Meeting agenda

DS 5165:

Name: GatesProject_DL_OC_Fall2019_Practice_Cleaned_AddGaming

Step 1: run iAFM models

Model iAFM: with opportunity as the fixed effect

Formula: glmer(response ~ opportunity0 + (opportunity0|KC) + (opportunity0|individual), data=., family=binomial(), nAGQ = 0))

Model summary model

ds5165	# records	AIC	BIC	Pseudo-R^2 (fixed effects)	Pseudo-R^2 (total)	Intercept	coefficient
iAFM	32458	37774.13	37841.23	0.04	0.56	-0.27	0.04

Model params (ranef(model_iafm))
model_iafm_param.xlsx

Model reversed iAFM: iAFM with reverse_opportunity as the fixed effect

Formula: glmer(response ~ reverse_opportunity + (reverse_opportunity|KC) + (reverse_opportunity|individual), data=., family=binomial(), nAGQ = 0))

Model summary model

ds5165	# records	AIC	BIC	Pseudo-R^2 (fixed effects)	Pseudo-R^2 (total)	Intercept	coefficient
iAFM_reverse	32458	37811.43	37878.53	0.09	0.52	0.1	-0.06

Model params (ranef(model_iafm_reverse))

model_iafm_reverse_param.xlsx

Step 1.5: get PredAvgiAFM

Uses all of the parameter estimates (from step 1) and their maximum opportunity on each KC to predict an end of instruction state for each student on each KC. And then averages across KCs to get a single predicted value per student (PredAvgiAFM).

Code

[10]:
```
# Maximum opportunity on each KC
predict_data = my_data %>%
      group_by(individual, KC) %>%
      slice(which.max(opportunity0))
```

[11]:
```
# Predict an end of instruction state for each student on each KC
predict_data$pred_iafm = predict(model_iafm, predict_data,type="response", allow.new.levels=TRUE)
```

[12]:
```
# Average across KCs to get a single predicted value per student
PredictedScores = predict_data %>%
  group_by(individual) %>%
  summarise(
    PredAvgiAFM = mean(pred_iafm),
)
```

[14]:
```
# Export the predicted value to a CSV file
write.csv(PredictedScores, file = "/kaggle/working/predicted.csv")
```

Predicted value dataframe

predicted.xlsx

get TotalOpportunity

Sum-up the max opportunity for each student on each KC

Code

```
# Total opportunity per student
total_opportunity = predict_data %>%
  group_by(individual) %>%
  summarise(
    TotalOpportunity = sum(opportunity),
)
# Export the predicted value to a CSV file
write.csv(total_opportunity, file = "/kaggle/working/total_opportunity.csv")
```

Total Opportunity dataframe

total_opportunity.xlsx

Step 2: Do iAFM or reverse iAFM student parameters and prediction better predict the post-test?

```
# Model 1: pretest only
test_scores %>%
  lm(Posttest ~ Pretest, data = .) %>%
  summ()

# Model 2: pretest + PredAvgiAFM
test_scores %>%
  lm(Posttest ~ PredAvgiAFM + Pretest, data = .) %>%
  summ()

# Model 3: pretest + int_iAFM
test_scores %>%
  lm(Posttest ~ int_iAFM + Pretest, data = .) %>%
  summ()

# Model 4: pretest + int_iAFM_reverse
test_scores %>%
  lm(Posttest ~ int_iAFM_reverse + Pretest, data = .) %>%
  summ()

# Model 5: pretest + int_iAFM + int_iAFM_reverse
test_scores %>%
  lm(Posttest ~ int_iAFM + int_iAFM_reverse + Pretest, data = .) %>%
  summ()
```

Summary of models

Model	# students	F-statistic	R-squared	Adjusted R-squared	p	AIC	BIC	log-likelihood
1: pretest	129	71.18	0.36	0.35	0.00	-97.6802212647543	-89.1007840516693	51.84011 (df=3)
2: pretest + PredAvgiAFM	129	49.19	0.44	0.43	0.00	-112.715518804335	-101.276269186888	60.35776 (df=4)
3: pretest + int_iAFM	129	**84.03**	**0.57**	**0.56**	0.00	**-147.600241987266**	**-136.160992369819**	77.80012 (df=4)
4: pretest + int_iAFM_reverse	129	75.66	0.55	0.54	0.00	-140.04307108267	-128.603821465223	74.02154 (df=4)
5: pretest + int_iAFM + int_iAFM_reverse	129	55.73	**0.57**	**0.56**	0.00	-145.806850497986	-131.507788476178	77.90343 (df=5)

Model	AIC	BIC	log-likelihood
1: pretest	-97.6802212647543	-89.1007840516693	51.84011 (df=3)
2: pretest + PredAvgiAFM	-112.715518804335	-101.276269186888	60.35776 (df=4)
3: pretest + int_iAFM	-147.600241987266	-136.160992369819	77.80012 (df=4)
4: pretest + int_iAFM_reverse	-140.04307108267	-128.603821465223	74.02154 (df=4)
5: pretest + int_iAFM + int_iAFM_reverse	-145.806850497986	-131.507788476178	77.90343 (df=5)

Model Statistics

Model 1: pretest only

MODEL INFO:
Observations: 129
Dependent Variable: Posttest
Type: OLS linear regression

MODEL FIT:
$F(1,127) = 71.18$, $p = 0.00$
$R^2 = 0.36$
Adj. $R^2 = 0.35$

Standard errors: OLS

	Est.	S.E.	t val.	p
(Intercept)	0.26	0.03	8.60	0.00
Pretest	0.65	0.08	8.44	0.00

el

Model 2: pretest + PredAvgiAFM

MODEL INFO:
Observations: 129
Dependent Variable: Posttest
Type: OLS linear regression

MODEL FIT:
$F(2,126) = 49.19$, $p = 0.00$
$R^2 = 0.44$
Adj. $R^2 = 0.43$

Standard errors: OLS

	Est.	S.E.	t val.	p
(Intercept)	0.09	0.05	1.86	0.07
PredAvgiAFM	0.41	0.10	4.22	0.00
Pretest	0.46	0.09	5.34	0.00

Model 3: pretest + int_iAFM

MODEL FIT:
F(2,126) = 84.03, p = 0.00
$R^2 = 0.57$
Adj. $R^2 = 0.56$

Standard errors: OLS

	Est.	S.E.	t val.	p
(Intercept)	0.40	0.03	13.12	0.00
int_iAFM	0.12	0.01	7.90	0.00
Pretest	0.25	0.08	3.08	0.00

MODEL INFO:
Observations: 129
Dependent Variable: Posttest
Type: OLS linear regression

Model 4: pretest + int_iAFM_reverse

MODEL FIT:
F(2,126) = 75.66, p = 0.00
$R^2 = 0.55$
Adj. $R^2 = 0.54$

Standard errors: OLS

	Est.	S.E.	t val.	p
(Intercept)	0.39	0.03	12.46	0.00
int_iAFM_reverse	0.12	0.02	7.19	0.00
Pretest	0.28	0.08	3.34	0.00

Model 5: pretest + int_iAFM + int_iAFM_reverse

MODEL FIT:
F(3,125) = 55.73, p = 0.00
$R^2 = 0.57$
Adj. $R^2 = 0.56$

Standard errors: OLS

	Est.	S.E.	t val.	p
(Intercept)	0.40	0.03	13.04	0.00
int_iAFM	0.10	0.04	2.78	0.01

```
int_iAFM_reverse         0.02   0.04     0.45   0.66
Pretest                  0.24   0.08     2.99   0.00
--------------------------------------------------
```

Pairwise ANOVA Tests

Model 1: pretest only v.s. Model 2: pretest + PredAvgiAFM

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	127	3.381139	NA	NA	NA
2	126	2.962863	1	0.4182759	2.46964e-05

Model 1: pretest only v.s. Model 3: pretest + int_iAFM

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	127	3.381139	NA	NA	NA
2	126	2.260830	1	1.120309	2.751333e-15

Model 1: pretest only v.s. Model 4: pretest + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	127	3.381139	NA	NA	NA
2	126	2.397232	1	0.9839074	6.417744e-13

Model 1: pretest only v.s. Model 5: pretest + int_iAFM + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	127	3.381139	NA	NA	NA
2	125	2.257212	2	1.123927	3.051798e-14

Model 2: pretest + PredAvgiAFM v.s. Model 3: pretest + int_iAFM

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	126	2.962863	NA	NA	NA
2	126	2.260830	0	0.7020335	NA

Model 2: pretest + PredAvgiAFM v.s. Model 4: pretest + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	126	2.962863	NA	NA	NA
2	126	2.397232	0	0.5656315	NA

Model 2: pretest + PredAvgiAFM v.s. Model 5: pretest + int_iAFM + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	126	2.962863	NA	NA	NA
2	125	2.257212	1	0.7056516	4.072877e-10

Model 3: pretest + int_iAFM v.s. Model 4: pretest + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	126	2.260830	NA	NA	NA
2	126	2.397232	0	-0.1364019	NA

Model 3: pretest + int_iAFM v.s. Model 5: pretest + int_iAFM + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	126	2.260830	NA	NA	NA
2	125	2.257212	1	0.003618084	0.6544284

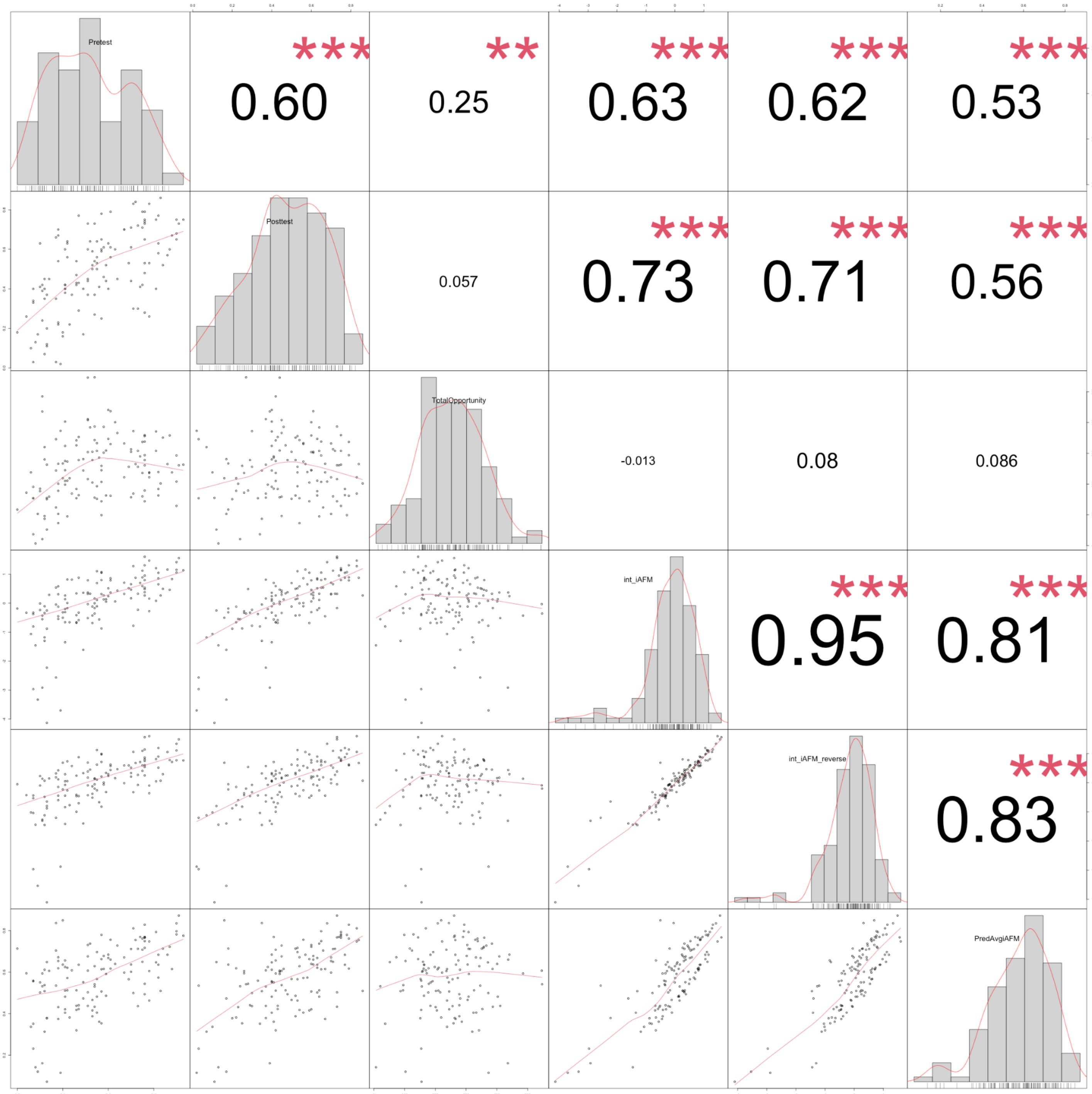
Model 4: pretest + int_iAFM_reverse v.s. Model 5: pretest + int_iAFM + int_iAFM_reverse

	Res.Df	RSS	Df	Sum of Sq	Pr(>Chi)
1	126	2.397232	NA	NA	NA
2	125	2.257212	1	0.14002	0.00535926

Correlation Matrix

	Pretest	Posttest	TotalOpportunity	int_iAFM	int_iAFM_reverse	PredAvgiAFM
Pretest	1.0000000	0.5993182	0.24872107	0.62535623	0.62235234	0.53417269
Posttest	0.5993182	1.0000000	0.05715290	0.73436133	0.71099652	0.55816121
TotalOpportunity	0.2487211	0.0571529	1.00000000	-0.01257679	0.08021127	0.08569131
int_iAFM	0.6253562	0.7343613	-0.01257679	1.00000000	0.94752788	0.81418203
int_iAFM_reverse	0.6223523	0.7109965	0.08021127	0.94752788	1.00000000	0.82673920
PredAvgiAFM	0.5341727	0.5581612	0.08569131	0.81418203	0.82673920	1.00000000

Correlation Chart



Alternative - 1 parameter fit

1. Create a table with both pre and post in separate rows for each student

Student Test-Time Test-Score Process-Model-Prediction1 Process-Model-Prediction2
S1 Pre .4 prob(-1.1) [intercept_iAFM] prob(-1.1) [intercept_iAFM]
S1 Post .6 prob(.4) [intercept_iAFM_reverse] prob(.34) [max-Opp-iAFM??]
S2 …

Insert a link to the resulting cvs table:
https://drive.google.com/file/d/11GUuKK5f3DzxHrInLmFGknuOyv4KBC_t/view?usp=drive_link

2. Run analyses
   a. Two parameter version:
      Model1: Test-Score ~ Process-Model-Prediction1 [+ Intercept]
      lm(TestScore ~ ProcessModelPrediction1, data = .)

```
MODEL INFO:
Observations: 258
Dependent Variable: TestScore
Type: OLS linear regression

MODEL FIT:
F(1,256) = 212.07, p = 0.00
R² = 0.45
Adj. R² = 0.45

Standard errors: OLS
------------------------------------------------------------
                                  Est.   S.E.   t val.      p
----------------------------- ------- ------ -------- ------
(Intercept)                      -0.00   0.03    -0.15   0.88
ProcessModelPrediction1           0.69   0.05    14.56   0.00
------------------------------------------------------------
```

      Model2: Test-Score ~ Process-Model-Prediction2 [+ Intercept]

   b. One parameter version:
      Model3: Test-Score ~ 1* Process-Model-Prediction1 [+ Intercept]
      Model4: Test-Score ~ 1* Process-Model-Prediction2 [+ Intercept]

Model	# students	F-statistic	R-squared	Adjusted R-squared	p	AIC	BIC	log-likelihood
1	129	212.07	0.45	0.45	0.00	-282.640448 455066	-271.981569 700301.	144.3202 (df=3)
2	129	145.04	0.36	0.36	0.00	-242.764374 115557	-232.105495 360792	124.3822 (df=3)
3	129	NA	0.36	NA	NA	-244.845176	-237.739257	124.4226

						738824	568981	(df=2)
4	129	NA	0.24	NA	NA	-198.729029494008	-191.623110324164	101.3645 (df=2)

Interpretation

Which is better using reverse_opportunity or avg_max_opportunity?
Reverse_opportunity (Process-Model-Prediction1) is “probably better” avg_max_opportunity (Process-Model-Prediction2)

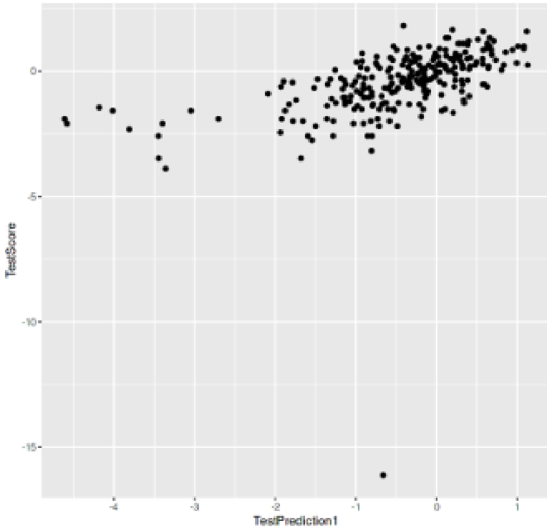
- Higher R2 and lower AIC and BIC

3. Re-run Analysis With Log-Odds
   Model1: LogOdds(Test-Score) ~ Process-Model-Prediction1 [+ Intercept]
   Model2: LogOdds(Test-Score) ~ Process-Model-Prediction2 [+ Intercept]
   Model3: LogOdds(Test-Score) ~ 1* Process-Model-Prediction1 [+ Intercept]
   Model4: LogOdds(Test-Score) ~ 1* Process-Model-Prediction2 [+ Intercept]

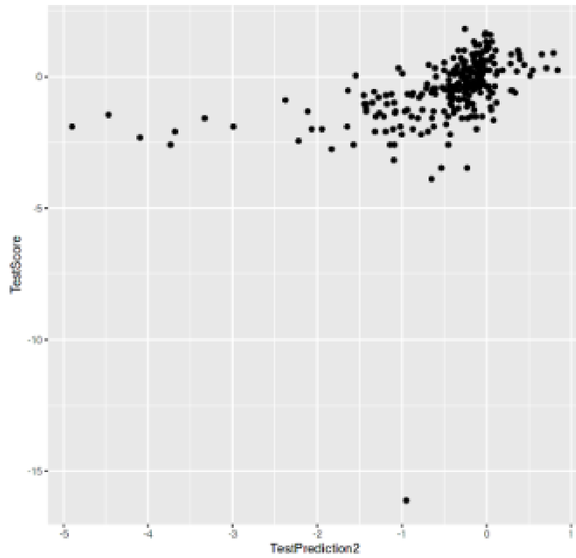
Model	# students	F-statistic	R-squared	Adjusted R-squared	p	AIC	BIC	log-likelihood
1	129	77.50	0.23	0.23	0.00	852.955767299429	863.614646054194	-423.4779 (df=3)
2	129	56.38	0.18	0.18	0.00	869.835902329846	880.494781084611	-431.918 (df=3)
3	129	NA	0.195	NA	NA	863.096944814295	870.202863984138	-429.5485 (df=2)
4	129	NA	0.165	NA	NA	872.419728399524	879.525647569367	-434.2099 (df=2)

Scatter Plots of TestScore vs Prediction

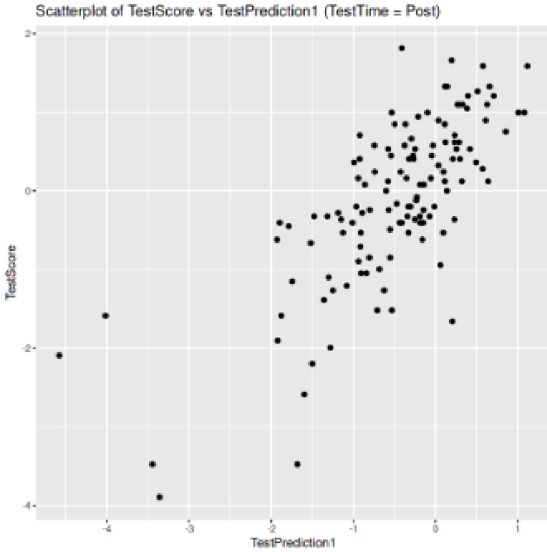
Scatterplot of TestScore vs TestPrediction1 (Overall)



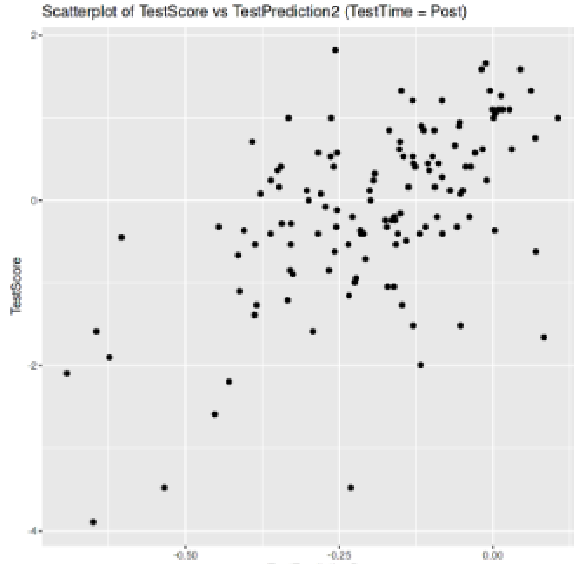
Scatterplot of TestScore vs TestPrediction2 (Overall)



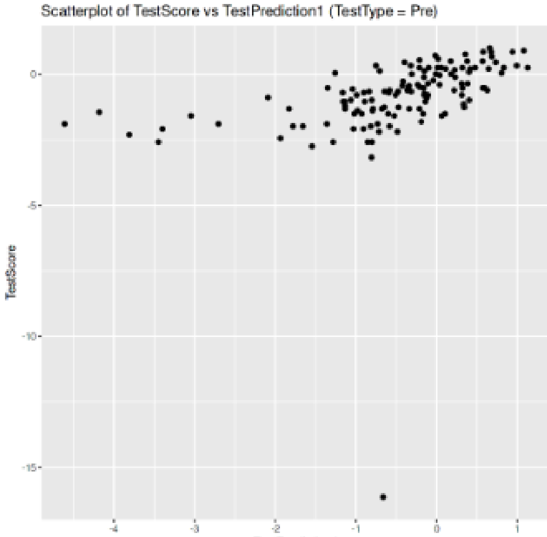
Scatterplot of TestScore vs TestPrediction1 (TestTime = Post)



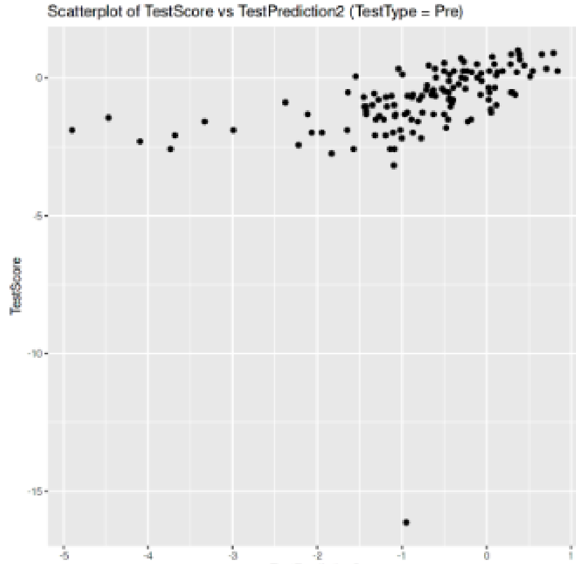
Scatterplot of TestScore vs TestPrediction2 (TestTime = Post)



Scatterplot of TestScore vs TestPrediction1 (TestType = Pre)



Scatterplot of TestScore vs TestPrediction2 (TestType = Pre)



Does adding total opportunity better predict the post-test?

```
# Model 1.2: pretest + TotalOpportunity
test_scores %>%
  lm(Posttest ~ TotalOpportunity + Pretest, data = .) %>%
  summ()

# Model 2.2: pretest + PredAvgiAFM + TotalOpportunity
test_scores %>%
  lm(Posttest ~ TotalOpportunity + PredAvgiAFM + Pretest, data = .) %>%
  summ()

# Model 3.2: pretest + int_iAFM + TotalOpportunity
test_scores %>%
  lm(Posttest ~ TotalOpportunity + int_iAFM + Pretest, data = .) %>%
  summ()

# Model 4.2: pretest + int_iAFM_reverse + TotalOpportunity
test_scores %>%
  lm(Posttest ~ TotalOpportunity + int_iAFM_reverse + Pretest, data = .) %>%
  summ()
```

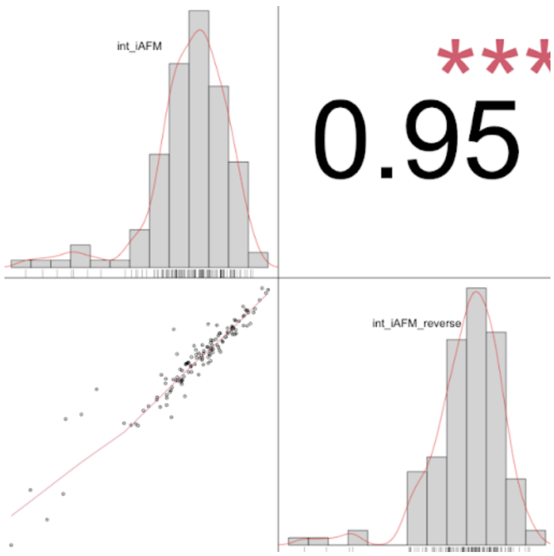
Model	# students	F-statistic	R-squared	Adjusted R-squared	p
pretest + totalopp	129	36.71	0.37	0.36	0.00
pretest + PredAvgiAFM+ totalopp	129	33.36	0.44	0.43	0.00
pretest + int_iAFM+ totalopp	129	55.59	0.57	0.56	0.00
pretest + int_iAFM_reverse+ totalopp	129	50.60	0.55	0.54	0.00

Compared to results of models without total opportunity, the R-squared are basically the same, but the F-statistic is significantly lower. "TotalOpportunity" does not significantly improve the model's ability to predict Posttest scores when controlling for the other predictors.

Log-likelihood AIC BIC

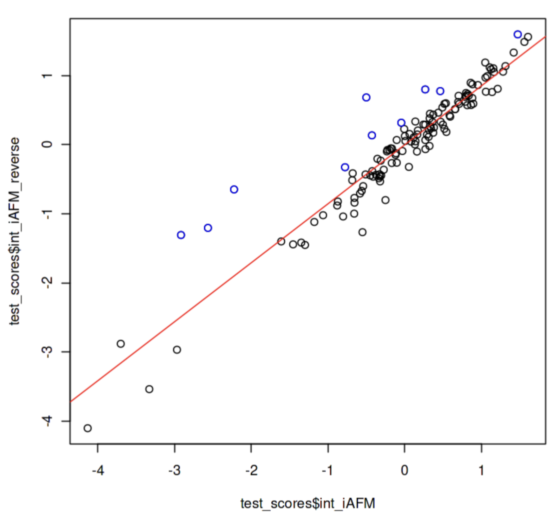
Identify and analyze “overachievers”

- Background: *int_iAFM* and *int_iAFM_rev*erse are highly correlated: students with good initial scores will have better final scores



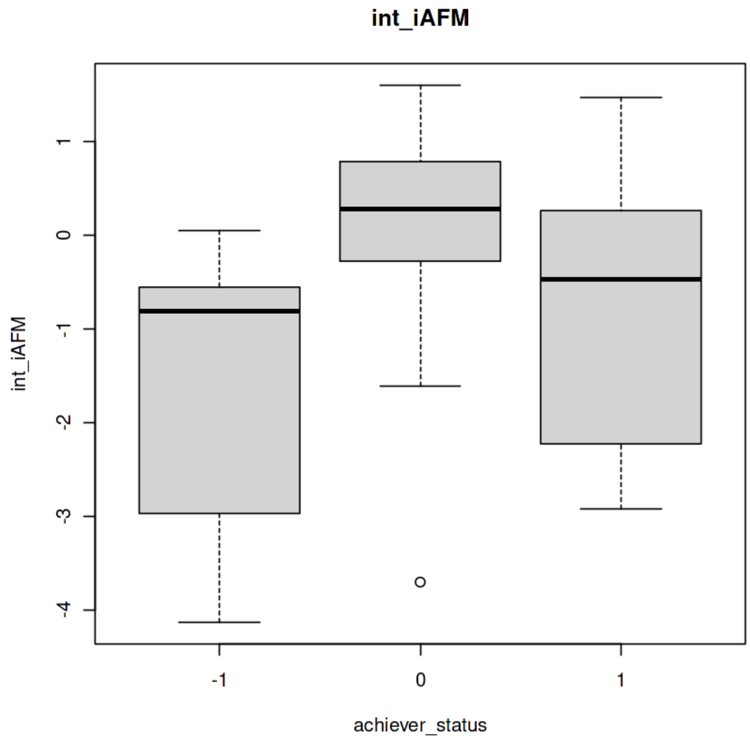
- Definition of overachievers and underachievers

```r
# Assuming you have a linear model fit
linear_model <- lm(int_iAFM_reverse ~ int_iAFM, data = test_scores)
# Predict values using the linear model
predicted_values <- predict(linear_model, newdata = test_scores)
# Set a threshold
threshold_difference <- 0.3
# Calculate the absolute difference between actual and predicted values
difference <- test_scores$int_iAFM_reverse - predicted_values
# Create a achiever status column
# 1 - overachiever, -1 - underachiever, 0 - otherwise
test_scores$achiever_status <- ifelse(difference > threshold_difference, 1,
                                      ifelse(difference < -threshold_difference, -1, 0))
```

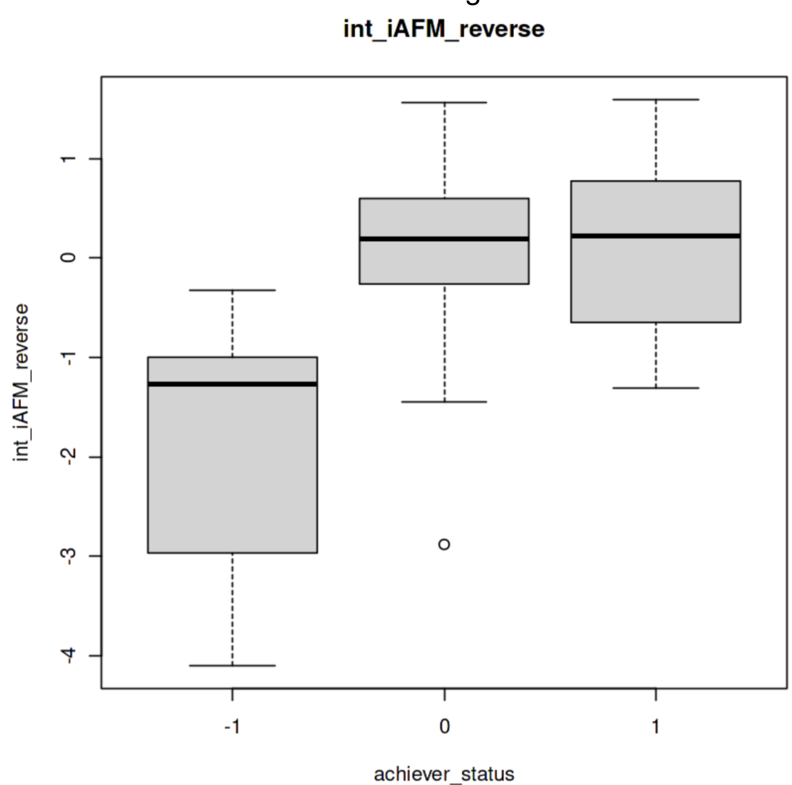


draw y = x

- Analysis:
    - Key variables between among different achiever status
        - Overachievers have lower initial knowledge: more room to improve

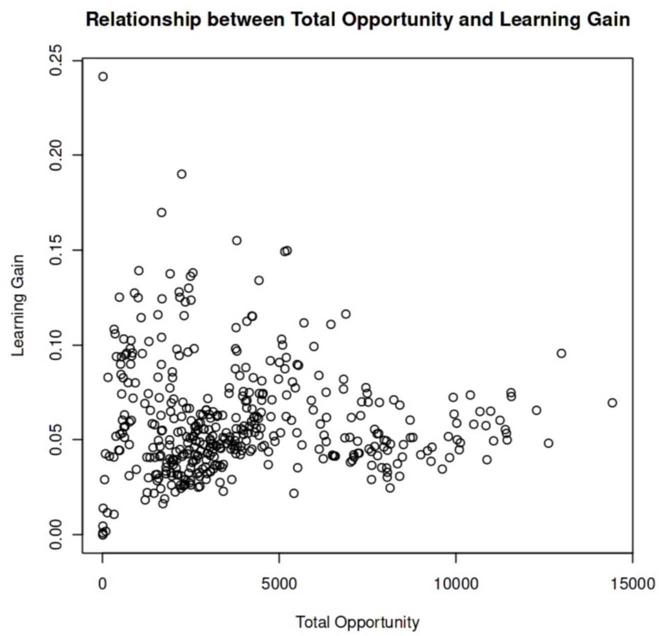


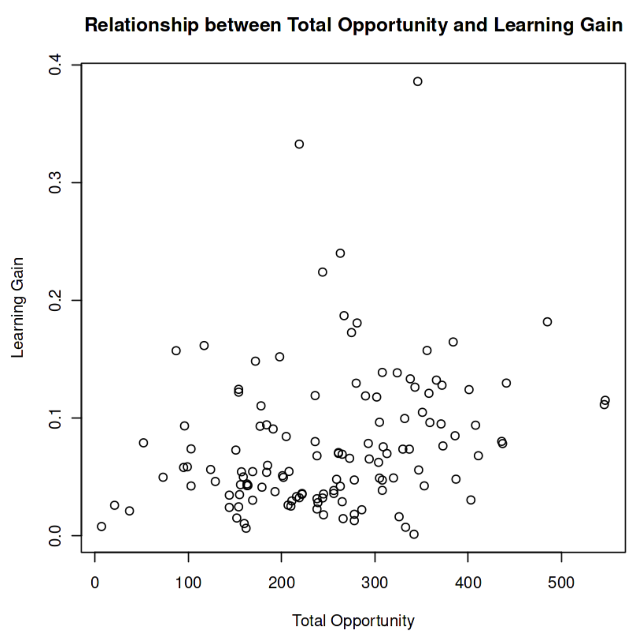
        - Overachievers have similar knowledge in the end as normal students

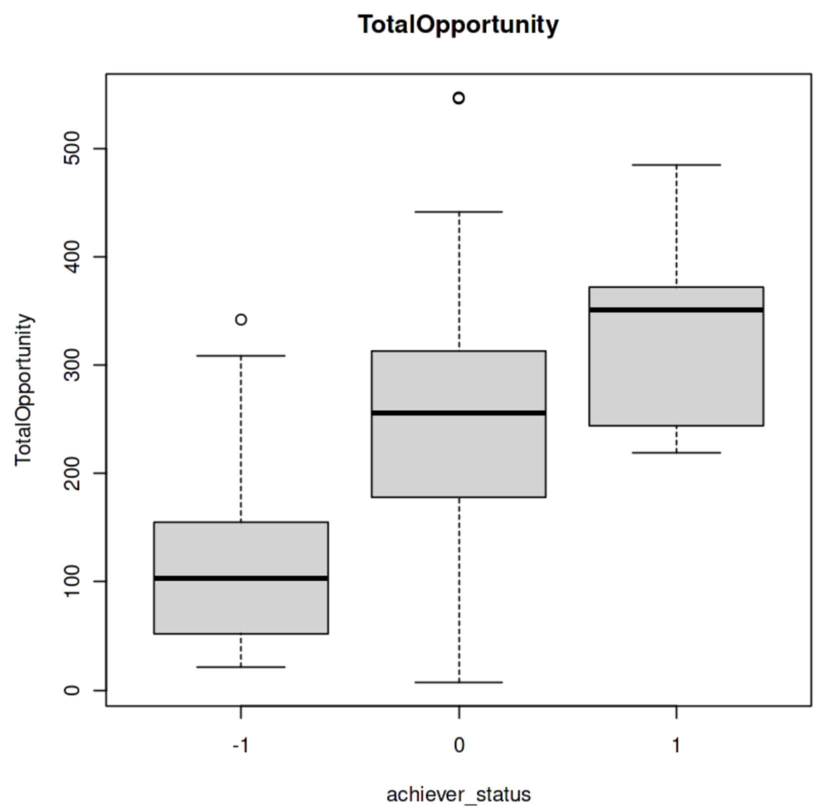


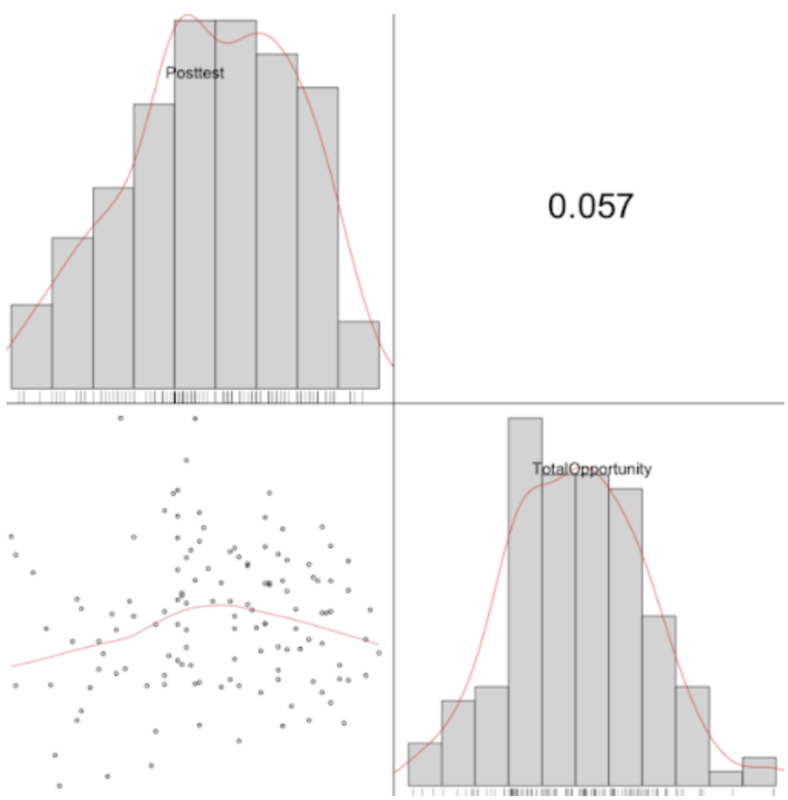
- Total opportunity: doing more problems makes a student an overachiever
  Measure of learning: post - pre
  Int_reverse - int
  preiAfm(max_opp) - avg(preiAfm(0) | KC)

```
pred_initial = predict(model_iafm,initial_data,type="response",allow.new.levels=TRUE)
pred_iafm = predict(model_iafm,ds_predict,type="response",allow.new.levels=TRUE)
```

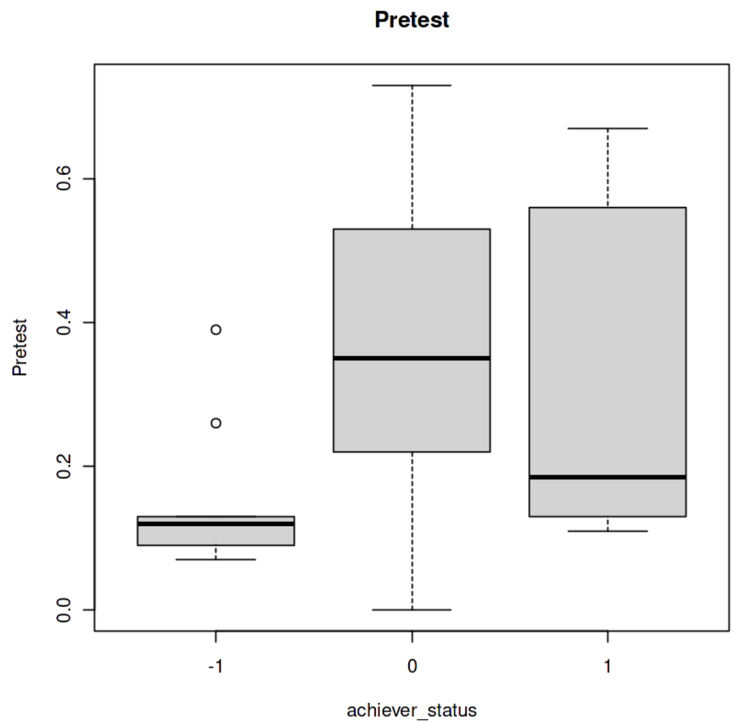




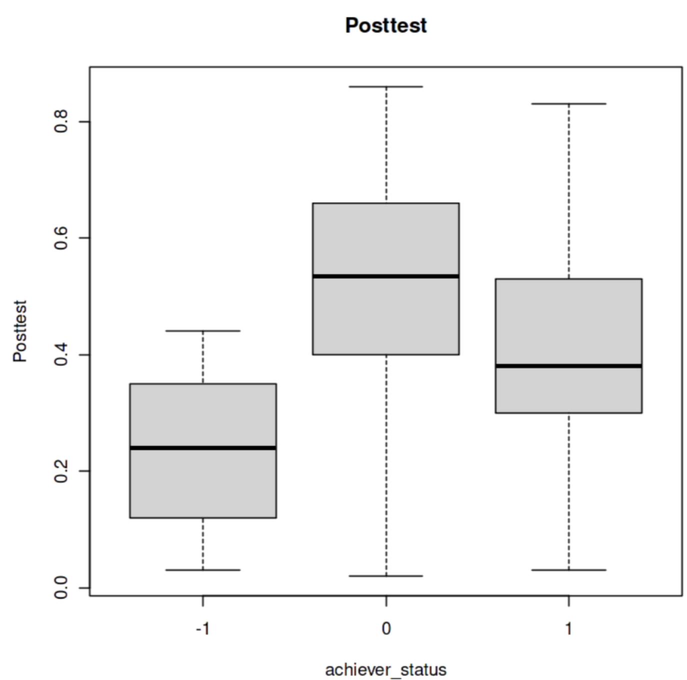




- Maybe we could identify the potential underachiever at the beginning of the semester according to the pretest score

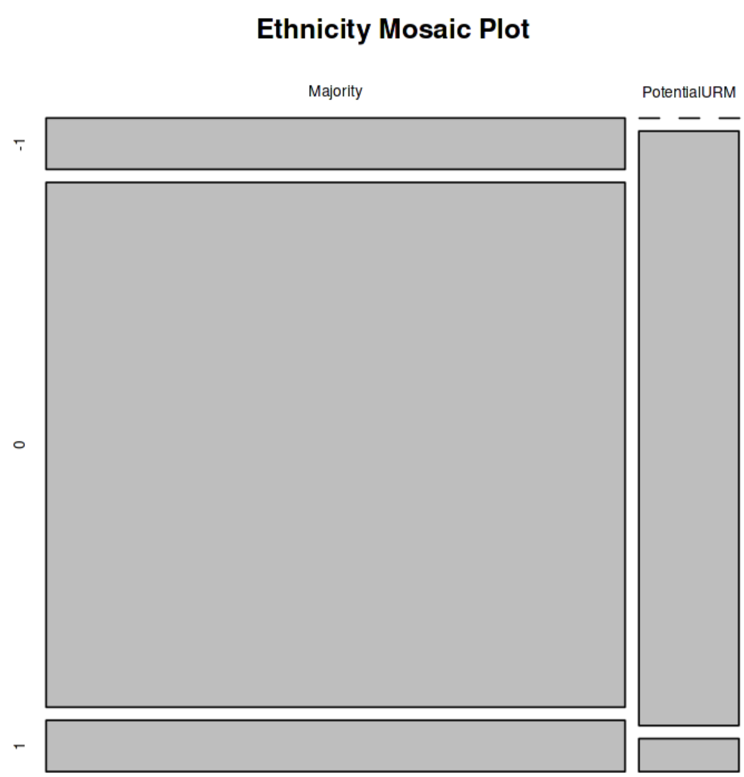


- Post test score



- Categorical: mosaic plot


Gender Mosaic Plot
F
M
-1
0
1


Ethnicity Mosaic Plot
Majority
PotentialURM
-1
0
1

Other datasets

Datasets that have “pretest” “posttest” in problem hierarchy or problem name: [datasets]

Datasets in Gates Project ([project id 527])

aset	dataset_name	pretest	posttest	others
3	GatesProject_CentralCatholic_Spring2019	yes	yes	
4	GatesProjectTest	yes	yes	
0	GatesProject_CentralCatholic_Spring2019 (Cleaned)	yes	yes	
2	GatesProject_CentralCatholic_Spring2019 (Cleaned) Less Advanced (LA) Students	yes	yes	
3	GatesProject_CentralCatholic_Spring2019 (Cleaned) More Advanced (MA) Students	yes	yes	
3	GatesProject_DL_CC_Fall2019	yes	yes	
1	GatesProject_OC_Fall2019	yes	yes	
1	GatesProject_WM_Spring2020	yes	yes	
8	GatesProject_NKA_Spring2020	yes	yes	
1	GatesProject_SV_Spring2020	yes	no	
0	GatesProject_LB_Spring2020	yes	no	
9	GatesSpring20VersionPublic	no	no	
1	GatesProject_DL_CC_OC_Fall2019_Practice_Cleaned	no	no	
4	GatesProject_WM_NKA_Spring2020_Practice_Cleaned	no	no	
5	GatesProject_DL_OC_Fall2019_Practice_Cleaned	no	no	
4	MC Pilot Testing	no	no	
5	GatesProject_CityCharter_Summer2020	no	no	
5	GatesProject_DL_OC_Fall2019_Practice_Cleaned_WithRefinedKCM	no	no	
7	GatesProject_BV_Spring2021	yes	no	
4	GatesProject_DL_Fall2021	yes	yes	

9	GatesProject_CC_Fall2021	yes	yes	
5	GatesProject_DL_OC_Fall2019_Practice_Cleaned_AddGaming	yes	yes	extracted from transactions
3	GatesProject_MA_Spring2022	yes	yes	mid-test, school test…
0	GatesProject_BV_Spring2022	yes	yes	mid-test
7	Mathtutor Problem Set 6.01 (Demo)	no	no	
4	GatesProject_BV_Spring2022_Practice_Cleaned_AddGaming	no	no	
7	GatesProject_NKA_Fall2022	yes	yes	mid-test

Summary by school and semesters

ar semester	school	prestest	posttest
ring 2019	CC	yes	yes
l 2019	DL	yes	yes
l 2019	CC	yes	yes
l 2019	OC	yes	yes
ring 2020	WM	yes	yes
ring 2020	NKA	yes	yes
ring 2020	SV	yes	no
ring 2020	LB	yes	no
ring 2021	BV	yes	no
l 2021	DL	yes	yes
l 2021	CC	yes	yes
ring 2022	MA	yes	yes
ring 2022	BV	yes	yes
l 2022	NKA	yes	yes

DS 613

Name: Bernachi

DS 3093

Name:GatesProject_DL_CC_Fall2019
Goal: separate pretest and posttest data from transaction data; compute student pretest and post test scores
Results: https://drive.google.com/file/d/134Ivv-EN6PBUcSHfi6-LVgDUw1L3rifI/view?usp=sharing

DS 3151

Name: GatesProject_OC_Fall2019

Goal: separate pretest and posttest data from transaction data; compute student pretest and post test scores

Results: https://drive.google.com/file/d/1eFAe5GOZA5eVeXKAPETFeVbBj1xuLuJ0/view?usp=sharing