Anchor Skills Project

Robert Jackson

5/24/2018

Notes:

There are common core standard in the game. I standard per episode.

There are Anchor standards as well that are key to close reading and improving your writing.

All students starts at moderate with relation to Anchor Standards. You aren't graded until you get to an "activity".

After you get to **80%** correct the difficulty increases to **CHALLENGE** If a student drops to **70%** the difficulty drops to **SUPPORT** * The drop to support is specific to the Anchor standard. * CHALLENGE - has less clues and the task is a little more complex.

Nicole created an Excel file that tracks the leveling (CHALLENGE, MODERATE, SUPPORT)

Main interest in these data sets. * Interested in looking at the the Anchor skill trajectory of students through episodes.

- Interested to see how students perform in relations to the teacher activity level within the platform
  - We could group teacher activity by tiers (get started activities, reporting activities, feedback)

Note Ask Nicole, Why they "log into the platform" column is either "0" or "null" for a lot of the educators.

Part One:

Main Question We'll start by looking at student performanc as it relates to Anchor skills. I first want to figure out the data struture, which means I'll need to figure out some summary statistics as it relates the the "student leveling and task" data.

Note: The excel spreadsheet with the "student leveling" data has two tabs one titled: not null, and one titled: null, I believe this is something we asked Nicole to do weeks ago. Just as a note there are 28,873 null rows, versus the 271,814 rows in the other data-set. Just to be sure I'll run a missing value analysis before moving forward.

- not null: 271,814
- null: 28,873
- total: 300,687

```
x <- 28873 + 271814
x
```

```
## [1] 300687
```

```
X<-28873/300687
X
```

```
## [1] 0.09602344
```

Its just under the %10. I'll run the Litte'sMCar test.

```
library("BaylorEdPsych")
library("mvnmle")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("ggplot2")
```

```
Long_A <-read.csv("StudentLevelingLongform.noNull.csv")
Long_B <- read.csv("StudentLevelingLongform.null.csv")
Long_ALL <- rbind(Long_A, Long_B)
```

```
# Binding the two data sets, one with no nulls and the one with nulls. Were going to use the MissMech p
# missing1 <- LittleMCAR(Long_ALL)

#e
#[1] 176052.8

#$df
#[1] 10

#$p.value
#[1] 0

#$missing.patterns
#[1] 2

#$amount.missing
#                user_id play_iteration_id episode_id standard     instance support_level taskcount tas
#Number Missing        0                 0          0        0 2.887200e+04             0         0
#Percent Missing       0                 0          0        0 9.602075e-02             0         0
#                task_description why_important time_completed
#Number Missing                 0             0              0
#Percent Missing                0             0              0

#$data
#$data$DataSet1

#$data$DataSet2
```

I ran little's MCAR test and found that the p value is 0 which is good enough to reject the null hypothesis. While this does not prove the missing data is random, it does support that claim. I figured as much but its important to confirm. Moving on to analysis of the main data.

```
# Defining the function
my.summary <- function(x, na.rm=TRUE){
  result <- c(Mean=mean(x, na.rm=na.rm),
              SD=sd(x, na.rm=na.rm),
              Median=median(x, na.rm=na.rm),
              Min=min(x, na.rm=na.rm),
              Max=max(x, na.rm=na.rm),
              N=length(x))
```

```
}

# identifying numeric columns
ind <- sapply(Long_A, is.numeric)


# applying the function to numeric columns only
sapply(Long_A[, ind], my.summary)
```

```
##           user_id play_iteration_id   episode_id     instance    taskcount
## Mean    39643.386         34989.077 4.170695e+00 1.915390e+00 1.667713e+00
## SD       4267.243          3441.949 3.302641e+00 1.272868e+00 1.251006e+00
## Median  39931.000         34940.000 3.000000e+00 1.000000e+00 1.000000e+00
## Min      1534.000            73.000 1.000000e+00 1.000000e+00 1.000000e+00
## Max     46996.000         99999.000 1.200000e+01 1.300000e+01 2.400000e+01
## N      271813.000        271813.000 2.718130e+05 2.718130e+05 2.718130e+05
##             task_id
## Mean      113.52858
## SD         94.12627
## Median     90.00000
## Min         1.00000
## Max       327.00000
## N      271813.00000
```

```
rapply(Long_A,function(x)length(unique(x)))
```

```
##           user_id play_iteration_id        episode_id          standard
##              2021              9368                12                 3
##          instance     support_level         taskcount           task_id
##                13                 3                24               304
##  task_description     why_important    time_completed
##               155                12             36995
```

SUMMARY STATISTICS TAKEAWAYS

- There are 2021 users
- Max 12 episodes to play (which makes sense)
- There are only 3 Unique standards being assesed
- Instance is the categorical (dummy) variable for the support level
- There are 304 unique tasks
- This will be a summary table of the numerical statistics. It doesn't have any information on the factors data which has some great information as well. I'll have to create categorical variables that are dummy coded with that data.

Things we can look at:

- The time each student commits to completing each episode (tasks)
- What percentage of students stuggle with which standard
- Are there any variables that correlate to changes in support levels

Nice to Ponder: Could we use the variables we have to build out factor analysis/PCA for student profiles?

```
test <- filter(Long_A, user_id == 10899)

rapply(test,function(x)length(unique(x)))
```

```
##           user_id play_iteration_id        episode_id          standard
##                 1                 7                 7                 3
##          instance     support_level         taskcount           task_id
##                 3                 2                 6               114
##  task_description     why_important    time_completed
##                73                 6               116
```

The time each student commits to completing each episode (tasks)

- Were first going to look into the amount of time students commit to each episode.
- Then we'll run some exploratory viz to see trends if possible
- After that we'll start to look at how students perform and how that changes over time.

I'll build out the variables needed to visualize them. Things I want clarify:

- How many episodes were completed?
- How are students shifting in their anchor scores?
- Which episodes are leading to the shifts in score?
- Are those shifts 21st century skill specific?

```
#testing whether or not I can count how many episodes students have completed.  Success! Now to add the
test$Completed <- ifelse(test$task_description == "Complete Episode Decision", 1, 0)
```

```
Long_A$Completed <- ifelse(Long_A$task_description == "Complete Episode Decision", 1, 0)
```

```
# Time to make an indicator of when a student's anchor skill shifts from main to another support level.

# 1st - I need to make a variable that counts the differences in an instance against the previous insta

# 2nd - Use dyplr to create another variable that actually counts that differnce and indicate that ever

# * How are students shifting in their anchor scores?
# * Which eps/ 21st century skills are being assessed in those score shifts?
```

Test out the steps above.

```
test2 <-test %>%
  group_by(user_id, idx = cumsum(instance == "main")) %>%
  mutate(counter = row_number()) %>%
  ungroup %>%
  select(-idx)
```

```
test2 <- dplyr::select(test2, user_id, episode_id, standard, instance, support_level, Completed , counte
```

```
test2 <- test2 %>%
    group_by(user_id) %>%
    mutate(Diff = counter - lag(counter))
```

Try it with actual data. I'll need to see if it actual recognizes the id variable as well and restart the counter.

```r
Long_A <-Long_A %>%
  group_by(user_id, idx = cumsum(support_level == "main")) %>%
  mutate(counter = row_number()) %>%
  ungroup %>%
  select(-idx)
```

```r
Long_A2 <- dplyr::select(Long_A, user_id, episode_id, standard, instance, support_level, Completed , cou
```

```r
Long_A3 <- Long_A2 %>%
    group_by(user_id) %>%
    mutate(Diff = counter - lag(counter))
```

```r
Long_A3 <- Long_A3 %>%
  mutate(
  Challenge = ifelse(
  support_level == "challenge" & counter == 2,
  1,
  ifelse(
  support_level == "support 1" & counter == 2, -1, 0)
  ))

# If the challenge column has a 1 this means that a student has moved from "main" to "challenge" on a p

# If the challenge column has a -1 this mean that a student ahs moved from "main" to "support" on a par
```

```r
# Defining the function
my.summary <- function(x, na.rm=TRUE){
  result <- c(Mean=mean(x, na.rm=na.rm),
              SD=sd(x, na.rm=na.rm),
              Median=median(x, na.rm=na.rm),
              Min=min(x, na.rm=na.rm),
              Max=max(x, na.rm=na.rm),
              N=length(x))
}

# identifying numeric columns
ind <- sapply(Long_A3, is.numeric)


# applying the function to numeric columns only
sapply(Long_A3[, ind], my.summary)
```

```
##           user_id   episode_id     instance    Completed      counter
## Mean    39643.386 4.170695e+00 1.915390e+00 2.173553e-02     18.23017
## SD       4267.243 3.302641e+00 1.272868e+00 1.458190e-01     38.12396
## Median  39931.000 3.000000e+00 1.000000e+00 0.000000e+00      1.00000
## Min      1534.000 1.000000e+00 1.000000e+00 0.000000e+00      1.00000
## Max     46996.000 1.200000e+01 1.300000e+01 1.000000e+00    475.00000
## N      271813.000 2.718130e+05 2.718130e+05 2.718130e+05 271813.00000
##                 Diff     Challenge
## Mean    1.782521e-01 -1.894685e-03
## SD      3.772928e+00  8.823198e-02
## Median  0.000000e+00  0.000000e+00
## Min    -3.830000e+02 -1.000000e+00
## Max     1.000000e+00  1.000000e+00
```

```
## N       2.718130e+05  2.718130e+05
```

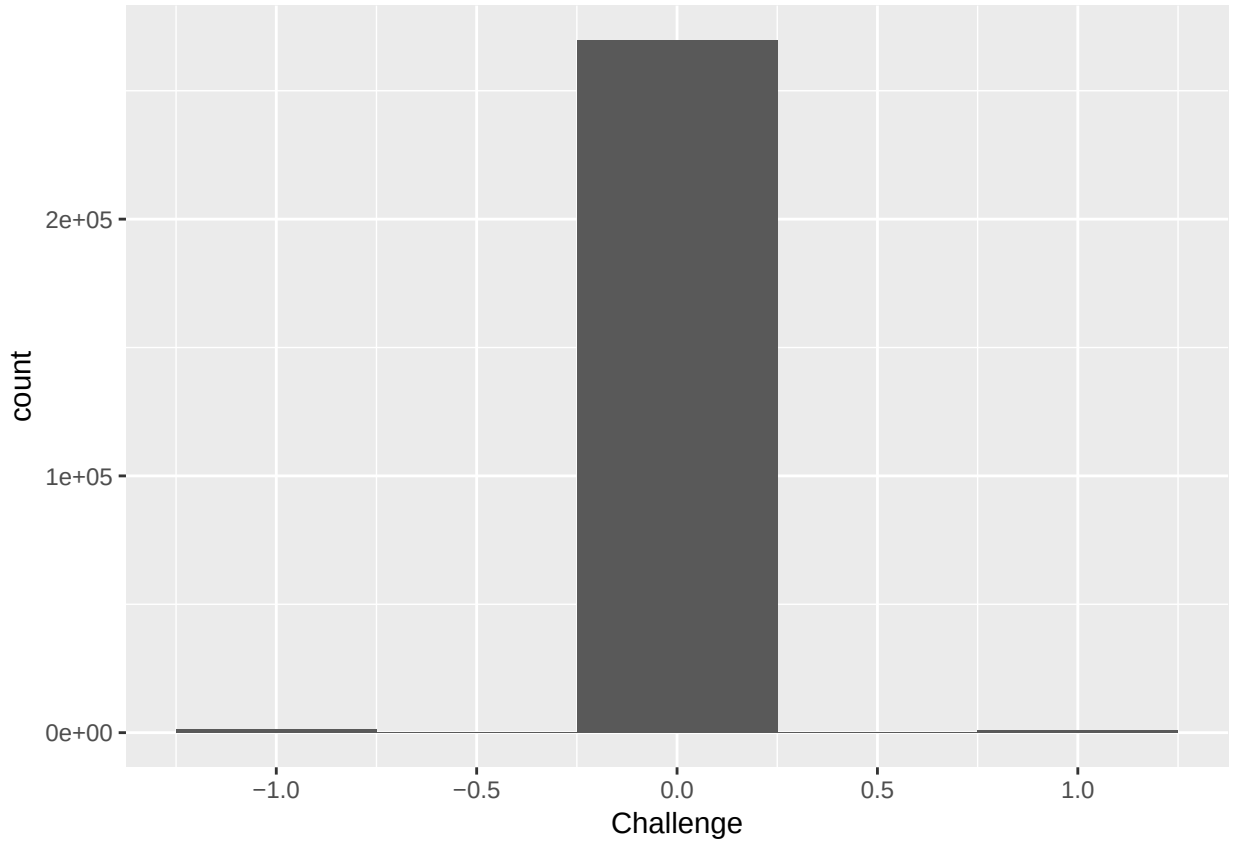
```
Long_Viz <- dplyr::filter(Long_A3, Challenge != 0)
```

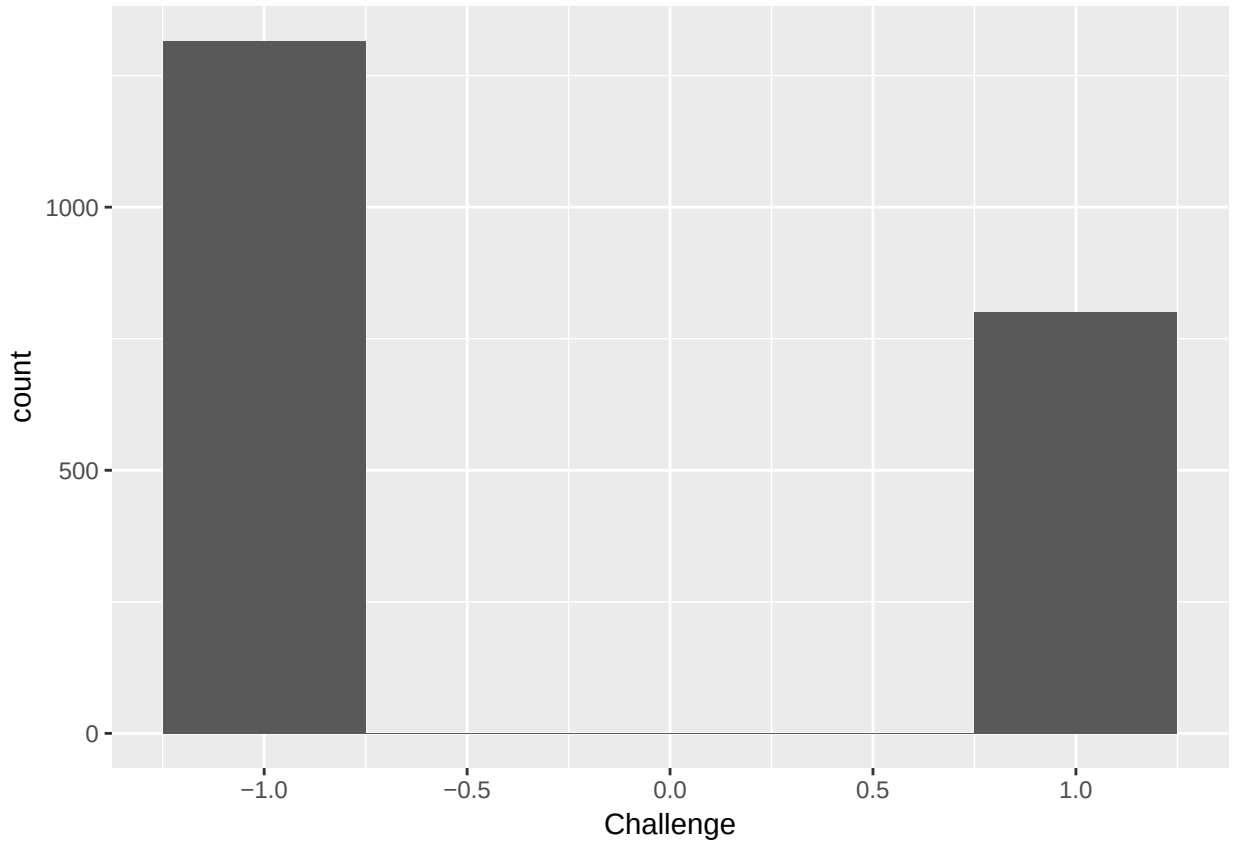
Summary Statistics

```
##          user_id  episode_id     instance Completed counter Diff
## Mean   39566.488    4.949929    2.3660841         0       2    1
## SD      3963.619    2.722230    0.8764092         0       0    0
## Median 39838.000    5.000000    2.0000000         0       2    1
## Min    10899.000    1.000000    1.0000000         0       2    1
## Max    46811.000   12.000000   12.0000000         0       2    1
## N       2117.000 2117.000000 2117.0000000      2117    2117 2117
##           Challenge
## Mean     -0.2432688
## SD        0.9701881
## Median   -1.0000000
## Min      -1.0000000
## Max       1.0000000
## N      2117.0000000
```

```
# The majority of student actions are in normal support mode.  This is evident by the histograms below
```

```
ggplot(data = Long_A3) + geom_histogram(mapping = aes(x = Challenge), binwidth = 0.5)
```






```
rapply(Long_Viz,function(x)length(unique(x)))
```

```
##       user_id    episode_id      standard      instance support_level
##          1194            12             3            12             2
##     Completed       counter          Diff     Challenge
##             1             1             1             2
```

```
# Below are the standards that had a support level shift.

with(Long_Viz, table(standard, Challenge))
```

```
##           Challenge
## standard    -1   1
##   CCRA.R.4 451 592
##   CCRA.R.8 515 134
##   CCRA.R.9 350  75
```

```
Long_M <- Long_A %>%
    group_by(user_id) %>%
    mutate(Diff = counter - lag(counter))
```

```
Long_M <- Long_M %>%
  mutate(
  Challenge = ifelse(
  support_level == "challenge" & counter == 2,
  1,
  ifelse(
  support_level == "support 1" & counter == 2, -1, 0)
  ))
```

```
Long_M<-Long_M %>%
  group_by(user_id) %>%
  mutate(count = n_distinct(episode_id))

Long_M<-dplyr::rename(Long_M, Ep_played=count)
```

```
ep5 <- dplyr::filter(Long_M, episode_id == 5)
```

```
rapply(ep5,function(x)length(unique(x)))
```

```
##           user_id play_iteration_id        episode_id          standard
##               673               698                 1                 1
##          instance     support_level         taskcount           task_id
##                 9                 3                 8                26
##  task_description     why_important    time_completed         Completed
##                26                 2              5231                 2
##           counter              Diff         Challenge         Ep_played
##               207                32                 3                12
```

To see how many students are either in "support" or "challenge" by episode

- -1 are the students who are in support.
- 1 are the students who are in challenge.

```
test4 <- Long_M

Long_Chall <- dplyr::filter(Long_M, Challenge != 0)

with(Long_Chall, table(episode_id, Challenge))
```

```
##           Challenge
## episode_id  -1   1
##          1   95  16
##          2   31  34
##          3  340 468
##          4   12  10
##          5  427  71
##          6   56  69
##          7   43  26
##          8  143  20
##          9   33  27
##          10  57  17
##          11  24  21
##          12  55  22
```

```
Long_M <-Long_M %>%
  group_by(episode_id) %>%
  mutate(count = n_distinct(user_id))
```

Number of students playing each episode *Open dataframe StudentsWhoPlayed to see how many student have played each episode.*

```
# Open dataframe StudentsWhoPlayed to see how many student have played each episode.
Long_M <- dplyr::select(Long_M, episode_id, count)
StudentsWhoPlayed <- dplyr::distinct(Long_M)

StudentsWhoPlayed <- dplyr::rename(StudentsWhoPlayed, NumStudents = count)
```

```
print(StudentsWhoPlayed)
```

```
## # A tibble: 12 x 2
## # Groups:   episode_id [12]
##    episode_id NumStudents
##         <int>       <int>
##  1          1        1805
##  2          2        1436
##  3          3        1139
##  4          4         829
##  5          5         673
##  6          6         666
##  7          7         479
##  8          8         498
##  9          9         332
## 10         10         365
## 11         11         284
## 12         12         305
```

ANSWERED: How are students shifting in their anchor scores?

As evident by the first histogram above (Lines 287 - 288) majority of student actions are in main.

As evident by the second histogram above (lines 293 -294) out of the 1194 students who change in support level, majority are moving into support.

There are 2117 instances where a student moves from main to either "support" or "challenge". This could be the same student multiple times not that each moment is a new student moving standards.

There are 1194 students who's support level changes out of the 2021 total unique ids. (59% shift in their support level)

*Overwhelmingly students that shift in their support levels are moving towards "support 1" for CCRA.R.8 & CCRA.R.9.

ANSWERED: Which eps/ 21st century skills are being assessed in those score shifts?

Anchor support level shifts by episode

```
Challenge
```

episode_id -1 1 1 95 16 2 31 34 3 340 468 4 12 10 5 427 71 6 56 69 7 43 26 8 143 20 9 33 27 10 57 17 11 24 21 12 55 22

The most students who have a shift in support completed episode 3 there is a balanced split between moving to "support" and moving to "challenge".

Ep. 5 has a large disparity in support level shifts. There are 497 students who have completed this episode and 86% of students shift to support out of main.

There is no real connection I can see between anchor skill support change and 21st century skills, but I'll need to dig deeper to confirm.

ANSWERED: How many students played each episode episode_id NumStudents

1 1805
2 1436
3 1139
4 829
5 673
6 666

7 479
8 498
9 332
10 365 11 284
12 305