well.Name)) %>%   
 mutate(Well.Name = gsub('B\_', '', Well.Name)) %>%   
 select(-Well.Type,-Replicate,-Threshold..dRn.,-Well) %>%   
 mutate(subject.id = gsub('\_[0-9]\*','',Well.Name)) %>%  
 rename(copies.in = Quantity..copies.)  
pcr\_B1[pcr\_B1 == "No Ct"] <- ''  
pcr\_B1$copies.in <- as.numeric(pcr\_B1$copies.in)  
  
# Number of rows in part1 (gii and 2rd or 3rd NP samples) influenza B PCR data  
nrow(pcr\_B1)

## [1] 340

# Number of columns in part1 (gii and 2rd or 3rd NP samples) influenza B PCR data  
ncol(pcr\_B1)

## [1] 5

#### READ in "2016.06.24 GII and repeat NP swab FluB part II.csv" ####  
pcr\_B2 <- read.csv('UMD\_Raw\_Data/PCR data/PCR results/2016.06.24 GII and repeat NP swab FluB part II.csv',as.is=T)  
names(pcr\_B2)

## [1] "Experiment" "Well" "Well.Name"   
## [4] "Well.Type" "Threshold..dRn." "Ct..dRn."   
## [7] "Quantity..copies."

pcr\_B2 <- pcr\_B2 %>%  
 filter(Ct..dRn. != 'Reference') %>%  
 filter(!grepl('\_Low', Well.Name)) %>%   
 filter(!grepl('NTC', Well.Name)) %>%   
 filter(grepl('\_', Well.Name)) %>%   
 filter(!grepl('Standard', Well.Type)) %>%  
 filter(!grepl('High', Well.Name)) %>%   
 mutate(Well.Name = gsub('\_B', '', Well.Name)) %>%  
 mutate(Well.Name = gsub('B\_', '', Well.Name)) %>%  
 mutate(subject.id = gsub('\_[0-9]\*', '', Well.Name)) %>%   
 select(-Well.Type, -Threshold..dRn., -Well) %>%  
 rename(copies.in = Quantity..copies.)  
pcr\_B2[pcr\_B2 == "No Ct"] <- ''  
pcr\_B2$copies.in <- as.numeric(pcr\_B2$copies.in)  
  
# Number of rows in part2 (gii and 2rd or 3rd NP samples) influenza B PCR data  
nrow(pcr\_B2)

## [1] 348

# Number of columns in part2 (gii and 2rd or 3rd NP samples) influenza B PCR data  
ncol(pcr\_B2)

## [1] 5

#### Merge the flu B pcr data ####  
  
pcr\_B <- arrange(rbind(pcr\_B1, pcr\_B2))  
pcr\_B <- pcr\_B %>%  
 mutate(type = 'B')  
pcr\_B <- pcr\_B[order(pcr\_B$Well.Name), ]  
pcr\_B <- pcr\_B %>%  
 mutate(copy.num = copies.in\*272)  
  
# Number of rows of total (gii and 2rd or 3rd NP samples) influenza B PCR data  
nrow(pcr\_B)

## [1] 688

# Number of columns of total (gii and 2rd or 3rd NP sampless) influenza B PCR data  
ncol(pcr\_B)

## [1] 7

pcr\_B <- pcr\_B[order(pcr\_B$subject.id), ]  
pcr\_B <- pcr\_B %>%   
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
  
pcr\_Bfinal <- pcr\_B %>%  
 left\_join(fluBcali, by = 'date')  
pcr\_Bfinal$virus.copies <- pcr\_Bfinal$copy.num\*pcr\_Bfinal$cfactor  
  
#### Merge the fluA and fluB pcr data ####  
  
total.pcr <- rbind(pcr\_Afinal, pcr\_Bfinal)  
total.pcr$Well.Name[total.pcr$Well.Name == '42\_16'] <- '42\_18'  
total.pcr$Well.Name[total.pcr$Well.Name == '2110\_2'] <- '210\_2'  
total.pcr$subject.id[total.pcr$subject.id == '2110'] <- '210'  
total.pcr$Well.Name[total.pcr$Well.Name == '297\_10'] <- '297\_9'  
total.pcr$Well.Name[total.pcr$Well.Name == '194\_10'] <- '194\_3'  
total.pcr$Well.Name[total.pcr$Well.Name == '27\_12'] <- '27\_11'  
total.pcr$Well.Name[total.pcr$Well.Name == '117\_4'] <- '117\_3'  
total.pcr$Well.Name[total.pcr$Well.Name == '161\_8'] <- '161\_11'  
total.pcr$Well.Name[total.pcr$Well.Name == '292\_11' & total.pcr$Ct..dRn. == 30.82] <- '292\_9'  
total.pcr$Well.Name[total.pcr$Well.Name == '292\_11' & total.pcr$Ct..dRn. == 30.71] <- '292\_9'  
  
# There are a bunch of UK aerosol samples in this pcr file. They have "sd" in their Well.Name.  
# I can remove these.   
total.pcr <- total.pcr %>%   
 filter(!grepl('sd', Well.Name))  
  
# Convert the subject.id to numeric class and order from least to highest subject ID  
total.pcr$subject.id <- as.numeric(total.pcr$subject.id)  
total.pcr <- total.pcr %>%  
 arrange(subject.id)  
  
# Number of rows of total (gii and 2rd or 3rd NP samples) PCR data  
nrow(total.pcr)

## [1] 1769

# Number of columns of total (gii and 2rd or 3rd NP samples) PCR data  
ncol(total.pcr)

## [1] 10

#### Before merging PCR data with subtype, need the subtypes ####  
## In order to do this, we will insert the subtype script  
  
#### \*\*\*\* Using Script: Jing and Dr. Milton's "Subtype analysis.R" \*\*\*\* ####  
  
###  
## Original file information:  
  
# "Subtype analysis.R  
# by Jing Yan & Don Milton  
# Jan 24, 2016 -   
# Purpose: Sorting and analyze the sample subtypes from Lab data.   
# ... The types were based on the subtypes table on box in ...  
# ... folder: EMIT\_Date\_Analysis.   
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# Procedures:   
# 1. We input the subtype part I and subtype part II from the folder InputFiles\_UMD/PCR\_9.16.2015   
# Files: '2015.09.15 Subtyping part I.csv' and '2015.09.15 Subtyping part II.csv'  
# 2. We combine the part1 and part2, the rows equal to the sum of part1 and part2  
# 3. We remove the rows within the data (e.g. reference, calibration, etc.) not needed for analysis  
# 4. Assign a new column for subject ID, and another column for sample type   
# 5. Assign any subtype with "No Ct" as False, and the other ones with value as True (after checking for very high Ct values)  
# 6. Determine the sample type based on the combination of T and F  
# 7. Assign each sample type a different integer (1-10),check if all the subject has a number assigned based on subtype  
# 8. We remove the subject with duplicated data (with same subject ID and sample type)  
# 9. For the subject with multiple experiments but different sample type, pick them out and do case by case analysis  
# 10. Combine the selected experiments with the other subjects with unique experiment( final\_subtype)  
# 11. Output (save) dataframe as EMIT\_subtypes.RDS with limited number of variables.   
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
###  
  
#### READ in and work with "2016.06.17 1st visit NP swab subtyping.csv" ####  
  
part1 <- read.csv('UMD\_Raw\_Data/PCR Data/PCR Results/2016.06.17 1st visit NP swab subtyping.csv', as.is = T)  
part1 <- part1 %>%   
 select(-Well, -Well.Type, -Threshold..dR.)  
  
# Number of rows in part11 file  
print(nrow(part1))

## [1] 3774

# Number of columns in part11 file  
print(ncol(part1))

## [1] 3

# Remove the rows within the data (e.g. reference, calibration, etc.) not needed for analysis  
m1 <- part1 %>%   
 filter(Ct..dR. != 'Reference') %>%   
 filter(!grepl('[A-Z]\_[A-Z]', Well.Name)) %>%   
 filter(!grepl('PosControl', Well.Name)) %>%   
 mutate(subject\_id = gsub('\_1\_[0-Z]\*[0-Z]\*[0-Z]\*', '', Well.Name)) %>% # gen new col for subject\_ID which can be obtained from sample\_ID   
 mutate(subject\_id = gsub('\_[0-Z]\*', '', subject\_id))  
m1$Experiment <- as.factor(m1$Experiment)  
  
# Number of subj-exp in merged data  
print(nrow(m1 %>%   
 distinct(subject\_id)))

## [1] 183

# Identify each subtype e.g. A, B, H3,give a new column called types  
m7 <- m1 %>%   
 mutate(types = gsub('[0-9]\*\_1\_', '', Well.Name)) %>%   
 mutate(types = gsub('1\_', '', types))  
  
m7$subject\_id <- as.numeric(m7$subject\_id)  
  
m7 <- m7 %>%  
 arrange(subject\_id) # order the data based on subject\_id  
  
# Number of rows in the sorted subtype file  
print(nrow(m7))

## [1] 1407

# Number of columns in the sorted subtype file  
print(ncol(m7))

## [1] 5

# Examine the Ct Values by type  
m7$Ct <- as.double(m7$Ct..dR.)

## Warning: NAs introduced by coercion

print(filter(m7, !is.na(Ct)) %>%   
 group\_by(types) %>%   
 summarise(avg = mean(Ct), mx = max(Ct)))

## # A tibble: 6 x 3  
## types avg mx  
## <chr> <dbl> <dbl>  
## 1 A 22.3 37.7  
## 2 B 20.1 38.5  
## 3 H3 23.9 37.9  
## 4 PA 22.4 37.4  
## 5 PH1 22.4 23.2  
## 6 RP 24.0 28.4

# Reactions with very high Ct values (>= 40) -- double check these, if there are any:  
suspects <- filter(m7, Ct >= 40)  
print(suspects)

## [1] Experiment Well.Name Ct..dR. subject\_id types Ct   
## <0 rows> (or 0-length row.names)

m7$pos <- with(m7, ifelse(Ct..dR. == "No Ct", F, T)) # Considered Positive if any Ct value  
  
m7 <- m7 %>%  
 arrange(subject\_id, types, Ct)  
  
m7.a <- m7 %>%   
 group\_by(subject\_id, types) %>%   
 summarise(n = n())  
  
m7.b <- m7.a %>%   
 filter(n >= 2) # find subj with more than one result  
  
m7.c <- m7.b %>%  
 inner\_join(m7, by = c('subject\_id', 'types')) #get type data for the subj with more than one type result  
  
## Getting a df with true or false listed for each subtype for each subject\_id, where the vars are the subtypes and the observations are the subject\_ids  
  
m7.c <- m7.c %>%  
 arrange(subject\_id, types, Ct) %>%   
 select(-n) %>%   
 distinct(subject\_id, types, .keep\_all = TRUE)  
  
m7.c2 <- m7 %>%  
 anti\_join(m7.c, by = c('subject\_id', 'types'))  
  
m7 <- m7.c %>%  
 full\_join(m7.c2) %>%  
 arrange(subject\_id, types, Ct)

## Joining, by = c("subject\_id", "types", "Experiment", "Well.Name", "Ct..dR.", "Ct", "pos")

m7$Experiment <- as.factor(m7$Experiment)  
  
m7 <- m7 %>%   
 distinct(subject\_id, types, .keep\_all = T)  
  
m7\_1 <- m7 %>%   
 select(subject\_id, types, pos)   
  
m7.s <- spread(m7\_1, key = types, value = pos) # cols = types, expt = rows  
  
# Number of subject - experiments  
print(nrow(m7.s))

## [1] 183

# This m7.s gives the final set of subject\_ids with their subtype classification  
  
## Adding some additional variables to the m7.s df to further classify the observations  
  
m7.s$type.sub.H1 <- with(m7.s, ifelse(A & H1, T, F)) # if A and H1 then subtype = H1  
  
m7.s$type.sub.H3 <- with(m7.s, ifelse(A & H3, T, F)) # if A and H3 then subtype = H3  
  
m7.s$type.sub.PH1 <- with(m7.s, ifelse(A & PA & PH1, T, F))  
  
m7.s$type.B <- with(m7.s, ifelse(B, T, F))  
  
m7.s$type.H3N2.and.B <- with(m7.s, ifelse(B & type.sub.H3, T, F))  
  
m7.s$type.H3N2.and.PH1 <- with(m7.s, ifelse(type.sub.H3 & type.sub.PH1, T, F))  
  
m7.s$type.B.and.PH1 <- with(m7.s, ifelse(type.B & type.sub.PH1, T, F))  
  
m7.s$type.sub.indet <- with(m7.s, ifelse(((A | H3 | H1 | PH1 | PA) # any one of the A reactions  
 & !(type.sub.H3|type.sub.H1|type.sub.PH1)), T, F)) # & did not meet a subt def  
  
m7.s$type.neg <- with(m7.s, ifelse(RP & # RP positive w/o another pos is a true negative reaction.  
 !(A|B|H3|PH1|PA), T, F))  
  
m7.s$type.badass <- with(m7.s, ifelse(!RP & # RP negative and everything else neg is a bad assay.  
 !(A | H3 | H1 | PH1 | PA | B ), T, F))  
  
## Assign a number code to each possible combination of results (among those observed)  
  
m7.s$num <- NA  
m7.s$num[m7.s$type.sub.H1 == T] <- 1  
m7.s$num[m7.s$type.sub.H3 == T] <- 2  
m7.s$num[m7.s$type.sub.PH1 == T] <- 3  
m7.s$num[m7.s$type.B == T] <- 4  
m7.s$num[m7.s$type.neg == T] <- 5  
m7.s$num[m7.s$type.H3N2.and.B == T] <- 6  
m7.s$num[m7.s$type.H3N2.and.PH1 == T] <- 7  
m7.s$num[m7.s$type.B.and.PH1 == T] <- 8  
m7.s$num[m7.s$type.sub.indet == T] <- 9  
m7.s$num[m7.s$type.badass == T] <- 10  
  
# Number of rows without a number assigned  
print(nrow(filter(m7.s, is.na(num))))

## [1] 0

# Number of rows without a number assigned - assigned to an object  
check <- m7.s %>%   
 filter(is.na(num))  
  
# Create a subset of the data (m7.s1) with subj who have only one assay or have more than one assay but the same result each time having only one row. And, with more than one row for those subj with different results for repeat assays  
# Keep only rows that are different in both result 'num' and subject name  
m7.s1 <- m7.s[order(m7.s$subject\_id), ]  
  
# Count the number of rows n for each subject\_id and find out the subjects with multiple experiments but different sample type  
m7.s2 <- m7.s1 %>%   
 group\_by(subject\_id) %>%   
 summarise(n = n())  
# Number of subjects with 1, 2, or more obs that are different additional obs for single subjects that were not different have been deleted  
print(with(m7.s2, addmargins(table(n, exclude = c()))))

## n  
## 1 Sum   
## 183 183

## Move to classify a final subtype for each subject\_id  
  
finalsubtype <- m7.s1 %>%  
 rename(subject.id = subject\_id, type.inf = num) %>%   
 select(subject.id, type.inf, A, B, H1, H3, PH1, PA, RP)  
finalsubtype$subject.id <- as.integer(finalsubtype$subject.id)  
  
# Assign labels to type.inf  
finalsubtype$type.inf <-   
 factor(finalsubtype$type.inf, levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),  
 labels = c('seasonal H1','H3N2','Pandemic H1','B','Negative','H3N2 and B','H3N2 and PH1','B and PH1','Indeterminate','bad assay'))  
finalsubtype$type.inf <- as.character(finalsubtype$type.inf)  
  
indeterminate <- finalsubtype %>%   
 filter(type.inf == 'Indeterminate')  
  
finalsubtype$type.inf[finalsubtype$subject.id == 95] <- 'B and unsubtypable A'  
finalsubtype$type.inf[finalsubtype$subject.id == 176] <- 'H3N2'   
finalsubtype$type.inf[finalsubtype$subject.id == 335] <- 'B'   
finalsubtype$type.inf[finalsubtype$subject.id == 64] <- 'Unsubtypable A'   
  
# There are 183 entries in the finalsubtype df  
# Let's check how many negative, indeterminate, and others there are - also, wondering why 183 as opposed to 178 enrolled?  
  
finalsubtype\_pos <- finalsubtype %>%  
 filter(type.inf != "Negative")  
# 149 subjects that were positive were represented here in the finalsubtype df  
# Which ones of these were ignored in the final set of 142 subject IDs?  
# Note that this finalsubtype df is later updated in this script and this may address this question. So I'll hold off on answering this now and see what the updated finalsubtype df looks like.  
  
# Write out this finalsubtype  
saveRDS(finalsubtype, file = "Curated Data/Cleaned Data/EMIT\_subtypes.RDS")  
  
#### READ in and work with "2016.06.17 1st visit NP swab subtyping II.csv" ####  
  
part2 <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.06.17 1st visit NP swab subtyping II.csv', as.is = T)  
part2 <- part2 %>%   
 select(-Well, -Well.Type, -Threshold..dR.)  
  
# Number of rows in part2 file  
print(nrow(part2))

## [1] 206

# Number of columns in part2 file  
print(ncol(part2))

## [1] 3

# Remove the rows within the data (e.g. reference, calibration, etc.) not needed for analysis  
n2 <- part2 %>%   
 filter(Ct..dR. != 'Reference') %>%   
 filter(!grepl('[A-Z]\_[A-Z]', Well.Name)) %>%   
 filter(!grepl('PosControl', Well.Name))   
  
# Generate a new column for subject\_ID which can be obtained from sample\_ID  
n2 <- n2 %>%   
 mutate(subject\_id = gsub('\_[1-9]\*\_[0-Z]\*[0-Z]\*[0-Z]\*[0-Z]\*', '', Well.Name)) %>%   
 mutate(subject\_id = gsub('nf[0-Z]\*', '', subject\_id)) %>%   
 mutate(subject\_id = gsub('dm[0-Z]\*', '', subject\_id)) %>%  
 mutate(types = gsub('[0-9]\*\_[0-9]\*', '', Well.Name))  
n2$subject\_id <- as.numeric(n2$subject\_id)  
n2 <- n2 %>%   
 arrange(subject\_id)  
  
#### Nothing ever gets done with the above n2 object. Is this correct? ####  
# It looks like all of the data from this n2 df gets incorporated elsewhere - probably from other pcr assays.  
# The borderline H3 detection (ct >41) for 226\_7 (NPS) is not reflected in the final data and instead is marked as Unsubtypable A  
  
#### Merge the PCR data with sample virus subtype ####  
  
flu.types <- readRDS("Curated Data/Cleaned Data/EMIT\_subtypes.RDS")  
flu.types <- select(flu.types, subject.id, type.inf)  
# Remember here from above that this df has data from 183 subjects, 149 of whom are positive.   
  
# Include subtype (from the flu.types df, formerly an object called finalsubtype) with the total.pcr file  
# Note that the flu.types file has 156 subject IDs (presumably the enrolled and flu positive individuals)  
# But how many subject IDs does the total.pcr file have?  
total.pcr\_subjectID\_count <- total.pcr %>%  
 group\_by(subject.id) %>%  
 summarise(count = n())  
# Note that there are actually 202 subjectIDs here in this total.pcr file!  
  
# Were 46 of these 202 with pcr data negative and that's how we got the 156 positive cases?   
# There are only 183 subjects included in the subtype analysis, 149 of which are positive.  
# Do the 149 positives match the 156 positive cases that we ultimately get?  
# If so, then what happened to the other 7 that are positive but not included in the subtype df?  
  
includetype <- inner\_join(total.pcr, flu.types, by = "subject.id")  
includetype <- rename(includetype, sample.id = Well.Name)  
  
# The includetype df has 1713 observations  
includetype\_subjectID\_check <- includetype %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
# Those 1713 observations are made of data from 178 individuals (the 178 enrolled)  
  
includetype1 <- includetype %>%   
 filter(type.inf == 'Negative')  
# The includetype1 df has 407 observations  
  
# However, we can already look at this set and see that what might have been classified previously as negative sometimes has a positive for an aerosol sample, or on a subsequent day of testing. We see lots of observations in this set of 407 that have ct values!  
# We need to check to see what happens to these data moving forward.   
# The "negative" df generated below addresses this!  
  
# How many subject IDs are in this includetype1 df?  
includetype1\_subjectID\_check <- includetype1 %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
print(nrow(includetype1\_subjectID\_check))

## [1] 29

# Those 407 observations are made of data from 29 individuals  
# It looks like many of these "subtype negative" but positive pcr individuals will later get their subtype status updated based on the pcr data.  
  
#### Merge PCR data with sample type ####  
  
allsamples <- readRDS("Curated Data/Cleaned Data/EMIT\_samples.cc.RDS")  
sampletype <- allsamples %>%   
 select(subject.id, sample.id, sample.type)  
# This includes 1938 observations, but from how many subject IDs? Let's check.  
sampletype\_subjectID\_check <- sampletype %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
print(nrow(sampletype\_subjectID\_check))

## [1] 355

# There were 355 subject IDs here that produce the 1938 observations.   
  
# It looks like there are some samples where the sample type is missing - need to fill these in!  
sampletype\_na\_check <- allsamples %>%   
 select(subject.id, sample.id, sample.type) %>%  
 filter(is.na(sample.type))  
# This shows that there are 19 instances (samples) where we are missing the sample type  
# This could have implications for downstream data cleaning steps when we merge with this sampletype df!  
  
# Find out the negative virus subtype but have positive pcr results. From these we can define the virus subtypes of some of the negative cases  
# I don't understand what the above comment is trying to get at. If a case was negative, then there would be no subtype, correct?  
# Unless we are actually talking about negative aerosol samples, which seems to make more sense, since the includetype df includes pcr data only for the aerosol samples  
negative <- includetype %>%   
 filter(type.inf == 'Negative' & Ct..dRn. > 0) %>%   
 select(Experiment, sample.id, Ct..dRn., subject.id, type, type.inf) %>%  
 inner\_join(sampletype, by = c("subject.id", "sample.id"))  
print(nrow(negative))

## [1] 25

# This gives 27 observations from 9 subject IDs (hmm, really?)  
# In this instance the inner\_join with the sampletype df does not change the resulting "negative" df - still 27 observations from 9 subject IDs  
  
# Subjects with negative sample type but positive pcr results from either GII sample or 2rd/3rd NP swab.  
print(negative)

## Experiment  
## 1 15. 2015.07.29.013.243 2012-13 samples FluA\_B and FluA PCR  
## 2 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 3 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 4 15. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 5 15. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 6 18. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 7 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 8 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 9 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 10 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 11 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 12 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 13 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 14 23. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 15 19. 2015.10.18.014.120 2012-2013 Samples PCR Flu A&B  
## 16 19. 2015.10.18.014.120 2012-2013 Samples PCR Flu A&B  
## 17 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 18 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 19 16. 2015.08.21.014.116 2012-2013 Samples PCR Flu A&B  
## 20 16. 2015.08.21.014.116 2012-2013 Samples PCR Flu A&B  
## 21 16. 2015.08.21.014.116 2012-2013 Samples PCR Flu A&B  
## 22 15. 2015.08.20.014.114 2012-2013 Samples PCR Flu A&B  
## 23 15. 2015.08.20.014.114 2012-2013 Samples PCR Flu A&B  
## 24 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 25 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## sample.id Ct..dRn. subject.id type type.inf sample.type  
## 1 47\_11 44.85 47 B Negative Impactor 5 um NO mask  
## 2 52\_17 37.52 52 B Negative Impactor 5 um NO mask  
## 3 58\_2 35.6 58 B Negative Impactor 5 um NO mask  
## 4 105\_3 34.05 105 A Negative GII condensate NO mask  
## 5 105\_3 33.55 105 A Negative GII condensate NO mask  
## 6 223\_9 37.81 223 A Negative GII condensate NO mask  
## 7 223\_3 35.63 223 B Negative GII condensate NO mask  
## 8 223\_3 33.56 223 B Negative GII condensate NO mask  
## 9 223\_3 33.45 223 B Negative GII condensate NO mask  
## 10 223\_8 30.97 223 B Negative Nasopharyngeal swab  
## 11 223\_8 31.25 223 B Negative Nasopharyngeal swab  
## 12 223\_9 30.62 223 B Negative GII condensate NO mask  
## 13 223\_9 30.38 223 B Negative GII condensate NO mask  
## 14 226\_3 37.49 226 A Negative GII condensate NO mask  
## 15 231\_3 36.75 231 B Negative GII condensate NO mask  
## 16 231\_3 37.67 231 B Negative GII condensate NO mask  
## 17 231\_8 28.21 231 B Negative Nasopharyngeal swab  
## 18 231\_8 27.88 231 B Negative Nasopharyngeal swab  
## 19 327\_2 36.2 327 B Negative Impactor 5 um NO mask  
## 20 327\_3 31.79 327 B Negative GII condensate NO mask  
## 21 327\_3 31.9 327 B Negative GII condensate NO mask  
## 22 329\_2 36.82 329 B Negative Impactor 5 um NO mask  
## 23 329\_3 34.2 329 B Negative GII condensate NO mask  
## 24 365\_3 35.19 365 B Negative GII condensate NO mask  
## 25 365\_3 31.95 365 B Negative GII condensate NO mask

saveRDS(negative, "Curated Data/Cleaned Data/negative subtype sample with positive pcr.RDS")  
  
## Based on the pcr results from GII samples or 2rd/3rd np, we have modified a few subjects' subtype  
  
#### READ in "negative subtype sample with positive pcr.RDS" ####  
  
updatetype <- readRDS("Curated Data/Cleaned Data/negative subtype sample with positive pcr.RDS")  
  
updatetype1 <- updatetype %>%   
 select(subject.id, type) %>%   
 distinct(subject.id, type, .keep\_all = TRUE)  
  
updatetype2 <- updatetype1 %>%   
 filter(type == 'A')  
  
updatetype3 <- updatetype1 %>%   
 filter(type == 'B')  
  
finalsubtype$type.inf[finalsubtype$subject.id == 52] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 58] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 105] <- 'Unsubtypable A'   
finalsubtype$type.inf[finalsubtype$subject.id == 223] <- 'B and unsubtypable A'  
finalsubtype$type.inf[finalsubtype$subject.id == 226] <- 'Unsubtypable A'   
finalsubtype$type.inf[finalsubtype$subject.id == 327] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 231] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 329] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 365] <- 'B'  
  
# Note: preveiously Jing had commented out the lines that gave subject IDs 52 and 58 the subtype B designation  
# However, there is no evidence to do this, and the pcr files do show that these subject had samples that were positive for flu B  
# Jing also had indicated that 223 was a B type of influenza, however, this is not true - 223 is actually type A and B  
# Perhaps this was an error of misreading the updatetype df?  
  
#### READ in "EMIT\_samples.cc.RDS" ####  
  
enrollcheck <- readRDS('Curated Data/Cleaned Data/EMIT\_samples.cc.RDS') %>%   
 select(subject.id, enrolled) %>%   
 distinct(subject.id, enrolled, .keep\_all = TRUE) %>%   
 filter(enrolled == TRUE)  
  
finalenrolltype <- finalsubtype %>%  
 semi\_join(enrollcheck, by = 'subject.id') %>%  
 arrange(subject.id) %>%  
 select(subject.id, type.inf)  
  
saveRDS(finalenrolltype, "Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled.RDS")  
  
finalenrolledpositive <- finalenrolltype %>%   
 filter(!type.inf == 'Negative')  
# This shows that there were actually 158 positive cases (when we added the 2 flu B's on that were previously commented out: SID 52 and 58)  
  
saveRDS(finalenrolledpositive, "Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled\_positive.RDS")  
  
negative <- finalenrolltype %>%   
 filter(type.inf == 'Negative')  
  
h3n2 <- finalenrolltype %>%   
 filter(type.inf == 'H3N2')  
  
B <- finalenrolltype %>%   
 filter(type.inf == 'B')  
  
Pandemic.H1 <- finalenrolltype %>%   
 filter(type.inf == 'Pandemic H1')  
  
#### Check that all the negative subjects do not have any positive GII or 2nd/3rd np positive PCR samples ####  
  
flu.typesenroll <- readRDS("Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled.RDS")  
flu.typesenroll <- flu.typesenroll %>%  
 select(subject.id, type.inf)  
  
# Add the type of infection to the total.pcr df and create new df called includetypeenroll  
includetypeenroll <- total.pcr %>%  
 inner\_join(flu.typesenroll, by = "subject.id") %>%  
 rename(sample.id = Well.Name)  
# This gives a df with 1,713 observations that should be from the 178 enrolled participants  
  
# Let's check to make sure that these 1,713 observations do indeed come from 178 enrolled participants  
includetypeenroll\_subjectID\_check <- includetypeenroll %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
# This is correct! We see that there is data on 178 enrolled participants  
  
includetypeenrollneg <- includetypeenroll %>%   
 filter(type.inf == 'Negative') %>%  
 inner\_join(allsamples, by = c("subject.id", "sample.id")) %>%   
 filter(focus.ct > 0)  
# 50, 234, 306 are negative cases with all samples negative for PCR, but 50\_3, 234\_3, 306\_3 are positive for focus assay  
  
#### READ in "EMIT\_subtypes\_enrolled\_positive.RDS" ####  
  
flu.typepositive <- readRDS("Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled\_positive.RDS")  
  
#### Check the unmatched cases between sample virus subtype and PCR results ####  
  
# Add the infection types from the positive cases to the total.pcr df and add the sample types to each of the sample ids  
includetypepostive <- total.pcr %>%  
 inner\_join(flu.typepositive, by = "subject.id") %>%   
 rename(sample.id = Well.Name) %>%  
 inner\_join(sampletype, by = c("subject.id", "sample.id"))  
# \*\*\* Note: using an inner\_join here for the sampletype eliminates 2 observations in the includetypepositive df that would otherwise be there were the sampletype df complete and did not contain NAs - there are 19 observations in the sampletype df that are missing their sample type (i.e., NPS, Fine, Coarse, Throat, Ant Nares, etc.)  
# I may be able to locate these samples in the freezer and determine the sample types.   
  
# If the inner\_join with the sampletype df with missing sample types is used, then we get a "includetypepositive" df with 1461 observations  
# How many subject IDs contribute to these 1461 observations.  
# \*Remember that if we fix the sampletype file (i.e., fill in the 19 missing sample types), there is potential for 1463 observations  
includetypepositive\_subjectID\_check <- includetypepostive %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
print(nrow(includetypepositive\_subjectID\_check))

## [1] 158

# There are 158 subjectIDs represented in this includetypepostive df  
  
# Let's check to see if there are any intances where the type is B but the subtype (type.inf variable) is not B  
  
unmatched1 <- includetypepostive %>%   
 filter(type == 'B' & !grepl('B', type.inf))  
print(unmatched1)

## Experiment  
## 1 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 2 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 3 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 4 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 5 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 6 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 7 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 8 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 9 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 10 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 11 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 12 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 13 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 14 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 15 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 16 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 17 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 18 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 19 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 20 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 21 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 22 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 23 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 24 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 25 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 26 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 27 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 28 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 29 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 30 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 31 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 32 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 33 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 34 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 35 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 36 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 37 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 38 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 39 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 40 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 41 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 42 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 43 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 44 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 45 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 46 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 47 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 48 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 49 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 50 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 51 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 52 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 53 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 54 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 55 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 56 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 57 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 58 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 59 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 60 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 61 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 62 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 63 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 64 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 65 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 66 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 67 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 68 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 69 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 70 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 71 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 72 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 73 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 74 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## sample.id Ct..dRn. copies.in subject.id type copy.num  
## 1 31\_12 NA 31 B NA  
## 2 31\_12 NA 31 B NA  
## 3 31\_13 NA 31 B NA  
## 4 31\_13 NA 31 B NA  
## 5 31\_15 NA 31 B NA  
## 6 31\_15 NA 31 B NA  
## 7 31\_16 NA 31 B NA  
## 8 31\_16 NA 31 B NA  
## 9 31\_7 NA 31 B NA  
## 10 31\_7 NA 31 B NA  
## 11 31\_9 NA 31 B NA  
## 12 31\_9 NA 31 B NA  
## 13 64\_10 NA 64 B NA  
## 14 64\_10 NA 64 B NA  
## 15 64\_11 NA 64 B NA  
## 16 64\_11 NA 64 B NA  
## 17 64\_2 NA 64 B NA  
## 18 64\_2 NA 64 B NA  
## 19 64\_3 NA 64 B NA  
## 20 64\_3 NA 64 B NA  
## 21 64\_7 NA 64 B NA  
## 22 64\_7 NA 64 B NA  
## 23 94\_2 NA 94 B NA  
## 24 94\_2 NA 94 B NA  
## 25 94\_3 NA 94 B NA  
## 26 94\_3 NA 94 B NA  
## 27 94\_6 NA 94 B NA  
## 28 94\_6 NA 94 B NA  
## 29 105\_2 NA 105 B NA  
## 30 105\_2 NA 105 B NA  
## 31 105\_3 NA 105 B NA  
## 32 105\_3 NA 105 B NA  
## 33 114\_12 NA 114 B NA  
## 34 114\_12 NA 114 B NA  
## 35 114\_7 NA 114 B NA  
## 36 114\_7 NA 114 B NA  
## 37 114\_9 NA 114 B NA  
## 38 114\_9 NA 114 B NA  
## 39 127\_14 NA 127 B NA  
## 40 127\_14 NA 127 B NA  
## 41 127\_15 NA 127 B NA  
## 42 127\_15 NA 127 B NA  
## 43 127\_18 NA 127 B NA  
## 44 127\_18 NA 127 B NA  
## 45 136\_11 NA 136 B NA  
## 46 136\_11 NA 136 B NA  
## 47 136\_2 NA 136 B NA  
## 48 136\_2 NA 136 B NA  
## 49 136\_3 NA 136 B NA  
## 50 136\_3 NA 136 B NA  
## 51 136\_8 NA 136 B NA  
## 52 136\_8 NA 136 B NA  
## 53 136\_9 NA 136 B NA  
## 54 136\_9 NA 136 B NA  
## 55 176\_11 NA 176 B NA  
## 56 176\_11 NA 176 B NA  
## 57 176\_12 NA 176 B NA  
## 58 176\_12 NA 176 B NA  
## 59 176\_14 NA 176 B NA  
## 60 176\_14 NA 176 B NA  
## 61 176\_16 NA 176 B NA  
## 62 176\_16 NA 176 B NA  
## 63 176\_2 NA 176 B NA  
## 64 176\_2 NA 176 B NA  
## 65 176\_3 NA 176 B NA  
## 66 176\_3 NA 176 B NA  
## 67 176\_7 NA 176 B NA  
## 68 176\_7 NA 176 B NA  
## 69 176\_9 NA 176 B NA  
## 70 176\_9 NA 176 B NA  
## 71 226\_2 NA 226 B NA  
## 72 226\_2 NA 226 B NA  
## 73 226\_3 NA 226 B NA  
## 74 226\_3 NA 226 B NA  
## date  
## 1 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 2 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 3 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 4 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 5 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 6 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 7 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 8 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 9 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 10 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 11 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 12 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 13 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 14 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 15 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 16 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 17 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 18 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 19 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 20 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 21 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 22 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 23 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 24 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 25 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 26 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 27 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 28 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 29 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 30 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 31 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 32 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 33 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 34 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 35 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 36 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 37 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 38 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 39 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 40 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 41 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 42 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 43 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 44 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 45 2015.03.04.013.209 2012-13 samples FluB PCR  
## 46 2015.03.04.013.209 2012-13 samples FluB PCR  
## 47 2015.03.04.013.209 2012-13 samples FluB PCR  
## 48 2015.03.04.013.209 2012-13 samples FluB PCR  
## 49 2015.03.04.013.209 2012-13 samples FluB PCR  
## 50 2015.03.04.013.209 2012-13 samples FluB PCR  
## 51 2015.03.04.013.209 2012-13 samples FluB PCR  
## 52 2015.03.04.013.209 2012-13 samples FluB PCR  
## 53 2015.03.04.013.209 2012-13 samples FluB PCR  
## 54 2015.03.04.013.209 2012-13 samples FluB PCR  
## 55 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 56 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 57 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 58 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 59 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 60 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 61 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 62 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 63 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 64 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 65 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 66 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 67 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 68 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 69 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 70 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 71 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 72 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 73 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 74 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## cfactor virus.copies type.inf sample.type  
## 1 1.2888043 NA H3N2 Impactor 5 um NO mask  
## 2 1.2888043 NA H3N2 Impactor 5 um NO mask  
## 3 1.2888043 NA H3N2 Nasopharyngeal swab  
## 4 1.2888043 NA H3N2 Nasopharyngeal swab  
## 5 1.2888043 NA H3N2 GII condensate NO mask  
## 6 1.2888043 NA H3N2 GII condensate NO mask  
## 7 1.2888043 NA H3N2 Impactor 5 um NO mask  
## 8 1.2888043 NA H3N2 Impactor 5 um NO mask  
## 9 1.2888043 NA H3N2 Nasopharyngeal swab  
## 10 1.2888043 NA H3N2 Nasopharyngeal swab  
## 11 1.2888043 NA H3N2 GII condensate NO mask  
## 12 1.2888043 NA H3N2 GII condensate NO mask  
## 13 1.2888043 NA Unsubtypable A GII condensate NO mask  
## 14 1.2888043 NA Unsubtypable A GII condensate NO mask  
## 15 1.2888043 NA Unsubtypable A Impactor 5 um NO mask  
## 16 1.2888043 NA Unsubtypable A Impactor 5 um NO mask  
## 17 2.8832812 NA Unsubtypable A Impactor 5 um NO mask  
## 18 2.8832812 NA Unsubtypable A Impactor 5 um NO mask  
## 19 2.8832812 NA Unsubtypable A GII condensate NO mask  
## 20 2.8832812 NA Unsubtypable A GII condensate NO mask  
## 21 1.2888043 NA Unsubtypable A Nasopharyngeal swab  
## 22 1.2888043 NA Unsubtypable A Nasopharyngeal swab  
## 23 1.6774462 NA Pandemic H1 Impactor 5 um NO mask  
## 24 1.6774462 NA Pandemic H1 Impactor 5 um NO mask  
## 25 1.6774462 NA Pandemic H1 GII condensate NO mask  
## 26 1.6774462 NA Pandemic H1 GII condensate NO mask  
## 27 1.6774462 NA Pandemic H1 Nasopharyngeal swab  
## 28 1.6774462 NA Pandemic H1 Nasopharyngeal swab  
## 29 6.5092955 NA Unsubtypable A Impactor 5 um NO mask  
## 30 6.5092955 NA Unsubtypable A Impactor 5 um NO mask  
## 31 6.5092955 NA Unsubtypable A GII condensate NO mask  
## 32 6.5092955 NA Unsubtypable A GII condensate NO mask  
## 33 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 34 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 35 0.9444983 NA H3N2 Nasopharyngeal swab  
## 36 0.9444983 NA H3N2 Nasopharyngeal swab  
## 37 0.9444983 NA H3N2 GII condensate NO mask  
## 38 0.9444983 NA H3N2 GII condensate NO mask  
## 39 0.9444983 NA H3N2 Nasopharyngeal swab  
## 40 0.9444983 NA H3N2 Nasopharyngeal swab  
## 41 0.9444983 NA H3N2 GII condensate NO mask  
## 42 0.9444983 NA H3N2 GII condensate NO mask  
## 43 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 44 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 45 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 46 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 47 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 48 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 49 1.4690044 NA H3N2 GII condensate NO mask  
## 50 1.4690044 NA H3N2 GII condensate NO mask  
## 51 1.4690044 NA H3N2 Nasopharyngeal swab  
## 52 1.4690044 NA H3N2 Nasopharyngeal swab  
## 53 1.4690044 NA H3N2 GII condensate NO mask  
## 54 1.4690044 NA H3N2 GII condensate NO mask  
## 55 1.6774462 NA H3N2 Impactor 5 um NO mask  
## 56 1.6774462 NA H3N2 Impactor 5 um NO mask  
## 57 3.6584504 NA H3N2 Nasopharyngeal swab  
## 58 3.6584504 NA H3N2 Nasopharyngeal swab  
## 59 3.6584504 NA H3N2 GII condensate NO mask  
## 60 3.6584504 NA H3N2 GII condensate NO mask  
## 61 3.6584504 NA H3N2 Impactor 5 um NO mask  
## 62 3.6584504 NA H3N2 Impactor 5 um NO mask  
## 63 6.5092955 NA H3N2 Impactor 5 um NO mask  
## 64 6.5092955 NA H3N2 Impactor 5 um NO mask  
## 65 6.5092955 NA H3N2 GII condensate NO mask  
## 66 6.5092955 NA H3N2 GII condensate NO mask  
## 67 1.6774462 NA H3N2 Nasopharyngeal swab  
## 68 1.6774462 NA H3N2 Nasopharyngeal swab  
## 69 1.6774462 NA H3N2 GII condensate NO mask  
## 70 1.6774462 NA H3N2 GII condensate NO mask  
## 71 0.8392489 NA Unsubtypable A Impactor 5 um NO mask  
## 72 0.8392489 NA Unsubtypable A Impactor 5 um NO mask  
## 73 0.8392489 NA Unsubtypable A GII condensate NO mask  
## 74 0.8392489 NA Unsubtypable A GII condensate NO mask

# Even though there are a bunch of observations here that are labeled type == 'B' but type.inf not 'B', none of these observations have pcr evidence for B virus.  
# Thus, the type.inf appears to be not incorrect for each of these observations. We would have to check to see that there was pcr evidence for the type of infection labeled by type.inf in order to prove that these type.inf labels were correct.   
  
# Let's also check to see if there are any instances where the type is A but the subtype (type.inf variable) is not A   
unmatched2 <- includetypepostive %>%  
 filter( type == 'A' & type.inf == 'B')  
  
# Similarly, there were no instances where there was a type.inf of B where any samples were pcr positive for A virus.   
  
# Does this mean we can eliminate these observations in the unmatched1 and unmatched2 dfs? Yes.  
  
includetypepostiveupdate <- includetypepostive %>%  
 anti\_join(unmatched1, by = c("Experiment", "sample.id", "Ct..dRn.", "copies.in", "subject.id", "type", "copy.num", "date", "cfactor", "virus.copies", "type.inf", "sample.type")) %>%   
 anti\_join(unmatched2 , by = c("Experiment", "sample.id", "Ct..dRn.", "copies.in", "subject.id", "type", "copy.num", "date", "cfactor", "virus.copies", "type.inf", "sample.type"))  
  
# SubjectID check on includetypepositiveupdate  
  
includetypepostiveupdate\_subjectID\_check <- includetypepostiveupdate %>%  
 group\_by(subject.id) %>%  
 filter(!is.na(subject.id)) %>%  
 count()  
# We see that we have observations for all 158 enrolled and positive subjects  
# We notice that subjectID 52 has extraneous observations because there are 2 identical pcr experiments that have slightly different names  
# One is called: "15. 2015.07.29.013.243 2012-13 samples FluA\_B and FluA PCR"  
# Another is called: "17. 2015.7.29.013.243 2012-13 samples FluA\_B and FluA PCR"  
  
# \*\*\* We can go back and fix this in the pcr files  
  
#### Pick pos fluA, pos fluB, join, assign copy number ####  
  
# Pick out all the PCR with A assay results  
allpcrA <- includetypepostiveupdate %>%   
 filter(type == 'A') %>%  
 rename(virus.copiesA = virus.copies, typeA = type) %>%  
 rename(CtA = Ct..dRn.)  
  
# Pick out all the PCR with B assay results  
allpcrB <- includetypepostiveupdate %>%   
 filter(type == 'B') %>%  
 rename(virus.copiesB = virus.copies, typeB = type) %>%  
 rename(CtB = Ct..dRn.)  
  
# Join both A and B assay, and also add sample type in the data list, assign the final RNA copies number for each sample type  
allPCR <- allpcrA %>%  
 full\_join(allpcrB, by = c('subject.id', 'Experiment', 'type.inf', 'sample.id', 'sample.type', "copies.in", 'copy.num', 'date', 'cfactor')) %>%  
 arrange(subject.id) %>%   
 filter(!sample.type == 'Throat Swab')  
  
## DATA EDITING (dilution factors different for a few samples) ##  
# Flagged samples all NP swabs, dilution factor is 50  
# 66\_7 120\_7 184\_8 188\_7 189\_7 192\_7 196\_7 262\_7 277\_7 284\_12 284\_7 296\_12 296\_7  
  
allPCR1 <- allPCR %>%   
 filter(sample.id == '66\_7' | sample.id == '120\_7' | sample.id == '184\_8' | sample.id == '188\_7' | sample.id == '189\_7' | sample.id == '192\_7' |   
 sample.id == '196\_7' | sample.id == '262\_7' | sample.id == '277\_7' | sample.id == '284\_12' | sample.id == '284\_7' |   
 sample.id == '296\_12'| sample.id=='296\_7')  
  
allPCR2 <- allPCR %>%  
 anti\_join(allPCR1)

## Joining, by = c("Experiment", "sample.id", "CtA", "copies.in", "subject.id", "typeA", "copy.num", "date", "cfactor", "virus.copiesA", "type.inf", "sample.type", "CtB", "typeB", "virus.copiesB")

allPCR3 <- allPCR1 %>%   
 mutate(final.copiesA = virus.copiesA\*50, final.copiesB = virus.copiesB\*50)  
  
allPCR4 <- allPCR2 %>%   
 filter(!sample.type == 'Nasopharyngeal swab') %>%   
 mutate(final.copiesA = virus.copiesA\*25, final.copiesB = virus.copiesB\*25)  
  
allPCR5 <- allPCR2 %>%   
 filter(sample.type == 'Nasopharyngeal swab') %>%  
 mutate(final.copiesA = virus.copiesA\*100, final.copiesB = virus.copiesB\*100)  
  
allPCRtotal <- rbind(allPCR3, allPCR4, allPCR5) %>%   
 ungroup  
  
allPCR.A <- allPCRtotal %>%   
 filter(typeA == 'A') %>%  
 select(-CtB, -virus.copiesB, -typeB, -final.copiesB) %>%  
 rename(Ct = CtA, virus.copies = virus.copiesA, type = typeA, final.copies = final.copiesA)  
  
allPCR.B <- allPCRtotal %>%   
 filter(typeB == 'B') %>%  
 select(-CtA, -virus.copiesA, -typeA, -final.copiesA) %>%  
 rename(Ct = CtB, virus.copies = virus.copiesB, type = typeB, final.copies = final.copiesB)  
  
# 2. merge the seperated FILE FOR A and B back together, successfully make,- one row for one subject.id  
allPCRfinal <- rbind(allPCR.A, allPCR.B) %>%   
 ungroup  
  
allPCRfinal\_subjectID\_check <- allPCRfinal %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(allPCRfinal\_subjectID\_check))

## [1] 158

# Data exists here for the 158 positive enrolled cases  
  
saveRDS(allPCRfinal, "Curated Data/Cleaned Data/pcr data for gii and 2nd&3rd np.RDS")  
  
#### Check to make sure no repeats ####  
  
no.repeats.pos <- allPCRfinal %>%   
 group\_by(sample.id) %>%  
 summarise(n = n())  
# This is useful, there should be 2 repeats per sample (becasue samples were run in duplicate)  
# However, in cases where a sample was positive for both fluA and B, then there will be 4 pcr data repeats per sample (a duplicate for flu A assay and a duplicate for flu B assay)  
# However, in the case where we ran out of sample and went to a new tube and the new tube had a slighly different label (for example, samples with subscripts \_1 and \_7 are often NPS for the first day of sampling for a particular subject)  
  
# Subjects with known sample type should only have 2 PCR result  
no.repeats.pos.1 <- no.repeats.pos %>%   
 filter(n == 1) %>%   
 left\_join(allPCR, by = 'sample.id') %>%   
 select(Experiment, sample.id, CtA, copies.in, typeA, type.inf, CtB, typeB)  
# Subjects with known sample type have only one PCR result (Hmm - I'm don't understand this)  
print(no.repeats.pos.1)

## # A tibble: 0 x 8  
## # ... with 8 variables: Experiment <chr>, sample.id <chr>, CtA <chr>,  
## # copies.in <dbl>, typeA <chr>, type.inf <chr>, CtB <chr>, typeB <chr>

no.repeats.pos.2 <- no.repeats.pos %>%   
 filter(n == 4) %>%  
 left\_join(allPCR, by = 'sample.id') %>%  
 select(Experiment, sample.id, CtA, copies.in, typeA, type.inf, CtB, typeB)  
# Subjects with known sample type have two PCR result  
print(no.repeats.pos.2)

## # A tibble: 92 x 8  
## Experiment sample.id CtA copies.in typeA type.inf CtB typeB  
## <chr> <chr> <chr> <dbl> <chr> <chr> <chr> <chr>  
## 1 2. 2014.03.31.013… 107\_2 <NA> NA <NA> B "" B   
## 2 2. 2014.03.31.013… 107\_2 <NA> NA <NA> B "" B   
## 3 4. 2015.01.26.014… 107\_2 <NA> NA <NA> B "" B   
## 4 4. 2015.01.26.014… 107\_2 <NA> NA <NA> B "" B   
## 5 2. 2014.03.31.013… 107\_3 <NA> 0.0913 <NA> B 37.03 B   
## 6 2. 2014.03.31.013… 107\_3 <NA> 0.102 <NA> B 36.87 B   
## 7 4. 2015.01.26.014… 107\_3 <NA> 0.0675 <NA> B 37.45 B   
## 8 4. 2015.01.26.014… 107\_3 <NA> 0.173 <NA> B 36.16 B   
## 9 2. 2014.03.31.013… 110\_2 <NA> NA <NA> B "" B   
## 10 2. 2014.03.31.013… 110\_2 <NA> NA <NA> B "" B   
## # ... with 82 more rows

## This is interesting because it shows when a sample was repeated on a pcr assay (not talking about duplicates here, but rather repeat assays).   
# Is there a rule about which of the repeats to keep and which to exclude? For example, should the most recent assay be taken and the other excluded?  
# Does this ever get addressed later in any code?  
  
#### \*\*\*\* Using Script: Jing and Dr. Milton's "1st np swab quantity.R" \*\*\*\* ####  
  
###  
## Original file information:  
  
# Author: Jing Yan & Don Milton  
# Date: September 17, 2015  
# Revision Date: Jun 30, 2016  
# Title: 1st np swab quantity.R  
# Purpose: To sort the data files from the lab (Michael Grantham) PCR data for 1st np samples   
  
# Input files:  
# InputFiles\_UMD/PCR results/2016.06.17 1st visit NP swab FluA quant.csv  
# InputFiles\_UMD/PCR results/2016.06.17 1st visit NP swab FluB quant.csv  
# Output files: R\_output  
  
### Procedures:  
  
# 1. Sort out first NP swab PCR results(includes A and B, combine the two parts after sorting)  
# 2. For the first NP swab, seperate the subjects with flu A infection, flu B infection,   
# negative on the swab and dual infection of A and B  
# 3. Question: How to treat the samples with multiple PCR results?   
  
## Method 1  
# 1 obs---use  
# >=2 obs---take mean(if one is No Ct, Tobit fitted value)  
  
## Method 2  
# Take a tobit model(obs~sample\_id)---get fitted data for all the subjects  
# (check if fitted value match with the method1 values)  
  
## Method 3   
# Tobit(obs~sample.type+subject.id)---get fitted data for all the subjects (may be different from Method 1)  
  
# Question about the above notes (the notes above are from Jing) = how is Method 2 different from Method 3? They both say that they are meant to "get fitted data for all the subjects". Is Method 3 trying to get fitted data for all of the samples?  
  
###  
#### READ in and work with "EMIT\_subtypes\_enrolled\_positive.RDS" ####  
## Note that this df is the product of other script and in Jing's original setup, was saved to R\_output  
# The original script from Jing that produced this df was: "subtype analysis.R"  
## However, in this new setup, we have saved the EMIT\_subtypes\_enrolled\_positive.RDS in:   
# ... EMIT/EMIT\_Data\_Analysis\_Jake/EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data  
  
flu.types <- readRDS("Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled\_positive.RDS")  
flu.types <- select(flu.types, subject.id, type.inf)  
  
#### READ in "fluA\_1np\_calibration.RDS" ####  
npAcali <- readRDS("Curated Data/Cleaned Data/fluA\_1np\_calibration.RDS")  
  
#### READ in "2016.08.05 1st visit NP swab FluA quant.csv" ####  
npA <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.05 1st visit NP swab FluA quant.csv', as.is = T)  
  
#### Work with the above 2 dfs ####  
names(npA)

## [1] "Experiment" "Well" "Well.Name"   
## [4] "Well.Type" "Threshold..dRn." "Ct..dRn."   
## [7] "Quantity..copies."

npA <- npA %>%  
 filter(Ct..dRn. != 'Reference') %>%  
 filter(!grepl('\_Low', Well.Name)) %>%  
 filter(!grepl('NTC', Well.Name)) %>%  
 filter(!grepl('Standard', Well.Type)) %>%  
 filter(!grepl('High', Well.Name)) %>%  
 filter(grepl('\_', Well.Name)) %>%  
 select(-Well, -Well.Type, -Threshold..dRn.) %>%  
 rename(copies.in = Quantity..copies.)  
  
# Number of rows in part1 first NP influenza A PCR data  
nrow(npA)

## [1] 209

# Number of column in part1 first NP influenza A PCR data  
ncol(npA)

## [1] 4

npA <- npA %>%   
 mutate(subject.id = gsub('\_[1-9]\*\_[0-Z]\*', '', Well.Name)) %>%   
 mutate(subject.id = gsub('\_A', '', subject.id)) %>%   
 mutate(subject.id = gsub('nfA', '', subject.id))  
npA[npA == "No Ct"] <- ''  
npA$copies.in <- as.numeric(npA$copies.in)  
npA <- npA %>%  
 mutate(copy.num = copies.in\*100\*250) %>%  
 arrange(subject.id)  
npA$result.type <- 'A'  
npA <- npA %>%  
 mutate(sample.id = gsub('\_A', '', Well.Name)) %>%   
 mutate(sample.id = gsub('\_InfA', '', sample.id))  
npA$sample.id[npA$sample.id == '118\_1\_1'] <- '118\_1'   
npA$sample.id[npA$sample.id == '69\_1\_1'] <- '69\_1'   
npA$sample.id[npA$sample.id == '70\_1\_1'] <- '70\_1'   
npA$sample.id[npA$sample.id == '210\_1\_1'] <- '210\_1'  
npA$sample.id[npA$sample.id == '339\_1\_1'] <- '339\_1'  
npA$sample.id[npA$sample.id == '356\_1\_1'] <- '356\_1'  
npA <- npA %>%  
 distinct(subject.id, Ct..dRn., .keep\_all = TRUE)  
  
# Number of rows in first NP influenza A PCR data  
nrow(npA)

## [1] 190

# Number of columns in first NP influenza A PCR data  
ncol(npA)

## [1] 8

npA <- npA %>%  
 rename(Ct = Ct..dRn., type = result.type) %>%  
 select(-Well.Name) %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
  
npAfinal <- npA %>%  
 left\_join(npAcali, by = 'date')  
# The following samples were done without inter-run calibrators in the experiments, so the adjustment calibrators are missing  
npAfinal$cfactor[npAfinal$sample.id == '176\_6'] <- 1  
npAfinal$cfactor[npAfinal$sample.id == '64\_6'] <- 1  
npAfinal$virus.copies <- npAfinal$copy.num\*npAfinal$cfactor  
  
#### Now working with the flu B calibrations ####  
  
#### READ in "fluB\_1np\_calibration.RDS" ####  
npBcali <- readRDS("Curated Data/Cleaned Data/fluB\_1np\_calibration.RDS")  
  
#### READ in "2016.08.05 1st visit NP swab FluB quant.csv" ####  
npB <- read.csv('UMD\_Raw\_Data/PCR Data/PCR results/2016.08.05 1st visit NP swab FluB quant.csv', as.is = T)  
  
names(npB)

## [1] "Experiment" "Well" "Well.Name"   
## [4] "Well.Type" "Threshold..dRn." "Ct..dRn."   
## [7] "Quantity..copies."

npB <- npB %>%  
 filter(Ct..dRn. != 'Reference') %>%  
 filter(!grepl('\_Low', Well.Name)) %>%  
 filter(!grepl('NTC', Well.Name)) %>%  
 filter(!grepl('Standard', Well.Type)) %>%  
 filter(!grepl('High', Well.Name)) %>%  
 filter(grepl('\_', Well.Name)) %>%  
 select(-Well, -Well.Type, -Threshold..dRn.) %>%   
 rename(copies.in = Quantity..copies.)  
  
# Number of rows in part1 first NP influenza A PCR data  
nrow(npB)

## [1] 209

# Number of column in part1 first NP influenza A PCR data  
ncol(npB)

## [1] 4

npB <- npB %>%  
 mutate(subject.id = gsub('\_[1-9]\*\_[0-Z]\*', '', Well.Name)) %>%  
 mutate(subject.id = gsub('\_B', '', subject.id)) %>%  
 mutate(subject.id = gsub('nfB', '', subject.id))  
npB[npB == "No Ct"] <- ''  
npB$copies.in <- as.numeric(npB$copies.in)  
npB <- npB %>%   
 mutate(copy.num = copies.in\*100\*272) %>%  
 arrange(subject.id)  
npB$result.type <-'B'  
npB <- npB %>%  
 mutate(sample.id = gsub('\_B', '', Well.Name)) %>%  
 mutate(sample.id = gsub('\_InfB', '', sample.id))  
npB$sample.id[npB$sample.id == '118\_1\_1'] <- '118\_1'   
npB$sample.id[npB$sample.id == '210\_1\_1'] <- '210\_1'   
npB$sample.id[npB$sample.id == '339\_1\_1'] <- '339\_1'   
npB$sample.id[npB$sample.id == '356\_1\_1'] <- '356\_1'   
npB$sample.id[npB$sample.id == '69\_1\_1'] <- '69\_1'   
npB$sample.id[npB$sample.id == '70\_1\_1'] <- '70\_1'   
npB <- npB %>%  
 distinct(subject.id, Ct..dRn., .keep\_all = TRUE)  
  
# Number of rows in first NP influenza A PCR data  
nrow(npB)

## [1] 188

# Number of columns in first NP influenza A PCR data  
ncol(npB)

## [1] 8

npB <- npB %>%  
 rename(Ct = Ct..dRn., type = result.type) %>%  
 select(-Well.Name) %>%  
 mutate(date = gsub('^[0-9]\*.', '',Experiment))  
  
npBfinal <- npB %>%  
 left\_join(npBcali, by = 'date')  
npBfinal$virus.copies <- npBfinal$copy.num\*npBfinal$cfactor  
  
#### Merging together the fluA and fluB data ####  
  
npfirst <- rbind(npAfinal, npBfinal)  
npfirst$subject.id = as.integer(npfirst$subject.id)  
npfirst <- npfirst[order(npfirst$subject.id), ]  
  
# Check the number of subjectIDs, sampleIDs, and type of infection  
npfirst\_sampleID\_check <- npfirst %>%  
 group\_by(sample.id, type) %>%  
 count()  
  
# npfirst has 378 observations and now let's see how many subject IDs there are  
  
npfirst\_subjectID\_check <- npfirst %>%  
 group\_by(subject.id) %>%  
 count()  
# There are 183 subject IDs here. Most have 2 pcr results, some have more  
  
# Let's look at the subjectIDs with more than 2 pcr results  
npfirst\_subjectID\_check\_observations <- npfirst %>%  
 group\_by(subject.id) %>%  
 count() %>%  
 filter(n >2)  
# There are 11 subjects with multiple day 1 NP swabs that were run for either flu A or flu B.   
# I'm not talking about duplicates in the pcr assay - these are completely new experiments with new dates.  
# We will use all of the data, including that from repeat experiments  
  
npfirstpositive <- npfirst %>%   
 inner\_join(flu.types, by = c("subject.id"))  
  
npfirstpositive\_subjectID\_check <- npfirstpositive %>%  
 group\_by(subject.id) %>%  
 count()  
  
# Note: It appears that many of the 183 in the "npfirst" df were actually positive for flu virus, yet these were not included in the "npfirstpositive" df that includes data from the 158 enrolled and positive flu cases.  
# We need to do a review to better understand why there were some of these samples that were not included in the final dataset.   
  
unmatchedfirstnp1 <- npfirstpositive %>%   
 filter( type == 'B' & type.inf == 'H3N2')  
  
unmatchedfirstnp2 <- npfirstpositive %>%   
 filter( type == 'B' & type.inf == 'Pandemic H1')  
  
unmatchedfirstnp3 <- npfirstpositive %>%   
 filter( type == 'B' & type.inf == 'Unsubtypable A')  
  
unmatchedfirstnp4 <- npfirstpositive %>%   
 filter( type == 'B' & type.inf == 'H3N2 and PH1')  
  
unmatchedfirstnp5 <- npfirstpositive %>%   
 filter( type == 'A' & type.inf == 'B')  
  
unmatchedtotal <- rbind(unmatchedfirstnp1, unmatchedfirstnp2, unmatchedfirstnp3, unmatchedfirstnp4, unmatchedfirstnp5)  
  
# npfirstpositiveupdate <- setdiff(npfirstpositive, unmatchedtotal)  
# I actually think what Jing means here is to take the anti\_join, to remove the unmatchedtotal observations from the npfirstpositive df. Let's try this and comment out the setdiff command  
  
npfirstpositiveupdate <- npfirstpositive %>%  
 anti\_join(unmatchedtotal) %>%  
 arrange(subject.id)

## Joining, by = c("Experiment", "Ct", "copies.in", "subject.id", "copy.num", "type", "sample.id", "date", "cfactor", "virus.copies", "type.inf")

npfirstpositiveupdate$sample.type <- 'Nasopharyngeal swab'  
  
# Let's look at the number of subject IDs in this df and also check to if any samples have repeat assays  
npfirstpositiveupdate\_subjectID\_check <- npfirstpositiveupdate %>%  
 group\_by(subject.id) %>%  
 summarise(n = n()) %>%  
 arrange(n)  
# Samples that are duplicated (more than 1 assay) in the "npfirstpositiveupdate" df at least once are:   
# 64, 81, 97, 100, 104, 118, 123, 174, 176, 223, 230, 357, 95  
# Need to decide how to deal with these - average?  
# However some are negative on one run and positive on separate run - take the positive? the negative?  
# Could treat it like when there is one/two positive detections for a replicate on the PCR plate?  
  
# Not sure the following bit of code makes changes the df at all and not really sure of the intended purpose of this bit of code. I will comment it out for now.   
# I actually think the below code is responsible for misclassification of flu virus type (A or B) for samples that had both A and B infection!  
# npfirstpositiveupdate <- npfirstpositiveupdate %>%  
# mutate(AorB = ifelse(type.inf == "B", "B",  
# ifelse(type.inf == "B and unsubtypable A", "A and B",  
# ifelse(type.inf == "H3N2 and B", "A and B", "A")))) %>%  
# filter(AorB == type | AorB == "A and B") %>%  
# select(-AorB)  
  
saveRDS(npfirstpositiveupdate, "Curated Data/Cleaned Data/EMIT\_np\_quantity.RDS")  
  
# Seems like there should be 2 PCR results for each of these samples (replicates in the PCR assay) but most of these only have a single pcr copy number. Is this intentional?   
  
  
#### Working with the "EMIT\_UMD\_Final\_Data\_Summary.R" script ####  
# Title: EMIT\_UMD\_Final\_Data\_Summary.R  
# Moving this file, originally produced by Jing and Don Milton, to git lab  
# Also organizing it and cleaning it up some.   
  
## Original File information  
  
# Author: Jing Yan  
# Date: May 03, 2016  
# Revision Date: 2016  
# Title: final UMD data summary.R  
# Purpose: clean data and set files for tobit models  
# Input files:pcr data for gii and 2nd&3rd np.RDS  
# EMIT\_np\_quantity.RDS  
# EMIT\_subtypes\_enrolled\_positive.RDS  
# EMIT\_samples.cc.RDS  
# ???EMITClinicalUMD2013.csv  
# Output file: tables.txt  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_  
# Setup all I/O  
# setwd('C:/Users/Jing/Box Sync/Box Sync/EMIT/EMIT\_Data\_Analysis')  
# setwd('/Users/dmilton/Box Sync/0\_DKM/Lab/Biodefense/EMIT/EMIT\_Data\_Analysis')  
# setwd('/Volumes/Internal RAID Set 1/Box Sync/0\_DKM/Lab/Biodefense/EMIT/EMIT\_Data\_Analysis')  
# sink(file="R\_output/tables.txt",split=TRUE)  
# Out.dir <- "R\_output/"  
#\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
pcr1 <- readRDS("Curated Data/Cleaned Data/pcr data for gii and 2nd&3rd np.RDS")  
  
pcr1 <- pcr1 %>%   
 select(-Ct, -copies.in, -copy.num, -date, -cfactor, -virus.copies)  
  
nppcr1 <- readRDS("Curated Data/Cleaned Data/EMIT\_np\_quantity.RDS")  
  
nppcr1 <- nppcr1 %>%   
 select(-Ct, -copies.in, -copy.num, -date, -cfactor) %>%   
 rename(final.copies = virus.copies) # This seems to be incorrect. To get from virus.copies to final.copies one must multiply virus.copies by the dilution factor. For NP swabs the dilution factor was 100 (exept for a few NPS where 100ul instead of 50 ul was used to extract, in which case the dilution factor was 50).  
  
# Merge the preceding 2 dfs  
allpcr <- rbind(pcr1, nppcr1) %>%   
 ungroup  
  
# Check number of subjectIDs here  
allpcr\_subjectID\_check <- allpcr %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(allpcr\_subjectID\_check))

## [1] 158

# 114\_1, 127\_1, 335\_1 (These three subjects were enrolled on the second visit, these three samples are from their first screen visit)   
# I presume we are removing these because they don't have aerosol samples on their first day of sample collection, while they do have NPS data?  
# Subject 333 was removed, because the coarse sample was lost  
allpcr <- allpcr[!(allpcr$sample.id == "114\_1"| allpcr$sample.id == "127\_1" | allpcr$sample.id == "335\_1" | allpcr$sample.id == "333\_3" | allpcr$sample.id == "333\_1"), ]  
  
# Now that some data has been removed, again check number of subjectIDs here  
allpcr\_subjectID\_check <- allpcr %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(allpcr\_subjectID\_check))

## [1] 157

# I'm confused why the data for 114, 127, and 335 was simply removed for the first visit, but not the subsequent visits - wasn't the point of removing this data to eliminate the full pcr data for these subjects, because they don't have aerosol data on the first day of study where they do have NPS data? This is because they had data for dpo = 0 so that dpo = 0 visit was removed, while other data that was for dpo=1 or dpo=2 or dpo=3 was kept.  
  
## Read EMIT\_samples.cc.RDS  
all <- readRDS("Curated Data/Cleaned Data/EMIT\_samples.cc.RDS")  
  
subgroup <- all %>%   
 select(subject.id, sample.id, date.visit, date\_on\_sx)  
subgroup$date.visit <- as.Date(subgroup$date.visit)  
  
passagefocus <- all %>%   
 select(subject.id, sample.id, date.visit, passpos, validp, focus.ct)  
  
coughgiivisits <- all %>%   
 select(subject.id, sample.id, date.visit, cough\_number, sneeze\_number, sample.type, g2.run)  
# Does this cough\_number have all the data in it?  
# There seems to be quite a few NA's in there!  
# May need to find a cough\_number variable that has all of the data in it.  
# Note: that in earlier dfs that were used to create this object, there were missing cough data with notes that the audio recordings should be used to identify cough number for quite a few GII sampling instances.   
  
# Merge and clean  
pcrgiivisit <- allpcr %>%  
 inner\_join(coughgiivisits, by = c('subject.id', 'sample.id', 'sample.type'))  
pcrgiivisit$date.visit <- as.Date(pcrgiivisit$date.visit, "%m/%d/%Y")  
  
pcrgiivisit\_subjectID\_check <- pcrgiivisit %>%  
 group\_by(subject.id) %>%  
 count()  
  
## Read daypostonset.csv   
# (note: it is unclear how this file was created. It was found in Jing's R\_output folder so presumably it was created with some script - unfortunately a search of files yielded none that might have produced this file)  
daypick1 <- read.csv("UMD\_Raw\_Data/REDCAP/daypostonset.csv")  
  
# New plan (Januaray 29, 2019) regarding use of daypostonset data  
# It looks like the daypostonset.csv file that was in the raw data only has observations from 149 subject IDs and this is actually a major source of lost subject ID data between the 158 enrolled and positive cases and the 142 that made it into Jing's PNAS\_df.   
  
# Let's go back to the samples.cc df to recreate a new dayposonset df that has the data for all 178 screened individuals.  
  
samples.cc\_dpo <- samples.cc %>%  
 distinct(subject.id, date.visit, date\_on\_sx)  
samples.cc\_dpo$date.visit <- as.Date(samples.cc\_dpo$date.visit)  
samples.cc\_dpo$date\_on\_sx <- as.Date(samples.cc\_dpo$date\_on\_sx)  
samples.cc\_dpo <- samples.cc\_dpo %>%  
 mutate(dpo = date.visit - date\_on\_sx) %>%  
 arrange(subject.id)  
  
# How many subject IDs is this for?  
samples.cc\_dpo\_subjectID\_check <- samples.cc\_dpo %>%  
 group\_by(subject.id) %>%  
 count()  
  
# Need to add in the dpo where they are NA based on the date of symptom onset!  
  
sub <- unique(samples.cc\_dpo$subject.id)  
samples.cc\_dpo\_full <- samples.cc\_dpo %>%  
 filter(is.na(subject.id))  
  
for (i in 1:length(sub)) {  
 subid <- sub[i]  
 temp <- samples.cc\_dpo[samples.cc\_dpo$subject.id == subid, ]  
 for (j in 1:(nrow(temp))) {  
 temp$dpo[j] = temp$date.visit[j] - temp$date\_on\_sx[1]  
 samples.cc\_dpo\_full <- rbind(samples.cc\_dpo\_full, temp)  
 }  
}  
  
# This loop works ok, but there is a glitch with the NAs  
# To overcome this, we will filter out where dpo = NA and then take only the unique rows (to eliminate duplication that occurred in the loop)  
samples.cc\_dpo <- samples.cc\_dpo\_full %>%  
 filter(!is.na(dpo)) %>%  
 distinct(subject.id, dpo, .keep\_all = TRUE)  
  
# Note there were a few times where there was no date\_on\_sx and so dpo could not be determined.  
# This set of individuals was excluded from this version of samples.cc\_dpo  
  
# check the subjectIDs remaining in samples.cc\_dpo  
samples.cc\_dpo\_subjectID\_check <- samples.cc\_dpo %>%  
 group\_by(subject.id) %>%  
 count()  
# There are 331 subjectIDs here - this means that we lost 24 that didn't have a date\_on\_sx recorded  
# Hopefully these 24 subjectIDs are part of the unenrolled group, however we will be able to check this.  
  
# Make the samples.cc\_dpo df take on the daypick1 df  
daypick1 <- samples.cc\_dpo  
  
## old version  
# daypick <- daypick1 %>%  
# distinct(subject.id, date.visit, dpo, .keep\_all = TRUE) %>%  
# arrange(subject.id, dpo)  
#   
# daypickfinal <- daypick %>%   
# filter(!(dpo == 0 | dpo == 4 |dpo == 7))  
# daypickfinal$date.visit <- as.Date(daypickfinal$date.visit, format = "%m/%d/%Y")  
# daypickfinal$date\_on\_sx <- as.Date(daypickfinal$date\_on\_sx, format = "%m/%d/%Y")  
## end old version  
  
daypickfinal\_full <- daypick1  
daypickfinal\_full$dpo <- as.numeric(daypickfinal\_full$dpo)  
daypickfinal\_full <- daypickfinal\_full %>%  
 arrange(subject.id, dpo)  
  
# SubjectID count check for daypickfinal\_full  
daypickfinal\_full\_subjectID\_check <- daypickfinal\_full %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(daypickfinal\_full\_subjectID\_check))

## [1] 331

# 331 subjectIDs  
# But how many sampling instances?  
daypickfinal\_full\_sampling\_instance\_check <- daypickfinal\_full %>%  
 distinct(subject.id, date.visit)  
print(nrow(daypickfinal\_full\_sampling\_instance\_check))

## [1] 440

# 440 sampling instances when we use all of the daypickfinal data (inclusive of dpo day)  
  
# But when we restrict the dpo day to just dpo == 1, 2, or 3, then how many sampling instances do we get?  
  
# Jing's final PNAS dataset requires this daypickfinal to only include the dpo=1-3, however we left the daypickfinal\_full version above to help recreate and understand all the places we whittled down subjectIDs and/or sampling instances  
daypickfinal\_dpo1\_through\_3 <- daypick1 %>%   
 filter(dpo == 1 | dpo == 2 | dpo == 3)  
daypickfinal\_dpo1\_through\_3$dpo <- as.numeric(daypickfinal\_dpo1\_through\_3$dpo)  
daypickfinal\_dpo1\_through\_3 <- daypickfinal\_dpo1\_through\_3 %>%  
 arrange(subject.id, dpo)  
  
# SubjectID count check for daypickfinal\_dpo1\_through\_3 (before merge)  
daypickfinal\_dpo1\_through\_3\_subjectID\_check <- daypickfinal\_dpo1\_through\_3 %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(daypickfinal\_dpo1\_through\_3\_subjectID\_check))

## [1] 309

# This new daypickfinal\_dpo1\_through\_3 df has 309 subjectIDs (the filter command to select only those with dpo == 1, 2, or 3 eliminated 22 subjectID entries from daypickfinal\_full)  
# But how many sampling instances do we get now with the daypickfinal\_dpo1\_through\_3 df?  
daypickfinal\_dpo1\_through\_3\_sampling\_instance\_check <- daypickfinal\_dpo1\_through\_3 %>%  
 distinct(subject.id, date.visit)  
print(nrow(daypickfinal\_dpo1\_through\_3\_sampling\_instance\_check))

## [1] 399

# we get 399 sampling instances when restricting the dpo df to dpo=1-3  
  
# Merge and clean  
withdpo <- pcrgiivisit %>%  
 inner\_join(daypickfinal\_dpo1\_through\_3, by = c('subject.id', 'date.visit'))  
  
# Subject ID count check for withdpo  
withdpo\_subjectID\_check <- withdpo %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(withdpo\_subjectID\_check))

## [1] 148

# This merge reduces the number of subjectIDs from 157 in the pcrgiivisit df to 148 in the withdpo df (the daypickfinal\_dpo1\_through\_3 df had 309)  
# Let's examine to see how we lost these 9 subjectIDs  
# Any subjectIDs in pcrgiivisit that weren't in daypickfinal\_dpo1\_through\_3?  
pcrgiivisit\_but\_not\_in\_daypickfinal\_dpo1\_through\_3 <- pcrgiivisit %>%  
 anti\_join(daypickfinal\_dpo1\_through\_3, by = c('subject.id')) %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(pcrgiivisit\_but\_not\_in\_daypickfinal\_dpo1\_through\_3))

## [1] 8

# In checking the dpo df before excluding for dpo == less than 1 or greater than 3, we see that:  
# 122 only had a dpo 4 so was excluded  
# 166 only had a dpo 4 so was excluded  
# 249 only had a dpo 7 so was excluded  
# 299 only had a dpo 4 so was excluded  
# 302 only had a dpo 4 so was excluded  
# 318 only had a dpo 4 so was excluded  
# 327 only had a dpo 0 so was excluded  
# 350 only had a dpo 0 so was excluded  
  
# But what about the 9th missing subjectID?  
# Let's compare the merged df with the pcrgiivisit df to see who is missing  
pcrgiivisit\_but\_not\_in\_withdpo <- pcrgiivisit %>%  
 anti\_join(withdpo, by = "subject.id") %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(pcrgiivisit\_but\_not\_in\_withdpo))

## [1] 9

ninth\_missing\_subjectID <- pcrgiivisit\_but\_not\_in\_withdpo %>%  
 anti\_join(pcrgiivisit\_but\_not\_in\_daypickfinal\_dpo1\_through\_3)

## Joining, by = c("subject.id", "n")

# This shows that 114 was the other missing subjectID  
# Why was 114 missing? Was it missing from the daypickfinal\_dpo1\_through\_3 df?  
  
record\_for\_114 <- pcrgiivisit %>%  
 filter(subject.id == 114)  
# We see that 114 had positive pcr detections on dpo 4 but there were no pcr records for 114 from the only other day of sample collection, which was on dpo 1. There shoud be more pcr records for 114 from dpo 1. Even if these were all negative, this case could be included in the final PNAS\_df just like 174 was! Otherwise there is inconsistency in the analytical inclusion criteria!  
  
withdpo <- withdpo %>%   
 filter(!(is.na(cough\_number)))  
# But does this cough\_number include all the data from the recordings? We assume so.  
  
# Subject ID count check for withdpo  
withdpo\_subjectID\_check <- withdpo %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(withdpo\_subjectID\_check))

## [1] 146

# Thus - there were 2 records among the 148 that didn't have cough data so those were removed and now we are down to 146.  
  
## Jing says:  
# Remove experiment 9. 2015.06.19.014.74 2012-2013 Samples PCR Flu B   
# Then we removed subjects 322 and 337 and remove 182 second visit  
## End of Jing comment  
  
# But, what is the explanation for this? We solved the case - see note below - but it was due to bad interrun calibrator in the pcr assay.  
  
withdpo\_322\_record <- withdpo %>%  
 filter(subject.id == 322)  
# Only thing out of place is that the first sample was taken on dpo=0, but this shouldn't be reason to exclude the dpo=1 samples, remaining here, from inclusion!  
# I haven't seen any reason to support exclusion of this case.   
  
withdpo\_337\_record <- withdpo %>%  
 filter(subject.id == 337)  
# I haven't seen any reason to support exclusion of this case.   
# Is it because there is only a dpo=3 record here?  
  
withdpo\_182\_record <- withdpo %>%  
 filter(subject.id == 182)  
# I haven't seen any reason to support exclusion of dpo = 2 data for this subject.   
# Is it because there is no dpo = 1 data here? If so, then why keep the dpo=3 data?  
  
withdpo <- withdpo[!(withdpo$subject.id == 322 | withdpo$subject.id == 337), ]  
withdpo <- withdpo[!(withdpo$subject.id == 182 & withdpo$g2.run == 2), ]  
  
# Subject ID count check for withdpo  
withdpo\_subjectID\_check <- withdpo %>%  
 group\_by(subject.id) %>%  
 count()  
  
# After check some more notes, we see that 322, 337 and the second day of pcr data for 182 were excluded because the interrun-calibrator in the pcr assay was not working properly.   
  
clinical\_in\_file <- 'UMD\_Raw\_Data/REDCAP/EMITClinicalUMD2013.csv'  
clinical\_umd <- read.csv(clinical\_in\_file)  
  
# Check out the format of the date\_visit variable  
clinical\_umd\_date <- clinical\_umd %>%  
 select(date\_visit)  
  
clinical\_umd$date\_visit <- as.Date(clinical\_umd$date\_visit, format = "%m/%d/%y")  
  
clinical\_umd1 <- clinical\_umd %>%   
 select(field\_subj\_id, body\_temp, date\_visit, nose\_run, nose\_stuf, sneeze, throat\_sr, earache, malaise, headache, mj\_ache, sw\_fever\_chill, lymph\_node, chest\_tight, sob, cough, fluvac\_cur, sex, asthma, anitviral\_24h, smoke) %>%  
 rename(subject.id = field\_subj\_id, date.visit = date\_visit)  
  
clinical\_umd\_1 <- clinical\_umd1 %>%  
 select(subject.id, body\_temp, date.visit) %>%  
 filter(!is.na(body\_temp))  
  
# Subject ID count check for "clinical\_umd\_1" df  
clinical\_umd\_1\_subjectID\_check <- clinical\_umd\_1 %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(clinical\_umd\_1\_subjectID\_check))

## [1] 355

clinical\_umd\_2 <- clinical\_umd1 %>%   
 select(subject.id, date.visit, nose\_run, nose\_stuf, sneeze, throat\_sr, earache, malaise, headache, mj\_ache, sw\_fever\_chill, lymph\_node, chest\_tight, sob, cough) %>%   
 filter(!is.na(nose\_run))  
  
# Subject ID count check for "clinical\_umd\_2" df  
clinical\_umd\_2\_subjectID\_check <- clinical\_umd\_2 %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(clinical\_umd\_2\_subjectID\_check))

## [1] 355

clinical\_umd\_3 <- clinical\_umd1 %>%   
 select(subject.id, asthma) %>%   
 filter(!is.na(asthma))  
# Save this object for later merge  
  
# Subject ID count check for "clinical\_umd\_3" df  
clinical\_umd\_3\_subjectID\_check <- clinical\_umd\_3 %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(clinical\_umd\_3\_subjectID\_check))

## [1] 178

# Read in new df  
subtype <- readRDS("Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled\_positive.RDS")  
  
# Merge and clean  
comfirmcases <- clinical\_umd\_2 %>%  
 inner\_join(subtype, by = 'subject.id')  
# Subject ID count check  
comfirmcases\_subjectID\_check <- comfirmcases %>%  
 group\_by(subject.id) %>%  
 count()  
  
comfirmcases1 <- comfirmcases %>%   
 mutate(upper\_sym = nose\_run + nose\_stuf + sneeze + throat\_sr + earache) %>%   
 mutate(lower\_sym = chest\_tight + sob + cough) %>%   
 mutate(systemic\_sym = malaise + headache + mj\_ache + lymph\_node + sw\_fever\_chill)  
  
# Merge  
comfirmcases2 <- comfirmcases1 %>%  
 inner\_join(subgroup, by = c('subject.id', 'date.visit'))  
# Subject ID count check  
comfirmcases2\_subjectID\_check <- comfirmcases2 %>%  
 group\_by(subject.id) %>%  
 count()  
  
## What if we merge withdpo with comfirmcases2 directly, instead of involving the allpcr df? To do this, we will comment out the below merge that merges the allpcr and comfirmcases2 dfs.  
  
# pcr\_body\_temp\_symptoms <- allpcr %>%  
# inner\_join(comfirmcases2, by = c('subject.id', 'sample.id', 'type.inf')) %>%  
# select(-date\_on\_sx)  
# pcr\_body\_temp\_symptoms$date\_on\_sx <- as.Date(pcr\_body\_temp\_symptoms$date\_on\_sx)  
  
finaldata <- withdpo %>%  
 inner\_join(comfirmcases2, by = c("sample.id", "subject.id", "type.inf", "date.visit")) %>%  
 select(-date\_on\_sx.y) %>%  
 mutate(date\_on\_sx = date\_on\_sx.x) %>%  
 select(-date\_on\_sx.x)  
  
# Subject ID count check on the finaldata df  
finaldata\_subjectID\_check <- finaldata %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(finaldata\_subjectID\_check))

## [1] 144

# finaldata <- inner\_join(withdpo, pcr\_body\_temp\_symptoms, by = c("subject.id", "date.visit", "type", "type.inf", "sample.type", "final.copies", "Experiment"))  
  
# Add body temp variable with the clinical\_umd\_1 df  
finaldata2 <- finaldata %>%  
 left\_join(clinical\_umd\_1, by = c('date.visit', 'subject.id'))  
  
# Add asthma variable with the clinical\_umd\_3 df  
finaldata3 <- finaldata2 %>%  
 inner\_join(clinical\_umd\_3, by = 'subject.id')  
  
enrolled <- readRDS("Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled.RDS")  
enrolledcase <- enrolled %>%   
 select(subject.id)  
# So far this 'enrolled' object doesn't is not incoporated at all.  
# It shows the list of 178 enrolled study participants  
  
#### G2 LOG DATA ####  
g2\_in\_file <- 'UMD\_Raw\_Data/GII/EMITGIILogUMD2013.csv'  
g2\_log <- read.csv(g2\_in\_file)  
  
##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Date Entry Error Correction  
print(select(filter(g2\_log, subject\_id == 284), subject\_id, redcap\_event\_name, start\_dt))

## subject\_id redcap\_event\_name start\_dt  
## 1 284 baseline\_and\_colle\_arm\_1 2013-02-16  
## 2 284 collection\_2\_arm\_1 2013-03-17  
## 3 284 collection\_3\_arm\_1 2013-02-18

# Subject\_id 284 g2 collection\_2\_arm\_1 was entered as 2013-03-17 but baseline was on 2013-02-16 and collection\_3 was 2013-02-18."  
# Therefore recode collection\_2 date to February from March (i.e. to 2013-02-17).  
  
g2\_log$start\_dt[which(g2\_log$subject\_id == 284 & g2\_log$start\_dt == '2013-03-17', arr.ind = TRUE)] <- '2013-02-17'  
g2\_log <- g2\_log %>%   
 filter(!(subject\_id == 81 & redcap\_event\_name == 'collection\_2\_arm\_1'))  
  
g2\_log1 <- g2\_log %>%   
 select(subject\_id, start\_dt, g2\_unit, chiller\_t1, chiller\_t2, chiller\_t3, elbow\_rh1, elbow\_rh2, elbow\_rh3, elbow\_t1, elbow\_t2, elbow\_t3, cond\_tin1, cond\_tin2, cond\_tin3, cond\_tout1, cond\_tout2, cond\_tout3)  
  
g2\_log1$chiller.t <- (g2\_log1$chiller\_t1 + g2\_log1$chiller\_t2 + g2\_log1$chiller\_t3) / 3  
g2\_log1$elbow\_rh <- (g2\_log1$elbow\_rh1 + g2\_log1$elbow\_rh2 + g2\_log1$elbow\_rh3) / 3  
g2\_log1$elbow\_t <- (g2\_log1$elbow\_t1 + g2\_log1$elbow\_t2 + g2\_log1$elbow\_t3) / 3  
g2\_log1$cond\_tin <- (g2\_log1$cond\_tin1 + g2\_log1$cond\_tin2 + g2\_log1$cond\_tin3) / 3  
g2\_log1$cond\_tout <- (g2\_log1$cond\_tout1 + g2\_log1$cond\_tout2 + g2\_log1$cond\_tout3) / 3  
  
g2\_log1 <- g2\_log1 %>%   
 select(subject\_id, start\_dt, g2\_unit,chiller.t, elbow\_rh,elbow\_t, cond\_tin,cond\_tout) %>%  
 rename(subject.id = subject\_id, date.visit = start\_dt)  
g2\_log1$date.visit <- as.Date(g2\_log1$date.visit)  
  
# Subject ID count check for g1\_log1  
g2\_log1\_subject\_ID\_check <- g2\_log1 %>%  
 group\_by(subject.id) %>%  
 count()  
  
# Merge and clean  
# Add the g2 log data variables with the g2\_log1 df  
finaldata4 <- finaldata3 %>%  
 inner\_join(g2\_log1, by = c('subject.id', 'date.visit'))  
  
# The below line of code seems to be incorrect and causes issues downstream.  
# This is because this line of script only takes into consideration H3N2, Unsubtypable A, and Pandemic H1, but it doesn't take into consideration the instances where there are 2 types! (like with subject)  
# finaldata4$typeAB <- ifelse(finaldata3$type.inf =='H3N2' | finaldata3$type.inf == 'Unsubtypable A' | finaldata3$type.inf == 'Pandemic H1', "A", "B")  
# finaldata4$typeAB[finaldata3$subject.id == 55] <- 'A'  
# finaldata4$typeAB[finaldata3$subject.id == 230] <- 'A'  
  
# SubjectID count check for finaldata4  
finaldata4\_subjectID\_check <- finaldata4 %>%  
 group\_by(subject.id) %>%  
 count()  
  
VSAS <- read.csv("UMD\_Raw\_Data/REDCAP/vacine\_smoker\_antiviral\_sex.csv")  
  
# Add sex, fluvac\_cur, antiviral\_24h, Smoker variables from the VSAS df   
finaldata142 <- finaldata4 %>%  
 inner\_join(VSAS, by = 'subject.id')  
  
# SubjectID count check for finaldata142  
finaldata142\_subjectID\_check <- finaldata142 %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(finaldata142\_subjectID\_check))

## [1] 144

saveRDS(finaldata142, "Curated Data/Cleaned Data/finaldata142.RDS")   
  
totalsamples <- finaldata142 %>%   
 distinct(subject.id, sample.id, sample.type, .keep\_all = TRUE)  
totalsamples\_subjectID\_check <- totalsamples %>%  
 group\_by(subject.id) %>%  
 count()  
  
totalsubjects <- finaldata142 %>%   
 distinct(subject.id, .keep\_all = TRUE)  
  
sex <- finaldata142 %>%   
 distinct(subject.id, sex, .keep\_all = TRUE) %>%   
 filter(sex == 1)  
  
flushot <- finaldata142 %>%   
 distinct(subject.id, fluvac\_cur, .keep\_all = TRUE) %>%   
 filter(fluvac\_cur == 1)  
  
asthma <- finaldata142 %>%   
 distinct(subject.id, asthma, .keep\_all = TRUE) %>%   
 filter(asthma == 1)  
  
smoker <- finaldata142 %>%   
 distinct(subject.id, Smoker, .keep\_all = TRUE) %>%   
 filter(Smoker == 1)  
  
antiviral <- finaldata142 %>%   
 distinct(subject.id, anitviral\_24h, .keep\_all = TRUE) %>%  
 filter(anitviral\_24h == 1)  
  
npsubgroup <- finaldata142 %>%   
 filter(sample.type == 'Nasopharyngeal swab') %>%   
 select(Experiment, subject.id, sample.id, type, type.inf, sample.type, final.copies, date.visit, cough\_number, sneeze\_number, g2.run, dpo, upper\_sym, lower\_sym, systemic\_sym, body\_temp, asthma, g2\_unit, chiller.t, elbow\_rh, elbow\_t, cond\_tin, cond\_tout, sex, fluvac\_cur, anitviral\_24h, Smoker) %>%  
 arrange(subject.id, date.visit)  
  
npsubgroup\_subjectID\_check <- npsubgroup %>%  
 group\_by(subject.id) %>%  
 count()  
  
count1 <- npsubgroup %>%   
 distinct(subject.id, .keep\_all = TRUE)  
  
count2 <- npsubgroup %>%   
 group\_by(sample.id) %>%   
 summarise(n = n())  
  
count3 <- count2 %>%   
 filter(n > 2)  
  
finesubgroup <- finaldata142 %>%   
 filter(sample.type == 'GII condensate NO mask')  
  
symptom <- finesubgroup %>%   
 select(subject.id, sample.id, dpo, systemic\_sym, upper\_sym, lower\_sym, nose\_run, nose\_stuf, sneeze, throat\_sr, earache, chest\_tight, sob, cough, malaise, headache, mj\_ache, sw\_fever\_chill, lymph\_node) %>%   
 distinct(subject.id, sample.id, dpo, systemic\_sym, upper\_sym, lower\_sym, nose\_run, nose\_stuf, sneeze, throat\_sr, earache, chest\_tight, sob, cough, malaise, headache, mj\_ache, sw\_fever\_chill, lymph\_node)  
  
# write.csv(symptom, "C:/Users/Jing/Desktop/symptomscoreupdate.csv")   
  
finesubgroup <- finesubgroup %>%   
 select(Experiment, subject.id, sample.id, type, type.inf, sample.type, final.copies, date.visit, cough\_number, sneeze\_number, g2.run, dpo, upper\_sym, lower\_sym, systemic\_sym, body\_temp, asthma, g2\_unit, chiller.t, elbow\_rh, elbow\_t, cond\_tin, cond\_tout, sex, fluvac\_cur, anitviral\_24h, Smoker)  
  
count4 <- finesubgroup %>%   
 distinct(subject.id)  
count5 <- finesubgroup %>%   
 group\_by(sample.id) %>%   
 summarise(n = n())  
count6 <- count5 %>%   
 filter(n > 2)  
  
coarsesubgroup <- finaldata142 %>%   
 filter(sample.type == 'Impactor 5 um NO mask') %>%   
 select(Experiment, subject.id, sample.id, type, type.inf, sample.type, final.copies, date.visit, cough\_number, sneeze\_number, g2.run, dpo, upper\_sym, lower\_sym, systemic\_sym, body\_temp, asthma, g2\_unit, chiller.t, elbow\_rh, elbow\_t, cond\_tin, cond\_tout, sex, fluvac\_cur, anitviral\_24h, Smoker)  
  
count7 <- coarsesubgroup %>%   
 distinct(subject.id)  
count8 <- coarsesubgroup %>%   
 group\_by(sample.id) %>%   
 summarise(n = n())  
count9 <- count8 %>%   
 filter(n > 2)  
# Repeated format  
  
finesubgroup <- finesubgroup %>%   
 mutate(sampleid = gsub('^[0-9]\*\_', '', sample.id))  
finesubgroup$NPswab <- 0  
finesubgroup$Coarse <- 0  
finesubgroup$Fine <- 1  
  
coarsesubgroup <- coarsesubgroup %>%   
 mutate(sampleid = gsub('^[0-9]\*\_', '', sample.id))  
coarsesubgroup$NPswab <- 0  
coarsesubgroup$Coarse <- 1  
coarsesubgroup$Fine <- 0  
  
npsubgroup<-npsubgroup %>%   
 mutate(sampleid = gsub('^[0-9]\*\_', '', sample.id))  
npsubgroup$NPswab <- 1  
npsubgroup$Coarse <- 0  
npsubgroup$Fine <- 0  
  
finaldataset <- rbind(finesubgroup, npsubgroup, coarsesubgroup) %>%   
 ungroup   
finaldataset$final.copies <- as.numeric(finaldataset$final.copies)  
finaldataset <- finaldataset %>%  
 arrange(subject.id, date.visit, type, sample.type, final.copies)  
  
# finaldataset <- finaldataset[order(finaldataset$subject.id,finaldataset$dpo), ]  
# finaldataset$final.copies[is.na(finaldataset$final.copies)] <- '.'  
  
# Need to add some manipulation to the df right here to replicate what Jing produced for final input to the SAS program  
  
# Assess the df  
finaldataset\_count <- finaldataset %>%  
 arrange(sample.type, subject.id, date.visit, Experiment) %>%  
 group\_by(sample.type, subject.id, date.visit, Experiment, type.inf) %>%  
 summarise(count = n())  
print(nrow(finaldataset\_count))

## [1] 697

finaldataset\_subjectID\_check <- finaldataset %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(finaldataset\_subjectID\_check))

## [1] 144

finaldataset\_sampleID\_check <- finaldataset %>%  
 group\_by(sample.id) %>%  
 count()  
print(nrow(finaldataset\_sampleID\_check))

## [1] 668

# I figured out why there are only single PCR replicate results (singlicate as opposed to duplicate) - it’s because these were part of the subtyping assays and there was only enough material to run the CDC panel in singlicate (or a second extraction would have been required). For this reason, the 1st day visit NP swabs were run in singlicate (many of them). Others got put on par assays later on and have duplicates. This explains the lack of consistency in the number of pcr results reported for first visit NP swabs here. We will move forward with the dataframe that we have generated.   
  
# We also note that we have kept all of the data where there are multiple assays on the same sample. Tobit models will help us interpret these data, especially when there were 2 assays and one was positive while the other was negative (or there were a combination of replicates that were positive/negative)  
  
# This df is missing the following variables (compared with Jing's output):  
# centimeterheight  
# kilogramweight  
# BMI  
# cur\_asthma (how is this different than the asthma variable that is already in the df?)  
# lung\_sym\_2pos (not sure what this is or how to compute?)  
# fluvac\_last2y  
# bothyear  
# age  
  
# Not sure what lung\_symp\_2pos means, but the other variables can be taken directly from the clinical\_umd df or derived from variables in the clinical\_umd df.   
  
clinical\_umd\_height\_weight\_BMI\_vax\_age <- clinical\_umd %>%  
 filter(!is.na(age)) %>%  
 mutate(height\_inches = height\_in + 12\*(height\_ft)) %>%  
 mutate(height\_cm = 2.54\*(height\_inches)) %>%  
 mutate(weight\_kg = 0.453592\*weight) %>%  
 mutate(BMI = weight\_kg/((height\_cm/100)^2)) %>%  
 mutate(vax\_bothyear = ifelse(fluvac\_cur == 1 & fluvac\_last2y == 1, 1, 0)) %>%  
 rename(subject.id = field\_subj\_id, date.visit = date\_visit) %>%  
 select(subject.id, date.visit, height\_cm, weight\_kg, BMI, fluvac\_last2y, vax\_bothyear, fluvac\_10y, age)  
  
missing\_fluvac\_last2y <- clinical\_umd\_height\_weight\_BMI\_vax\_age %>%  
 filter(is.na(fluvac\_last2y))  
print(nrow(missing\_fluvac\_last2y))

## [1] 121

finaldataset <- finaldataset %>%  
 left\_join(clinical\_umd\_height\_weight\_BMI\_vax\_age, by = c("subject.id")) %>%  
 select(-date.visit.y) %>%  
 rename(date.visit = date.visit.x) %>%  
 filter(subject.id != 52) %>% # It was decided by the lab that subject 52 was a false positives and should be removed  
 filter(subject.id != 58) %>% # It was decided by the lab that subject 58 was a false positives and should be removed  
 arrange(subject.id, date.visit, type, sample.type, Experiment, final.copies)  
  
finaldataset\_subjectID\_check <- finaldataset %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(finaldataset\_subjectID\_check))

## [1] 142

write.csv(finaldataset, "Curated Data/Analytical Datasets/finaldatasetrepeatupdate.csv")  
  
finaldataset\_missing\_fluvac\_last2y <- finaldataset %>%  
 filter(is.na(fluvac\_last2y)) %>%  
 distinct(subject.id)  
print(nrow(finaldataset\_missing\_fluvac\_last2y))

## [1] 55

# Looks like 55 out of 142 subjects are missing the fluvac\_last2y variable!  
# There simply isn't data on this in the raw data. Is there a different raw datafile that should be used?  
# Somehow in the original data that was used in the PNAS analysis, we have data for this variable on all 142 subjects.  
# It looks like all of these 55 subject IDs are marked in the PNAS final dataset as having a 0 for the fluvac\_last2y as opposed to an NA. I'm not sure if this is correct.   
  
#### Creating the "all\_data" df that has all the data from all of the screened and enrolled participants ####  
  
# Need to merge together the pcr data into a definitive set called allPCRfinal  
npfirst1 <- npfirst %>%  
 select(subject.id, sample.id, type, Experiment, Ct, cfactor, virus.copies) %>%  
 mutate(final.copies = virus.copies) %>%  
 ungroup()  
# The difference between copies.in and copy.num in the npfirst object is a factor of 8,000 for flu A assays and 41,100 for flu B assays.  
# If we break down these factors of 8,000 (for flu A) and 41,100 (for flu B) we see that it is probably a combination of what Jing used for the conversion factor from virus particles and RNA copies (80 for flu A and 411 for flu B although it should be 250 for flu A and 272 for flu B), and the dilution factor (100 for NP swabs, except for a few where 100ul instead of 50ul was used to extract)  
# So the virus.copies variable is actually the finalized pcr variable and all we needed to do in the above step was rename virus.copies as final.copies.  
# However, this whole process is under review because the conversion factors between virus particles and RNA copies are not what we believe that they should be.   
  
total.pcr1 <- total.pcr %>%  
 select(subject.id, Well.Name, type, Experiment, Ct..dRn., cfactor, virus.copies) %>%  
 rename(sample.id = Well.Name)  
# Unlike the first visit NP swab data discussed above in the npfirst object, the total.pcr object has undergone a different data manipulation process with respect to applying calibration factors.  
# For total.pcr, the copies.in variable was multiplied by the RNA copies to virus particle conversion factors of 80 (for flu A) and 411 (for flu B) in order to get out copy.num (note - we are reviewing this because we believe that the conversion factors should really be 250 and 272 for flu A and B, respectively). Then, copy.num was multiplied by the pcr assay calibration factor to get virus.copies. Unlike in the NP swab first visit data where the dilution factor was applied as part of the process of getting from copies.in to copy.num, with the aerosols and post first visit NP swabs, the dilution factor has not yet been applied and, thus, the following code is written to get from virus.copies to a final.copies variable that has the dilution factor applied to it.   
  
## Need the below code (copied from above) to manipulate the pcr data from the total.pcr object in order to apply the dilution factors for the aerosol data and NP swabs that were run on visits 2 or 3 (not first visit NPS).  
  
samples.cc\_type <- samples.cc %>%  
 select(sample.id, sample.type)  
  
total.pcr1 <- total.pcr1 %>%  
 left\_join(enrolled, by = "subject.id") %>%  
 left\_join(samples.cc\_type, by = "sample.id") %>%  
 ungroup()  
  
# Pick out all the PCR with A assay results  
allpcrA <- total.pcr1 %>%   
 filter(type == 'A') %>%  
 rename(virus.copiesA = virus.copies, typeA = type) %>%  
 rename(CtA = Ct..dRn.)  
  
# Pick out all the PCR with B assay results  
allpcrB <- total.pcr1 %>%   
 filter(type == 'B') %>%  
 rename(virus.copiesB = virus.copies, typeB = type) %>%  
 rename(CtB = Ct..dRn.)  
  
# Join both A and B assay, and also add sample type in the data list, assign the final RNA copies number for each sample type  
allPCR <- allpcrA %>%  
 full\_join(allpcrB, by = c('subject.id', 'Experiment', 'type.inf', 'sample.id', 'sample.type', 'cfactor')) %>%  
 arrange(subject.id) %>%   
 filter(!sample.type == 'Throat Swab')  
  
## DATA EDITING (dilution factors different for a few samples) ##  
# Flagged samples all NP swabs, dilution factor is 50  
# 66\_7 120\_7 184\_8 188\_7 189\_7 192\_7 196\_7 262\_7 277\_7 284\_12 284\_7 296\_12 296\_7  
# Note that none of these samples were first visit NPS samples, so this data edit applied to only the data from non-first visit NPS.   
  
allPCR1 <- allPCR %>%   
 filter(sample.id == '66\_7' | sample.id == '120\_7' | sample.id == '184\_8' | sample.id == '188\_7' | sample.id == '189\_7' | sample.id == '192\_7' |   
 sample.id == '196\_7' | sample.id == '262\_7' | sample.id == '277\_7' | sample.id == '284\_12' | sample.id == '284\_7' |   
 sample.id == '296\_12'| sample.id=='296\_7')  
  
allPCR2 <- allPCR %>%  
 anti\_join(allPCR1)

## Joining, by = c("subject.id", "sample.id", "typeA", "Experiment", "CtA", "cfactor", "virus.copiesA", "type.inf", "sample.type", "typeB", "CtB", "virus.copiesB")

allPCR3 <- allPCR1 %>%   
 mutate(final.copiesA = virus.copiesA\*50, final.copiesB = virus.copiesB\*50)  
  
allPCR4 <- allPCR2 %>%   
 filter(!sample.type == 'Nasopharyngeal swab') %>%   
 mutate(final.copiesA = virus.copiesA\*25, final.copiesB = virus.copiesB\*25)  
  
allPCR5 <- allPCR2 %>%   
 filter(sample.type == 'Nasopharyngeal swab') %>%  
 mutate(final.copiesA = virus.copiesA\*100, final.copiesB = virus.copiesB\*100)  
  
allPCRtotal <- rbind(allPCR3, allPCR4, allPCR5) %>%   
 ungroup()  
  
allPCR.A <- allPCRtotal %>%   
 filter(typeA == 'A') %>%  
 select(-CtB, -virus.copiesB, -typeB, -final.copiesB) %>%  
 rename(Ct = CtA, virus.copies = virus.copiesA, type = typeA, final.copies = final.copiesA)  
  
allPCR.B <- allPCRtotal %>%   
 filter(typeB == 'B') %>%  
 select(-CtA, -virus.copiesA, -typeA, -final.copiesA) %>%  
 rename(Ct = CtB, virus.copies = virus.copiesB, type = typeB, final.copies = final.copiesB)  
  
# merge the seperated FILE FOR A and B back together, successfully make - one row for one subject.id  
allPCRfinal <- rbind(allPCR.A, allPCR.B) %>%   
 ungroup() %>%  
 select(-type.inf, -sample.type)  
  
allPCRfinal\_subjectID\_check <- allPCRfinal %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(allPCRfinal\_subjectID\_check))

## [1] 202

# 202 subject IDs have some sort of pcr data for aerosols or post day 1 NPS  
  
## Now, bind the allPCRfinal and npfirst1 objects together  
  
pcr\_full <- rbind(allPCRfinal, npfirst1)  
  
pcr\_full\_subjectID\_check <- pcr\_full %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(pcr\_full\_subjectID\_check))

## [1] 207

# 207 subjects  
  
pcr\_full\_merge <- pcr\_full %>%  
 select(subject.id, sample.id, type, Experiment, Ct, final.copies)  
  
## Now, get the samples.cc variable ready to merge.  
  
samples.cc\_date\_on\_sx\_subjectID\_check <- samples.cc %>%  
 select(subject.id, sample.id, date.visit, date\_on\_sx) %>%  
 filter(!is.na(date\_on\_sx)) %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(samples.cc\_date\_on\_sx\_subjectID\_check))

## [1] 331

# We see that the samples.cc df has date\_on\_sx for 331 subjects (same as for the daypickfinal\_full df).   
  
# Now let's check how many sampling instances the samples.cc has  
samples.cc\_sampling\_instances\_check <- samples.cc %>%  
 distinct(subject.id, date.visit)  
print(nrow(samples.cc\_sampling\_instances\_check))

## [1] 473

# 473 sampling instances. But what about if we exclude to only those sampling instances that are associated with g2 visits?  
samples.cc\_gii\_sampling\_instances\_check <- samples.cc %>%  
 filter(g2.run != 0) %>%  
 distinct(subject.id, date.visit)  
print(nrow(samples.cc\_gii\_sampling\_instances\_check))

## [1] 276

# There are 276 sampling instances.   
  
## Now get the enrolled df ready to merge  
enrolled\_type.inf <- enrolled  
print(nrow(enrolled\_type.inf))

## [1] 178

## Now get the daypickfinal\_full df ready to merge  
daypickfinal\_dpo <- daypickfinal\_full %>%  
 select(subject.id, date.visit, dpo)  
print(nrow(daypickfinal\_dpo))

## [1] 440

# Note that the date\_on\_sx variable in the samples.cc df is more comprehensive than the one from daypickfinal\_dpo because it includes data on more subjectIDs.  
# samples.cc contains 473 sampling instances, while daypickfinal\_full contains 440 sampling instances.  
  
## Now get the clinical\_umd\_body\_temp df ready to merge  
clinical\_umd\_body\_temp <- clinical\_umd\_1   
print(nrow(clinical\_umd\_body\_temp))

## [1] 474

# Oddly enough, the clinical\_umd\_1 df is actually more comprehensive than the body\_temp variable from the samples.cc df (this could be due to the way merges were done in the original script where samples.cc was created - this script is embedded in the current script now)  
  
# Now get the clinical\_umd\_symptoms df ready to merge  
clinical\_umd\_symptoms <- clinical\_umd\_2  
print(nrow(clinical\_umd\_symptoms))

## [1] 475

# Now get the clinical\_umd\_asthma df ready to merge  
clinical\_umd\_asthma <- clinical\_umd\_3  
print(nrow(clinical\_umd\_asthma))

## [1] 178

# Now get the g2 log ready to merge  
g2\_log1 # Already in good form

## subject.id date.visit g2\_unit chiller.t elbow\_rh elbow\_t cond\_tin  
## 1 7 2012-12-04 0 28.00000 77.03333 84.00000 30.00000  
## 2 8 2012-12-04 0 28.00000 79.63333 85.23333 30.00000  
## 3 13 2012-12-10 0 28.00000 77.96667 84.36667 31.33333  
## 4 13 2012-12-11 0 30.73333 71.70000 85.76667 31.00000  
## 5 13 2012-12-12 0 29.26667 78.03333 83.83333 30.00000  
## 6 17 2012-12-11 0 32.06667 74.46667 87.06667 32.66667  
## 7 19 2012-12-11 0 33.26667 86.80000 87.33333 34.33333  
## 8 20 2012-12-12 0 28.50000 76.23333 84.00000 29.66667  
## 9 22 2012-12-12 0 28.46667 67.86667 85.00000 29.00000  
## 10 27 2012-12-14 0 28.16667 70.76667 83.33333 29.00000  
## 11 27 2012-12-15 0 29.10000 72.16667 83.56667 30.00000  
## 12 31 2012-12-17 0 31.73333 81.43333 84.76667 32.33333  
## 13 31 2012-12-18 0 31.76667 84.43333 84.66667 32.00000  
## 14 31 2012-12-19 0 28.06667 74.90000 82.56667 29.00000  
## 15 32 2012-12-18 0 28.36667 76.43333 83.73333 29.00000  
## 16 32 2012-12-19 0 28.96667 70.76667 82.96667 30.00000  
## 17 35 2012-12-19 0 29.36667 80.96667 84.63333 30.00000  
## 18 36 2012-12-19 0 28.56667 66.96667 83.16667 29.33333  
## 19 38 2012-12-21 0 28.06667 68.36667 81.33333 29.00000  
## 20 40 2012-12-28 0 28.10000 76.10000 78.96667 29.00000  
## 21 42 2013-01-02 0 28.06667 74.86667 80.00000 29.00000  
## 22 42 2013-01-03 0 28.10000 72.70000 79.03333 29.00000  
## 23 42 2013-01-04 0 28.06667 76.53333 79.56667 29.00000  
## 24 43 2013-01-02 0 28.03333 63.56667 80.93333 29.00000  
## 25 44 2013-01-02 0 28.10000 65.40000 81.83333 29.00000  
## 26 45 2013-01-02 0 28.06667 72.86667 82.43333 29.00000  
## 27 46 2013-01-04 0 28.06667 76.16667 79.30000 29.00000  
## 28 47 2013-01-04 0 28.06667 72.10000 80.70000 29.00000  
## 29 47 2013-01-05 0 28.06667 70.86667 80.06667 29.00000  
## 30 48 2013-01-05 0 28.10000 69.86667 80.03333 29.00000  
## 31 49 2013-01-05 0 28.06667 74.06667 80.00000 29.00000  
## 32 50 2013-01-07 0 28.06667 66.00000 81.60000 29.00000  
## 33 51 2013-01-08 0 28.06667 78.13333 82.23333 29.00000  
## 34 52 2013-01-08 0 28.13333 73.33333 84.23333 29.00000  
## 35 52 2013-01-09 0 28.10000 64.40000 84.36667 29.00000  
## 36 52 2013-01-10 1 28.10000 81.03333 83.30000 29.00000  
## 37 54 2013-01-10 0 28.10000 73.53333 81.73333 29.00000  
## 38 55 2013-01-10 1 28.73333 79.53333 81.13333 29.33333  
## 39 57 2013-01-10 0 28.13333 65.56667 83.53333 29.00000  
## 40 58 2013-01-10 0 28.06667 65.76667 83.20000 29.00000  
## 41 59 2013-01-10 0 28.06667 73.60000 83.83333 29.00000  
## 42 62 2013-01-11 0 28.10000 74.83333 84.00000 29.00000  
## 43 62 2013-01-12 0 28.03333 80.03333 81.96667 27.00000  
## 44 62 2013-01-13 0 28.20000 79.10000 83.03333 29.00000  
## 45 63 2013-01-11 0 28.40000 88.10000 83.50000 29.00000  
## 46 63 2013-01-12 1 28.23333 91.60000 82.73333 49.66667  
## 47 63 2013-01-13 1 29.50000 87.73333 82.76667 50.00000  
## 48 64 2013-01-12 1 28.40000 78.06667 83.93333 50.33333  
## 49 64 2013-01-13 1 28.96667 78.40000 83.83333 50.00000  
## 50 66 2013-01-13 0 29.10000 81.30000 83.76667 29.66667  
## 51 66 2013-01-14 0 28.33333 87.40000 82.53333 29.00000  
## 52 66 2013-01-15 0 28.46667 85.70000 83.03333 29.00000  
## 53 68 2013-01-14 1 28.06667 81.06667 83.00000 50.00000  
## 54 68 2013-01-15 1 28.03333 83.90000 82.43333 42.00000  
## 55 68 2013-01-16 1 28.00000 82.40000 82.56667 47.33333  
## 56 69 2013-01-15 0 28.10000 84.66667 82.30000 29.00000  
## 57 70 2013-01-16 0 28.03333 78.83333 82.40000 29.00000  
## 58 70 2013-01-17 0 28.10000 78.16667 82.50000 28.00000  
## 59 70 2013-01-18 1 28.16667 75.23333 81.60000 44.00000  
## 60 75 2013-01-18 0 28.10000 71.30000 82.63333 29.00000  
## 61 78 2013-01-22 1 28.03333 74.20000 80.73333 28.66667  
## 62 78 2013-01-23 0 28.06667 74.90000 77.86667 28.00000  
## 63 79 2013-01-22 0 28.10000 84.53333 80.00000 30.00000  
## 64 79 2013-01-23 0 28.06667 71.80000 79.03333 28.66667  
## 65 79 2013-01-24 0 28.10000 78.16667 81.10000 28.00000  
## 66 80 2013-01-22 1 28.13333 79.06667 81.93333 48.00000  
## 67 81 2013-01-22 0 28.03333 70.70000 80.90000 30.00000  
## 68 84 2013-01-22 1 28.13333 73.10000 82.03333 47.33333  
## 69 84 2013-01-23 1 28.10000 84.33333 79.50000 48.00000  
## 70 88 2013-01-24 1 28.06667 82.43333 79.73333 46.00000  
## 71 90 2013-01-24 1 28.06667 77.23333 81.36667 42.00000  
## 72 92 2013-01-25 0 28.06667 74.56667 80.50000 29.00000  
## 73 94 2013-01-25 1 28.06667 88.30000 79.76667 42.00000  
## 74 95 2013-01-25 0 28.10000 77.90000 79.90000 30.00000  
## 75 96 2013-01-25 1 28.06667 79.73333 81.06667 46.00000  
## 76 97 2013-01-28 0 28.10000 81.60000 79.46667 29.66667  
## 77 97 2013-01-29 0 28.10000 82.26667 80.20000 29.00000  
## 78 97 2013-01-30 1 29.66667 90.40000 81.56667 50.00000  
## 79 100 2013-01-28 1 28.06667 75.50000 81.56667 48.00000  
## 80 101 2013-01-28 0 28.10000 77.23333 80.83333 30.00000  
## 81 103 2013-01-28 0 28.10000 72.56667 81.20000 30.00000  
## 82 103 2013-01-29 1 28.00000 72.76667 81.66667 47.33333  
## 83 104 2013-01-28 1 28.13333 75.80000 81.83333 46.00000  
## 84 105 2013-01-28 0 28.36667 80.83333 80.23333 30.00000  
## 85 106 2013-01-28 1 28.10000 78.83333 84.16667 46.00000  
## 86 106 2013-01-29 1 28.76667 85.36667 83.76667 46.00000  
## 87 107 2013-01-28 0 28.06667 79.13333 81.26667 29.00000  
## 88 108 2013-01-29 0 28.10000 84.66667 81.50000 29.00000  
## 89 110 2013-01-29 0 28.06667 80.93333 82.60000 30.00000  
## 90 111 2013-01-29 1 29.16667 76.50000 85.10000 46.00000  
## 91 112 2013-01-29 0 28.13333 94.23333 82.00000 29.00000  
## 92 113 2013-01-30 0 29.50000 85.16667 83.10000 30.00000  
## 93 113 2013-01-31 1 28.93333 76.43333 84.06667 50.00000  
## 94 114 2013-02-02 1 28.06667 78.10000 80.56667 46.00000  
## 95 115 2013-01-30 1 30.56667 82.20000 83.23333 50.66667  
## 96 117 2013-01-30 0 30.40000 82.10000 82.76667 32.00000  
## 97 117 2013-01-31 0 33.32000 76.90000 85.36667 34.66667  
## 98 118 2013-01-30 1 30.06667 82.26667 82.53333 52.66667  
## 99 118 2013-01-31 1 29.13333 79.50000 84.90000 52.66667  
## 100 118 2013-02-01 0 32.60000 81.43333 81.23333 32.00000  
## 101 119 2013-01-31 0 28.06667 69.73333 82.80000 28.00000  
## 102 120 2013-01-31 1 28.06667 81.60000 83.83333 50.00000  
## 103 120 2013-02-01 1 28.20000 72.10000 84.20000 46.66667  
## 104 121 2013-01-31 1 29.20000 77.36667 84.26667 52.00000  
## 105 121 2013-02-01 1 28.06667 80.26667 81.70000 48.00000  
## 106 122 2013-01-31 0 32.54000 79.76667 84.56667 33.33333  
## 107 123 2013-01-31 1 28.23333 71.40000 85.26667 50.00000  
## 108 124 2013-01-31 0 32.66000 79.33333 85.16667 32.00000  
## 109 124 2013-02-01 0 32.54000 80.10000 83.33333 32.00000  
## 110 124 2013-02-02 0 32.60000 80.03333 79.93333 32.00000  
## 111 127 2013-02-04 0 32.54000 86.60000 78.96667 34.00000  
## 112 128 2013-02-01 1 28.13333 71.83333 83.33333 48.66667  
## 113 129 2013-02-01 0 32.60000 84.23333 83.46667 34.00000  
## 114 129 2013-02-02 0 32.54000 78.40000 79.60000 34.00000  
## 115 129 2013-02-03 1 28.06667 81.80000 80.13333 46.66667  
## 116 130 2013-02-01 1 28.26667 73.30000 83.93333 48.00000  
## 117 130 2013-02-02 1 28.03333 82.26667 81.40000 44.00000  
## 118 130 2013-02-03 0 32.54000 88.56667 76.33333 34.00000  
## 119 131 2013-02-01 0 32.54000 75.60000 83.66667 32.00000  
## 120 131 2013-02-02 1 28.06667 79.80000 79.76667 47.33333  
## 121 131 2013-02-03 1 28.06667 78.40000 79.76667 46.00000  
## 122 136 2013-02-03 0 32.54000 81.86667 55.08667 34.00000  
## 123 136 2013-02-04 0 32.60000 83.83333 81.70000 34.00000  
## 124 138 2013-02-04 0 32.54000 79.13333 78.40000 34.00000  
## 125 138 2013-02-05 0 32.57000 82.50000 78.16667 34.00000  
## cond\_tout  
## 1 29.00000  
## 2 29.00000  
## 3 29.66667  
## 4 29.00000  
## 5 28.66667  
## 6 30.33333  
## 7 31.66667  
## 8 28.33333  
## 9 28.00000  
## 10 28.00000  
## 11 29.00000  
## 12 30.00000  
## 13 30.00000  
## 14 28.00000  
## 15 28.00000  
## 16 28.00000  
## 17 29.00000  
## 18 28.33333  
## 19 28.00000  
## 20 28.00000  
## 21 28.00000  
## 22 28.00000  
## 23 28.00000  
## 24 28.00000  
## 25 28.00000  
## 26 28.00000  
## 27 28.00000  
## 28 28.00000  
## 29 28.00000  
## 30 28.00000  
## 31 28.00000  
## 32 27.00000  
## 33 27.00000  
## 34 27.00000  
## 35 27.00000  
## 36 27.00000  
## 37 27.00000  
## 38 27.33333  
## 39 27.00000  
## 40 27.00000  
## 41 27.00000  
## 42 27.00000  
## 43 29.00000  
## 44 27.00000  
## 45 27.00000  
## 46 30.00000  
## 47 30.66667  
## 48 30.00000  
## 49 30.00000  
## 50 27.66667  
## 51 27.00000  
## 52 27.00000  
## 53 30.00000  
## 54 36.00000  
## 55 30.00000  
## 56 27.00000  
## 57 27.00000  
## 58 26.00000  
## 59 30.00000  
## 60 26.33333  
## 61 47.33333  
## 62 26.00000  
## 63 26.00000  
## 64 26.00000  
## 65 26.00000  
## 66 28.00000  
## 67 27.33333  
## 68 26.00000  
## 69 26.00000  
## 70 26.00000  
## 71 26.00000  
## 72 27.00000  
## 73 28.00000  
## 74 26.00000  
## 75 26.00000  
## 76 26.33333  
## 77 27.00000  
## 78 26.66667  
## 79 26.66667  
## 80 26.00000  
## 81 26.00000  
## 82 26.00000  
## 83 26.00000  
## 84 27.00000  
## 85 26.00000  
## 86 26.00000  
## 87 26.00000  
## 88 26.00000  
## 89 26.00000  
## 90 27.33333  
## 91 26.00000  
## 92 28.00000  
## 93 27.00000  
## 94 26.00000  
## 95 26.00000  
## 96 28.00000  
## 97 32.66667  
## 98 26.00000  
## 99 26.00000  
## 100 30.00000  
## 101 26.00000  
## 102 26.00000  
## 103 26.00000  
## 104 26.66667  
## 105 26.00000  
## 106 30.66667  
## 107 26.00000  
## 108 30.00000  
## 109 30.00000  
## 110 30.00000  
## 111 32.00000  
## 112 26.00000  
## 113 32.00000  
## 114 30.00000  
## 115 26.00000  
## 116 26.00000  
## 117 24.00000  
## 118 32.00000  
## 119 30.00000  
## 120 26.00000  
## 121 26.00000  
## 122 32.00000  
## 123 32.00000  
## 124 32.00000  
## 125 32.00000  
## [ reached getOption("max.print") -- omitted 151 rows ]

print(nrow(g2\_log1))

## [1] 276

# Now get the vacine\_smoker\_antiviral\_sex df ready to merge  
VSAS # Already in good form

## subject.id sex fluvac\_cur anitviral\_24h Smoker  
## 1 116 1 0 0 0  
## 2 17 1 1 0 1  
## 3 32 0 0 1 1  
## 4 45 1 0 0 1  
## 5 46 0 0 0 1  
## 6 47 0 1 0 1  
## 7 58 1 0 0 1  
## 8 66 0 0 0 1  
## 9 80 1 0 1 1  
## 10 110 1 0 0 1  
## 11 114 1 0 0 1  
## 12 186 0 0 0 1  
## 13 187 0 1 0 1  
## 14 189 1 0 0 1  
## 15 194 1 0 1 1  
## 16 195 0 0 0 1  
## 17 196 1 0 0 1  
## 18 210 0 0 0 1  
## 19 249 1 0 0 1  
## 20 252 1 0 0 1  
## 21 253 0 1 0 1  
## 22 255 1 0 0 1  
## 23 265 1 0 0 1  
## 24 285 0 0 0 1  
## 25 303 1 0 0 1  
## 26 306 1 0 0 1  
## 27 313 1 1 0 1  
## 28 321 1 0 0 1  
## 29 326 1 0 0 1  
## 30 356 1 0 0 1  
## 31 161 1 0 0 1  
## 32 7 0 0 0 0  
## 33 8 0 0 0 0  
## 34 9 0 0 0 0  
## 35 10 0 0 0 0  
## 36 12 1 0 0 0  
## 37 13 1 0 0 0  
## 38 14 0 1 0 0  
## 39 15 0 0 0 0  
## 40 16 0 1 0 0  
## 41 18 0 0 0 0  
## 42 19 1 0 0 0  
## 43 20 1 0 0 0  
## 44 21 1 0 0 0  
## 45 22 1 0 0 0  
## 46 23 1 0 0 0  
## 47 24 1 0 0 0  
## 48 25 0 0 0 0  
## 49 26 0 0 0 0  
## 50 27 0 0 0 0  
## 51 29 0 0 0 0  
## 52 30 0 0 0 0  
## 53 31 1 0 0 0  
## 54 33 0 1 0 0  
## 55 34 1 0 0 0  
## 56 35 1 0 0 0  
## 57 36 0 0 0 0  
## 58 37 1 0 0 0  
## 59 38 0 0 0 0  
## 60 39 1 0 0 0  
## 61 40 0 0 0 0  
## 62 41 0 1 0 0  
## 63 42 0 0 0 0  
## 64 43 0 0 0 0  
## 65 44 0 1 0 0  
## 66 48 0 0 0 0  
## 67 49 0 0 0 0  
## 68 50 0 1 0 0  
## 69 51 1 0 0 0  
## 70 52 1 0 0 0  
## 71 54 1 0 0 0  
## 72 55 0 0 0 0  
## 73 56 0 0 0 0  
## 74 57 0 1 0 0  
## 75 59 0 1 0 0  
## 76 60 1 1 0 0  
## 77 61 0 1 0 0  
## 78 62 1 0 0 0  
## 79 63 1 0 0 0  
## 80 64 0 0 0 0  
## 81 65 0 0 0 0  
## 82 67 1 0 0 0  
## 83 68 0 0 0 0  
## 84 69 1 0 0 0  
## 85 70 1 1 0 0  
## 86 71 0 0 0 0  
## 87 72 1 1 0 0  
## 88 74 1 0 0 0  
## 89 75 0 1 0 0  
## 90 77 1 1 0 0  
## 91 78 0 0 0 0  
## 92 79 1 0 0 0  
## 93 81 1 0 0 0  
## 94 82 1 0 0 0  
## 95 83 1 0 0 0  
## 96 84 0 0 0 0  
## 97 85 1 0 0 0  
## 98 86 0 0 0 0  
## 99 87 1 0 0 0  
## 100 88 1 1 0 0  
## 101 89 0 0 0 0  
## 102 90 0 1 0 0  
## 103 91 0 0 0 0  
## 104 92 1 0 0 0  
## 105 93 1 0 0 0  
## 106 94 0 0 0 0  
## 107 95 1 0 0 0  
## 108 96 0 0 0 0  
## 109 97 0 0 0 0  
## 110 98 0 0 0 0  
## 111 99 0 0 0 0  
## 112 100 1 0 0 0  
## 113 101 1 0 0 0  
## 114 102 1 0 0 0  
## 115 103 0 1 0 0  
## 116 104 0 0 0 0  
## 117 105 1 0 0 0  
## 118 106 1 0 0 0  
## 119 107 1 0 0 0  
## 120 108 0 0 0 0  
## 121 109 1 0 0 0  
## 122 111 0 0 0 0  
## 123 112 0 1 0 0  
## 124 113 0 0 0 0  
## 125 115 0 0 0 0  
## 126 117 0 0 0 0  
## 127 118 0 1 0 0  
## 128 119 1 0 0 0  
## 129 120 1 1 0 0  
## 130 121 0 0 0 0  
## 131 122 1 0 0 0  
## 132 123 0 1 0 0  
## 133 124 1 0 0 0  
## 134 125 0 1 0 0  
## 135 126 0 0 0 0  
## 136 127 0 0 0 0  
## 137 128 0 0 0 0  
## 138 129 1 0 0 0  
## 139 130 1 0 0 0  
## 140 131 1 1 1 0  
## 141 132 1 1 0 0  
## 142 133 1 0 0 0  
## 143 134 1 1 0 0  
## 144 135 1 1 0 0  
## 145 136 1 0 0 0  
## 146 137 0 1 0 0  
## 147 138 0 1 0 0  
## 148 139 0 0 0 0  
## 149 140 0 0 0 0  
## 150 141 1 1 0 0  
## 151 142 1 0 0 0  
## 152 143 0 1 0 0  
## 153 144 1 1 0 0  
## 154 145 0 0 0 0  
## 155 146 1 0 0 0  
## 156 147 0 0 0 0  
## 157 148 1 1 0 0  
## 158 149 0 0 0 0  
## 159 150 1 0 0 0  
## 160 151 0 0 0 0  
## 161 152 1 0 0 0  
## 162 153 1 0 0 0  
## 163 154 1 0 0 0  
## 164 155 1 0 0 0  
## 165 156 0 0 0 0  
## 166 157 1 0 0 0  
## 167 158 1 0 0 0  
## 168 159 1 0 0 0  
## 169 160 1 0 0 0  
## 170 162 1 1 0 0  
## 171 163 0 0 0 0  
## 172 164 0 1 0 0  
## 173 165 0 0 0 0  
## 174 166 1 0 0 0  
## 175 167 0 0 0 0  
## 176 168 0 0 0 0  
## 177 169 1 0 0 0  
## 178 170 0 1 0 0  
## 179 171 0 1 0 0  
## 180 172 0 1 0 0  
## 181 173 0 0 0 0  
## 182 174 1 0 0 0  
## 183 175 1 0 0 0  
## 184 176 0 1 0 0  
## 185 177 1 0 0 0  
## 186 178 0 0 1 0  
## 187 179 0 0 0 0  
## 188 180 1 0 0 0  
## 189 181 0 1 0 0  
## 190 182 1 1 0 0  
## 191 183 0 1 0 0  
## 192 184 1 0 0 0  
## 193 185 1 0 0 0  
## 194 188 1 0 0 0  
## 195 190 1 0 0 0  
## 196 191 0 0 0 0  
## 197 192 1 0 0 0  
## 198 193 0 0 0 0  
## 199 197 1 0 0 0  
## 200 198 1 1 1 0  
## [ reached getOption("max.print") -- omitted 155 rows ]

print(nrow(VSAS))

## [1] 355

# Oddly the body temp variable from the clinical\_umd object appears to be more complete than the one in the samples.cc df  
# Thus we will elimnate the body\_temp variable from samples.cc and use the one from clinical\_umd\_body\_temp  
samples.cc <- samples.cc %>%  
 select(-body\_temp)  
print(nrow(samples.cc))

## [1] 1938

## Now can begin to bind all of the pieces together  
all\_data <- samples.cc %>%  
 left\_join(enrolled\_type.inf, by = "subject.id") %>%  
 left\_join(daypickfinal\_dpo, by = c("subject.id", "date.visit")) %>%  
 left\_join(pcr\_full\_merge, by = c("subject.id", "sample.id")) %>%  
 left\_join(clinical\_umd\_body\_temp, by = c("subject.id", "date.visit")) %>%   
 left\_join(clinical\_umd\_symptoms, by = c("subject.id", "date.visit")) %>%  
 left\_join(clinical\_umd\_asthma, by = "subject.id") %>%  
 left\_join(g2\_log1, by = c("subject.id", "date.visit")) %>%  
 left\_join(VSAS, by = "subject.id") %>%  
 arrange(subject.id, date.visit, type, sample.type, final.copies)  
print(nrow(all\_data))

## [1] 3216

all\_data$final.copies <- as.numeric(all\_data$final.copies)  
  
# Check number of subjectIDs in all\_data df  
all\_data\_subjectID\_check <- all\_data %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(all\_data\_subjectID\_check))

## [1] 355

all\_data\_gii\_sample\_instance\_check <- all\_data %>%  
 filter(g2.run == 1 | g2.run == 2 | g2.run == 3) %>%  
 distinct(subject.id, date.visit)  
print(nrow(all\_data\_gii\_sample\_instance\_check))

## [1] 276

# Hmm, I'm getting 276 entries here but we should have 278 according to PNAS SI Table S1.   
# Looking back through the G2 log and the rest of the data, I keep seeing 276 as the correct number here.  
# I'm not able to replicate the 278 number here.   
  
# This df is missing the following variables (compared with Jing's output):  
# centimeterheight  
# kilogramweight  
# BMI  
# cur\_asthma (how is this different than the asthma variable that is already in the df?)  
# lung\_sym\_2pos (not sure what this is or how to compute?)  
# fluvac\_last2y  
# bothyear  
# age  
  
# Not sure what lung\_symp\_2pos means, but the other variables can be taken directly from the clinical\_umd df or derived from variables in the clinical\_umd df.   
  
clinical\_umd\_height\_weight\_BMI\_vax\_age <- clinical\_umd %>%  
 filter(!is.na(age)) %>%  
 mutate(height\_inches = height\_in + 12\*(height\_ft)) %>%  
 mutate(height\_cm = 2.54\*(height\_inches)) %>%  
 mutate(weight\_kg = 0.453592\*weight) %>%  
 mutate(BMI = weight\_kg/((height\_cm/100)^2)) %>%  
 mutate(vax\_bothyear = ifelse(fluvac\_cur == 1 & fluvac\_last2y == 1, 1, 0)) %>%  
 rename(subject.id = field\_subj\_id, date.visit = date\_visit) %>%  
 select(subject.id, height\_cm, weight\_kg, BMI, fluvac\_last2y, vax\_bothyear, fluvac\_10y, age)  
  
missing\_fluvac\_last2y <- clinical\_umd\_height\_weight\_BMI\_vax\_age %>%  
 filter(is.na(fluvac\_last2y))  
print(nrow(missing\_fluvac\_last2y))

## [1] 121

# Note there are 121 missing fluvac\_last2y data on 121 subjects. It looks like in Jing's final dataset used in tobit regression, these NAs were turned into 0s.   
  
all\_data <- all\_data %>%  
 left\_join(clinical\_umd\_height\_weight\_BMI\_vax\_age, by = c("subject.id")) %>%  
 arrange(subject.id, date.visit, type, sample.type, Experiment, final.copies)  
  
# Note that the finaldataset only has data for NPS, fine, and coarse, so note that most of the variables for the sample.type vars levels of throat swab and anterior nasal swab are NA.   
  
# Write out the all\_data df  
write.csv(all\_data, "Curated Data/Analytical Datasets/all\_screened.csv")  
  
#### Creating the all\_cases dfs ####  
  
## Now, to make the all\_cases df, we take only the subjects that are in the enrolled\_type.inf object and also cut to only those with g2 data  
all\_cases <- enrolled\_type.inf %>%  
 select(subject.id) %>%  
 inner\_join(all\_data)

## Joining, by = "subject.id"

print(nrow(all\_cases))

## [1] 2764

write.csv(all\_cases, "Curated Data/Analytical Datasets/all\_cases.csv")  
  
all\_cases\_gii\_samples <- enrolled\_type.inf %>%  
 select(subject.id) %>%  
 inner\_join(all\_data) %>%  
 filter(g2.run != 0)

## Joining, by = "subject.id"

print(nrow(all\_cases\_gii\_samples))

## [1] 2751

# Check number of subjectIDs in all\_cases df  
all\_cases\_gii\_samples\_subjectID\_check <- all\_cases\_gii\_samples %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(all\_cases\_gii\_samples\_subjectID\_check))

## [1] 178

# There are 178 subjects  
  
all\_cases\_gii\_sample\_instance\_check <- all\_cases\_gii\_samples %>%  
 distinct(subject.id, date.visit)  
print(nrow(all\_cases\_gii\_sample\_instance\_check))

## [1] 276

# There are 276 gii sampling instances on these 178 subjects.   
  
# Write out the all\_cases df  
write.csv(all\_cases\_gii\_samples, "Curated Data/Analytical Datasets/all\_cases\_gii\_samples.csv")  
  
#### Creating flu\_cases dfs ####  
## Now, to make the flu\_cases df, we take the all the data where we have an enrolled participant (i.e., data on subtype used as a positive key for if someone enrolled and was positive for influenza virus)  
flu\_cases <- subtype %>%  
 select(subject.id) %>%  
 inner\_join(all\_data)

## Joining, by = "subject.id"

print(nrow(flu\_cases))

## [1] 2422

write.csv(flu\_cases, "Curated Data/Analytical Datasets/flu\_cases.csv")  
  
# We will also make another version of this that includes only the data where gii sampling instances occurred.   
flu\_cases\_gii\_samples <- subtype %>%  
 select(subject.id) %>%  
 inner\_join(all\_data) %>%  
 filter(g2.run != 0)

## Joining, by = "subject.id"

print(nrow(flu\_cases\_gii\_samples))

## [1] 2409

# Check number of subjectIDs in all\_cases df  
flu\_cases\_gii\_samples\_subjectID\_check <- flu\_cases\_gii\_samples %>%  
 group\_by(subject.id) %>%  
 count()  
print(nrow(flu\_cases\_gii\_samples\_subjectID\_check))

## [1] 158

# 158 subjectIDs  
  
# Check the number of gii sampling instances  
flu\_cases\_gii\_samples\_gii\_sampling\_instances\_check <- flu\_cases\_gii\_samples %>%  
 distinct(subject.id, date.visit)  
print(nrow(flu\_cases\_gii\_samples\_gii\_sampling\_instances\_check))

## [1] 250

# 250 gii sampling instances for the 150 subjects  
  
# Let's examine this set of sampling instances that was just eliminated when we went from all\_cases\_gii\_samples to flu\_cases\_gii\_samples  
all\_cases\_to\_flu\_cases\_gii\_samples <- all\_cases\_gii\_samples %>%  
 anti\_join(flu\_cases\_gii\_samples)

## Joining, by = c("subject.id", "date.visit", "date\_on\_sx", "sample.id", "sample.type", "g2.run", "visit.num", "passpos", "validp", "focus.ct", "enrolled", "rapid\_flu\_\_\_3", "rapid\_flu\_loc", "cough\_number", "sneeze\_number", "type.inf", "dpo", "type", "Experiment", "Ct", "final.copies", "body\_temp", "nose\_run", "nose\_stuf", "sneeze", "throat\_sr", "earache", "malaise", "headache", "mj\_ache", "sw\_fever\_chill", "lymph\_node", "chest\_tight", "sob", "cough", "asthma", "g2\_unit", "chiller.t", "elbow\_rh", "elbow\_t", "cond\_tin", "cond\_tout", "sex", "fluvac\_cur", "anitviral\_24h", "Smoker", "height\_cm", "weight\_kg", "BMI", "fluvac\_last2y", "vax\_bothyear", "fluvac\_10y", "age")

# Check to make sure all of these instances were negative (for the instances where there was pcr data)  
all\_cases\_to\_flu\_cases\_gii\_samples\_pcr <- all\_cases\_to\_flu\_cases\_gii\_samples %>%  
 filter(!is.na(final.copies))  
# Correct.   
# So, now let's count how many gii sampling instances are part of this all\_cases\_to\_flu\_cases\_gii\_samples df  
  
all\_cases\_to\_flu\_cases\_gii\_samples\_gii\_sampling\_instance\_check <- all\_cases\_to\_flu\_cases\_gii\_samples %>%  
 distinct(subject.id, date.visit)  
print(nrow(all\_cases\_to\_flu\_cases\_gii\_samples\_gii\_sampling\_instance\_check))

## [1] 26

# There were 26 instances here that were removed due to "not confirmed by PCR"  
# Were these 26 instances just among the 20 subjects excluded or were there others gii instances excluded from some subjects outside of the grouop of 20?  
# The 26 should just be from the 20 subjects excluded but let's double check  
all\_cases\_to\_flu\_cases\_gii\_samples\_gii\_sampling\_instance\_check\_subID <- all\_cases\_to\_flu\_cases\_gii\_samples %>%  
 distinct(subject.id, date.visit) %>%  
 distinct(subject.id)  
print(nrow(all\_cases\_to\_flu\_cases\_gii\_samples\_gii\_sampling\_instance\_check\_subID))

## [1] 20

# Sure enough, the 26 gii instances came from the 20 subjects excluded.  
  
# Write out the flu\_cases\_gii\_samples df  
write.csv(flu\_cases\_gii\_samples, "Curated Data/Analytical Datasets/flu\_cases\_gii\_samples.csv")  
  
#### Getting from the 158 subjects and 250 sampling instances down to the numbers in the final PNAS dataset ####  
  
## Exclude visits on dpo = 0  
flu\_cases\_gii\_samples\_exclude\_day0 <- flu\_cases\_gii\_samples %>%  
 filter(dpo != 0)  
  
# Check subject IDs  
flu\_cases\_gii\_samples\_exclude\_day0\_subjectID\_check <- flu\_cases\_gii\_samples\_exclude\_day0 %>%  
 distinct(subject.id)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_subjectID\_check))

## [1] 156

# 156 subjects (2 subjects fewer than before this round of exclusion)  
  
# Check gii sampling instances  
flu\_cases\_gii\_samples\_exclude\_day0\_gii\_sampling\_instance\_check <- flu\_cases\_gii\_samples\_exclude\_day0 %>%  
 distinct(subject.id, date.visit)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_gii\_sampling\_instance\_check))

## [1] 242

# 242 gii sampling instances (8 subjects fewer than before this round of exclusion)  
  
# Who are the subjects that contributed to thes 8 gii sampling instances excluded and how many of these instances do they account for each?  
flu\_cases\_gii\_samples\_exclude\_day0\_gii\_sampling\_instance\_check\_subID <- flu\_cases\_gii\_samples\_gii\_sampling\_instances\_check %>%  
 anti\_join(flu\_cases\_gii\_samples\_exclude\_day0\_gii\_sampling\_instance\_check) %>%  
 group\_by(subject.id, date.visit) %>%  
 count()

## Joining, by = c("subject.id", "date.visit")

print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_gii\_sampling\_instance\_check\_subID))

## [1] 8

# This shows that there were 8 subjects that each lost a single gii sampling instance, however 6 of these had multiple gii sampling instances and thus, not all of their data was excluded by this exclusion step.  
# To check which subjects were the 2 that were completely excluded via this data exclusion step...  
dpo0\_subjects <- flu\_cases\_gii\_samples\_subjectID\_check %>%  
 anti\_join(flu\_cases\_gii\_samples\_exclude\_day0\_subjectID\_check) %>%  
 count()

## Joining, by = "subject.id"

print(nrow(dpo0\_subjects))

## [1] 2

## Exclude visits after dpo = 3  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus <- flu\_cases\_gii\_samples\_exclude\_day0 %>%  
 filter(dpo == 1 | dpo == 2 | dpo == 3)  
  
# Check subject IDs  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_subjectID\_check <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus %>%  
 distinct(subject.id, .keep\_all = TRUE)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_subjectID\_check))

## [1] 149

# 149 subjects  
  
# Check gii sampling instances  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus %>%  
 distinct(subject.id, date.visit)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check))

## [1] 232

# 232 gii sampling instances  
  
# Who are the subjects that contributed to thes 10 gii sampling instances excluded and how many of these instances do they account for each?  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check\_subID <- flu\_cases\_gii\_samples\_exclude\_day0\_gii\_sampling\_instance\_check %>%  
 anti\_join(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check) %>%  
 group\_by(subject.id, date.visit) %>%  
 count()

## Joining, by = c("subject.id", "date.visit")

print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check\_subID))

## [1] 10

# This shows that there were 10 subjects that each lost a single gii sampling instance, however 3 of these had multiple gii sampling instances and thus, not all of their data was excluded by this exclusion step.  
# To check which subjects were the 7 that were completely excluded via this data exclusion step...  
dpo0\_and\_4plus\_subjects <- flu\_cases\_gii\_samples\_exclude\_day0\_subjectID\_check %>%  
 anti\_join(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_subjectID\_check) %>%  
 count()

## Joining, by = "subject.id"

print(nrow(dpo0\_and\_4plus\_subjects))

## [1] 7

# To check which subjects were the other 3 that were not completely excluded via this data exclusion step...  
remaining <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check\_subID %>%  
 anti\_join(dpo0\_and\_4plus\_subjects)

## Joining, by = c("subject.id", "n")

print(nrow(remaining))

## [1] 3

## Exclude where cough data is missing  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus %>%  
 filter(!is.na(cough\_number))  
  
# Check subject IDs  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_subjectID\_check <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough %>%  
 distinct(subject.id)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_subjectID\_check))

## [1] 147

# 147 subjects  
  
# Check gii sampling instances  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_gii\_sampling\_instance\_check <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough %>%  
 distinct(subject.id, date.visit)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_gii\_sampling\_instance\_check))

## [1] 225

# 225 gii sampling instances  
  
# Who are the subjects that contributed to these 7 gii sampling instances excluded and how many of these instances do they account for each?  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_gii\_sampling\_instance\_check\_subID <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_gii\_sampling\_instance\_check %>%  
 anti\_join(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_gii\_sampling\_instance\_check) %>%  
 distinct(subject.id)

## Joining, by = c("subject.id", "date.visit")

print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_gii\_sampling\_instance\_check\_subID))

## [1] 7

# This shows that there were 7 subjects that each lost a single gii sampling instance, however 5 of these had multiple gii sampling instances and thus, not all of their data was excluded by this exclusion step.  
# To check which subjects were the 2 that were completely excluded via this data exclusion step...  
dpo0\_and\_4plus\_cough\_subjects <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_subjectID\_check %>%  
 anti\_join(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_subjectID\_check) %>%  
 count()

## Joining, by = "subject.id"

print(nrow(dpo0\_and\_4plus\_cough\_subjects))

## [1] 2

# To check which subjects were the other 5 that were not completely excluded via this data exclusion step...  
remaining <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_gii\_sampling\_instance\_check\_subID %>%  
 anti\_join(dpo0\_and\_4plus\_cough\_subjects, by = "subject.id")  
print(nrow(remaining))

## [1] 5

## Exclude where there is incomplete PCR data  
  
# DATA EXCLUSION DUE TO PCR & OTHER ISSUES ##  
# Drop 333 only gii visit because lost coarse aerosol sample (although good data exists for the NP and the fine aerosol)  
# Drop 52 because false positive (this makes 52 a negative case and thus we exclude all 2/2 gii sampling instances)  
# Drop 58 because false positive (58 only had 1 gii sampling instance so this was excluded)  
# Drop 182 2nd gii visit because bad interrun calibrator on the PCR (there is still a 1st gii visit for 182 so this subject is not excluded entirely)  
# Drop 322 because bad interrun calibrator on the PCR (this was the only gii sampling instance so this subject is excluded entirely)  
# Drop 337 because bad interrun calibrator on the PCR (this was the only gii sampling instance so this subject is excluded entirely)  
  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough %>%  
 filter(!subject.id %in% c(333, 52, 58, 322, 337)) %>%  
 filter(!(subject.id == 182 & date.visit == "2013-02-08"))  
  
## END DATA EXCLUSION DUE TO PCR & OTHER ISSUES ##  
  
# Check subject IDs  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR\_subjectID\_check <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR %>%  
 distinct(subject.id)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR\_subjectID\_check))

## [1] 142

# 142 subjects  
  
# Check gii sampling instances  
flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR\_gii\_sampling\_instance\_check <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR %>%  
 filter(g2.run != 0) %>%  
 distinct(subject.id, date.visit)  
print(nrow(flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR\_gii\_sampling\_instance\_check))

## [1] 218

# 218 gii sampling instances  
  
# Let's make the name of this df a little better to understand and remove superfluous observations to get out what we got for the finaldataset df  
PNAS\_data\_full <- flu\_cases\_gii\_samples\_exclude\_day0\_and\_dpo4plus\_and\_missingcough\_incompletePCR  
  
## Remove the throat swab and anterior nasal swab information  
## Also remove all the times that the type.inf doesn't match with the type  
  
table(PNAS\_data\_full$type.inf)

##   
## B B and unsubtypable A H3N2   
## 600 29 1288   
## H3N2 and B H3N2 and PH1 Pandemic H1   
## 33 9 43   
## Unsubtypable A   
## 53

table(PNAS\_data\_full$type)

##   
## A B   
## 967 513

PNAS\_data\_full <- PNAS\_data\_full %>%  
 filter(sample.type != "Throat Swab") %>%  
 filter(sample.type != "anterior nasal swab") %>%  
 filter((type.inf == "H3N2" & type == "A") |   
 (type.inf == "Pandemic H1" & type == "A") |  
 (type.inf == "Unsubtypable A" & type == "A") |  
 (type.inf == "H3N2 and PH1" & type == "A") |  
 (type.inf == "B" & type == "B") |  
 (type.inf == "B and unsubtypable A" & (type == "B" | type == "A")) |  
 (type.inf == "H3N2 and B" & (type == "B" | type == "A")))  
  
## Compare the PNAS\_data\_full df with the finaldataset that was already produced.  
# The only difference should be that the PNAS\_data\_full has some vars that aren't in finaldataset.  
# Also note that the finaldataset df has indicator variables for fine, coarse, and NPS, symptoms, height, weight, BMI, vax vars, age (all variables that were added to the df at the very end of manipulation), that the PNAS\_data\_full doesn't have, but could be added easily.   
  
compare(PNAS\_data\_full, finaldataset)

## Compare Object  
##   
## Function Call:   
## compare.data.frame(x = PNAS\_data\_full, y = finaldataset)  
##   
## Shared: 32 variables and 1232 observations.  
## Not shared: 29 variables and 0 observations.  
##   
## Differences found in 0/31 variables compared.  
## 0 variables compared have non-identical attributes.

summary(compare(PNAS\_data\_full, finaldataset))

##   
##   
## Table: Summary of data.frames  
##   
## version arg ncol nrow  
## -------- --------------- ----- -----  
## x PNAS\_data\_full 53 1232  
## y finaldataset 38 1232  
##   
##   
##   
## Table: Variables not shared  
##   
## version variable position class   
## -------- --------------- --------- ----------  
## x date\_on\_sx 3 factor   
## x visit.num 7 numeric   
## x passpos 8 logical   
## x validp 9 logical   
## x focus.ct 10 numeric   
## x enrolled 11 logical   
## x rapid\_flu\_\_\_3 12 integer   
## x rapid\_flu\_loc 13 integer   
## x Ct 20 character   
## x nose\_run 23 integer   
## x nose\_stuf 24 integer   
## x sneeze 25 integer   
## x throat\_sr 26 integer   
## x earache 27 integer   
## x malaise 28 integer   
## x headache 29 integer   
## x mj\_ache 30 integer   
## x sw\_fever\_chill 31 integer   
## x lymph\_node 32 integer   
## x chest\_tight 33 integer   
## x sob 34 integer   
## x cough 35 integer   
## y upper\_sym 13 integer   
## y lower\_sym 14 integer   
## y systemic\_sym 15 integer   
## y sampleid 28 character   
## y NPswab 29 numeric   
## y Coarse 30 numeric   
## y Fine 31 numeric   
##   
##   
##   
## Table: Other variables not compared  
##   
## | |  
## |:-------------------------------|  
## |No other variables not compared |  
##   
##   
##   
## Table: Observations not shared  
##   
## | |  
## |:--------------------------|  
## |No observations not shared |  
##   
##   
##   
## Table: Differences detected by variable  
##   
## var.x var.y n NAs  
## -------------- -------------- --- ----  
## subject.id subject.id 0 0  
## date.visit date.visit 0 0  
## sample.id sample.id 0 0  
## sample.type sample.type 0 0  
## g2.run g2.run 0 0  
## cough\_number cough\_number 0 0  
## sneeze\_number sneeze\_number 0 0  
## type.inf type.inf 0 0  
## dpo dpo 0 0  
## type type 0 0  
## Experiment Experiment 0 0  
## final.copies final.copies 0 0  
## body\_temp body\_temp 0 0  
## asthma asthma 0 0  
## g2\_unit g2\_unit 0 0  
## chiller.t chiller.t 0 0  
## elbow\_rh elbow\_rh 0 0  
## elbow\_t elbow\_t 0 0  
## cond\_tin cond\_tin 0 0  
## cond\_tout cond\_tout 0 0  
## sex sex 0 0  
## fluvac\_cur fluvac\_cur 0 0  
## anitviral\_24h anitviral\_24h 0 0  
## Smoker Smoker 0 0  
## height\_cm height\_cm 0 0  
## weight\_kg weight\_kg 0 0  
## BMI BMI 0 0  
## fluvac\_last2y fluvac\_last2y 0 0  
## vax\_bothyear vax\_bothyear 0 0  
## fluvac\_10y fluvac\_10y 0 0  
## age age 0 0  
##   
##   
##   
## Table: differences detected  
##   
## | |  
## |:-----------------------|  
## |No differences detected |  
##   
##   
##   
## Table: Non-identical attributes  
##   
## | |  
## |:---------------------------|  
## |No non-identical attributes |

# In fact, these dataset observations are identical!  
  
# Note that the PNAS\_data\_full is called PNAS becasue it has additional variables, such as rapid test results, that the finaldataset (and the PNAS set that Jing used) don't have  
  
write\_csv(PNAS\_data\_full, "Curated Data/Analytical Datasets/PNAS\_data\_full.csv")  
  
# The EMIT Main Quarantine manuscript requires fine and coarse aerosol virus RNA copy GM and SD for comparison  
# To facilitate this analysis for the EMIT Main Q manuscript, we will write out a copy of this dataset to the EMIT\_Data\_Analysis\_Jake/EMIT\_Quarantine/Curated Data/Analytical Datasets directory  
write.csv(PNAS\_data\_full, "/Users/jbueno/Box Sync/EMIT/EMIT\_Data\_Analysis\_Jake/EMIT\_Quarantine/Curated Data/Analytical Datasets/EMIT\_UMD\_PNAS\_data\_full.csv")  
  
# The EMIT Natural Versus Artificial Inoculation manuscript requires this PNAS\_data\_full df as well. Thus we will write it out to the appropriate directory as well.  
write.csv(PNAS\_data\_full, "/Users/jbueno/Box Sync/EMIT/EMIT\_Data\_Analysis\_Jake/Natural\_vs\_Artificial\_Infection/Analytical Datasets/EMIT\_UMD\_PNAS\_data\_full.csv")  
  
#### Exploring the PNAS\_data\_full df and the flu\_cases df a little more ####  
  
## Let's do some summary analysis   
  
table(PNAS\_data\_full$sample.type)

##   
## GII condensate NO mask Impactor 5 um NO mask Nasopharyngeal swab   
## 452 452 328

# Number of subjects with at least one positive aerosol sample on at least one day of sampling  
table(PNAS\_data\_full$sample.type)

##   
## GII condensate NO mask Impactor 5 um NO mask Nasopharyngeal swab   
## 452 452 328

PNAS\_data\_full\_pos\_aerosol <- PNAS\_data\_full %>%  
 filter(sample.type == "GII condensate NO mask" | sample.type == "Impactor 5 um NO mask") %>%  
 filter(!is.na(final.copies)) %>%  
 distinct(subject.id)  
print(nrow(PNAS\_data\_full\_pos\_aerosol))

## [1] 119

# Number of subjects with at least one positive fine particle aerosol sample on at least one day of sampling  
PNAS\_data\_full\_pos\_fine <- PNAS\_data\_full %>%  
 filter(sample.type == "GII condensate NO mask") %>%  
 filter(!is.na(final.copies)) %>%  
 distinct(subject.id)  
print(nrow(PNAS\_data\_full\_pos\_fine))

## [1] 118

# Number of subjects with at least one positive coarse particle aerosol sample on at least one day of sampling  
PNAS\_data\_full\_pos\_coarse <- PNAS\_data\_full %>%  
 filter(sample.type == "Impactor 5 um NO mask") %>%  
 filter(!is.na(final.copies)) %>%  
 distinct(subject.id)  
print(nrow(PNAS\_data\_full\_pos\_coarse))

## [1] 67

# Total number of subjects  
PNAS\_data\_full\_subjects <- PNAS\_data\_full %>%  
 distinct(subject.id)  
print(nrow(PNAS\_data\_full\_subjects))

## [1] 142

# Fraction positive on at least one day  
print(nrow(PNAS\_data\_full\_pos\_aerosol)/nrow(PNAS\_data\_full\_subjects))

## [1] 0.8380282

# Table with number of subjects with 1 gii instance, 2 gii instances, 3 gii instances  
gii\_instances <- PNAS\_data\_full %>%  
 distinct(subject.id, g2.run) %>%  
 mutate(one\_instance = ifelse(g2.run == 1 & g2.run != 2 & g2.run != 3, 1, 0)) %>%  
 mutate(two\_instances = ifelse(g2.run == 2 & g2.run != 1 & g2.run != 3, 1, 0)) %>%  
 mutate(three\_instances = ifelse(g2.run == 3 & g2.run != 1 & g2.run != 2, 1, 0))  
print(sum(gii\_instances$one\_instance))

## [1] 137

print(sum(gii\_instances$two\_instances))

## [1] 61

print(sum(gii\_instances$three\_instances))

## [1] 20

table(gii\_instances$g2.run)

##   
## 1 2 3   
## 137 61 20

#lapply(gii\_instances, function(x) data.frame(table(x)))  
  
# Person-visits with negative NPS  
gii\_person\_visit\_negative\_NPS <- PNAS\_data\_full %>%  
 filter(sample.type == "Nasopharyngeal swab") %>%  
 filter(is.na(final.copies)) %>%  
 distinct(subject.id, date.visit, .keep\_all = TRUE) %>%  
 select(subject.id, date.visit, type.inf)  
print(gii\_person\_visit\_negative\_NPS)

## # A tibble: 16 x 3  
## # Groups: subject.id [13]  
## subject.id date.visit type.inf   
## <dbl> <date> <chr>   
## 1 64 2013-01-12 Unsubtypable A   
## 2 64 2013-01-13 Unsubtypable A   
## 3 81 2013-01-22 B   
## 4 105 2013-01-28 Unsubtypable A   
## 5 129 2013-02-02 H3N2   
## 6 130 2013-02-02 H3N2   
## 7 130 2013-02-03 H3N2   
## 8 174 2013-02-07 B   
## 9 176 2013-02-07 H3N2   
## 10 189 2013-02-09 H3N2   
## 11 223 2013-02-12 B and unsubtypable A  
## 12 226 2013-02-11 Unsubtypable A   
## 13 230 2013-02-12 H3N2 and B   
## 14 329 2013-02-26 B   
## 15 329 2013-02-27 B   
## 16 365 2013-03-13 B

# However a closer look at these 16 instances revealed that there were only sampling instances from 5 subjects where there were positive results but negative NPS data.   
  
# Person-visits with all negative samples  
ever\_positive <- PNAS\_data\_full %>%  
 filter(!is.na(final.copies)) %>%  
 distinct(subject.id, date.visit)  
person\_visit\_negative <- PNAS\_data\_full %>%  
 anti\_join(ever\_positive) %>%  
 distinct(subject.id, date.visit)

## Joining, by = c("subject.id", "date.visit")

print(person\_visit\_negative)

## # A tibble: 2 x 2  
## # Groups: subject.id [2]  
## subject.id date.visit  
## <dbl> <date>   
## 1 174 2013-02-07  
## 2 329 2013-02-27