Raw Data Merge

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(googlesheets4)
library(qualtRics)
library(lubridate)
```

Note: the Qualtrics pretest data still has participant names, so this script cannot be fully executed without the Qualtrics credentials, which are stored elsewhere.

Load list of valid IDs

```
gs4_auth(path=Sys.getenv("SHINY_PROTOCOL"))

#gs4_auth(email="russell.g.almond@gmail.com", scopes="https://www.googleapis.com/auth/spreadsheets")
#gs4_deauth()

ValidIDs <- read_sheet("1lBpOLhNdSfOr1OtwC2cyqInB2npLbjHOKUMh4z_nzfM")$StudyID
```

```
## v Reading from "ValidIDs-Deidentified".
```

```
## v Range 'ValidIDs-Deidentified'.
```

```
AllIDs <- read_sheet("1BPkBMEE5DOtZ3MBt5C1ab4djddAOhUyAIPMiR4WRhBQ")
```

```
## v Reading from "UserIDs2".
```

```
## v Range 'Sheet1'.
```

```
AllIDs %>% filter(StudyID %in% ValidIDs) -> PPIDs
AllIDs %>% filter(!(StudyID %in% ValidIDs)) -> InvIDs
```

Survey Metadata

Experiment in trying to mine the QSF information to automagically build data dictionary.

```
qualtricsIDs <- c(PRE="SV_bkALP8OIYWnEgkK",ECT="SV_3BKybDb3f2v4QU6",
                  POT="SV_1MotNHu1dPEAJnM",Assent="SV_07esYssbhDccXs1",
          Consent="SV_8vsIO1klt7DSPMG", PosttestA="SV_aeBB1Tw7bYKeIqq",
```

```
          PosttestB="SV_4ISgzDcDm6BLiom", PretestA="SV_4IpHb99rMIEKtIq",
          PretestB="SV_6eTzhLhhnhXrPmK")
```

Question Titles

```
pretestQ <- survey_questions(qualtricsIDs["PRE"])
potQ <- survey_questions(qualtricsIDs["POT"])
ectQ <- survey_questions(qualtricsIDs["ECT"])
posttestB <- survey_questions(qualtricsIDs["PosttestB"])
```

```
pretestQuestions <- gsub("</?p>","",pretestQ$question)
names(pretestQuestions) <- pretestQ$qname
potQuestions <- gsub("</?p>","",potQ$question)
names(potQuestions) <- potQ$qname
ectQuestions <- gsub("</?p>","",ectQ$question)
names(ectQuestions) <- ectQ$qname
postBQuestions <- gsub("</?p>","",posttestB$question)
names(postBQuestions) <- posttestB$qname
```

Qualtrics QSF

```
preQSF <- jsonlite::fromJSON("PP_IES_Pretest.qsf",FALSE)
potQSF <- jsonlite::fromJSON("PP_IES_Posttest_POT (2).qsf",FALSE)
ectQSF <- jsonlite::fromJSON("PP_IES_Posttest_ECT.qsf",FALSE)
postBQSF <- jsonlite::fromJSON("Posttest_B_Summer_Camp.qsf",FALSE)
```

Fetch the elements marked "SQ". Use the DataExportTag as the name, this should allow searching the metadata by name.

```
preSQ <- lapply(preQSF$SurveyElements[which(sapply(preQSF$SurveyElements,
                                    function(el) el$Element=="SQ"))],
                function(sq) sq$Payload)
names(preSQ) <- sapply(preSQ, function(q) q$DataExportTag)
potSQ <- lapply(potQSF$SurveyElements[which(sapply(potQSF$SurveyElements,
                                    function(el) el$Element=="SQ"))],
                function(sq) sq$Payload)
names(potSQ) <- sapply(potSQ, function(q) q$DataExportTag)
ectSQ <- lapply(ectQSF$SurveyElements[which(sapply(ectQSF$SurveyElements,
                                    function(el) el$Element=="SQ"))],
                function(sq) sq$Payload)
names(ectSQ) <- sapply(ectSQ, function(q) q$DataExportTag)
postBSQ <- lapply(postBQSF$SurveyElements[which(sapply(postBQSF$SurveyElements,
                                    function(el) el$Element=="SQ"))],
                function(sq) sq$Payload)
names(postBSQ) <- sapply(postBSQ, function(q) q$DataExportTag)
```

Names for IMI questions

```
choices <- ectSQ$IMI$Choices
IMI.choices <- rep(NA_character_, max(as.numeric(names(choices))))
for (choi in names(choices)) {
  IMI.choices[as.numeric(choi)] <- choices[[choi]]$Display
}
```

```
chorder <- as.numeric(unlist(ectSQ$IMI$ChoiceOrder))
IMI.stems <- IMI.choices[chorder]
names(IMI.stems) <- paste("IMI",1L:length(IMI.stems),sep="_")
IMI.stems <- gsub("\t","",IMI.stems)  # Clean extra tabs.
```

Names for PA questions.

PA questions only appear on ECD form.

```
PA.names <- paste("PA",1:7,sep="")
PA.stems <- sapply(ectSQ[PA.names],function (sq) sq$QuestionText)
PA.labels <- sapply(ectSQ[PA.names],function(sq) unlist(sq$Choices))
PA.labels
```

```
##           PA1                           PA2
## 1.Display "Strongly agree"              "Strongly agree"
## 2.Display "Somewhat agree"              "Somewhat agree"
## 3.Display "Neither agree nor disagree" "Neither agree nor disagree"
## 4.Display "Somewhat disagree"           "Somewhat disagree"
## 5.Display "Strongly disagree"           "Strongly disagree"
##           PA3                           PA4
## 1.Display "Strongly agree"              "Strongly agree"
## 2.Display "Somewhat agree"              "Somewhat agree"
## 3.Display "Neither agree nor disagree" "Neither agree nor disagree"
## 4.Display "Somewhat disagree"           "Somewhat disagree"
## 5.Display "Strongly disagree"           "Strongly disagree"
##           PA5                           PA6
## 1.Display "Strongly agree"              "Strongly agree"
## 2.Display "Somewhat agree"              "Somewhat agree"
## 3.Display "Neither agree nor disagree" "Neither agree nor disagree"
## 4.Display "Somewhat disagree"           "Somewhat disagree"
## 5.Display "Strongly disagree"           "Strongly disagree"
##           PA7
## 1.Display "Strongly agree"
## 2.Display "Somewhat agree"
## 3.Display "Neither agree nor disagree"
## 4.Display "Somewhat disagree"
## 5.Display "Strongly disagree"
```

Names for the ethnicity questions

```
ethnames <- preSQ$Ethnicity$Choices
ethcolnames <- paste("Ethnicity",names(ethnames),sep="_")
ethnames <- unlist(ethnames)
ethnames <- ethnames[grepl("Display",names(ethnames))]
names(ethnames) <- ethcolnames
ethnames
```

```
##                          Ethnicity_1                          Ethnicity_4
##    "American Indian or Alaska Native"                              "Asian"
##                          Ethnicity_5                          Ethnicity_6
##          "Black or African American"                             "Hispanic"
##                          Ethnicity_7                          Ethnicity_8
## "Native Hawaiian or Pacific Islander"                              "White"
```

```
##                          Ethnicity_9                         Ethnicity_10
##                    "Other (enter)"                "Prefer not to say"
```

Extract the keys

```
preKey <- sapply(preSQ,function(sq) {
  if (is.null(sq$GradingData) || length(sq$GradingData)==0L) return (NA)
  as.numeric(sq$GradingData[[1]]$ChoiceID)
})
preKey <- preKey[!is.na(preKey)]
potKey <- sapply(potSQ,function(sq) {
  if (is.null(sq$GradingData) || length(sq$GradingData)==0L) return (NA)
  as.numeric(sq$GradingData[[1]]$ChoiceID)
})
potKey <- potKey[!is.na(potKey)]
ectKey <- sapply(ectSQ,function(sq) {
  if (is.null(sq$GradingData) || length(sq$GradingData)==0L) return (NA)
  as.numeric(sq$GradingData[[1]]$ChoiceID)
})
ectKey <- ectKey[!is.na(ectKey)]
postBKey <- sapply(postBSQ,function(sq) {
  if (is.null(sq$GradingData) || length(sq$GradingData)==0L) return (NA)
  as.numeric(sq$GradingData[[1]]$ChoiceID)
})
postBKey <- postBKey[!is.na(postBKey)]
```

```
keyID <- "1fagyGKc30Fx2ORwcWoULJ_dYZWqwPwJWCc-xo1pfV2o"
physics.key <- read_sheet(keyID,"Physics")
```

```
## v Reading from "KeysAndCodes".
```

```
## v Range ''Physics''.
```

```
IMI.scales <- read_sheet(keyID,"IMI")
```

```
## v Reading from "KeysAndCodes".
```

```
## v Range ''IMI''.
```

```
PA.rev <- read_sheet(keyID,"PA")
```

```
## v Reading from "KeysAndCodes".
```

```
## v Range ''PA''.
```

I have key information from two sources, Qualtrics and external file. Check to see if they are the same.

```
formA.keys <- filter(physics.key,Form=="A") %>% select(ID,Key)
formA.keys$prekey <- preKey[formA.keys$ID]
formA.keys$potKey <- potKey[formA.keys$ID]
formA.keys$ectKey <- ectKey[formA.keys$ID]
head(formA.keys)
```

```
## # A tibble: 6 x 5
##   ID    Key   prekey potKey ectKey
##   <chr> <chr>  <dbl>  <dbl>  <dbl>
## 1 A-NQ1 a          1      1      1
## 2 A-NQ2 b          2      2      2
```

```
## 3 A-NQ3 b          2      2      2
## 4 A-NQ4 a          1      1      1
## 5 A-NQ5 c          3      3      3
## 6 A-NQ6 c          3      3      3
```

No key mismatches, but column labeling issue. For some reason, `A-FQ10` is called `AFQ10` in pretest.

```
choicesToFactor <- function (sq, col) {
  levels <- as.numeric(names(sq$Choices))
  labels <- sapply(sq$Choices, function(c) c$Display)
  factor(col,levels,labels)
}
```

Load The Qualtrics Data

Exclude metadata fields, fields containing PID and "DO" fields which give presentation ordering.

```
pretest <- fetch_survey(qualtricsIDs["PRE"],force=TRUE,
                        label=FALSE,convert=FALSE) %>%
  filter(ID %in% ValidIDs) %>%
  select(!(StartDate:IPAddress)) %>%
  select(!(ResponseId:UserLanguage)) %>%
  select(!(First:Q80)) %>%
  select(!contains("DO"))
```

```
##   |                                                                      |
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   StartDate = col_datetime(format = ""),
##   EndDate = col_datetime(format = ""),
##   IPAddress = col_character(),
##   RecordedDate = col_datetime(format = ""),
##   ResponseId = col_character(),
##   RecipientLastName = col_logical(),
##   RecipientFirstName = col_logical(),
##   RecipientEmail = col_logical(),
##   ExternalReference = col_logical(),
##   DistributionChannel = col_character(),
##   UserLanguage = col_character(),
##   ID = col_character(),
##   First = col_character(),
##   Last = col_character(),
##   Q80 = col_character(),
##   Age = col_character(),
##   Ethnicity_9_TEXT = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
names(pretest)[5] <- "StudyID"
pretest <- filter(pretest,!is.na(StudyID))

ecttest <- fetch_survey(qualtricsIDs["ECT"],force=TRUE,
                        label=FALSE, convert=FALSE) %>%
```

```
  filter(UserID1 %in% ValidIDs) %>%
  select(!(StartDate:IPAddress)) %>%
  select(!(ResponseId:UserLanguage)) %>%
  select(!contains("DO"))
```

```
##   |                                                                      |
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   StartDate = col_datetime(format = ""),
##   EndDate = col_datetime(format = ""),
##   IPAddress = col_character(),
##   RecordedDate = col_datetime(format = ""),
##   ResponseId = col_character(),
##   RecipientLastName = col_logical(),
##   RecipientFirstName = col_logical(),
##   RecipientEmail = col_logical(),
##   ExternalReference = col_logical(),
##   DistributionChannel = col_character(),
##   UserLanguage = col_character(),
##   UserID1 = col_character(),
##   Q90 = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
names(ecttest)[5] <- "StudyID"
ecttest <- filter(ecttest,!is.na(StudyID))

pottest <- fetch_survey(qualtricsIDs["POT"], force=TRUE,
                        label=FALSE, convert=FALSE) %>%
  filter(UserID1 %in% ValidIDs) %>%
  select(!(StartDate:IPAddress)) %>%
  select(!(ResponseId:UserLanguage)) %>%
  select(!contains("DO"))
```

```
##   |                                                                      |
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   StartDate = col_datetime(format = ""),
##   EndDate = col_datetime(format = ""),
##   IPAddress = col_character(),
##   RecordedDate = col_datetime(format = ""),
##   ResponseId = col_character(),
##   RecipientLastName = col_logical(),
##   RecipientFirstName = col_logical(),
##   RecipientEmail = col_logical(),
##   ExternalReference = col_logical(),
##   DistributionChannel = col_character(),
##   UserLanguage = col_character(),
##   UserID1 = col_character(),
##   Q90 = col_character()
```

```
## )
## i Use `spec()` for the full column specifications.
```

```
names(pottest)[5] <- "StudyID"
pottest <- filter(pottest,!is.na(StudyID))
```

Clean duplicates.

```
dupes.pre <- pretest$StudyID[duplicated(pretest$StudyID)]
"Pretest:"
```

```
## [1] "Pretest:"
```

```
dupes.pre
```

```
##  [1] "B2035" "F3733" "D3508" "F1450" "A1663" "F3069" "B2114" "D1988" "C1225"
## [10] "C1225" "D1371" "A0471" "B1731" "B1146" "C2473" "B1146"
```

```
"ECT:"
```

```
## [1] "ECT:"
```

```
dupes.ect <- ecttest$StudyID[duplicated(ecttest$StudyID)]
dupes.ect
```

```
## [1] "A3799" "B0088" "C3586" "D0314"
```

```
"POT:"
```

```
## [1] "POT:"
```

```
dupes.pot <- pottest$StudyID[duplicated(pottest$StudyID)]
dupes.pot
```

```
## [1] "A2079" "D3654" "F1562" "B1731" "B2754" "D2697"
```

POT

```
pottest %>% filter(StudyID%in%dupes.pot)
```

```
## # A tibble: 12 x 45
##    Progress `Duration (in seconds)` Finished RecordedDate        StudyID IMI_1
##       <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <dbl>
##  1      100                     462        1 2022-11-30 10:38:18 D3654       6
##  2      100                     198        1 2022-11-30 13:56:13 B2754       4
##  3      100                    1111        1 2022-12-01 12:09:17 B1731       5
##  4      100                     390        1 2022-12-02 09:26:02 A2079       4
##  5      100                     574        1 2022-12-02 09:35:58 A2079       3
##  6      100                     854        1 2022-12-02 10:14:13 D3654       4
##  7      100                     144        1 2022-12-02 12:22:54 F1562       7
##  8      100                     122        1 2022-12-02 12:25:21 F1562       7
##  9       78                     550        0 2022-12-07 10:42:51 D2697       4
## 10       41                     625        0 2022-12-07 12:25:06 B1731       5
## 11        3                      27        0 2022-12-07 13:23:34 B2754      NA
## 12        3                    1250        0 2022-12-07 14:00:14 D2697      NA
## # i 39 more variables: IMI_2 <dbl>, IMI_3 <dbl>, IMI_4 <dbl>, IMI_5 <dbl>,
## #   IMI_6 <dbl>, IMI_7 <dbl>, IMI_8 <dbl>, IMI_9 <dbl>, IMI_10 <dbl>,
## #   IMI_11 <dbl>, IMI_12 <dbl>, IMI_13 <dbl>, IMI_14 <dbl>, IMI_15 <dbl>,
## #   IMI_16 <dbl>, IMI_17 <dbl>, IMI_18 <dbl>, `A-NQ9` <dbl>, `A-NQ10` <dbl>,
```

```
## #   `A-NQ11` <dbl>, `A-NQ12` <dbl>, `A-NQ13` <dbl>, `A-NQ14` <dbl>,
## #   `A-FQ9` <dbl>, `A-FQ10` <dbl>, `A-FQ11` <dbl>, `A-FQ12` <dbl>,
## #   `A-FQ13` <dbl>, `A-FQ14` <dbl>, Q90 <chr>, Q74 <dbl>, PA1 <dbl>, ...
```

```
pottest <- filter(pottest,!(StudyID %in% dupes.pot) | Finished)
dupes.pot1 <- pottest$StudyID[duplicated(pottest$StudyID)]
pottest %>% filter(StudyID%in%dupes.pot1)
```

```
## # A tibble: 6 x 45
##   Progress `Duration (in seconds)` Finished RecordedDate        StudyID IMI_1
##      <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <dbl>
## 1      100                     462        1 2022-11-30 10:38:18 D3654       6
## 2      100                     390        1 2022-12-02 09:26:02 A2079       4
## 3      100                     574        1 2022-12-02 09:35:58 A2079       3
## 4      100                     854        1 2022-12-02 10:14:13 D3654       4
## 5      100                     144        1 2022-12-02 12:22:54 F1562       7
## 6      100                     122        1 2022-12-02 12:25:21 F1562       7
## # i 39 more variables: IMI_2 <dbl>, IMI_3 <dbl>, IMI_4 <dbl>, IMI_5 <dbl>,
## #   IMI_6 <dbl>, IMI_7 <dbl>, IMI_8 <dbl>, IMI_9 <dbl>, IMI_10 <dbl>,
## #   IMI_11 <dbl>, IMI_12 <dbl>, IMI_13 <dbl>, IMI_14 <dbl>, IMI_15 <dbl>,
## #   IMI_16 <dbl>, IMI_17 <dbl>, IMI_18 <dbl>, `A-NQ9` <dbl>, `A-NQ10` <dbl>,
## #   `A-NQ11` <dbl>, `A-NQ12` <dbl>, `A-NQ13` <dbl>, `A-NQ14` <dbl>,
## #   `A-FQ9` <dbl>, `A-FQ10` <dbl>, `A-FQ11` <dbl>, `A-FQ12` <dbl>,
## #   `A-FQ13` <dbl>, `A-FQ14` <dbl>, Q90 <chr>, Q74 <dbl>, PA1 <dbl>, ...
```

For A2079 and D3654 take the row with non-missing PA1 For F1562 Flip a coin.

```
pottest <- filter(pottest,!(StudyID %in% c("A2079","D3654")) | !is.na(PA1))
dupes.pot2 <- pottest$StudyID[duplicated(pottest$StudyID)]
for (dupID in dupes.pot2) {
  dupindex <- which(pottest$StudyID==dupID)
  dropindix <- sample(dupindex,length(dupindex)-1,replace=FALSE)
  pottest <- pottest[-dropindix,]
}
anyDuplicated(pottest$StudyID)
```

```
## [1] 0
```

ECT

```
filter(ecttest, StudyID %in% dupes.ect) %>% arrange(StudyID)
```

```
## # A tibble: 8 x 52
##   Progress `Duration (in seconds)` Finished RecordedDate        StudyID IMI_1
##      <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <dbl>
## 1      100                     429        1 2022-11-30 09:39:04 A3799       7
## 2        2                       4        0 2022-12-07 09:46:18 A3799      NA
## 3      100                     524        1 2022-12-02 08:45:03 B0088       5
## 4        2                      20        0 2022-12-09 08:13:31 B0088      NA
## 5      100                     653        1 2022-11-30 09:42:51 C3586       6
## 6        2                     185        0 2022-12-09 09:05:05 C3586      NA
## 7      100                     194        1 2022-12-02 13:57:23 D0314       5
## 8       84                     229        0 2022-12-09 13:55:01 D0314       6
## # i 46 more variables: IMI_2 <dbl>, IMI_3 <dbl>, IMI_4 <dbl>, IMI_5 <dbl>,
## #   IMI_6 <dbl>, IMI_7 <dbl>, IMI_8 <dbl>, IMI_9 <dbl>, IMI_10 <dbl>,
## #   IMI_11 <dbl>, IMI_12 <dbl>, IMI_13 <dbl>, IMI_14 <dbl>, IMI_15 <dbl>,
```

```
## #   IMI_16 <dbl>, IMI_17 <dbl>, IMI_18 <dbl>, `A-NQ1` <dbl>, `A-NQ2` <dbl>,
## #   `A-NQ3` <dbl>, `A-NQ4` <dbl>, `A-NQ5` <dbl>, `A-NQ6` <dbl>, `A-NQ7` <dbl>,
## #   `A-NQ8` <dbl>, `A-NQ12` <dbl>, `A-NQ13` <dbl>, `A-NQ14` <dbl>,
## #   `A-FQ1` <dbl>, `A-FQ2` <dbl>, `A-FQ3` <dbl>, `A-FQ4` <dbl>, ...
```

Select the ones which are finished.

```
ecttest <- filter(ecttest,!(StudyID %in% dupes.ect) | Finished)
ecttest$StudyID[duplicated(ecttest$StudyID)]
```

```
## character(0)
```

PRE

```
filter(pretest, StudyID %in% dupes.pre) %>%
  arrange(StudyID) %>% select(Progress, Finished, StudyID)
```

```
## # A tibble: 30 x 3
##    Progress Finished StudyID
##       <dbl>    <dbl> <chr>
##  1      100        1 A0471
##  2        3        0 A0471
##  3      100        1 A1663
##  4      100        1 A1663
##  5       82        0 B1146
##  6      100        1 B1146
##  7        2        0 B1146
##  8       88        0 B1731
##  9      100        1 B1731
## 10      100        1 B2035
## # i 20 more rows
```

```
pretest <- filter(pretest,!(StudyID %in% dupes.pre) | Finished)
dupes.pre1 <- pretest$StudyID[duplicated(pretest$StudyID)]
filter(pretest,StudyID %in% dupes.pre1)
```

```
## # A tibble: 2 x 47
##   Progress `Duration (in seconds)` Finished RecordedDate        StudyID Age
##      <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <chr>
## 1      100                    1292        1 2022-11-28 12:15:13 A1663   15
## 2      100                     643        1 2022-11-28 12:30:57 A1663   15
## # i 41 more variables: Sex <dbl>, Ethnicity_1 <dbl>, Ethnicity_4 <dbl>,
## #   Ethnicity_5 <dbl>, Ethnicity_6 <dbl>, Ethnicity_7 <dbl>, Ethnicity_8 <dbl>,
## #   Ethnicity_9 <dbl>, Ethnicity_10 <dbl>, Ethnicity_9_TEXT <chr>,
## #   Gaming <dbl>, Physics <dbl>, `A-NQ1` <dbl>, `A-NQ2` <dbl>, `A-NQ3` <dbl>,
## #   `A-NQ4` <dbl>, `A-NQ5` <dbl>, `A-NQ6` <dbl>, `A-NQ7` <dbl>, `A-NQ8` <dbl>,
## #   `A-NQ9` <dbl>, `A-NQ10` <dbl>, `A-NQ11` <dbl>, `A-NQ12` <dbl>,
## #   `A-NQ13` <dbl>, `A-NQ14` <dbl>, `A-FQ1` <dbl>, `A-FQ2` <dbl>, ...
```

Still one ambiguous case. Take the one with the longest duration.

```
for (dupID in dupes.pre1) {
  dupindex <- which(pretest$StudyID==dupID)
  dur <- pretest[dupindex,"Duration (in seconds)",drop=TRUE]
  dropindix <- dupindex[dur < max(dur)]
  pretest <- pretest[-dropindix,]
}
```

```
anyDuplicated(pretest$StudyID)
```

```
## [1] 0
```

Covert to factors and add labels.

Demographics

Only a few students checked Nonbinary or Other, so put them together. Also, call prefer not to say as NA.

A couple of students put XX years old, instead of their age, so fix.

```
pretest$Sex <- choicesToFactor(preSQ$Sex,pretest$Sex)
pretest$Gender <- case_match(pretest$Sex,
                             "Male"~"Male",
                             "Female"~"Female",
                             c("Other","Nonbinary") ~ "Other",
                             .default=NA) %>%
  factor(levels=c("Male","Female","Other"))
age <- pretest$Age
age[!is.na(age)] <- regmatches(age[!is.na(age)],
                        regexpr("[[:digit:]]*",
                                age[!is.na(age)]))
pretest$Age <- as.numeric(age)
pretest$Gaming <- choicesToFactor(preSQ$Gaming,pretest$Gaming)
pretest$Physics <- choicesToFactor(preSQ$Physics,pretest$Physics)
```

Ethnicity. Collapse a couple of categories.

```
pretest$Ethnicity_9_TEXT[!is.na(pretest$Ethnicity_9)]
```

```
## [1] "Part Hawaiian"    "Jamaican behamin" "italian"          NA
```

Qualtrics "helpfully" codes this value as 1/NA. Fix to true logicals.

```
ethids <- grep("Ethnicity_",names(pretest))
for (eth in ethids) {
  name <- names(pretest)[eth]
  if (any(grepl("TEXT",name))) {
    names(pretest)[eth] <- "Other_TEXT"
  } else {
    names(pretest)[eth] <- ethnames[name]
    pretest[[eth]] <- !is.na(pretest[[eth,drop=TRUE]])
  }
}
```

```
## Warning: Extra arguments ignored.
## Extra arguments ignored.
## Extra arguments ignored.
## Extra arguments ignored.
## Extra arguments ignored.
## Extra arguments ignored.
## Extra arguments ignored.
## Extra arguments ignored.
```

```
head(pretest[,ethids])
```

```
## # A tibble: 6 x 9
```

```
##   `American Indian or Alaska Native` Asian `Black or African American` Hispanic
##   <lgl>                              <lgl> <lgl>                        <lgl>
## 1 FALSE                              FALSE FALSE                        FALSE
## 2 FALSE                              FALSE FALSE                        FALSE
## 3 FALSE                              TRUE  FALSE                        FALSE
## 4 FALSE                              TRUE  FALSE                        FALSE
## 5 FALSE                              FALSE FALSE                        FALSE
## 6 FALSE                              TRUE  FALSE                        FALSE
## # i 5 more variables: `Native Hawaiian or Pacific Islander` <lgl>, White <lgl>,
## #   `Other (enter)` <lgl>, `Prefer not to say` <lgl>, Other_TEXT <chr>
```

```
ethids1 <- ethids[-length(ethids)] ## Remove Other_TEXT
ethcols <- as.matrix(as.data.frame(pretest[,ethids1]))
ethnicity <- sapply(1L:nrow(pretest), function(irow) {
    paste(ethnames[ethcols[irow,]], collapse=",")
})
unique(ethnicity)
```

```
##  [1] "White,Other (enter)"
##  [2] "White"
##  [3] "Asian"
##  [4] "Asian,White"
##  [5] "Black or African American"
##  [6] "Black or African American,White"
##  [7] "Hispanic,White"
##  [8] "Asian,Hispanic"
##  [9] "Asian,Black or African American"
## [10] "Hispanic"
## [11] "Black or African American,Other (enter)"
## [12] "White,Prefer not to say"
## [13] "Black or African American,Hispanic"
## [14] "American Indian or Alaska Native,Black or African American,White"
## [15] "Black or African American,Native Hawaiian or Pacific Islander"
```

To get a single factor, collapse any combination into "mixed"

```
ethnicity[grep(",",ethnicity)] <- "Mixed"
is.na(ethnicity) <- pretest$`Prefer not to say`
ethnicity <- as.factor(ethnicity)
summary(ethnicity)
```

```
##                    Asian Black or African American                 Hispanic
##                        5                       51                       10
##                    Mixed                    White                    NA's
##                       39                      137                        1
```

```
pretest$Ethnicity <- ethnicity
```

Physics Questions

Need to fix naming issue with Column A-FQ10. Also A-NQ1 and A-FQ4 metadata missing.

```
names(pretest)[names(pretest)=="AFQ10"] <- "A-FQ10"
names(preKey)[names(preKey)=="AFQ10"] <- "A-FQ10"
names(preSQ)[names(preSQ)=="AFQ10"] <- "A-FQ10"
preSQ["A-NQ1"] <- potSQ["A-NQ1"]
preSQ["A-FQ4"] <- potSQ["A-FQ4"]
```

```
for (q in grep("A-",names(pretest),value=TRUE)) {
  vals <- pretest[[q]]
  pretest[[paste(q,"scored",sep="_")]] <- as.numeric(vals==preKey[q])
  pretest[[q]] <- choicesToFactor(preSQ[[q]],vals)
}
for (q in grep("A-",names(ecttest),value=TRUE)) {
  vals <- ecttest[[q]]
  ecttest[[paste(q,"scored",sep="_")]] <- as.numeric(vals==ectKey[q])
  ecttest[[q]] <- choicesToFactor(preSQ[[q]],vals)
}
for (q in grep("A-",names(pottest),value=TRUE)) {
  vals <- pottest[[q]]
  pottest[[paste(q,"scored",sep="_")]] <- as.numeric(vals==preKey[q])
  pottest[[q]] <- choicesToFactor(preSQ[[q]],vals)
}
head(pretest)
```

```
## # A tibble: 6 x 77
##   Progress `Duration (in seconds)` Finished RecordedDate        StudyID   Age
##      <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <dbl>
## 1      100                     660        1 2022-11-28 08:24:13 D3317      12
## 2      100                     905        1 2022-11-28 08:25:38 C3115      13
## 3      100                     757        1 2022-11-28 08:25:41 E3384      13
## 4      100                     632        1 2022-11-28 08:25:47 C3148      12
## 5      100                     748        1 2022-11-28 08:26:43 A3294      12
## 6      100                     564        1 2022-11-28 08:26:51 E0224      13
## # i 71 more variables: Sex <fct>, `American Indian or Alaska Native` <lgl>,
## #   Asian <lgl>, `Black or African American` <lgl>, Hispanic <lgl>,
## #   `Native Hawaiian or Pacific Islander` <lgl>, White <lgl>,
## #   `Other (enter)` <lgl>, `Prefer not to say` <lgl>, Other_TEXT <chr>,
## #   Gaming <fct>, Physics <fct>, `A-NQ1` <fct>, `A-NQ2` <fct>, `A-NQ3` <fct>,
## #   `A-NQ4` <fct>, `A-NQ5` <fct>, `A-NQ6` <fct>, `A-NQ7` <fct>, `A-NQ8` <fct>,
## #   `A-NQ9` <fct>, `A-NQ10` <fct>, `A-NQ11` <fct>, `A-NQ12` <fct>, ...
```

```
NearECTcols <- paste(physics.key$ID[physics.key$Form=="A" &
                                    physics.key$`Near/Far`=="Near" &
                                    physics.key$`HL Concept`=="EcT"],
                     "scored",sep="_")
FarECTcols <- paste(physics.key$ID[physics.key$Form=="A" &
                                   physics.key$`Near/Far`=="Far" &
                                   physics.key$`HL Concept`=="EcT"],
                    "scored",sep="_")
NearPOTcols <- paste(physics.key$ID[physics.key$Form=="A" &
                                    physics.key$`Near/Far`=="Near" &
                                    physics.key$`HL Concept`=="PoT"],
                     "scored",sep="_")
FarPOTcols <- paste(physics.key$ID[physics.key$Form=="A" &
                                   physics.key$`Near/Far`=="Far" &
                                   physics.key$`HL Concept`=="PoT"],
                    "scored",sep="_")
```

```
pretest %>%
  mutate(NearECT=rowSums(pretest[,NearECTcols],na.rm=TRUE),
         FarECT=rowSums(pretest[,FarECTcols],na.rm=TRUE),
```

```
         NearPOT=rowSums(pretest[,NearPOTcols],na.rm=TRUE),
         FarPOT=rowSums(pretest[,FarPOTcols],na.rm=TRUE)) %>%
  mutate(Near=NearECT+NearPOT,Far=FarECT+FarPOT,
         ECT=NearECT+FarECT, POT=NearPOT+FarPOT,
         PhysicsScore=NearECT+FarECT+NearPOT+FarPOT) ->
  pretest
```

```
pottest %>%
  mutate(NearPOTpost=rowSums(pretest[,NearPOTcols],na.rm=TRUE),
         FarPOTpost=rowSums(pretest[,FarPOTcols],na.rm=TRUE)) %>%
  mutate(POTpost = NearPOTpost+FarPOTpost) ->
  pottest
ecttest %>%
  mutate(NearECTpost=rowSums(ecttest[,NearECTcols],na.rm=TRUE),
         FarECTpost=rowSums(ecttest[,FarECTcols],na.rm=TRUE)) %>%
  mutate(ECTpost = NearECTpost+FarECTpost) ->
  ecttest
```

IMI Questions

IMI questions are only in the posttest.

```
for (col in IMI.scales$ID[IMI.scales$Reverse]) {
  pottest[[col]] <- 8-pottest[[col]]
  ecttest[[col]] <- 8-ecttest[[col]]
}
```

```
pottest$IMI <- rowSums(pottest[,unique(IMI.scales$ID)],na.rm=TRUE)
ecttest$IMI <- rowSums(ecttest[,unique(IMI.scales$ID)],na.rm=TRUE)

for (scale in unique(IMI.scales$Scale)) {
  pottest[[paste("IMI",scale,sep="_")]] <-
    rowSums(pottest[,IMI.scales$ID[IMI.scales$Scale==scale]],na.rm=TRUE)
  ecttest[[paste("IMI",scale,sep="_")]] <-
    rowSums(ecttest[,IMI.scales$ID[IMI.scales$Scale==scale]],na.rm=TRUE)
}
```

PA questions

Again, only in ECT and POT

UGH! Qualtrics was not consistent in the numeric values for these columns. Some were 1–5, some were 11–15, and one was 8–12. WTF. I went back and fixed the coding in Qualtrics.

```
for(paq in PA.rev$ID[PA.rev$Reversed]) {
  ecttest[[paq]] <- 5-ecttest[[paq]]
  pottest[[paq]] <- 5-pottest[[paq]]
}
ecttest$PA <- rowSums(ecttest[,PA.rev$ID],na.rm=TRUE)
pottest$PA <- rowSums(pottest[,PA.rev$ID],na.rm=TRUE)
head(pottest)
```

```
## # A tibble: 6 x 68
##   Progress `Duration (in seconds)` Finished RecordedDate        StudyID IMI_1
##      <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <dbl>
## 1      100                     319        1 2022-11-30 08:39:38 E0224       4
```

```
## 2      100                     339        1 2022-11-30 08:39:41 E0549       4
## 3      100                     445        1 2022-11-30 08:41:12 E0550       5
## 4      100                     437        1 2022-11-30 08:41:13 D0213       6
## 5      100                     438        1 2022-11-30 08:41:22 B0190       4
## 6      100                     448        1 2022-11-30 08:41:38 B2462       4
## # i 62 more variables: IMI_2 <dbl>, IMI_3 <dbl>, IMI_4 <dbl>, IMI_5 <dbl>,
## #   IMI_6 <dbl>, IMI_7 <dbl>, IMI_8 <dbl>, IMI_9 <dbl>, IMI_10 <dbl>,
## #   IMI_11 <dbl>, IMI_12 <dbl>, IMI_13 <dbl>, IMI_14 <dbl>, IMI_15 <dbl>,
## #   IMI_16 <dbl>, IMI_17 <dbl>, IMI_18 <dbl>, `A-NQ9` <fct>, `A-NQ10` <fct>,
## #   `A-NQ11` <fct>, `A-NQ12` <fct>, `A-NQ13` <fct>, `A-NQ14` <fct>,
## #   `A-FQ9` <fct>, `A-FQ10` <fct>, `A-FQ11` <fct>, `A-FQ12` <fct>,
## #   `A-FQ13` <fct>, `A-FQ14` <fct>, Q90 <chr>, Q74 <dbl>, PA1 <dbl>, ...
```

```
head(ecttest)
```

```
## # A tibble: 6 x 82
##   Progress `Duration (in seconds)` Finished RecordedDate        StudyID IMI_1
##      <dbl>                   <dbl>    <dbl> <dttm>              <chr>   <dbl>
## 1      100                     481        1 2022-11-30 08:42:33 C0066       6
## 2      100                     514        1 2022-11-30 08:42:52 A0202       1
## 3      100                     556        1 2022-11-30 08:42:58 F0235       5
## 4      100                     560        1 2022-11-30 08:43:25 C0022       5
## 5      100                     585        1 2022-11-30 08:43:40 A0189       4
## 6      100                     530        1 2022-11-30 08:44:53 C3115       6
## # i 76 more variables: IMI_2 <dbl>, IMI_3 <dbl>, IMI_4 <dbl>, IMI_5 <dbl>,
## #   IMI_6 <dbl>, IMI_7 <dbl>, IMI_8 <dbl>, IMI_9 <dbl>, IMI_10 <dbl>,
## #   IMI_11 <dbl>, IMI_12 <dbl>, IMI_13 <dbl>, IMI_14 <dbl>, IMI_15 <dbl>,
## #   IMI_16 <dbl>, IMI_17 <dbl>, IMI_18 <dbl>, `A-NQ1` <fct>, `A-NQ2` <fct>,
## #   `A-NQ3` <fct>, `A-NQ4` <fct>, `A-NQ5` <fct>, `A-NQ6` <fct>, `A-NQ7` <fct>,
## #   `A-NQ8` <fct>, `A-NQ12` <fct>, `A-NQ13` <fct>, `A-NQ14` <fct>,
## #   `A-FQ1` <fct>, `A-FQ2` <fct>, `A-FQ3` <fct>, `A-FQ4` <fct>, ...
```

Do the join

```
FSUSFall2022 <- select(PPIDs,!Number) %>%
  full_join(pretest,by="StudyID",suffix=c("",".pre")) %>%
  full_join(ecttest,by="StudyID",suffix=c("",".ect")) %>%
  full_join(pottest,by="StudyID",suffix=c("",".pot"))
```

Check duplicates

```
dupes <- FSUSFall2022$StudyID[duplicated(FSUSFall2022$StudyID)]
dupes
```

```
## character(0)
```

Yay! Screening on Finished fixed the duplication problem.

Output the data

```
outID <- "1DJf7Iidg-GvEXHkrfNPwGTB7VerLSTAUohzZYQOov9E"
write_sheet(FSUSFall2022,outID, sheet="Data")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'Data'.
```

```
write_sheet(data.frame(StudyID=dupes),outID,"Duplicate IDs")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'Duplicate IDs'.
```

```
write_csv(FSUSFall2022,"data/PPIESFall2022PrePost.csv")
```

Generate cleaning script for the log data.

```
Xids <- c(InvIDs$ID1,InvIDs$ID2)
write_sheet(data.frame(Xids=Xids),outID,"DeleteThese")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'DeleteThese'.
```

This is now used to clean the log files.

Merge data from Log Files

BNScores

BNScores is keyed by uid, so we are ready to go. Kill the first two columns, which are unneeded.

```
BNscores <- read_csv("https://pluto.coe.fsu.edu/Proc4/dongle/data/stats-BigStudy.csv")
```

```
## New names:
## Rows: 203 Columns: 61
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (3): app, uid, context dbl (57): ...1, Physics_EAP, ForceAndMotion_EAP,
## LinearMomentum_EAP, Energy... dttm (1): timestamp
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
BNscores <- select(BNscores,!any_of(c("...1","app")))
```

```
write_sheet(BNscores, outID, "BNscores")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'BNscores'.
```

```
write_csv(BNscores,"data/PPIESFall2022BN.csv")
```

Observables

observables has (uid,Context) as key, need to pivot winder. Timestamp isn't really an observable, but keep it anyway.

```
observables <- read_csv("https://pluto.coe.fsu.edu/Proc4/dongle/data/ppObs.BigStudy.csv")
```

```
## New names:
## Rows: 7111 Columns: 24
```

```
## -- Column specification
## ------------------------------------------------------- Delimiter: "," chr
## (3): uid, context, TrophyLevel dbl (4): ...1, ObjectCount, NumberAttempts,
## bankBalance lgl (16): blowerManip, BouncinessRun, Agent, ApplicableAgent,
## PufferClicks,... dttm (1): timestamp
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
obsnames <- names(observables)[-(1:3)]
## Drop useless column and sort by timestamp
observables %>% select(!`...1`) %>% arrange(timestamp) -> observables
```

Add Quits

```
observables$TrophyLevel[is.na(observables$TrophyLevel)] <-
  ifelse(is.na(observables$NumberAttempts[is.na(observables$TrophyLevel)]),NA,"quit")
observables$TrophyLevel <- ordered(observables$TrophyLevel,
                                   c("quit","silver","gold"))
```

Multiple Attempts

Problem. What to do with multiple attempts. Two choices, take First, and Take Last

```
obsFirst <-
  pivot_wider(observables, id_cols=uid,
              names_from = context,
              values_from = all_of(obsnames),
              values_fn = function(x) first(x,na_rm=TRUE))
obsLast <-
  pivot_wider(observables, id_cols=uid,
              names_from = context,
              values_from = all_of(obsnames),
              values_fn = function(x) last(x,na_rm=TRUE))
obsMax <-
  pivot_wider(observables, id_cols=uid,
              names_from=context,
              values_from=TrophyLevel,
              values_fn = function(x) max(x,na.rm=FALSE))
```

A lot of N/A observables, so we have total NA columns. Drop these.

```
obsFirst<- obsFirst[,sapply(obsFirst,function(c) !all(is.na(c)))]
obsFirst <-
  obsFirst[,sapply(obsFirst,function(c) !is.numeric(c) || sum(c,na.rm=TRUE) != 0)]
obsLast<- obsLast[,sapply(obsLast,function(c) !all(is.na(c)))]
obsLast<-
  obsLast[,sapply(obsLast,function(c) !is.numeric(c) || sum(c,na.rm=TRUE) != 0)]
head(obsLast)
```

```
## # A tibble: 6 x 97
##   uid   timestamp_Nudge     timestamp_Lever     timestamp_Ramp
##   <chr> <dttm>              <dttm>              <dttm>
## 1 C3148 2022-12-01 08:15:25 2022-12-01 08:15:44 2022-12-01 08:16:08
## 2 D3418 2022-12-02 08:14:38 2022-12-02 08:16:08 2022-12-02 08:37:16
## 3 D3137 2022-12-01 08:22:55 2022-12-01 08:23:21 2022-12-01 08:23:39
```

```
## 4 A3430 2022-12-01 08:11:00 2022-11-29 08:19:26 2022-11-29 08:19:58
## 5 D3317 2022-12-02 08:28:32 2022-12-01 08:22:09 2022-12-01 08:22:35
## 6 A3294 2022-12-01 08:14:50 2022-12-01 08:15:54 2022-12-01 08:16:36
## # i 93 more variables: timestamp_Pendulum <dttm>, timestamp_Springboard <dttm>,
## #   `timestamp_Down Hill` <dttm>, `timestamp_Lead the Ball` <dttm>,
## #   `timestamp_Chocolate Factory` <dttm>, `timestamp_Around the Tree` <dttm>,
## #   `timestamp_Big Watermill` <dttm>, `timestamp_Sunny Day` <dttm>,
## #   timestamp_Wavy <dttm>, `timestamp_On the Upswing` <dttm>,
## #   timestamp_Shark <dttm>, `timestamp_One at a time` <dttm>,
## #   timestamp_Scale <dttm>, `timestamp_Cloudy Day` <dttm>, ...
```

Level names

We currently don't need this, but might later.

```
levels <- gsub("TrophyLevel_(.*)","\\1",grep("TrophyLevel_",names(obsFirst),value=TRUE))
levels
```

```
##  [1] "Nudge"                "Lever"                "Ramp"
##  [4] "Pendulum"             "Springboard"          "Down Hill"
##  [7] "Lead the Ball"        "Chocolate Factory"    "Around the Tree"
## [10] "Big Watermill"        "Sunny Day"            "Wavy"
## [13] "On the Upswing"       "Shark"                "One at a time"
## [16] "Scale"                "Cloudy Day"           "Crazy Seesaw"
## [19] "Roller coaster"       "Uphill Battle"        "Little Mermaid"
## [22] "Ultimate Pinball"     "Yippie!"              "Diving Board"
## [25] "Stiff Curtains"       "Diving Board World"   "Need Fulcrum"
## [28] "Timing is Everything" "Perfect Toss"         "Up in the Air"
## [31] "Spider Web"           "Can Opener"
```

Add Trophy Counts

```
obsMax %>% select(!uid) %>% as.matrix() %>% apply(1, function(row)
  table(ordered(row, c("quit","silver","gold")))) %>% t()  -> tMax
obsMax <- cbind(obsMax,tMax)
obsFirst %>% select(starts_with("Trophy")) %>% as.matrix() %>% apply(1, function(row)
  table(ordered(row, c("quit","silver","gold")))) %>% t()  -> tFirst
obsFirst <- cbind(obsFirst,tFirst)
obsLast %>% select(starts_with("Trophy")) %>% as.matrix() %>% apply(1, function(row)
  table(ordered(row, c("quit","silver","gold")))) %>% t()  -> tLast
obsLast <- cbind(obsLast,tLast)
```

Write it out

```
write_sheet(obsFirst,outID,sheet="ObsFirst")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'ObsFirst'.
```

```
write_sheet(obsLast,outID,sheet="ObsLast")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'ObsLast'.
```

```
write_sheet(obsMax,outID,sheet="ObsMax")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
```

```
## v Writing to sheet 'ObsMax'.
```

```
write_csv(obsFirst,"data/PPIESFall2022obsFirst.csv")
write_csv(obsLast,"data/PPIESFall2022obsLast.csv")
write_csv(obsMax,"data/PPIESFall2022obsMax.csv")
```

Learning Supports

Key is (uid,context,onWhat). Value is LS_duration. Can summarize with sum.

```
learningSupports <- read_csv("https://pluto.coe.fsu.edu/Proc4/dongle/data/ppLS.BigStudy.csv")
```

```
## New names:
## Rows: 3468 Columns: 11
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (5): uid, context, mess, onWhat, learningSupportType dbl (2): ...1, LS_duration
## lgl (3): currentMoney, appId, money dttm (1): timestamp
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
learningSupports <-
  mutate(learningSupports,duration=LS_duration) %>%
  select(!any_of(c("...1","appid","LS_duration"))) %>%
  filter(!is.na(onWhat))
```

```
lsWide <-
  pivot_wider(learningSupports,
              id_cols=uid,
              names_from=c(context,onWhat),
              values_from=c(timestamp,duration),
              names_glue="{context}_{onWhat}_{.value}",
              values_fn=list(timestamp=~min(.x, na.rm=TRUE),
                             duration=~ sum(.x,na.rm=TRUE)))
head(lsWide)
```

```
## # A tibble: 6 x 191
##   uid   Nudge_game_tools_timestamp Springboard_game_too~1 Big Watermill_hint_t~2
##   <chr> <dttm>                     <dttm>                 <dttm>
## 1 A3328 2022-11-29 08:18:59        NA                     NA
## 2 F3452 NA                         2022-11-29 08:25:19    2022-11-29 08:29:31
## 3 C3104 NA                         2022-11-29 08:26:30    NA
## 4 A3351 NA                         NA                     NA
## 5 C3115 NA                         NA                     NA
## 6 E0549 NA                         2022-11-29 08:32:14    NA
## # i abbreviated names: 1: Springboard_game_tools_timestamp,
## #   2: `Big Watermill_hint_timestamp`
## # i 187 more variables: `Chocolate Factory_physAnim_ramp_timestamp` <dttm>,
## #   `Sunny Day_game_tools_timestamp` <dttm>,
## #   `Big Watermill_physAnim_ramp_timestamp` <dttm>,
## #   `Big Watermill_game_tools_timestamp` <dttm>,
## #   Shark_physAnim_lever_timestamp <dttm>, Shark_hint_timestamp <dttm>, ...
```

Kill combinations that never occur.

```
all(sapply(lsWide,function(c) !all(is.na(c))))
```

```
## [1] TRUE
```

Get the names of the various learning support types.

```
lsNames <- unique(sub("^[^_]*_(.*)_duration$","\\1",
                      grep("_duration$",names(lsWide),value=TRUE)))
lsNames
```

```
## [1] "game_tools"           "hint"                 "physAnim_ramp"
## [4] "physAnim_lever"       "physAnim_pendulum"    "physAnim_springboard"
## [7] "happy"                "exercise"             "breathing"
```

```
durations <- select(lsWide,ends_with("duration"))
for (lsType in lsNames) {
  subset <- select(durations,contains(lsType))
  lsWide[[paste(lsType,"total",sep="_")]] <-
    rowSums(subset,na.rm=TRUE)
  lsWide[[paste(lsType,"count",sep="_")]] <-
    rowSums(!is.na(subset))
}
```

```
write_sheet(lsWide,outID,"LearningSupport")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
```

```
## v Writing to sheet 'LearningSupport'.
```

```
write_csv(lsWide,"data/PPIESFall2022ls.csv")
```

Join and Write

```
FSUSFall2022BigDaddy <- select(PPIDs,!Number) %>%
  full_join(pretest,by="StudyID",suffix=c("",".pre")) %>%
  full_join(ecttest,by="StudyID",suffix=c("",".ect")) %>%
  full_join(pottest,by="StudyID",suffix=c("",".pot")) %>%
  left_join(BNscores,by=join_by(StudyID==uid),suffix=c("",".bn")) %>%
  left_join(obsFirst,by=join_by(StudyID==uid),suffix=c("",".first")) %>%
  left_join(obsLast,by=join_by(StudyID==uid),suffix=c("",".last")) %>%
  left_join(obsMax,by=join_by(StudyID==uid),suffix=c("",".max")) %>%
  left_join(lsWide,by=join_by(StudyID==uid),suffix=c("",".ls"))
```

```
#write_sheet(FSUSFall2022BigDaddy,outID,"BigDaddy")
write_csv(FSUSFall2022BigDaddy,"data/PPIESFall2022Full.csv")
```

SPSS Outpt

```
names(FSUSFall2022BigDaddy) <- gsub("-","_",names(FSUSFall2022BigDaddy)) %>%
  gsub(" (in seconds)","",.,fixed=TRUE) %>%
  gsub(" ","",.) %>% gsub("(enter)","",.,fixed=TRUE) %>%
  gsub("Yippie!","Yippie",.,fixed=TRUE)
haven::write_sav(FSUSFall2022BigDaddy,"FSUSFall2022BigDaddy.sav")
```

Metadata Export

```
prechoices <- sapply(preSQ, function(sq) {
  if (length(sq$Choices) >0L) {
    choices <- sapply(sq$Choices,function(ch) ch$Display)
    c(choices,rep(NA_character_,8-length(choices)))
  } else {
    rep(NA_character_,8)
  }
})
prechoices <- t(prechoices)
colnames(prechoices) <- paste(1:ncol(prechoices))
```

```
premeta <- data.frame(
  name=names(preSQ),
  qid=sapply(preSQ,function(sq) sq$QuestionID),
  qtag=sapply(preSQ,function(sq) sq$DataExportTag),
  text=sapply(preSQ,function(sq) sq$QuestionText),
  description=sapply(preSQ,function(sq) {
    if (is.null(sq$QuestionDescription))
      ""
    else
      sq$QuestionDescription
  }),
  choices=prechoices
)
head(premeta)
```

```
##          name   qid   qtag
## Q27       Q27 QID27    Q27
## Q65       Q65 QID65    Q65
## A.NQ5   A-NQ5 QID16  A-NQ5
## A.FQ14 A-FQ14 QID62 A-FQ14
## A.FQ7   A-FQ7 QID48  A-FQ7
## A.FQ12 A-FQ12 QID58 A-FQ12
##
## Q27    <video class="qmedia" controls="true" height="480" preload="auto" width="774"><source src="ht
## Q65    <video class="qmedia" controls="true" height="480" preload="auto" width="854"><source src="ht
## A.NQ5
## A.FQ14
## A.FQ7
## A.FQ12
##                                                                                                  desc
## Q27
## Q65
## A.NQ5  A ball is dropped from each point shown above (A, B, C). When will the ball have the fastest s
## A.FQ14              A person is fishing in a stream. If the fishing pole was shorter, it would bend
## A.FQ7  A springboard is bent down by weight B. When the weight is released, the green ball flies up i
## A.FQ12 An acrobat needs to land on the platform above. At the top of another platform, two acrobats
##                             choices.1                    choices.2
## Q27                              <NA>                         <NA>
## Q65                              <NA>                         <NA>
## A.NQ5           Dropped from point A          Dropped from point B
## A.FQ14                           more                         less
```

```
## A.FQ7  Increase the mass of the ball Decrease the mass of the ball
## A.FQ12                                A                             B
##                                          choices.3
## Q27                                           <NA>
## Q65                                           <NA>
## A.NQ5                            Dropped from point C
## A.FQ14                                     the same
## A.FQ7                  Increase the mass of weight B
## A.FQ12 Both will have the same effect on the acrobat
##                                                 choices.4 choices.5 choices.6
## Q27                                                  <NA>      <NA>      <NA>
## Q65                                                  <NA>      <NA>      <NA>
## A.NQ5                                        No difference      <NA>      <NA>
## A.FQ14                         more information is needed      <NA>      <NA>
## A.FQ7  More information is needed to answer the question      <NA>      <NA>
## A.FQ12                            Not enough information      <NA>      <NA>
##        choices.7 choices.8
## Q27         <NA>      <NA>
## Q65         <NA>      <NA>
## A.NQ5       <NA>      <NA>
## A.FQ14      <NA>      <NA>
## A.FQ7       <NA>      <NA>
## A.FQ12      <NA>      <NA>
```

ECT

IMI meta-data is completely different from the rest, so needs special handling.

```
ectchoices <- sapply(ectSQ[names(ectSQ)!="IMI"], function(sq) {
  if (length(sq$Choices) >0L) {
    choices <- sapply(sq$Choices,function(ch) ch$Display)
    c(choices,rep(NA_character_,8-length(choices)))
  } else {
    rep(NA_character_,8)
  }
})
ectchoices <- t(ectchoices)
colnames(ectchoices) <- paste(1:ncol(ectchoices))
```

```
ectmeta <- data.frame(
  name=paste(names(ectSQ[names(ectSQ)!="IMI"]),"ect",sep="."),
  qid=sapply(ectSQ[names(ectSQ)!="IMI"],function(sq) sq$QuestionID),
  qtag=sapply(ectSQ[names(ectSQ)!="IMI"],function(sq) sq$DataExportTag),
  text=sapply(ectSQ[names(ectSQ)!="IMI"],function(sq) sq$QuestionText),
  description=sapply(ectSQ[names(ectSQ)!="IMI"],function(sq) {
    if (is.null(sq$QuestionDescription))
      ""
    else
      sq$QuestionDescription
  }),
  choices=ectchoices
)


head(ectmeta)
```

```
##              name   qid   qtag
## Q33       Q33.ect QID33    Q33
## A-NQ5   A-NQ5.ect QID22  A-NQ5
## A-FQ14 A-FQ14.ect QID68 A-FQ14
## A-FQ7   A-FQ7.ect QID54  A-FQ7
## A-FQ12 A-FQ12.ect QID64 A-FQ12
## A-NQ2   A-NQ2.ect QID16  A-NQ2
##
## Q33    <video class="qmedia" controls="true" height="480" preload="auto" width="774"><source src="ht
## A-NQ5
## A-FQ14
## A-FQ7
## A-FQ12
## A-NQ2
##                                                                                                  desc
## Q33
## A-NQ5  A ball is dropped from each point shown above (A, B, C). When will the ball have the fastest s
## A-FQ14              A person is fishing in a stream. If the fishing pole was shorter, it would bend
## A-FQ7  A springboard is bent down by weight B. When the weight is released, the green ball flies up
## A-FQ12 An acrobat needs to land on the platform above. At the top of another platform, two acrobats
## A-NQ2  An object is drawn resting on the right-hand side of the lever. It's just heavy enough to lif
##                             choices.1                     choices.2
## Q33                              <NA>                          <NA>
## A-NQ5           Dropped from point A          Dropped from point B
## A-FQ14                           more                          less
## A-FQ7  Increase the mass of the ball Decrease the mass of the ball
## A-FQ12                             A                             B
## A-NQ2             The same as before              More than before
##                                          choices.3
## Q33                                           <NA>
## A-NQ5                          Dropped from point C
## A-FQ14                                      the same
## A-FQ7                  Increase the mass of weight B
## A-FQ12 Both will have the same effect on the acrobat
## A-NQ2                               Less than before
##                                                   choices.4 choices.5 choices.6
## Q33                                                    <NA>      <NA>      <NA>
## A-NQ5                                        No difference      <NA>      <NA>
## A-FQ14                         more information is needed      <NA>      <NA>
## A-FQ7  More information is needed to answer the question      <NA>      <NA>
## A-FQ12                           Not enough information      <NA>      <NA>
## A-NQ2                            Not enough information      <NA>      <NA>
##        choices.7 choices.8
## Q33         <NA>      <NA>
## A-NQ5       <NA>      <NA>
## A-FQ14      <NA>      <NA>
## A-FQ7       <NA>      <NA>
## A-FQ12      <NA>      <NA>
## A-NQ2       <NA>      <NA>
```

```
IMISQ <- ectSQ[["IMI"]]
IMI.options <- c(sapply(IMISQ$Answers,function(ch) ch$Display),"8"=NA_character_)

imimeta <- data.frame(
```

```
  name=names(IMI.stems),
  qid=IMISQ$QuestionID,
  qtag=IMISQ$DataExportTag,
  text=IMI.stems,
  description=IMI.stems,
  choices=t(replicate(length(IMI.stems),IMI.options))
)
head(imimeta)
```

```
##        name   qid qtag                                          text
## IMI_1 IMI_1 QID77  IMI           I enjoyed playing the game very much
## IMI_2 IMI_2 QID77  IMI                      The game was fun to play.
## IMI_3 IMI_3 QID77  IMI                  I thought the game was boring
## IMI_4 IMI_4 QID77  IMI     The game did not hold my attention at all.
## IMI_5 IMI_5 QID77  IMI I would describe the game as very interesting.
## IMI_6 IMI_6 QID77  IMI          I think I am pretty good at the game.
##                                       description       choices.1 choices.2
## IMI_1           I enjoyed playing the game very much Not at all true     
## IMI_2                       The game was fun to play. Not at all true     
## IMI_3                   I thought the game was boring Not at all true     
## IMI_4     The game did not hold my attention at all. Not at all true     
## IMI_5 I would describe the game as very interesting. Not at all true     
## IMI_6          I think I am pretty good at the game. Not at all true     
##       choices.3     choices.4 choices.5 choices.6 choices.7 choices.8
## IMI_1      Somewhat true           Very true      <NA>
## IMI_2      Somewhat true           Very true      <NA>
## IMI_3      Somewhat true           Very true      <NA>
## IMI_4      Somewhat true           Very true      <NA>
## IMI_5      Somewhat true           Very true      <NA>
## IMI_6      Somewhat true           Very true      <NA>
```

POT metadata

```
potchoices <- sapply(potSQ[names(potSQ)!="IMI"], function(sq) {
  if (length(sq$Choices) >0L) {
    choices <- sapply(sq$Choices,function(ch) ch$Display)
    c(choices,rep(NA_character_,8-length(choices)))
  } else {
    rep(NA_character_,8)
  }
})
potchoices <- t(potchoices)
colnames(ectchoices) <- paste(1:ncol(ectchoices))
```

```
potmeta <- data.frame(
  name=paste(names(potSQ[names(potSQ)!="IMI"]),"pot",sep="."),
  qid=sapply(potSQ[names(potSQ)!="IMI"],function(sq) sq$QuestionID),
  qtag=sapply(potSQ[names(potSQ)!="IMI"],function(sq) sq$DataExportTag),
  text=sapply(potSQ[names(potSQ)!="IMI"],function(sq) sq$QuestionText),
  description=sapply(potSQ[names(potSQ)!="IMI"],function(sq) {
    if (is.null(sq$QuestionDescription))
      ""
    else
      sq$QuestionDescription
```

```
  }),
  choices=potchoices
)


head(potmeta)
```

```
##              name   qid   qtag
## Q33       Q33.pot QID33    Q33
## A-NQ5   A-NQ5.pot QID22  A-NQ5
## A-FQ14 A-FQ14.pot QID68 A-FQ14
## A-FQ7   A-FQ7.pot QID54  A-FQ7
## A-FQ12 A-FQ12.pot QID64 A-FQ12
## A-NQ2   A-NQ2.pot QID16  A-NQ2
##
## Q33    <video class="qmedia" controls="true" height="480" preload="auto" width="774"><source src="ht
## A-NQ5
## A-FQ14
## A-FQ7
## A-FQ12
## A-NQ2
##                                                                                                      desc
## Q33
## A-NQ5  A ball is dropped from each point shown above (A, B, C). When will the ball have the fastest s
## A-FQ14              A person is fishing in a stream. If the fishing pole was shorter, it would bend
## A-FQ7  A springboard is bent down by weight B. When the weight is released, the green ball flies up
## A-FQ12 An acrobat needs to land on the platform above. At the top of another platform, two acrobats
## A-NQ2  An object is drawn resting on the right-hand side of the lever. It's just heavy enough to lif
##                             choices.1                     choices.2
## Q33                              <NA>                          <NA>
## A-NQ5           Dropped from point A          Dropped from point B
## A-FQ14                           more                          less
## A-FQ7  Increase the mass of the ball Decrease the mass of the ball
## A-FQ12                             A                             B
## A-NQ2             The same as before              More than before
##                                         choices.3
## Q33                                          <NA>
## A-NQ5                          Dropped from point C
## A-FQ14                                    the same
## A-FQ7                  Increase the mass of weight B
## A-FQ12 Both will have the same effect on the acrobat
## A-NQ2                               Less than before
##                                                  choices.4 choices.5 choices.6
## Q33                                                   <NA>      <NA>      <NA>
## A-NQ5                                       No difference      <NA>      <NA>
## A-FQ14                        more information is needed      <NA>      <NA>
## A-FQ7  More information is needed to answer the question      <NA>      <NA>
## A-FQ12                           Not enough information      <NA>      <NA>
## A-NQ2                            Not enough information      <NA>      <NA>
##        choices.7 choices.8
## Q33         <NA>      <NA>
## A-NQ5       <NA>      <NA>
## A-FQ14      <NA>      <NA>
## A-FQ7       <NA>      <NA>
```

```
## A-FQ12       <NA>       <NA>
## A-NQ2        <NA>       <NA>
```

Form B metadata

```
bchoices <- sapply(postBSQ[names(postBSQ)!="IMI"], function(sq) {
  if (length(sq$Choices) >0L) {
    choices <- sapply(sq$Choices,function(ch) ch$Display)
    c(choices,rep(NA_character_,8-length(choices)))
  } else {
    rep(NA_character_,8)
  }
})
bchoices <- t(bchoices)
colnames(bchoices) <- paste(1:ncol(bchoices))
```

```
bmeta <- data.frame(
  name=paste(names(postBSQ[names(postBSQ)!="IMI"]),"pot",sep="."),
  qid=sapply(postBSQ[names(postBSQ)!="IMI"],function(sq) sq$QuestionID),
  qtag=sapply(postBSQ[names(postBSQ)!="IMI"],function(sq) sq$DataExportTag),
  text=sapply(postBSQ[names(postBSQ)!="IMI"],function(sq) sq$QuestionText),
  description=sapply(postBSQ[names(postBSQ)!="IMI"],function(sq) {
    if (is.null(sq$QuestionDescription))
      ""
    else
      sq$QuestionDescription
  }),
  choices=bchoices
)
```

Write it out

```
write_sheet(rbind(premeta,ectmeta,imimeta,potmeta),outID,"Metadata")
```

```
## v Writing to "_FSUS Fall 2022 Pre-post Data".
## v Writing to sheet 'Metadata'.
```