EMIT\_UMD\_Natural\_Infection\_Cleaning.R

jbueno

Mon Jan 7 14:47:16 2019

# EMIT UMD Natural Infection Study Data Curation - cleaning raw data to produce cleaned spreadsheets  
# 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 - January 2019  
# Summary:  
  
#### Load required packages and set working directory ####  
  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.7  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(RcppRoll)  
library(readxl)  
library(knitr)  
library(data.table)

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

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

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

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday,  
## week, yday, year

## The following object is masked from 'package:base':  
##   
## date

setwd("/Users/jbueno/Box Sync/EMIT/EMIT\_Data\_Analysis\_Jake")  
  
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: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.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] lubridate\_1.7.4 data.table\_1.11.8 knitr\_1.20   
## [4] readxl\_1.1.0 RcppRoll\_0.3.0 forcats\_0.3.0   
## [7] stringr\_1.3.1 dplyr\_0.7.7 purrr\_0.2.5   
## [10] readr\_1.1.1 tidyr\_0.8.2 tibble\_1.4.2   
## [13] ggplot2\_3.1.0 tidyverse\_1.2.1   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.19 cellranger\_1.1.0 pillar\_1.3.0 compiler\_3.5.1   
## [5] plyr\_1.8.4 bindr\_0.1.1 tools\_3.5.1 digest\_0.6.18   
## [9] jsonlite\_1.5 evaluate\_0.12 nlme\_3.1-137 gtable\_0.2.0   
## [13] lattice\_0.20-38 pkgconfig\_2.0.2 rlang\_0.3.0.1 cli\_1.0.1   
## [17] rstudioapi\_0.8 haven\_1.1.2 bindrcpp\_0.2.2 withr\_2.1.2   
## [21] xml2\_1.2.0 httr\_1.3.1 hms\_0.4.2 rprojroot\_1.3-2   
## [25] grid\_3.5.1 tidyselect\_0.2.5 glue\_1.3.0 R6\_2.3.0   
## [29] rmarkdown\_1.10 modelr\_0.1.2 magrittr\_1.5 backports\_1.1.2   
## [33] scales\_1.0.0 htmltools\_0.3.6 rvest\_0.3.2 assertthat\_0.2.0  
## [37] colorspace\_1.3-2 stringi\_1.2.4 lazyeval\_0.2.1 munsell\_0.5.0   
## [41] broom\_0.5.0 crayon\_1.3.4

# Now pasting code from Jing Yan and Don Milton that was used in previous work on the EMIT UMD data.  
# The goal here is to review their script and improve the clarity  
  
#### \*\*\*\* Using Script: Jing Yan and Dr. Milton's "Merge\_1-3.R-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 <- 'EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/REDCAP/EMITClinicalUMD2013.csv'  
clinical\_umd <- read.csv(clinical\_in\_file)  
  
# 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)  
  
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 <- rename(sum\_clinical, 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")  
print(head(tbl\_df(sum\_clinical)))

## # A tibble: 6 x 11  
## 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 5 more variables: visit\_num <dbl>, g2\_name <fct>, g2\_run <dbl>,  
## # enrolled <lgl>, clinical.i <lgl>

#### READ in and work with G2 LOG DATA ####  
  
g2\_in\_file <- 'EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/GII/EMITGIILogUMD2013.csv'  
g2\_log <- read.csv(g2\_in\_file)  
  
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 11  
## 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 5 more variables: 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 <- filter(g2\_log, !(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)  
  
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) %>%  
 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 5  
## field\_subj\_id date\_visit g2\_coll\_num subj\_min g2lm.i  
## <int> <date> <dbl> <int> <lgl>   
## 1 7 2012-12-04 1 30 TRUE   
## 2 8 2012-12-04 1 30 TRUE   
## 3 13 2012-12-10 1 30 TRUE   
## 4 13 2012-12-11 2 30 TRUE   
## 5 13 2012-12-12 3 30 TRUE   
## 6 17 2012-12-11 1 30 TRUE

#### 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,   
 g2\_coll\_num,   
 enrolled,   
 visit\_num,   
 g2\_run,   
 clinical.i,   
 g2lm.i,   
 rapid\_flu\_\_\_3,   
 rapid\_flu\_loc,   
 body\_temp)  
  
# 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)  
merge1 <- merge1 %>%   
 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 12  
## field\_subj\_id date\_visit g2\_coll\_num enrolled visit\_num g2\_run clinical.i  
## <int> <date> <dbl> <lgl> <dbl> <dbl> <lgl>   
## 1 7 2012-12-04 1 TRUE 1 1 TRUE   
## 2 8 2012-12-04 1 TRUE 1 1 TRUE   
## 3 9 2012-12-05 0 FALSE 1 0 TRUE   
## 4 10 2012-12-05 0 FALSE 1 0 TRUE   
## 5 12 2012-12-10 0 FALSE 1 0 TRUE   
## 6 13 2012-12-10 1 TRUE 1 1 TRUE   
## # ... with 5 more variables: g2lm.i <lgl>, rapid\_flu\_\_\_3 <int>,  
## # rapid\_flu\_loc <int>, body\_temp <dbl>, merge1.i <lgl>

#### READ in and work with the FIELD SAMPLE DATABASE ####  
  
# Input Field Sample Data  
field\_db\_in\_file <- 'EMIT\_UMD\_Natural\_Infection/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] "EMIT\_UMD\_Natural\_Infection/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)  
field.db1 <- filter(field.db1, field\_subj\_id != 28)  
field.db1 <- filter(field.db1, field\_subj\_id != 53)  
field.db1 <- filter(field.db1, field\_subj\_id != 73)  
field.db1 <- filter(field.db1, 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

#### 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)  
  
# Head of merge2  
print(head(tbl\_df(merge2)))

## # A tibble: 6 x 15  
## field\_subj\_id date\_visit g2\_coll\_num enrolled visit\_num g2\_run clinical.i  
## <int> <date> <dbl> <lgl> <dbl> <dbl> <lgl>   
## 1 7 2012-12-04 1 TRUE 1 1 TRUE   
## 2 7 2012-12-04 1 TRUE 1 1 TRUE   
## 3 7 2012-12-04 1 TRUE 1 1 TRUE   
## 4 7 2012-12-04 1 TRUE 1 1 TRUE   
## 5 7 2012-12-04 1 TRUE 1 1 TRUE   
## 6 8 2012-12-04 1 TRUE 1 1 TRUE   
## # ... with 8 more variables: g2lm.i <lgl>, rapid\_flu\_\_\_3 <int>,  
## # rapid\_flu\_loc <int>, body\_temp <dbl>, 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 g2\_coll\_num enrolled visit\_num g2\_run  
## 1 135 2013-02-05 0 FALSE 2 0  
## clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i  
## 1 TRUE NA 0 0 36.5 TRUE  
## sample\_id sample\_type field.db1.i  
## 1 <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 g2\_coll\_num enrolled visit\_num g2\_run  
## 1 135 2013-02-02 0 FALSE 1 0  
## 2 135 2013-02-02 0 FALSE 1 0  
## 3 135 2013-02-05 0 FALSE 2 0  
## clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i  
## 1 TRUE NA 0 0 37.0 TRUE  
## 2 TRUE NA 0 0 37.0 TRUE  
## 3 TRUE NA 0 0 36.5 TRUE  
## sample\_id sample\_type field.db1.i  
## 1 135\_1 Nasopharyngeal swab TRUE  
## 2 135\_6 Nasopharyngeal swab TRUE  
## 3 <NA> <NA> 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 <- filter(merge2, !(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 g2\_coll\_num enrolled visit\_num   
## [6] g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc  
## [11] body\_temp merge1.i 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 12  
## # ... with 12 variables: field\_subj\_id <int>, date\_visit <date>,  
## # 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>, 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 16  
## field\_subj\_id date\_visit g2\_coll\_num enrolled visit\_num g2\_run clinical.i  
## <int> <date> <dbl> <lgl> <dbl> <dbl> <lgl>   
## 1 7 2012-12-04 1 TRUE 1 1 TRUE   
## 2 7 2012-12-04 1 TRUE 1 1 TRUE   
## 3 7 2012-12-04 1 TRUE 1 1 TRUE   
## 4 7 2012-12-04 1 TRUE 1 1 TRUE   
## 5 7 2012-12-04 1 TRUE 1 1 TRUE   
## 6 8 2012-12-04 1 TRUE 1 1 TRUE   
## # ... with 9 more variables: g2lm.i <lgl>, rapid\_flu\_\_\_3 <int>,  
## # rapid\_flu\_loc <int>, body\_temp <dbl>, merge1.i <lgl>, sample\_id <chr>,  
## # sample\_type <chr>, field.db1.i <lgl>, merge2.i <lgl>

#### READ in and work with the UMD SAMPLES DATABASE (REDCAP DATA) ####  
sample\_in\_file <- 'EMIT\_UMD\_Natural\_Infection/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] "EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/REDCAP/EMITUMDSamples2013\_DATA.csv"

# 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 <- select(sample\_in %>%   
 filter(redcap\_event\_name == "collection\_arm\_1"),   
 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 <- select(passage, 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)  
  
focus1\_24g <- focus1\_24g %>%  
 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)  
  
focus1\_24w <- focus1\_24w %>%  
 rename(ct = ct\_24w)  
  
focus1\_96 <- focus1 %>%  
 filter(focus1$plate\_type == 2) %>%   
 select(-ct\_24w, -ct\_24g)  
  
focus1\_96 <- focus1\_96 %>%  
 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)  
  
focus2\_24g <- focus2\_24g %>%  
 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)  
  
focus2\_24w <- focus2\_24w %>%  
 rename(ct = ct\_24w)  
  
focus2\_96 <- focus2 %>%   
 filter(focus2$plate\_type == 2) %>%   
 select(-ct\_24w, -ct\_24g)  
  
focus2\_96 <- rename(focus2\_96, 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 <- select(focus, 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 ####  
  
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" "g2\_coll\_num" "enrolled"   
## [5] "visit\_num" "g2\_run" "clinical.i" "g2lm.i"   
## [9] "rapid\_flu\_\_\_3" "rapid\_flu\_loc" "body\_temp" "merge1.i"   
## [13] "sample\_id" "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"

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] 30

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

## # A tibble: 6 x 30  
## sample\_id field\_subj\_id.x date\_visit.x g2\_coll\_num enrolled visit\_num  
## <chr> <int> <date> <dbl> <lgl> <dbl>  
## 1 10\_1 10 2012-12-05 0 FALSE 1  
## 2 10\_6 10 2012-12-05 0 FALSE 1  
## 3 100\_1 100 2013-01-28 1 TRUE 1  
## 4 100\_2 100 2013-01-28 1 TRUE 1  
## 5 100\_3 100 2013-01-28 1 TRUE 1  
## 6 100\_4 100 2013-01-28 1 TRUE 1  
## # ... with 24 more variables: g2\_run <dbl>, clinical.i <lgl>,  
## # g2lm.i <lgl>, rapid\_flu\_\_\_3 <int>, rapid\_flu\_loc <int>,  
## # body\_temp <dbl>, 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 g2\_coll\_num enrolled visit\_num  
## 1 237\_1 237 2013-02-12 1 TRUE 1  
## 2 237\_2 237 2013-02-12 1 TRUE 1  
## 3 237\_3 237 2013-02-12 1 TRUE 1  
## 4 237\_4 237 2013-02-12 1 TRUE 1  
## 5 237\_5 237 2013-02-12 1 TRUE 1  
## 6 237\_6 237 2013-02-12 1 TRUE 1  
## g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i  
## 1 1 TRUE TRUE 1 NA 37.8 TRUE  
## 2 1 TRUE TRUE 1 NA 37.8 TRUE  
## 3 1 TRUE TRUE 1 NA 37.8 TRUE  
## 4 1 TRUE TRUE 1 NA 37.8 TRUE  
## 5 1 TRUE TRUE 1 NA 37.8 TRUE  
## 6 1 TRUE TRUE 1 NA 37.8 TRUE  
## sample\_type.x field.db1.i merge2.i field\_subj\_id.y  
## 1 Nasopharyngeal swab TRUE TRUE 237  
## 2 Impactor 5 um NO mask TRUE TRUE 237  
## 3 GII condensate NO mask TRUE TRUE 237  
## 4 anterior nasal swab TRUE TRUE 237  
## 5 Throat Swab TRUE TRUE 237  
## 6 TRUE TRUE 237  
## sample\_type.y date\_visit.y passpos validp ct np impactor  
## 1 Nasopharyngeal swab 2013-02-12 NA NA NA TRUE FALSE  
## 2 Impactor 5 um NO mask 2013-02-12 NA NA NA FALSE TRUE  
## 3 GII condensate NO mask 2013-02-12 NA NA NA FALSE FALSE  
## 4 anterior nasal swab 2013-02-12 NA NA NA FALSE FALSE  
## 5 Throat Swab 2013-02-12 NA NA NA FALSE FALSE  
## 6 Nasopharyngeal swab 2013-02-12 NA NA NA TRUE FALSE  
## condensate antnasal throat focus.i passage.i cr.i  
## 1 FALSE FALSE FALSE FALSE FALSE TRUE  
## 2 FALSE FALSE FALSE FALSE FALSE TRUE  
## 3 TRUE FALSE FALSE FALSE FALSE TRUE  
## 4 FALSE TRUE FALSE FALSE FALSE TRUE  
## 5 FALSE FALSE TRUE FALSE FALSE TRUE  
## 6 FALSE FALSE FALSE FALSE FALSE 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 g2\_coll\_num enrolled visit\_num  
## 1 301\_1 301 2013-02-19 0 FALSE 1  
## 2 301\_3 NA <NA> NA NA NA  
## 3 301\_5 NA <NA> NA NA NA  
## 4 301\_6 301 2013-02-19 0 FALSE 1  
## g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i  
## 1 0 TRUE NA 0 0 36.7 TRUE  
## 2 NA NA NA NA NA NA NA  
## 3 NA NA NA NA NA NA NA  
## 4 0 TRUE NA 0 0 36.7 TRUE  
## sample\_type.x field.db1.i merge2.i field\_subj\_id.y  
## 1 Nasopharyngeal swab TRUE TRUE 301  
## 2 <NA> NA NA NA  
## 3 <NA> NA NA NA  
## 4 Nasopharyngeal swab TRUE TRUE NA  
## sample\_type.y date\_visit.y passpos validp ct np impactor  
## 1 Nasopharyngeal swab 2013-02-19 TRUE TRUE NA TRUE FALSE  
## 2 <NA> <NA> FALSE TRUE NA NA NA  
## 3 <NA> <NA> TRUE TRUE NA NA NA  
## 4 <NA> <NA> NA NA NA NA NA  
## condensate antnasal throat focus.i passage.i cr.i  
## 1 FALSE FALSE FALSE FALSE TRUE TRUE  
## 2 NA NA NA FALSE TRUE TRUE  
## 3 NA NA NA FALSE TRUE TRUE  
## 4 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 g2\_coll\_num enrolled visit\_num  
## 1 301\_1 301 2013-02-19 0 FALSE 1  
## 2 301\_6 301 2013-02-19 0 FALSE 1  
## g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i  
## 1 0 TRUE NA 0 0 36.7 TRUE  
## 2 0 TRUE NA 0 0 36.7 TRUE  
## sample\_type.x field.db1.i merge2.i field\_subj\_id.y  
## 1 Nasopharyngeal swab TRUE TRUE 301  
## 2 Nasopharyngeal swab TRUE TRUE NA  
## sample\_type.y date\_visit.y passpos validp ct np impactor  
## 1 Nasopharyngeal swab 2013-02-19 TRUE TRUE NA TRUE FALSE  
## 2 <NA> <NA> NA NA NA NA NA  
## condensate antnasal throat focus.i passage.i cr.i  
## 1 FALSE FALSE FALSE FALSE TRUE TRUE  
## 2 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 g2\_coll\_num enrolled visit\_num  
## 1 288\_10 288 2013-02-19 2 TRUE 999  
## 2 288\_9 288 2013-02-19 2 TRUE 999  
## 3 290\_10 290 2013-02-19 2 TRUE 999  
## 4 290\_9 290 2013-02-19 2 TRUE 999  
## 5 292\_10 292 2013-02-19 2 TRUE 999  
## 6 292\_9 292 2013-02-19 2 TRUE 999  
## 7 296\_10 296 2013-02-19 2 TRUE 999  
## 8 296\_9 296 2013-02-19 2 TRUE 999  
## 9 297\_10 297 2013-02-19 2 TRUE 999  
## 10 297\_9 297 2013-02-19 2 TRUE 999  
## 11 298\_3 298 2013-02-19 1 TRUE 1  
## 12 298\_5 298 2013-02-19 1 TRUE 1  
## 13 299\_3 299 2013-02-19 1 TRUE 1  
## 14 299\_5 299 2013-02-19 1 TRUE 1  
## 15 302\_3 302 2013-02-19 1 TRUE 1  
## 16 302\_5 302 2013-02-19 1 TRUE 1  
## g2\_run clinical.i g2lm.i rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i  
## 1 2 TRUE TRUE NA NA NA TRUE  
## 2 2 TRUE TRUE NA NA NA TRUE  
## 3 2 TRUE TRUE NA NA NA TRUE  
## 4 2 TRUE TRUE NA NA NA TRUE  
## 5 2 TRUE TRUE NA NA NA TRUE  
## 6 2 TRUE TRUE NA NA NA TRUE  
## 7 2 TRUE TRUE NA NA NA TRUE  
## 8 2 TRUE TRUE NA NA NA TRUE  
## 9 2 TRUE TRUE NA NA NA TRUE  
## 10 2 TRUE TRUE NA NA NA TRUE  
## 11 1 TRUE TRUE 1 NA 38.1 TRUE  
## 12 1 TRUE TRUE 1 NA 38.1 TRUE  
## 13 1 TRUE TRUE 0 1 37.9 TRUE  
## 14 1 TRUE TRUE 0 1 37.9 TRUE  
## 15 1 TRUE TRUE 0 0 37.4 TRUE  
## 16 1 TRUE TRUE 0 0 37.4 TRUE  
## sample\_type.x field.db1.i merge2.i field\_subj\_id.y  
## 1 Throat Swab TRUE TRUE 288  
## 2 GII condensate NO mask TRUE TRUE 288  
## 3 Throat Swab TRUE TRUE 290  
## 4 GII condensate NO mask TRUE TRUE 290  
## 5 Throat Swab TRUE TRUE 292  
## 6 GII condensate NO mask TRUE TRUE 292  
## 7 Throat Swab TRUE TRUE 296  
## 8 GII condensate NO mask TRUE TRUE 296  
## 9 Throat Swab TRUE TRUE 297  
## 10 GII condensate NO mask TRUE TRUE 297  
## 11 GII condensate NO mask TRUE TRUE 298  
## 12 Throat Swab TRUE TRUE 298  
## 13 GII condensate NO mask TRUE TRUE 299  
## 14 Throat Swab TRUE TRUE 299  
## 15 GII condensate NO mask TRUE TRUE 302  
## 16 Throat Swab TRUE TRUE 302  
## sample\_type.y date\_visit.y passpos validp ct np  
## 1 Throat Swab 2013-02-19 TRUE TRUE 0.000000 FALSE  
## 2 GII condensate NO mask 2013-02-19 TRUE TRUE 0.000000 FALSE  
## 3 Throat Swab 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 4 GII condensate NO mask 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 5 Throat Swab 2013-02-19 TRUE TRUE 6.666667 FALSE  
## 6 GII condensate NO mask 2013-02-19 TRUE TRUE 0.000000 FALSE  
## 7 Throat Swab 2013-02-19 TRUE TRUE 0.000000 FALSE  
## 8 GII condensate NO mask 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 9 Throat Swab 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 10 GII condensate NO mask 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 11 GII condensate NO mask 2013-02-19 TRUE TRUE 866.666667 FALSE  
## 12 Throat Swab 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 13 GII condensate NO mask 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 14 Throat Swab 2013-02-19 TRUE TRUE 0.000000 FALSE  
## 15 GII condensate NO mask 2013-02-19 FALSE TRUE 0.000000 FALSE  
## 16 Throat Swab 2013-02-19 TRUE TRUE 286.666667 FALSE  
## impactor condensate antnasal throat focus.i passage.i cr.i  
## 1 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 2 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 3 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 4 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 5 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 6 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 7 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 8 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 9 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 10 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 11 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 12 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 13 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 14 FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## 15 FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## 16 FALSE FALSE 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 g2\_coll\_num enrolled visit\_num g2\_run clinical.i g2lm.i  
## 1 138\_16 3 TRUE 999 3 TRUE TRUE  
## 2 14\_6 0 FALSE 1 0 TRUE NA  
## 3 141\_12 2 TRUE 999 2 TRUE TRUE  
## 4 151\_6 0 FALSE 1 0 TRUE NA  
## 5 174\_16 3 TRUE 999 3 TRUE TRUE  
## 6 184\_7 1 TRUE 1 1 TRUE TRUE  
## 7 186\_6 1 TRUE 1 1 TRUE TRUE  
## 8 189\_12 2 TRUE 999 2 TRUE TRUE  
## 9 200\_7 0 FALSE 1 0 TRUE NA  
## 10 220\_6 0 FALSE 1 0 TRUE NA  
## 11 228\_7 1 TRUE 1 1 TRUE TRUE  
## 12 31\_17 3 TRUE 999 3 TRUE TRUE  
## 13 334\_6 0 FALSE 1 0 TRUE NA  
## 14 34\_7 0 FALSE 1 0 TRUE NA  
## 15 35\_7 1 TRUE 1 1 TRUE TRUE  
## 16 43\_7 1 TRUE 1 1 TRUE TRUE  
## 17 58\_6 1 TRUE 1 1 TRUE TRUE  
## 18 67\_6 0 FALSE 1 0 TRUE NA  
## 19 70\_12 3 TRUE 999 3 TRUE TRUE  
## rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i field.db1.i merge2.i  
## 1 NA NA NA TRUE TRUE TRUE  
## 2 1 NA 36.9 TRUE TRUE TRUE  
## 3 NA NA NA TRUE TRUE TRUE  
## 4 0 0 37.3 TRUE TRUE TRUE  
## 5 NA NA NA TRUE TRUE TRUE  
## 6 0 1 37.7 TRUE TRUE TRUE  
## 7 0 0 36.8 TRUE TRUE TRUE  
## 8 NA NA NA TRUE TRUE TRUE  
## 9 0 0 36.9 TRUE TRUE TRUE  
## 10 0 0 37.1 TRUE TRUE TRUE  
## 11 0 1 38.0 TRUE TRUE TRUE  
## 12 NA NA NA TRUE TRUE TRUE  
## 13 0 0 36.9 TRUE TRUE TRUE  
## 14 0 0 37.4 TRUE TRUE TRUE  
## 15 0 0 37.8 TRUE TRUE TRUE  
## 16 0 0 37.1 TRUE TRUE TRUE  
## 17 0 0 36.9 TRUE TRUE TRUE  
## 18 0 0 36.8 TRUE TRUE TRUE  
## 19 NA NA NA TRUE TRUE TRUE  
## passpos validp ct np impactor condensate antnasal throat focus.i  
## 1 NA NA NA NA NA NA NA NA NA  
## 2 NA NA NA NA NA NA NA NA NA  
## 3 NA NA NA NA NA NA NA NA NA  
## 4 NA NA NA NA NA NA NA NA NA  
## 5 NA NA NA NA NA NA NA NA NA  
## 6 NA NA NA NA NA NA NA NA NA  
## 7 NA NA NA NA NA NA NA NA NA  
## 8 NA NA NA NA NA NA NA NA NA  
## 9 NA NA NA NA NA NA NA NA NA  
## 10 NA NA NA NA NA NA NA NA NA  
## 11 NA NA NA NA NA NA NA NA NA  
## 12 NA NA NA NA NA NA NA NA NA  
## 13 NA NA NA NA NA NA NA NA NA  
## 14 NA NA NA NA NA NA NA NA NA  
## 15 NA NA NA NA NA NA NA NA NA  
## 16 NA NA NA NA NA NA NA NA NA  
## 17 NA NA NA NA NA NA NA NA NA  
## 18 NA NA NA NA NA NA NA NA NA  
## 19 NA NA NA NA NA NA NA NA NA  
## passage.i cr.i subject\_id sample\_type date\_visit  
## 1 NA NA 138 <NA> 2013-02-06  
## 2 NA NA 14 <NA> 2012-12-11  
## 3 NA NA 141 <NA> 2013-02-05  
## 4 NA NA 151 <NA> 2013-02-05  
## 5 NA NA 174 <NA> 2013-02-08  
## 6 NA NA 184 <NA> 2013-02-07  
## 7 NA NA 186 <NA> 2013-02-07  
## 8 NA NA 189 <NA> 2013-02-09  
## 9 NA NA 200 <NA> 2013-02-09  
## 10 NA NA 220 <NA> 2013-02-11  
## 11 NA NA 228 <NA> 2013-02-11  
## 12 NA NA 31 <NA> 2012-12-19  
## 13 NA NA 334 <NA> 2013-02-27  
## 14 NA NA 34 <NA> 2012-12-19  
## 15 NA NA 35 <NA> 2012-12-19  
## 16 NA NA 43 <NA> 2013-01-02  
## 17 NA NA 58 <NA> 2013-01-10  
## 18 NA NA 67 <NA> 2013-01-14  
## 19 NA NA 70 <NA> 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 g2\_coll\_num enrolled visit\_num g2\_run clinical.i g2lm.i  
## 1 247\_1 1 TRUE 1 1 TRUE TRUE  
## 2 247\_10 2 TRUE 999 2 TRUE TRUE  
## 3 247\_2 1 TRUE 1 1 TRUE TRUE  
## 4 247\_3 1 TRUE 1 1 TRUE TRUE  
## 5 247\_4 1 TRUE 1 1 TRUE TRUE  
## 6 247\_5 1 TRUE 1 1 TRUE TRUE  
## 7 247\_6 2 TRUE 999 2 TRUE TRUE  
## 8 247\_7 2 TRUE 999 2 TRUE TRUE  
## 9 247\_8 2 TRUE 999 2 TRUE TRUE  
## 10 247\_9 2 TRUE 999 2 TRUE TRUE  
## rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i field.db1.i merge2.i  
## 1 0 0 37.2 TRUE TRUE TRUE  
## 2 NA NA NA TRUE TRUE TRUE  
## 3 0 0 37.2 TRUE TRUE TRUE  
## 4 0 0 37.2 TRUE TRUE TRUE  
## 5 0 0 37.2 TRUE TRUE TRUE  
## 6 0 0 37.2 TRUE TRUE TRUE  
## 7 NA NA NA TRUE TRUE TRUE  
## 8 NA NA NA TRUE TRUE TRUE  
## 9 NA NA NA TRUE TRUE TRUE  
## 10 NA NA NA TRUE TRUE TRUE  
## passpos validp ct np impactor condensate antnasal throat  
## 1 NA NA NA TRUE FALSE FALSE FALSE FALSE  
## 2 NA NA NA FALSE TRUE FALSE FALSE FALSE  
## 3 NA NA NA FALSE TRUE FALSE FALSE FALSE  
## 4 NA NA NA FALSE FALSE TRUE FALSE FALSE  
## 5 NA NA NA FALSE FALSE FALSE TRUE FALSE  
## 6 NA NA NA FALSE FALSE FALSE FALSE TRUE  
## 7 TRUE TRUE 0.000000 TRUE FALSE FALSE FALSE FALSE  
## 8 NA NA NA TRUE FALSE FALSE FALSE FALSE  
## 9 FALSE TRUE 0.000000 FALSE FALSE TRUE FALSE FALSE  
## 10 TRUE TRUE 3.333333 FALSE FALSE FALSE FALSE TRUE  
## focus.i passage.i cr.i subject\_id sample\_type date\_visit  
## 1 FALSE FALSE TRUE 247 Nasopharyngeal swab 2013-02-12  
## 2 FALSE FALSE TRUE 247 Impactor 5 um NO mask 2013-02-13  
## 3 FALSE FALSE TRUE 247 Impactor 5 um NO mask 2013-02-12  
## 4 FALSE FALSE TRUE 247 GII condensate NO mask 2013-02-12  
## 5 FALSE FALSE TRUE 247 anterior nasal swab 2013-02-12  
## 6 FALSE FALSE TRUE 247 Throat Swab 2013-02-12  
## 7 TRUE TRUE TRUE 247 Nasopharyngeal swab 2013-02-13  
## 8 FALSE FALSE TRUE 247 Nasopharyngeal swab 2013-02-13  
## 9 TRUE TRUE TRUE 247 GII condensate NO mask 2013-02-13  
## 10 TRUE TRUE TRUE 247 Throat Swab 2013-02-13

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

## sample\_id g2\_coll\_num enrolled visit\_num g2\_run clinical.i g2lm.i  
## 1 225\_1 0 FALSE 1 0 TRUE NA  
## 2 225\_6 0 FALSE 1 0 TRUE NA  
## 3 250\_1 1 TRUE 2 1 TRUE TRUE  
## 4 250\_2 1 TRUE 2 1 TRUE TRUE  
## 5 250\_3 1 TRUE 2 1 TRUE TRUE  
## 6 250\_4 1 TRUE 2 1 TRUE TRUE  
## 7 250\_5 1 TRUE 2 1 TRUE TRUE  
## 8 250\_6 1 TRUE 2 1 TRUE TRUE  
## rapid\_flu\_\_\_3 rapid\_flu\_loc body\_temp merge1.i field.db1.i merge2.i  
## 1 0 0 36.7 TRUE TRUE TRUE  
## 2 0 0 36.7 TRUE TRUE TRUE  
## 3 0 0 36.8 TRUE TRUE TRUE  
## 4 0 0 36.8 TRUE TRUE TRUE  
## 5 0 0 36.8 TRUE TRUE TRUE  
## 6 0 0 36.8 TRUE TRUE TRUE  
## 7 0 0 36.8 TRUE TRUE TRUE  
## 8 0 0 36.8 TRUE TRUE TRUE  
## passpos validp ct np impactor condensate antnasal throat  
## 1 NA NA NA NA NA NA NA NA  
## 2 NA NA NA NA NA NA NA NA  
## 3 TRUE TRUE 1893.33333 TRUE FALSE FALSE FALSE FALSE  
## 4 NA NA NA FALSE TRUE FALSE FALSE FALSE  
## 5 NA FALSE 43.33333 FALSE FALSE TRUE FALSE FALSE  
## 6 NA NA NA FALSE FALSE FALSE TRUE FALSE  
## 7 TRUE TRUE 673.33333 FALSE FALSE FALSE FALSE TRUE  
## 8 NA NA NA NA NA NA NA NA  
## focus.i passage.i cr.i subject\_id sample\_type date\_visit  
## 1 NA NA NA 250 Nasopharyngeal swab 2013-02-11  
## 2 NA NA NA 250 Nasopharyngeal swab 2013-02-11  
## 3 TRUE TRUE TRUE 250 Nasopharyngeal swab 2013-02-13  
## 4 FALSE FALSE TRUE 250 Impactor 5 um NO mask 2013-02-13  
## 5 TRUE TRUE TRUE 250 GII condensate NO mask 2013-02-13  
## 6 FALSE FALSE TRUE 250 anterior nasal swab 2013-02-13  
## 7 TRUE TRUE TRUE 250 Throat Swab 2013-02-13  
## 8 NA NA NA 250 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,   
 sample.id,   
 sample.type,   
 g2.run,   
 visit.num,   
 passpos,   
 validp,   
 focus.ct,   
 enrolled,   
 rapid\_flu\_\_\_3,   
 rapid\_flu\_loc,   
 body\_temp)  
  
#### Write out EMIT\_samples.cc.RDS file from merge of Clin DB + G2 Log + Field Sample DB ####  
  
saveRDS(samples.cc, file = "EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_samples.cc.RDS")  
  
#### \*\*\*\* Using Script: Jing Yan and Dr. Milton's "Snippets analysis\_1.r" \*\*\*\* ####  
  
# Perhaps the earlier lines of script in this program address what "Snippets analysis\_1.r" was mostly getting at.  
# However, "Snippets analysis\_1.r" provides some, perhaps useful, summary information and looks at roommates.   
  
###  
# Original file information:  
  
# By Jing Yan & Don Milton  
# December 14, 2015 - December 21, 2015  
# Purpose: Combine clinical redcap data with GII data from redcap and check that all subject enrolled  
# according to redcap clinical data have the appropriate number of GII records and that  
# persons not enrolled only have screening visits (up to 3) and no GII records.  
# Will also generate a list of subjects showing whether they were enrolled and how many GII  
# sessions they completed.   
# Will identify roommate screenings and roommates enrolled after screening as roommates.   
  
###  
  
#### READ in and work with Clinical Database ####  
  
clinical\_umd <- read.csv(clinical\_in\_file)  
  
# clinical\_in\_file was an object that was taken from reading in "EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/REDCAP/EMITClinicalUMD2013.csv"  
# This was already read in with the "Merge\_1-3.R-update.r" script -- see above.  
  
## Check whether there was anyone with no visit 1 who has a record for having had a g2 run ##  
  
g2\_id <- unique(clinical\_umd$field\_subj\_id[grep('^g2', clinical\_umd$redcap\_event\_name)])  
  
visit1\_id <- unique(clinical\_umd$field\_subj\_id[grep('^visit', clinical\_umd$redcap\_event\_name)])  
  
# Number of subjects with at least 1 g2 visit who also have a visit 1  
print(sum(g2\_id %in% visit1\_id))

## [1] 178

# Number of subjects with at least one g2 visit regardless of having a visit 1  
print(length(g2\_id))

## [1] 178

# If the above two prints are the same, then everyone who gave a g2 sample had a visit 1 screening visit.  
  
## Check to see if there are persons with visits 2 or 3 and no visit 1 ##  
  
# Note that screening visits were only identified as screening in the redcap names for visits 2 and 3 but visit 1 was also screening  
  
visit2\_3gp <- clinical\_umd %>%   
 group\_by(field\_subj\_id, redcap\_event\_name) %>%   
 summarise(n = n())  
  
visit2\_3ID <- visit2\_3gp$field\_subj\_id[grepl('^screen', visit2\_3gp$redcap\_event\_name)]  
  
no\_visit2\_3ID <- unique(visit2\_3gp$field\_subj\_id[!grepl('^screen', visit2\_3gp$redcap\_event\_name)])  
  
# Subjects with Visit 2 or 3 who also have a Visit 1  
print(sum(visit2\_3ID %in% visit1\_id))

## [1] 21

# Number of subjects with at least one visit 2 or 3 arm regardless of having a visit 1  
print(length(visit2\_3ID))

## [1] 21

## Subjects receiving second and third screening visits ##  
  
visit2\_3DT <- clinical\_umd %>%   
 filter(grepl('^screen', redcap\_event\_name)) # all screening visit 2 and 3 records i.e. screened more than once  
  
visit2\_3DT2 <- clinical\_umd %>%   
 filter(field\_subj\_id %in% visit2\_3DT$field\_subj\_id) # all records for subj with a visit 2 or 3  
  
visit2\_3DT21 <- visit2\_3DT2 %>%   
 filter(grepl(0, is\_rmmate)) # select non roommate referrals  
  
# nonrr\_enroll\_visit2\_3 contains the records for people who were not roommate referrals and had visits 2 or 3.  
nonrr\_enroll\_visit2\_3 <- visit2\_3DT21 %>%   
 left\_join(visit2\_3DT2, by = "field\_subj\_id") %>%   
 select(field\_subj\_id, redcap\_event\_name.y)  
  
# Subjects 114, 127, and 250 have more than one screen visits but not referred by roomate.   
# All of them got enrolled. 114 has 2 screen visits, 127 has 3 screen visits, and 250 has 2 screen visits  
  
# The following subjects had more than one screening visit but were not roommates.  
unique(visit2\_3DT21$field\_subj\_id)

## [1] 114 127 250

# List of subjects and visits for repeatedly screened persons who were not roommate referrals  
print(nonrr\_enroll\_visit2\_3)

## field\_subj\_id redcap\_event\_name.y  
## 1 114 visit\_1\_part\_a\_arm\_1  
## 2 114 screen\_visit\_2\_arm\_1  
## 3 114 g2\_run\_1\_arm\_1  
## 4 127 visit\_1\_part\_a\_arm\_1  
## 5 127 screen\_visit\_2\_arm\_1  
## 6 127 screen\_visit\_3\_arm\_1  
## 7 127 g2\_run\_1\_arm\_1  
## 8 250 visit\_1\_part\_a\_arm\_1  
## 9 250 screen\_visit\_2\_arm\_1  
## 10 250 g2\_run\_1\_arm\_1

visit2\_3DT3 <- visit2\_3DT2 %>%   
 filter(!grepl('^screen', redcap\_event\_name)) # all records except visit 2 or 3 records for subj with a visit 2 or 3  
  
# check number of records including screening visits  
# Number of subjects with a visit 2 or 3 and a visit 1 or were also renrolled  
print(sum(unique(visit2\_3DT2$field\_subj\_id) %in% unique(visit2\_3DT3$field\_subj\_id)))

## [1] 20

# Number of all subject with a visit 2 or 3  
print(length(unique(visit2\_3DT2$field\_subj\_id)))

## [1] 20

## Roommates ##  
  
# identify the roommate referal subject   
rr <- clinical\_umd %>%   
 select(field\_subj\_id, redcap\_event\_name, is\_rmmate, indx\_id)  
  
# rr1 gives the the list of all the subjects referred by roomate with only the visit1\_arm  
rr1 <- rr %>%   
 filter(rr$is\_rmmate == 1)  
  
# Total number of roommate referrals screened  
print(length(rr1$field\_subj\_id))

## [1] 46

# rr2 shows all the arms and all fields  
rr2 <- rr1 %>%   
 left\_join(rr, by = c('field\_subj\_id'))  
rr3 <- rr2 %>%   
 select(field\_subj\_id, redcap\_event\_name.y, is\_rmmate.y, indx\_id.y) # limits fields from rr2  
  
# The following roommate referred subjects were screened  
print(unique(rr3$field\_subj\_id))

## [1] 133 134 135 144 147 152 160 178 180 190 199 205 206 209 217 218 219  
## [18] 220 221 224 237 239 246 254 259 263 265 268 269 275 278 279 280 283  
## [35] 286 295 307 312 324 325 330 332 334 335 354 362

rr4 <- rr3 %>%   
 filter(grepl('^g2', redcap\_event\_name.y))  
  
# The number of roommate referrals enrolled was  
print(length(rr4$field\_subj\_id))

## [1] 4

# The following roommate referred subjects were enrolled  
print(unique(rr4$field\_subj\_id))

## [1] 178 237 265 335

rr5 <- rr4 %>%   
 left\_join(rr3, by = c('field\_subj\_id')) %>%   
 select(field\_subj\_id, redcap\_event\_name.y.y, is\_rmmate.y.y, indx\_id.y.y)  
  
# Roommate referred subjects who were enrolled and their index ID  
print(rr5)

## field\_subj\_id redcap\_event\_name.y.y is\_rmmate.y.y indx\_id.y.y  
## 1 178 visit\_1\_part\_a\_arm\_1 1 143  
## 2 178 g2\_run\_1\_arm\_1 NA NA  
## 3 237 visit\_1\_part\_a\_arm\_1 1 223  
## 4 237 g2\_run\_1\_arm\_1 NA NA  
## 5 265 visit\_1\_part\_a\_arm\_1 1 255  
## 6 265 g2\_run\_1\_arm\_1 NA NA  
## 7 335 visit\_1\_part\_a\_arm\_1 1 328  
## 8 335 screen\_visit\_2\_arm\_1 NA NA  
## 9 335 g2\_run\_1\_arm\_1 NA NA

## Method 1, find the enrolled subjects and how many times they were enrolled ##  
  
enroll <- clinical\_umd %>%   
 filter(grepl('^g2\_run\_1', redcap\_event\_name)) %>%   
 group\_by(field\_subj\_id, redcap\_event\_name)  
  
# The total number of enrolled subjects was  
print(length(enroll$field\_subj\_id))

## [1] 178

enroll3 <- clinical\_umd %>%   
 filter(grepl('^g2\_run\_3', redcap\_event\_name)) %>%   
 group\_by(field\_subj\_id, redcap\_event\_name)  
  
# The total number of subjects that have three GII tests was  
print(length(enroll3$field\_subj\_id))

## [1] 27

enroll2 <- clinical\_umd %>%   
 filter(grepl('^g2\_run\_2', redcap\_event\_name)) %>%   
 group\_by(field\_subj\_id, redcap\_event\_name)  
  
# The total number of subjects that have two GII tests was  
print(length(enroll2$field\_subj\_id) - length(enroll3$field\_subj\_id))

## [1] 45

# The total number of subjects that have only one GII tests was  
print(length(enroll$field\_subj\_id) - length(enroll2$field\_subj\_id))

## [1] 106

# Subject 69 has 2 GII tests from redcap clinical record, but only have one GII record in the GII log on redcap, need further investigation.   
  
## More data exploration and manipulation ##  
  
# remove the screen arm from the data, leave the field\_subj\_id and redcap\_event\_name and mark n = 1 for each redcap\_event\_name  
tab0 <- clinical\_umd %>%   
 filter(!grepl('^screen', redcap\_event\_name)) %>%   
 group\_by(field\_subj\_id, redcap\_event\_name) %>%   
 summarise(n = n())  
  
# tab1: remove screen\_arm and left with visit\_1 and g2\_arm, and count the number of visit\_1 and g2\_arm  
tab1 <- clinical\_umd %>%   
 filter(!grepl('^screen', redcap\_event\_name)) %>%   
 group\_by(field\_subj\_id, date\_enroll) %>%   
 summarise(n = n())  
  
# tab1test, remove screen\_arm and only picked the field\_sub\_id and count the number of appearance(each subject should have at least 1 )  
tab1test <- clinical\_umd %>%   
 filter(!grepl('^screen', redcap\_event\_name)) %>%   
 group\_by(field\_subj\_id) %>%   
 summarise(n = n())  
  
# tab1new, add a colunm called enroll, and for n = 2 or > 2, means the subject need to have at least one time g2, so it is enrolled(1)  
# otherwise it is not enrolled(0). also add a new colunm called GII\_time, if if n = 2, means one g2\_arm, so GII\_time = n-1 = 1, apply   
# the same method for 2 and 3 GII\_times  
tab1new <- tab1test %>%   
 mutate(enroll = ifelse(n >= 2, 1, 0),   
 GII\_time = ifelse(n >= 2, n-1, 0))  
  
# tab1new1 sorted data for both enrolled and unenrolled subjects  
tab1new1 <- tab1new %>%   
 select(field\_subj\_id, enroll, GII\_time)  
names(tab1new1)[1] <- "subject\_id"  
  
# get the enrolldate correspond with the field\_subj\_id  
enrollDate <- clinical\_umd %>%   
 filter(grepl('^visit', redcap\_event\_name) & date\_enroll != '') %>%   
 select(field\_subj\_id, date\_enroll)  
  
# Number of duplicated enrolldate for the same subject id  
sum(duplicated(enrollDate$field\_subj\_id))

## [1] 0

# merge the enrolldate file with the ta1new which contained enroll, GII\_time, by subject\_id  
tab1link <- tab1new %>%   
 left\_join(enrollDate, by = 'field\_subj\_id') %>%   
 select(field\_subj\_id, enroll, GII\_time, date\_enroll)  
  
# check if tab1link has the same number of rows as tab1new,since the row numbers are the same, and there is no missing data for date\_enroll in tab1link  
# so the enrolldate match with the enrollcheck and GII\_times  
nrow(tab1link)

## [1] 355

nrow(tab1new)

## [1] 355

sum(tab1link$date\_enroll == '')

## [1] 0

names(tab1link)[1] <- "subject\_id"  
  
# subject which are enrolled  
tab2link <- tab1link %>%   
 filter(tab1link$enroll == 1)  
  
# subject came in only for screening visits  
tab3link <- tab1link %>%   
 filter(tab1link$enroll == 0)  
names(tab3link)[4] <- 'first\_visit\_date'  
  
#### READ in and work with the G2 Log Data ####  
  
g1 <- read.csv(g2\_in\_file)  
  
# g2\_in\_file was an object already read in earlier from 'EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/GII/EMITGIILogUMD2013.csv'  
# See above for where this was already read in with material from the "merge\_1-3.R-update.R" script  
  
# Sujbect 81 in GII file appear to have a collection\_2\_arm 1 but it actually is a one time GII subject  
sum((g1$start\_dt) == "")

## [1] 1

# Remove the second arm of subject 81 (REVISE TO ID ROWS WITHOUT START DATE)  
g2 <- filter(g1, !(g1$start\_dt) == "")  
  
# Pick the subject id and start date and order them  
g3 <- g2 %>%   
 select(subject\_id, start\_dt) %>%   
 arrange(subject\_id, start\_dt)  
names(g3)[2] <- "perform\_date"  
  
# Remove the duplicated and just save the first come in date  
g4 <- g3[!duplicated(g2$subject\_id), ]  
  
# Group the original subject and number them  
g5 <- g3 %>%   
 group\_by(subject\_id) %>%   
 summarise(n = n())  
  
# g6 tells us all many times the subject enrolled for gii study  
g6 <- g5 %>%   
 mutate(GII\_time = ifelse(n >= 1, n, 0))  
  
# Merge g4 (first come in date) wit the GII\_time  
g7 <- g4 %>%   
 left\_join(g6, by = 'subject\_id') %>%   
 select(subject\_id, GII\_time, perform\_date)  
  
# Change the start\_dt as date\_enroll  
names(g7)[3] <- "date\_enroll"  
  
#### Producing Enrollment Summary from Clinical Database and G2 Log Data ####  
  
# Merge the subject id with enroll time date\_enroll, sample perform date, only the enrolled the subject  
g8 <- g3 %>%   
 left\_join(g7, by = 'subject\_id') %>%   
 select(subject\_id, GII\_time, date\_enroll, perform\_date)  
  
# Check the gii subject list with the enrolled list from culture study  
m <- g7 %>%   
 left\_join(tab1link, by = c('subject\_id', 'GII\_time', 'date\_enroll'))

## Warning: Column `date\_enroll` joining factors with different levels,  
## coercing to character vector

sum(is.na(m$GII\_time))

## [1] 0

m2 <- tab1link %>%   
 left\_join(g8, by = 'subject\_id', 'GII\_time') %>%   
 select(subject\_id, enroll, perform\_date)  
  
## Write out the enrollment summary ##  
write.csv(m2, "EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/Enrollment\_Summary.csv")  
  
#### \*\*\*\* Using Script: Jing Yan's "field database with redcap culture.R" \*\*\*\* ####   
  
###  
## Original file information:  
  
# "field data base check with redcap culture.R"  
# by Jing Yan  
# December 16, 2015  
# Purpose:check if all the redcap culture sample id match with the field id   
  
###  
  
#### READ in and work with FIELD SAMPLE DATABASE ####  
  
a <- read.csv(field\_db\_in\_file, as.is = T)  
  
# field\_db\_in\_file was already read in earlier from 'EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/EMIT UMD Field\_db/field\_db.csv'  
  
names(a)

## [1] "SUBJECT\_IDENTIFIER" "SAMPLE\_ID" "COLLECTION\_DT"   
## [4] "ID" "TYPE\_NAME" "UNIT\_NAME"   
## [7] "RAPID\_TEST"

a1 <- a %>%   
 select(SUBJECT\_IDENTIFIER, SAMPLE\_ID, COLLECTION\_DT, TYPE\_NAME)  
names(a1)[2] = "sample\_id"  
names(a1)[3] = "Dates\_a"  
names(a1)[4] = "Sample.Type"  
  
# Add a new column as newdate which the same as collection\_dt but with format m/d/y  
a2 <- a1 %>%   
 mutate(newdate = as.Date(Dates\_a, format = '%m/%d/%Y'))  
  
#### READ in and work with UMD SAMPLES DATABASE (REDCAP DATA) ####  
  
# Read in redcap\_culture data  
b <- read.csv(sample\_in\_file, as.is = T)  
  
# sample\_in\_file is an object already read in from 'EMIT\_UMD\_Natural\_Infection/UMD\_Raw\_Data/REDCAP/EMITUMDSamples2013\_DATA.csv'  
  
#add a new colunm named as new date which is the same with date of sample collection  
b2 <- b %>%   
 mutate(newdate = as.Date(dt\_visit, format = '%m/%d/%Y'))  
  
#b3 proves that there is no missing date of sample collection in the redcap culture file  
b3 <- b2 %>%   
 filter(dt\_visit != '') %>%  
 rename(Sample.Type = sample\_type)  
  
m <- b3 %>%   
 left\_join(a2, by = c('sample\_id', 'newdate', 'Sample.Type'))  
  
m1 <- m %>%   
 select(sample\_id, newdate, Sample.Type, Dates\_a)  
  
m11 <- m1 %>%   
 select(sample\_id, newdate, Sample.Type)  
  
sum(is.na(m1$Dates\_a))

## [1] 1

m2 <- m1 %>%   
 filter(!grepl('.', Dates\_a))  
  
# This result suggests that all the redcap culture sample IDs and sample types and enrolled data match with the field ID data ...  
# ... Only 237\_6 was included in the field data base but not the redcap culture data ...  
# ... 237 is an enrolled subject for 1 time, it should not have a \_6 sample  
# From the above code we know all the culture redcap sample ID can be found in field ID, not sure if all the field ID can be found in culture redcap  
  
m3 <- a2 %>%   
 left\_join(b3, by = c('sample\_id', 'newdate', 'Sample.Type'))  
  
m4 <- m3 %>%   
 select(sample\_id, newdate, Sample.Type)  
  
sum(is.na(m4$Date.of.sample.collection))

## [1] 0

m5 <- m4 %>%   
 filter(!grepl('.', newdate))  
# m5 are the samples that were included in the field\_id but not included in the redcap culture  
  
# There is no output for this section - rather this is part of the data checking and exploratory analysis.   
# A report can be generated from the objects in this piece of the script if desired.   
  
#### \*\*\*\* 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('EMIT\_UMD\_Natural\_Infection/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  
fluAnp2 <- fluAnp2[order(fluAnp2$Experiment),]  
fluAnp2 <- fluAnp2 %>%   
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
fluAnp3 <- fluAnp2 %>%  
 group\_by(date) %>%  
 mutate(avgdiff = mean(dif))  
fluAnp3$cfactor <- 10^((fluAnp3$avgdiff) / 3.346)  
fluAnp3 <- fluAnp3 %>%   
 distinct(date, cfactor)  
fluAnp4 <- fluAnp3 %>%   
 select(date, cfactor)  
  
saveRDS(fluAnp4, "EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/fluA\_1np\_calibration.RDS")  
  
#### READ in \*"2016.08.05 1st visit NP swab FluB quant.csv"\* ####  
fluBnp <- read.csv('EMIT\_UMD\_Natural\_Infection/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  
fluBnp2 <- fluBnp2[order(fluBnp2$Experiment), ]  
fluBnp2 <-fluBnp2 %>%   
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
fluBnp3 <- fluBnp2 %>%   
 group\_by(date) %>%  
 mutate(avgdiff = mean(dif))  
fluBnp3$cfactor <- 10^((fluBnp3$avgdiff) / 3.297)  
fluBnp3 <- fluBnp3 %>%   
 distinct(date, cfactor)  
fluBnp4 <- fluBnp3 %>%  
 select(date, cfactor)  
  
saveRDS(fluBnp4, "EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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)  
fluA4 <- fluA3 %>%  
 select(date, cfactor)  
  
saveRDS(fluA4, "EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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)  
fluB4 <- fluB3 %>%  
 select(date, cfactor)  
  
saveRDS(fluB4, "EMIT\_UMD\_Natural\_Infection/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("EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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))  
pcr\_A1 <- rename(pcr\_A1, 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('EMIT\_UMD\_Natural\_Infection/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))  
pcr\_A2 <- rename(pcr\_A2, 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('EMIT\_UMD\_Natural\_Infection/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))  
pcr\_A3 <- rename(pcr\_A3, 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')  
pcr\_A <- pcr\_A %>% mutate(copy.num = copies.in\*80)  
  
# 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 <- pcr\_A[order(pcr\_A$subject.id), ]  
pcr\_A <- pcr\_A %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
pcr\_Afinal <- left\_join(pcr\_A,fluAcali,by = 'date')  
pcr\_Afinal$virus.copies <- pcr\_Afinal$copy.num\*pcr\_Afinal$cfactor  
  
#### READ in "fluB\_calibration.RDS" ####  
fluBcali <- readRDS("EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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('EMIT\_UMD\_Natural\_Infection/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."

pcr\_B1 <- pcr\_B1 %>%   
 filter(Ct..dRn. != 'Reference')  
pcr\_B1 <- pcr\_B1 %>%   
 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))  
pcr\_B1 <- rename(pcr\_B1, 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] 392

# 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('EMIT\_UMD\_Natural\_Infection/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)  
pcr\_B2 <- rename(pcr\_B2, 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\*411)  
  
# Number of rows of total (gii and 2rd or 3rd NP samples) influenza B PCR data  
nrow(pcr\_B)

## [1] 740

# 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$subject.id), ]  
pcr\_B <- pcr\_B %>%   
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
pcr\_Bfinal <- left\_join(pcr\_B, fluBcali, by = 'date')  
pcr\_Bfinal$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$subject.id <- as.numeric(total.pcr$subject.id)

## Warning: NAs introduced by coercion

total.pcr <- total.pcr[order(total.pcr$subject.id),]  
total.pcr <- rename(total.pcr, subject.id = subject.id)  
total.pcr$Well.Name[total.pcr$Well.Name == '42\_16'] <- '42\_18'  
total.pcr$Well.Name[total.pcr$Well.Name == '2110\_2'] <- '210\_2'  
total.pcr$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'  
  
# Number of rows of total (gii and 2rd or 3rd NP samples) PCR data  
nrow(total.pcr)

## [1] 1861

# 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 interrupt the flow of the: script to insert a section from 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('EMIT\_UMD\_Natural\_Infection/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'  
  
#### READ in and work with "2016.06.17 1st visit NP swab subtyping II.csv" ####  
  
part2 <- read.csv('EMIT\_UMD\_Natural\_Infection/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)  
  
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'   
  
# Write out this finalsubtype  
saveRDS(finalsubtype, file = "EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_subtypes.RDS")  
  
#### Merge the PCR data with sample virus subtype ####  
  
flu.types <- readRDS("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_subtypes.RDS")  
flu.types <- select(flu.types, subject.id, type.inf)  
  
# incldue subtype in the file  
includetype <- inner\_join(total.pcr, flu.types, by = "subject.id")  
includetype <- rename(includetype, sample.id = Well.Name)  
includetype1 <- includetype %>%   
 filter(type.inf == 'Negative')  
  
#### Merge PCR data with sample type ####  
  
allsamples <- readRDS("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_samples.cc.RDS")  
sampletype <- allsamples %>%   
 select(subject.id, sample.id, sample.type)  
  
#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  
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"))  
  
# 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 17. 2015.7.29.013.243 2012-13 samples FluA\_B and FluA PCR  
## 3 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 4 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 5 18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 6 15. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 7 15. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 8 18. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 9 10. 2015.08.07.014.104 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 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 15 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 16 23. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 17 19. 2015.10.18.014.120 2012-2013 Samples PCR Flu A&B  
## 18 19. 2015.10.18.014.120 2012-2013 Samples PCR Flu A&B  
## 19 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 20 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 21 16. 2015.08.21.014.116 2012-2013 Samples PCR Flu A&B  
## 22 16. 2015.08.21.014.116 2012-2013 Samples PCR Flu A&B  
## 23 16. 2015.08.21.014.116 2012-2013 Samples PCR Flu A&B  
## 24 15. 2015.08.20.014.114 2012-2013 Samples PCR Flu A&B  
## 25 15. 2015.08.20.014.114 2012-2013 Samples PCR Flu A&B  
## 26 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 27 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 47\_11 44.85 47 B Negative Impactor 5 um NO mask  
## 3 52\_17 37.52 52 B Negative Impactor 5 um NO mask  
## 4 58\_2 35.6 58 B Negative Impactor 5 um NO mask  
## 5 58\_2 35.6 58 B Negative Impactor 5 um NO mask  
## 6 105\_3 34.05 105 A Negative GII condensate NO mask  
## 7 105\_3 33.55 105 A Negative GII condensate NO mask  
## 8 223\_9 37.81 223 A Negative GII condensate NO mask  
## 9 223\_3 35.63 223 B Negative GII condensate NO mask  
## 10 223\_3 33.56 223 B Negative GII condensate NO mask  
## 11 223\_3 33.45 223 B Negative GII condensate NO mask  
## 12 223\_8 30.97 223 B Negative Nasopharyngeal swab  
## 13 223\_8 31.25 223 B Negative Nasopharyngeal swab  
## 14 223\_9 30.62 223 B Negative GII condensate NO mask  
## 15 223\_9 30.38 223 B Negative GII condensate NO mask  
## 16 226\_3 37.49 226 A Negative GII condensate NO mask  
## 17 231\_3 36.75 231 B Negative GII condensate NO mask  
## 18 231\_3 37.67 231 B Negative GII condensate NO mask  
## 19 231\_8 28.21 231 B Negative Nasopharyngeal swab  
## 20 231\_8 27.88 231 B Negative Nasopharyngeal swab  
## 21 327\_2 36.2 327 B Negative Impactor 5 um NO mask  
## 22 327\_3 31.79 327 B Negative GII condensate NO mask  
## 23 327\_3 31.9 327 B Negative GII condensate NO mask  
## 24 329\_2 36.82 329 B Negative Impactor 5 um NO mask  
## 25 329\_3 34.2 329 B Negative GII condensate NO mask  
## 26 365\_3 35.19 365 B Negative GII condensate NO mask  
## 27 365\_3 31.95 365 B Negative GII condensate NO mask

saveRDS(negative, "EMIT\_UMD\_Natural\_Infection/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("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/negative subtype sample with positive pcr.RDS")  
updatetype1 <- updatetype %>%   
 select(subject.id,type) %>% distinct(subject.id, type)  
updatetype2 <- updatetype1 %>%   
 filter(type == 'A')  
finalsubtype$type.inf[finalsubtype$subject.id == 105] <- 'Unsubtypable A'   
finalsubtype$type.inf[finalsubtype$subject.id == 226] <- 'Unsubtypable 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 == 223] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 231] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 327] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 329] <- 'B'  
finalsubtype$type.inf[finalsubtype$subject.id == 365] <- 'B'  
  
#### READ in "EMIT\_samples.cc.RDS" ####  
  
enrollcheck <- readRDS('EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_samples.cc.RDS') %>%   
 select(subject.id,enrolled) %>%   
 distinct(subject.id,enrolled) %>%   
 filter(enrolled == TRUE)  
  
finalenrolltype <- semi\_join(finalsubtype, enrollcheck, by = 'subject.id')  
finalenrolltype <- finalenrolltype[order(finalenrolltype$subject.id), ]  
finalenrolltype <- finalenrolltype %>%   
 select(subject.id, type.inf)  
  
saveRDS(finalenrolltype, "EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled.RDS")  
  
#### Check that all the negative subjects do not have any positive GII or 2nd/3rd np positive PCR samples ####  
  
flu.typesenroll <- readRDS("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled.RDS")  
flu.typesenroll <- select(flu.typesenroll,subject.id, type.inf)  
  
#incldue subtype in the file  
includetypeenroll <- inner\_join(total.pcr, flu.typesenroll, by = "subject.id") %>%  
 rename(sample.id = Well.Name)  
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  
  
finalenrollepositive <- finalenrolltype %>%   
 filter(!type.inf == 'Negative')  
  
saveRDS(finalenrollepositive, "EMIT\_UMD\_Natural\_Infection/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')  
  
#### READ in "EMIT\_subtypes\_enrolled\_positive.RDS" ####  
  
flu.typepositive <- readRDS("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_subtypes\_enrolled\_positive.RDS")  
  
#### Check the unmatched cases between sample virus subtype and PCR results ####  
  
flu.typepositive <- select(flu.typepositive, subject.id, type.inf)  
#incldue subtype in the file  
includetypepostive <- inner\_join(total.pcr, flu.typepositive, by = "subject.id") %>%   
 rename(sample.id = Well.Name) %>%  
 inner\_join(sampletype, by = c("subject.id", "sample.id"))  
  
# sampletypevirus <- inner\_join(includetypepostive, sampletype, by = c("subject.id", "sample.id"))  
# write.csv(sampletypevirus, "C:/Users/Jing/Desktop/output/sampletypevirus.csv")  
# no dual infection A and B found in the GII samples and 2rd or 3rd NP PCR data  
  
dual <- includetypepostive %>%   
 group\_by(subject.id) %>%   
 filter( type == 'A' & type == 'B')  
unmatched1 <- includetypepostive %>%   
 filter(type == 'B' & type.inf != 'B')  
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 18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 20 18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 21 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 22 16. 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 23 18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 24 18. 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 25 24. 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 26 24. 2015.10.15.014.118 Standard Curve & 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 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 30 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 31 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 32 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 33 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 34 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 35 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 36 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 37 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 38 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 39 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 40 19. 2015.08.04.014.100 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 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 46 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 47 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 48 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 49 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 50 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 51 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 52 10. 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 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 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 56 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 57 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 58 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 59 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 60 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 61 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 62 6. 2015.03.04.013.209 2012-13 samples FluB PCR  
## 63 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 64 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 65 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 66 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 67 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 68 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 69 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 70 12. 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 71 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 72 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 73 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 74 19. 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 75 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 76 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 77 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 78 11. 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 79 20. 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 80 20. 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 81 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 82 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 83 20. 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 84 20. 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 85 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 86 17. 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 87 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 88 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 89 25. 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 90 25. 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 91 25. 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 92 25. 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 93 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 94 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 95 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 96 14. 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 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\_2 NA 64 B NA  
## 20 64\_2 NA 64 B NA  
## 21 64\_3 NA 64 B NA  
## 22 64\_3 NA 64 B NA  
## 23 64\_3 NA 64 B NA  
## 24 64\_3 NA 64 B NA  
## 25 64\_7 NA 64 B NA  
## 26 64\_7 NA 64 B NA  
## 27 94\_2 NA 94 B NA  
## 28 94\_2 NA 94 B NA  
## 29 94\_3 NA 94 B NA  
## 30 94\_3 NA 94 B NA  
## 31 94\_6 NA 94 B NA  
## 32 94\_6 NA 94 B NA  
## 33 95\_2 NA 95 B NA  
## 34 95\_2 NA 95 B NA  
## 35 95\_3 NA 95 B NA  
## 36 95\_3 NA 95 B NA  
## 37 105\_2 NA 105 B NA  
## 38 105\_2 NA 105 B NA  
## 39 105\_3 NA 105 B NA  
## 40 105\_3 NA 105 B NA  
## 41 114\_12 NA 114 B NA  
## 42 114\_12 NA 114 B NA  
## 43 114\_7 NA 114 B NA  
## 44 114\_7 NA 114 B NA  
## 45 114\_9 NA 114 B NA  
## 46 114\_9 NA 114 B NA  
## 47 127\_14 NA 127 B NA  
## 48 127\_14 NA 127 B NA  
## 49 127\_15 NA 127 B NA  
## 50 127\_15 NA 127 B NA  
## 51 127\_18 NA 127 B NA  
## 52 127\_18 NA 127 B NA  
## 53 136\_11 NA 136 B NA  
## 54 136\_11 NA 136 B NA  
## 55 136\_2 NA 136 B NA  
## 56 136\_2 NA 136 B NA  
## 57 136\_3 NA 136 B NA  
## 58 136\_3 NA 136 B NA  
## 59 136\_8 NA 136 B NA  
## 60 136\_8 NA 136 B NA  
## 61 136\_9 NA 136 B NA  
## 62 136\_9 NA 136 B NA  
## 63 176\_11 NA 176 B NA  
## 64 176\_11 NA 176 B NA  
## 65 176\_12 NA 176 B NA  
## 66 176\_12 NA 176 B NA  
## 67 176\_14 NA 176 B NA  
## 68 176\_14 NA 176 B NA  
## 69 176\_16 NA 176 B NA  
## 70 176\_16 NA 176 B NA  
## 71 176\_2 NA 176 B NA  
## 72 176\_2 NA 176 B NA  
## 73 176\_3 NA 176 B NA  
## 74 176\_3 NA 176 B NA  
## 75 176\_7 NA 176 B NA  
## 76 176\_7 NA 176 B NA  
## 77 176\_9 NA 176 B NA  
## 78 176\_9 NA 176 B NA  
## 79 226\_2 NA 226 B NA  
## 80 226\_2 NA 226 B NA  
## 81 226\_2 NA 226 B NA  
## 82 226\_2 NA 226 B NA  
## 83 226\_3 NA 226 B NA  
## 84 226\_3 NA 226 B NA  
## 85 226\_3 NA 226 B NA  
## 86 226\_3 NA 226 B NA  
## 87 230\_11 NA 230 B NA  
## 88 230\_11 NA 230 B NA  
## 89 230\_2 NA 230 B NA  
## 90 230\_2 NA 230 B NA  
## 91 230\_3 NA 230 B NA  
## 92 230\_3 NA 230 B NA  
## 93 230\_8 NA 230 B NA  
## 94 230\_8 NA 230 B NA  
## 95 230\_9 NA 230 B NA  
## 96 230\_9 NA 230 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 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 20 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 21 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 22 2015.07.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 23 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 24 2015.7.30.013.245 2012-13 samples FluA\_B and FluA PCR  
## 25 2015.10.15.014.118 Standard Curve & 2012-2013 Samples PCR Flu A&B  
## 26 2015.10.15.014.118 Standard Curve & 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 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 30 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 31 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 32 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 33 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 34 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 35 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 36 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 37 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 38 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 39 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 40 2015.08.04.014.100 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 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 46 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 47 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 48 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 49 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 50 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 51 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 52 2015.08.07.014.104 2012-2013 Samples PCR Flu A&B  
## 53 2015.03.04.013.209 2012-13 samples FluB PCR  
## 54 2015.03.04.013.209 2012-13 samples FluB PCR  
## 55 2015.03.04.013.209 2012-13 samples FluB PCR  
## 56 2015.03.04.013.209 2012-13 samples FluB PCR  
## 57 2015.03.04.013.209 2012-13 samples FluB PCR  
## 58 2015.03.04.013.209 2012-13 samples FluB PCR  
## 59 2015.03.04.013.209 2012-13 samples FluB PCR  
## 60 2015.03.04.013.209 2012-13 samples FluB PCR  
## 61 2015.03.04.013.209 2012-13 samples FluB PCR  
## 62 2015.03.04.013.209 2012-13 samples FluB PCR  
## 63 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 64 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 65 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 66 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 67 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 68 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 69 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 70 2015.08.11.014.108 2012-2013 Samples PCR Flu A&B  
## 71 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 72 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 73 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 74 2015.08.04.014.100 2012-2013 Samples PCR Flu A&B  
## 75 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 76 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 77 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 78 2015.08.09.014.106 2012-2013 Samples PCR Flu A&B  
## 79 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 80 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 81 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 82 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 83 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 84 2015.08.05.013.248 2012-13 samples FluA\_B and FluA PCR  
## 85 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 86 2015.8.5.013.248 2012-13 samples FluA\_B and FluA PCR  
## 87 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 88 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 89 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 90 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 91 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 92 2016.03.17.013.286 2012-13 samples FluA\_B PCR  
## 93 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 94 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 95 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 96 2015.08.19.014.112 2012-2013 Samples PCR Flu A&B  
## 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 Impactor 5 um NO mask  
## 20 2.8832812 NA Unsubtypable A Impactor 5 um NO mask  
## 21 2.8832812 NA Unsubtypable A GII condensate NO mask  
## 22 2.8832812 NA Unsubtypable A GII condensate NO mask  
## 23 2.8832812 NA Unsubtypable A GII condensate NO mask  
## 24 2.8832812 NA Unsubtypable A GII condensate NO mask  
## 25 1.2888043 NA Unsubtypable A Nasopharyngeal swab  
## 26 1.2888043 NA Unsubtypable A Nasopharyngeal swab  
## 27 1.6774462 NA Pandemic H1 Impactor 5 um NO mask  
## 28 1.6774462 NA Pandemic H1 Impactor 5 um NO mask  
## 29 1.6774462 NA Pandemic H1 GII condensate NO mask  
## 30 1.6774462 NA Pandemic H1 GII condensate NO mask  
## 31 1.6774462 NA Pandemic H1 Nasopharyngeal swab  
## 32 1.6774462 NA Pandemic H1 Nasopharyngeal swab  
## 33 3.6584504 NA B and unsubtypable A Impactor 5 um NO mask  
## 34 3.6584504 NA B and unsubtypable A Impactor 5 um NO mask  
## 35 3.6584504 NA B and unsubtypable A GII condensate NO mask  
## 36 3.6584504 NA B and unsubtypable A GII condensate NO mask  
## 37 6.5092955 NA Unsubtypable A Impactor 5 um NO mask  
## 38 6.5092955 NA Unsubtypable A Impactor 5 um NO mask  
## 39 6.5092955 NA Unsubtypable A GII condensate NO mask  
## 40 6.5092955 NA Unsubtypable A GII condensate NO mask  
## 41 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 42 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 43 0.9444983 NA H3N2 Nasopharyngeal swab  
## 44 0.9444983 NA H3N2 Nasopharyngeal swab  
## 45 0.9444983 NA H3N2 GII condensate NO mask  
## 46 0.9444983 NA H3N2 GII condensate NO mask  
## 47 0.9444983 NA H3N2 Nasopharyngeal swab  
## 48 0.9444983 NA H3N2 Nasopharyngeal swab  
## 49 0.9444983 NA H3N2 GII condensate NO mask  
## 50 0.9444983 NA H3N2 GII condensate NO mask  
## 51 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 52 0.9444983 NA H3N2 Impactor 5 um NO mask  
## 53 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 54 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 55 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 56 1.4690044 NA H3N2 Impactor 5 um NO mask  
## 57 1.4690044 NA H3N2 GII condensate NO mask  
## 58 1.4690044 NA H3N2 GII condensate NO mask  
## 59 1.4690044 NA H3N2 Nasopharyngeal swab  
## 60 1.4690044 NA H3N2 Nasopharyngeal swab  
## 61 1.4690044 NA H3N2 GII condensate NO mask  
## 62 1.4690044 NA H3N2 GII condensate NO mask  
## 63 1.6774462 NA H3N2 Impactor 5 um NO mask  
## 64 1.6774462 NA H3N2 Impactor 5 um NO mask  
## 65 3.6584504 NA H3N2 Nasopharyngeal swab  
## 66 3.6584504 NA H3N2 Nasopharyngeal swab  
## 67 3.6584504 NA H3N2 GII condensate NO mask  
## 68 3.6584504 NA H3N2 GII condensate NO mask  
## 69 3.6584504 NA H3N2 Impactor 5 um NO mask  
## 70 3.6584504 NA H3N2 Impactor 5 um NO mask  
## 71 6.5092955 NA H3N2 Impactor 5 um NO mask  
## 72 6.5092955 NA H3N2 Impactor 5 um NO mask  
## 73 6.5092955 NA H3N2 GII condensate NO mask  
## 74 6.5092955 NA H3N2 GII condensate NO mask  
## 75 1.6774462 NA H3N2 Nasopharyngeal swab  
## 76 1.6774462 NA H3N2 Nasopharyngeal swab  
## 77 1.6774462 NA H3N2 GII condensate NO mask  
## 78 1.6774462 NA H3N2 GII condensate NO mask  
## 79 0.8392489 NA Unsubtypable A Impactor 5 um NO mask  
## 80 0.8392489 NA Unsubtypable A Impactor 5 um NO mask  
## 81 0.8392489 NA Unsubtypable A Impactor 5 um NO mask  
## 82 0.8392489 NA Unsubtypable A Impactor 5 um NO mask  
## 83 0.8392489 NA Unsubtypable A GII condensate NO mask  
## 84 0.8392489 NA Unsubtypable A GII condensate NO mask  
## 85 0.8392489 NA Unsubtypable A GII condensate NO mask  
## 86 0.8392489 NA Unsubtypable A GII condensate NO mask  
## 87 6.1303335 NA H3N2 and B Impactor 5 um NO mask  
## 88 6.1303335 NA H3N2 and B Impactor 5 um NO mask  
## 89 1.0268645 NA H3N2 and B Impactor 5 um NO mask  
## 90 1.0268645 NA H3N2 and B Impactor 5 um NO mask  
## 91 1.0268645 NA H3N2 and B GII condensate NO mask  
## 92 1.0268645 NA H3N2 and B GII condensate NO mask  
## 93 6.1303335 NA H3N2 and B Nasopharyngeal swab  
## 94 6.1303335 NA H3N2 and B Nasopharyngeal swab  
## 95 6.1303335 NA H3N2 and B GII condensate NO mask  
## 96 6.1303335 NA H3N2 and B GII condensate NO mask

unmatched2 <- includetypepostive %>%  
 filter( type == 'A' & type.inf == 'B')  
  
# Number of samples with positive ct value for A but stubtype for B  
nrow(unmatched2)

## [1] 79

includetypepostiveupdate <- anti\_join(includetypepostive, 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"))  
  
#### 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 <- full\_join(allpcrA, allpcrB, by = c('subject.id', 'Experiment', 'type.inf', 'sample.id', 'sample.type', "copies.in", 'copy.num', 'date', 'cfactor'))  
allPCR <- allPCR[order(allPCR$subject.id), ]  
allPCR <- allPCR %>%   
 filter(!sample.type == 'Throat Swab')  
# 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 <- anti\_join(allPCR, 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)  
allPCR.A <- rename(allPCR.A, 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)  
allPCR.B <- rename(allPCR.B, 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  
  
saveRDS(allPCRfinal, "EMIT\_UMD\_Natural\_Infection/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())  
  
# 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  
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: 108 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 98 more rows

#### \*\*\*\* 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)  
  
###  
#### 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("EMIT\_UMD\_Natural\_Infection/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("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/fluA\_1np\_calibration.RDS")  
  
#### READ in "2016.08.05 1st visit NP swab FluA quant.csv" ####  
npA <- read.csv('EMIT\_UMD\_Natural\_Infection/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')  
npA <- npA %>%  
 filter(!grepl('\_Low', Well.Name))  
npA <- npA %>%  
 filter(!grepl('NTC', Well.Name))  
npA <- npA %>%  
 filter(!grepl('Standard', Well.Type))  
npA <- npA %>%  
 filter(!grepl('High', Well.Name))  
npA <- npA %>%  
 filter(grepl('\_', Well.Name))  
npA <- npA %>%  
 select(-Well, -Well.Type, -Threshold..dRn.)  
npA <- rename(npA, 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))  
npA <- npA %>%   
 mutate(subject.id = gsub('\_A', '', subject.id))  
npA <- npA %>%   
 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\*80)  
npA <- npA[order(npA$subject.id), ]  
npA$result.type <- 'A'  
npA <- npA %>%  
 mutate(sample.id = gsub('\_A', '', Well.Name))  
npA <- npA %>%   
 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 <- rename(npA, Ct = Ct..dRn., type = result.type)  
npA <- npA %>%  
 select(-Well.Name)  
npA <- npA %>%  
 mutate(date = gsub('^[0-9]\*.', '', Experiment))  
npAfinal <- left\_join(npA, npAcali, by = 'date')  
#These samples were done by Jake and he did not put 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("EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/fluB\_1np\_calibration.RDS")  
  
#### READ in "2016.08.05 1st visit NP swab FluB quant.csv" ####  
npB <- read.csv('EMIT\_UMD\_Natural\_Infection/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')  
npB <- npB %>%  
 filter(!grepl('\_Low', Well.Name))  
npB <- npB %>%  
 filter(!grepl('NTC', Well.Name))  
npB <- npB %>%  
 filter(!grepl('Standard', Well.Type))  
npB <- npB %>%  
 filter(!grepl('High', Well.Name))  
npB <- npB %>%  
 filter(grepl('\_', Well.Name))  
npB <- npB %>%  
 select(-Well, -Well.Type, -Threshold..dRn.)  
npB <- rename(npB, 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))  
npB <- npB %>%  
 mutate(subject.id = gsub('\_B', '', subject.id))  
npB <- npB %>%  
 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\*411)  
npB <- npB[order(npB$subject.id),]  
npB$result.type <-'B'  
npB <- npB %>%  
 mutate(sample.id = gsub('\_B', '', Well.Name))  
npB <- npB %>%  
 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 <- rename(npB, Ct = Ct..dRn., type = result.type)  
npB <- npB %>%  
 select(-Well.Name)  
npB <- npB %>%  
 mutate(date = gsub('^[0-9]\*.', '',Experiment))  
npBfinal <- left\_join(npB ,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), ]  
  
npfirstpositive <- inner\_join(npfirst, flu.types, by = c("subject.id"))  
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)  
#Check if any replicates in the data and why  
check2 <- npfirstpositiveupdate %>%  
 group\_by(sample.id) %>%  
 summarise(n = n())  
check2 <- check2[order(check2$n), ]  
# samples that are duplicated at least once are 100\_1,104\_1,118\_1,123\_1,174\_1,230\_1,357\_1,81\_1,97\_1,95\_1  
#230 is dual infection has in both A and B, A has greater virus copies than B, 95 is dual infection in both A and B, B has greater virus copies than A   
npfirstpositiveupdate <- npfirstpositiveupdate[order(npfirstpositiveupdate$subject.id), ]  
  
npfirstpositiveupdate$sample.type <- 'Nasopharyngeal swab'  
  
saveRDS(npfirstpositiveupdate, "EMIT\_UMD\_Natural\_Infection/Curated Data/Cleaned Data/EMIT\_np\_quantity.RDS")