Online Education System Evaluation

Claire Jiaying Wu

07 Feb 2019

```
library(prettydoc)
library(data.table)
library(DT)
library(stringr)
library(ggpubr)
library(corrplot)
```

```
id.name <- "Student ID"
id.name.pattern <- "Student "
id.name.new <- "Numeric ID"

file.pk <- "../homework 1/Data/Prior Courses.csv"
file.kc1 <- "../homework 1/Data/Knowledge Check -- Level 2.csv"
file.kc2 <- "../homework 1/Data/Knowledge Check -- Level 5.csv"
file.grades <- "../homework 1/Data/Grades.csv"

pk.level.name <- "Prior Knowledge Level"

trig.name <- "Trigonometry"
calc.name <- "Calculus"

num.digits <- 2
threshold.kc1 <- 2
threshold.kc2 <- 5

homework.topics <- c("Homework Grade_Mechanics", "Homework Grade_Momentum", "Homework Grade_
                     "Homework Grade_Electricity", "Homework Grade_Magnetism", "Homework Gra

total.min.spent <- c("Level 2_Mechanics_Time Spend", "Level 2_Momentum_Time Spend", "Level 2
                     "Level 5_Mechanics_Time Spend", "Level 5_Momentum_Time Spend", "Level 5
                     "Level 5_Electricity_Time Spend", "Level 5_Magnetism_Time Spend", "Leve

score.kc2.names <- c("Level 5_Mechanics", "Level 5_Momentum", "Level 5_Gravity",
                     "Level 5_Electricity", "Level 5_Magnetism", "Level 5_Relativity")
```

```
max.with.na <- function(x){
  y <- as.numeric(x[!is.na(as.numeric(x))])
  if(length(y) == 0){
    return(NA_real_)
  }
  if(length(y) > 0){
    return(x = max(y, na.rm = TRUE))
  }
}

round.numerics <- function(x, digits){
  if(is.numeric(x)){
    x <- round(x = x, digits = digits)
  }
  return(x)
}

display.multiple.records <- function(x){
  y <- x[, N := .N, by = id.name]
  return(setorderv(y[N > 1]))
}

summarize.pk.class <- function(dataset, class_name){
  num_scores <- dataset[, .(num_scores = length(get(class_name)))]
  num_students <- dataset[, .(num_students = length(unique(get(id.name))))]
  mean_scores <- dataset[, .(mean_scores = round.numerics(mean(get(class_name), na.rm = TRUE
  sd_scores <- dataset[, .(sd_scores = round.numerics(sd(get(class_name), na.rm = TRUE), dig
  info <- data.table(class_name, num_scores, num_students, mean_scores, sd_scores)
  return(datatable(data = info, rownames = FALSE))
}
```

```
add.numeric.id <- function(dataset){
  dataset[, eval(id.name.new) := as.numeric(gsub(pattern = id.name.pattern, replacement = "
  return(dataset[c(1:3, 98:100),])
}

summarize.kc1.topic <- function(dataset, topic){
  num_students <- dataset[, .(num_students = (sum(!is.na(get(topic)))))]
  mean_score <- dataset[, .(mean_score = round.numerics(mean(get(topic), na.rm = TRUE), digi
  sd_score <- dataset[, .(sd_score = round.numerics(sd(get(topic), na.rm = TRUE), digits = 2
  pass_percent <- dataset[get(topic) >= threshold.kc1, .N] / dat.kc1[, .N] * 100
  mean_min <- dataset[, .(mean_min = round.numerics(mean(get(paste(topic,'_Time Spend', sep=
  sd_min <- dataset[, .(sd_min = round.numerics(sd(get(paste(topic,'_Time Spend', sep="")),
  return(data.table(topic, num_students, mean_score, sd_score, pass_percent, mean_min, sd_mi
}

summarize.kc2.topic <- function(dataset, topic){
  num_students <- dataset[, .(num_students = (sum(!is.na(get(topic)))))]
  mean_score <- dataset[, .(mean_score = round.numerics(mean(get(topic), na.rm = TRUE), digi
  sd_score <- dataset[, .(sd_score = round.numerics(sd(get(topic), na.rm = TRUE), digits = 2
  pass_percent <- dataset[get(topic) >= threshold.kc2, .N] / dat.kc1[, .N] * 100
  mean_min <- dataset[, .(mean_min = round.numerics(mean(get(paste(topic,'_Time Spend', sep=
  sd_min <- dataset[, .(sd_min = round.numerics(sd(get(paste(topic,'_Time Spend', sep="")),
  return(data.table(topic, num_students, mean_score, sd_score, pass_percent, mean_min, sd_mi
}

summarize.hw.class <- function(dataset, score_item){
  student_count <- dataset[, .(student_count = (sum(!is.na(get(score_item)))))]
  score_mean <- dataset[, .(score_mean = (mean(get(score_item), na.rm = TRUE)))]
  score_sd <- dataset[, .(score_sd = (sd(get(score_item), na.rm = TRUE)))]
  return(data.table(score_item, student_count, score_mean, score_sd))
}
```

Inspections - dimensions

```
dat.pk <- fread(input = file.pk)
dat.kc1 <- fread(input = file.kc1)
dat.kc2 <- fread(input = file.kc2)
dat.grades <- fread(input = file.grades)
```

Prior Knowledge

```
dim(dat.pk)
```

```
[1] 103   3
```

Knowledge Check 1

```
dim(dat.kc1)
```

```
[1] 100   7
```

Knowledge Check 2

```
dim(dat.kc2)
```

```
[1] 100  13
```

Grades

```
dim(dat.grades)
```

```
[1] 100  10
```

```
# How many unique students were in the class?
dat.pk[, length(unique(get(id.name)))]
```

```
[1] 100
```

```
dat.kc1[, length(unique(get(id.name)))]
```

```
[1] 100
```

```
dat.kc2[, length(unique(get(id.name)))]
```

```
[1] 100
```

```
dat.grades[, length(unique(get(id.name)))]
```

```
[1] 100
```

Inspections - multiple records

Which files (if any) contain more than 1 row per student?

Prior Knowledge

```
dup.pk = display.multiple.records(dat.pk)
datatable(data = dup.pk[,-'N'], rownames = FALSE)
```

Show 10 entries Search:

Student ID	Trigonometry	Calculus
Student 57	85	88
Student 57	88	86
Student 66	85	83
Student 66	86	86
Student 76	83	87
Student 76	84	88

Showing 1 to 6 of 6 entries Previous 1 Next

Knowledge Check 1

```
dup.kc1 = display.multiple.records(dat.kc1)
datatable(data = dup.kc1[,-'N'], rownames = FALSE)
```

Show 10 entries Search:

Student ID	Mechanics	Momentum	Gravity	Time: Mechanics	Time: Momentum	Time: Gravity
No data available in table						

Showing 0 to 0 of 0 entries Previous Next

Knowledge Check 2

```
dup.kc2 = display.multiple.records(dat.kc2)
datatable(data = dup.kc2[,-'N'], rownames = FALSE)
```

Show 10 entries Search:

Student ID	Mechanics	Momentum	Gravity	Electricity	Magnetism	Relativity	Time: Mechanics	Time: Momentum
No data available in table								

Showing 0 to 0 of 0 entries Previous Next

Grades

```
dup.grades = display.multiple.records(dat.grades)
datatable(data = dup.grades[,-'N'], rownames = FALSE)
```

Show 10 entries Search:

Student ID	HW 1: Mechanics	HW 2: Momentum	HW 3: Gravity	HW 4: Electricity	HW 5: Magnetism	HW 6: Relativity	Midterm	Final Exam	Total Score
No data available in table									

Showing 0 to 0 of 0 entries Previous Next

Cleaning - Reduction to a Single Record

To handle the students with multiple records, we decided to summarize their prior knowledge as follows:

- For each student, the highest score in a prior class will be used. If no numeric record is included, an NA value should be used. For reference, we have provided a function called **max.with.na** that can perform this calculation for a single student.
- We will also create an overall score called Prior Knowledge Level. For each student, this will be defined as the average of the student's highest score in Trigonometry and the student's highest score in Calculus. For students who did not take both of these classes, the overall score will be based on the measured values.

a) Summary Before the Reduction

Starting with the original table of Prior Knowledge scores, compute the following for each class: the number of measured records, the number of unique students with a measured record, the average score among all of the measured records, and the standard deviation of the scores among all of the measured records. Write a function called **summarize.pk.class** that will display the name of the prior class along with these figures using the **datatable** method from the **DT** package.

Trigonometry

```
summarize.pk.class(dat.pk, trig.name)
```

Show 10 entries Search:

class_name	num_scores	num_students	mean_scores	sd_scores
Trigonometry	103	100	84.04	4.05

Showing 1 to 1 of 1 entries Previous 1 Next

Calculus

```
summarize.pk.class(dat.pk, calc.name)
```

Show 10 entries Search:

class_name	num_scores	num_students	mean_scores	sd_scores
Calculus	103	100	84.36	4.07

Showing 1 to 1 of 1 entries Previous 1 Next

b) Reduction of Information

Create a new table called **pk.reduced** that will contain 1 record per student according to the criteria specified above. For the students with multiple records in the original file, display their records in the **pk.reduced** table using the **datatable** function in the **DT** package.

```
# Remove the multiple records first
pk.reduced <- dat.pk[-c(57,66,76,101,102,103), -'N']

# Process the function "max.with.na" and add back the records to "pk.reduced"
class.variables <- c(trig.name, calc.name)
st57 <- dat.pk[c(57,101), lapply(X = .SD, FUN = "max.with.na"), .SDcols = class.variables]
st66 <- dat.pk[c(66,102), lapply(X = .SD, FUN = "max.with.na"), .SDcols = class.variables]
st76 <- dat.pk[c(76,103), lapply(X = .SD, FUN = "max.with.na"), .SDcols = class.variables]
st57[, "Student ID" := "Student 57"]
st66[, "Student ID" := "Student 66"]
st76[, "Student ID" := "Student 76"]
pk.reduced <- rbindlist(l = list(pk.reduced, st57, st66, st76), fill = TRUE)

# Display the results by datatable function
datatable(data = pk.reduced[98:100,], rownames = FALSE)
```

Show 10 entries Search:

Student ID	Trigonometry	Calculus
Student 57	88	88
Student 66	86	86
Student 76	84	88

Showing 1 to 3 of 3 entries Previous 1 Next

c) Summary After the Reduction

Using the **pk.reduced** table, compute the following for each class and the Prior Knowledge Level: the number of measured records, the number of unique students with a measured record, the average score among all of the measured records, and the standard deviation of the scores among all of the measured records.

Trigonometry

```
summarize.pk.class(pk.reduced, trig.name)
```

Show 10 entries Search:

class_name	num_scores	num_students	mean_scores	sd_scores
Trigonometry	100	100	84.03	4.11

Showing 1 to 1 of 1 entries Previous 1 Next

Calculus

```
summarize.pk.class(pk.reduced, calc.name)
```

Show 10 entries Search:

class_name	num_scores	num_students	mean_scores	sd_scores
Calculus	100	100	84.33	4.11

Showing 1 to 1 of 1 entries Previous 1 Next

Prior Knowledge Level

```
pk.reduced[, Level := rowMeans(.SD, na.rm = TRUE), by = id.name, .SDcols = class.variables]
summarize.pk.class(pk.reduced, 'Level')
```

Show 10 entries Search:

class_name	num_scores	num_students	mean_scores	sd_scores
Level	100	100	84.19	3.89

Showing 1 to 1 of 1 entries Previous 1 Next

Combining the Tables - one comprehensive table called **dat**

```
# Add New (numeric) ID for each table using new defined function "add.numeric.id"
# Display the first and the last three rows as double check

add.numeric.id(pk.reduced)
```

```
   Student ID Trigonometry Calculus Level Numeric ID
1:  Student 1           87       90  88.5          1
2:  Student 2           89       85  87.0          2
3:  Student 3           87       84  85.5          3
4: Student 57           88       88  88.0         57
5: Student 66           86       86  86.0         66
6: Student 76           84       88  86.0         76
```

```
add.numeric.id(dat.kc1)
```

```
    Student ID Mechanics Momentum Gravity Time: Mechanics Time: Momentum
1:   Student 1   1.43789 1.073275  1.1047        48.36925       40.33339
2:   Student 2   1.43789 1.273275  1.1047        42.30118       41.48261
3:   Student 3   0.73789 1.473275  1.1047        41.71388       41.96426
4:  Student 98   2.63789 2.173275  1.6047        46.04264       45.62278
5:  Student 99   2.03789 1.673275  2.1047        44.09625       41.18611
6: Student 100   1.73789 1.373275  1.3047        31.57928       20.33776
   Time: Gravity N Numeric ID
1:      43.33428 1          1
2:      41.93193 1          2
3:      37.10438 1          3
4:      38.28007 1         98
5:      39.93279 1         99
6:      28.53840 1        100
```

```
add.numeric.id(dat.kc2)
```

```
    Student ID Mechanics Momentum  Gravity Electricity Magnetism
1:   Student 1  5.644334 4.725962 5.288509    5.341471  5.381294
2:   Student 2  5.344334 4.825962 4.388509    4.741471  4.981294
3:   Student 3  5.344334 5.325962 5.188509    5.441471  5.381294
4:  Student 98  5.144334       NA 4.788509    4.341471  5.381294
5:  Student 99  5.344334 5.025962 5.188509    4.741471  5.081294
6: Student 100  5.044334 5.225962 4.588509    5.241471  5.081294
   Relativity Time: Mechanics Time: Momentum Time: Gravity
1:   5.244868       35.724639       36.99167      39.46242
2:   4.744868       17.182767       40.95984      34.38783
3:   5.244868       23.435526       37.61321      32.41507
4:   4.644868       19.454932             NA      31.58007
5:   5.044868       -2.570931       16.17273      32.73389
6:   5.044868       27.115936       36.35036      46.28030
   Time: Electricity Time: Magnetism Time: Relativity N Numeric ID
1:          37.50984        39.16727         36.77232 1          1
2:          28.29130        45.13444         32.21368 1          2
3:          33.46466        37.97897         28.82718 1          3
4:          38.79300        41.65212         38.02863 1         98
5:          10.84618        28.42946         21.38664 1         99
6:          32.94506        45.58811         33.43606 1        100
```

```
add.numeric.id(dat.grades)
```

```
    Student ID HW 1: Mechanics HW 2: Momentum HW 3: Gravity
1:   Student 1              77             84            74
2:  Student 10              94             98            80
3: Student 100              84             92            80
4:  Student 97              91             96            83
5:  Student 98              91             89            83
6:  Student 99              99             93            88
   HW 4: Electricity HW 5: Magnetism HW 6: Relativity Midterm Final Exam
1:                98              96               81      87         80
2:                91              78               84      81         77
3:                96              90               87      87         90
4:                84              85               89      91         95
5:                85              89               85      84         88
6:                95              88               85      90         85
   Total Score N Numeric ID
```

```
1:    84.10000 1          1
2:    82.40000 1         10
3:    88.36667 1        100
4:    91.00000 1         97
5:    86.40000 1         98
6:    89.03333 1         99
```

```
# Set new column names for each table

# First table, Prior Knowledge
old.names = c("Trigonometry", "Calculus")
new.names = c("Prior Knowledge_Trigonometry", "Prior Knowledge_Calculus")
pk.reduced <- setnames(pk.reduced, old = old.names, new = new.names)

# Second table, Knowledge Check 1
old.names = c("Mechanics", "Momentum", "Gravity", "Time: Mechanics", "Time: Momentum", "Time
new.names = c("Level 2_Mechanics", "Level 2_Momentum", "Level 2_Gravity", "Level 2_Mechanics
              "Level 2_Momentum_Time Spend", "Level 2_Gravity_Time Spend")
dat.kc1 <- setnames(dat.kc1, old = old.names, new = new.names)

# Third table, Knowledge Check 2
old.names = c("Mechanics", "Momentum", "Gravity", "Electricity", "Magnetism", "Relativity",
              "Time: Mechanics", "Time: Momentum", "Time: Gravity",
              "Time: Electricity", "Time: Magnetism", "Time: Relativity")
new.names = c("Level 5_Mechanics", "Level 5_Momentum", "Level 5_Gravity",
              "Level 5_Electricity", "Level 5_Magnetism", "Level 5_Relativity",
              "Level 5_Mechanics_Time Spend", "Level 5_Momentum_Time Spend", "Level 5_Gravit
              "Level 5_Electricity_Time Spend", "Level 5_Magnetism_Time Spend", "Level 5_Rel
dat.kc2 <- setnames(dat.kc2, old = old.names, new = new.names)

# Last table, Grades
old.names = c("HW 1: Mechanics", "HW 2: Momentum", "HW 3: Gravity",
              "HW 4: Electricity", "HW 5: Magnetism", "HW 6: Relativity")
new.names = c("Homework Grade_Mechanics", "Homework Grade_Momentum", "Homework Grade_Gravity
              "Homework Grade_Electricity", "Homework Grade_Magnetism", "Homework Grade_Rela
dat.grades <- setnames(dat.grades, old = old.names, new = new.names)
```

```
# Merge the four tables using "merge" function
dat <- merge(x = pk.reduced, y = dat.kc1[, -'Student ID'], by = id.name.new, all.x = TRUE, a
dat <- merge(x = dat, y = dat.kc2[, -'Student ID'], by = id.name.new, all.x = TRUE, all.y =
dat <- merge(x = dat, y = dat.grades[, -'Student ID'], by = id.name.new, all.x = TRUE, all.y

# Set the table order by student id, increasing
dat <- setorderv(x = dat, cols = id.name.new, order = 1)
dat <- dat[, -c('Numeric ID', 'N', 'N.x', 'N.y')]

# Round the numeric variables to two digits
dat <- dat[, lapply(X = .SD, FUN = 'round.numerics', digits = 2)]

# Display the dat table by datatable function in DT
datatable(data = dat, rownames = FALSE)
```

Show 10 entries Search:

Student ID	Prior Knowledge_Trigonometry	Prior Knowledge_Calculus	Level	Level 2_Mechanics	Level 2_Momentum	Level 2_Gravity	2_Me
Student 1	87	90	88.5	1.44	1.07	1.1	
Student 2	89	85	87	1.44	1.27	1.1	
Student 3	87	84	85.5	0.74	1.47	1.1	
Student 4	86	86	86	1.84	1.77	1.3	
Student 5	82	84	83	2.44	1.87	1.8	
Student 6	79	81	80	1.94	2.07	1.8	
Student 7	86	85	85.5	2.04	1.97	1.6	
Student 8	84	86	85	2.04	2.17	1.6	
Student 9	88	87	87.5	2.34	2.57		
Student 10	79	84	81.5	2.34	1.77	1.8	

Showing 1 to 10 of 100 entries Previous 1 2 3 4 5 ... 10 Next

Evaluation - Knowledge Check 1

How did the students do on the first knowledge check?

```
# Compute the variables by the new defined function "summarize.kc1.topic"
topic1 <- summarize.kc1.topic(dat.kc1, 'Level 2_Mechanics')
topic2 <- summarize.kc1.topic(dat.kc1, 'Level 2_Momentum')
topic3 <- summarize.kc1.topic(dat.kc1, 'Level 2_Gravity')

# Combine the topics for kc1
summary_kc1 <- rbindlist(l = list(topic1, topic2, topic3), fill = TRUE)

# Display by the datatable function
datatable(data = summary_kc1, rownames = FALSE)
```

Show 10 entries Search:

topic	num_students	mean_score	sd_score	pass_percent	mean_min	sd_min
Level 2_Mechanics	98	2.04	0.47	62	37.41	9.25
Level 2_Momentum	97	1.88	0.46	42	38.5	8.11
Level 2_Gravity	94	1.6	0.44	16	37.67	8.43

Showing 1 to 3 of 3 entries Previous 1 Next

Evaluation - Knowledge Check 2

How did the students do on the second knowledge check?

```
# Compute the variables by the new defined function "summarize.kc2.topic"
topic2.1 <- summarize.kc2.topic(dat.kc2, 'Level 5_Mechanics')
topic2.2 <- summarize.kc2.topic(dat.kc2, 'Level 5_Momentum')
topic2.3 <- summarize.kc2.topic(dat.kc2, 'Level 5_Gravity')
topic2.4 <- summarize.kc2.topic(dat.kc2, 'Level 5_Electricity')
topic2.5 <- summarize.kc2.topic(dat.kc2, 'Level 5_Magnetism')
topic2.6 <- summarize.kc2.topic(dat.kc2, 'Level 5_Relativity')

# Combine the topics for kc1
summary_kc2 <- rbindlist(l = list(topic2.1, topic2.2, topic2.3, topic2.4, topic2.5, topic2.6

# Display by the datatable function
datatable(data = summary_kc2, rownames = FALSE)
```

Show 10 entries Search:

topic	num_students	mean_score	sd_score	pass_percent	mean_min	sd_min
Level 5_Mechanics	95	5.04	0.71	66	22.13	10.12
Level 5_Momentum	95	4.89	0.69	54	33.13	9.45
Level 5_Gravity	92	4.8	0.82	43	33.08	9.43
Level 5_Electricity	98	4.91	0.74	66	26.46	10.87
Level 5_Magnetism	96	4.96	0.76	66	36.14	9.65
Level 5_Relativity	99	4.87	0.7	52	30.42	10.18

Showing 1 to 6 of 6 entries Previous 1 Next

Evaluation - Is Time Spent Time Well Used?

```
# Calculate correlation for Knowledge Check 1
level2.1 <- cor(x = dat.kc1$`Level 2_Mechanics`, y = dat.kc1$`Level 2_Mechanics_Time Spend`,
level2.2 <- cor(x = dat.kc1$`Level 2_Momentum`, y = dat.kc1$`Level 2_Momentum_Time Spend`, u
level2.3 <- cor(x = dat.kc1$`Level 2_Gravity`, y = dat.kc1$`Level 2_Gravity_Time Spend`, use
Knowledge_Check <- "Level 2"
Topic <- c("Mechanics", "Momentum", "Gravity")
Correlation <- c(level2.1, level2.2, level2.3)
corr_kc2 <- data.table(Knowledge_Check, Topic, Correlation)

# Calculate correlation for Knowledge Check 2
level5.1 <- cor(x = dat.kc2$`Level 5_Mechanics`, y = dat.kc2$`Level 5_Mechanics_Time Spend`,
level5.2 <- cor(x = dat.kc2$`Level 5_Momentum`, y = dat.kc2$`Level 5_Momentum_Time Spend`, u
level5.3 <- cor(x = dat.kc2$`Level 5_Gravity`, y = dat.kc2$`Level 5_Gravity_Time Spend`, use
level5.4 <- cor(x = dat.kc2$`Level 5_Electricity`, y = dat.kc2$`Level 5_Electricity_Time Spe
level5.5 <- cor(x = dat.kc2$`Level 5_Magnetism`, y = dat.kc2$`Level 5_Magnetism_Time Spend`,
level5.6 <- cor(x = dat.kc2$`Level 5_Relativity`, y = dat.kc2$`Level 5_Relativity_Time Spend
Knowledge_Check5 <- "Level 5"
Topic5 <- c("Mechanics", "Momentum", "Gravity", "Electricity", "Magnetism", "Relativity")
Correlation5 <- c(level5.1, level5.2, level5.3, level5.4, level5.5, level5.6)
corr_kc5 <- data.table(Knowledge_Check5, Topic5, Correlation5)

# Combine the two correlations together and show by datatable function in DT
corr_kc <- rbindlist(l = list(corr_kc2, corr_kc5), fill = FALSE)
corr_kc <- corr_kc[, lapply(X = .SD, FUN = "round.numerics", digits = 2)]
datatable(data = corr_kc, rownames = FALSE)
```

Show 10 entries Search:

Knowledge_Check	Topic	Correlation
Level 2	Mechanics	-0.13
Level 2	Momentum	-0.01
Level 2	Gravity	-0.05
Level 5	Mechanics	-0.27
Level 5	Momentum	-0.16
Level 5	Gravity	-0.13
Level 5	Electricity	-0.15
Level 5	Magnetism	-0.16
Level 5	Relativity	-0.17

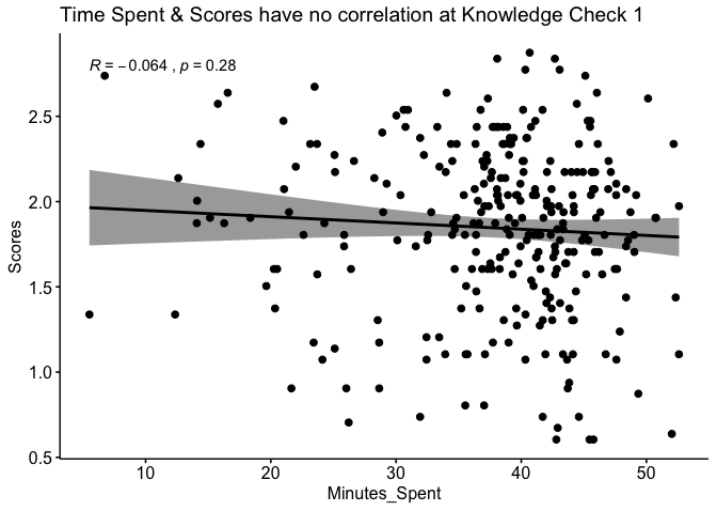
Showing 1 to 9 of 9 entries Previous 1 Next

Comment: Before I saw the data results, I don't think spending more time on the digital system is beneficial. And my guessing is confirmed by the data as above - all the small negative correlation coefficients showing no direct correlations between the time spent and the scores. Rather, we can see that the correlation coefficients of level 5 become smaller than that of level 2, which may indicate that the system potentially has negative influence on students' scores.

We could also examine the correlations between time spent and scores **through visualization**:

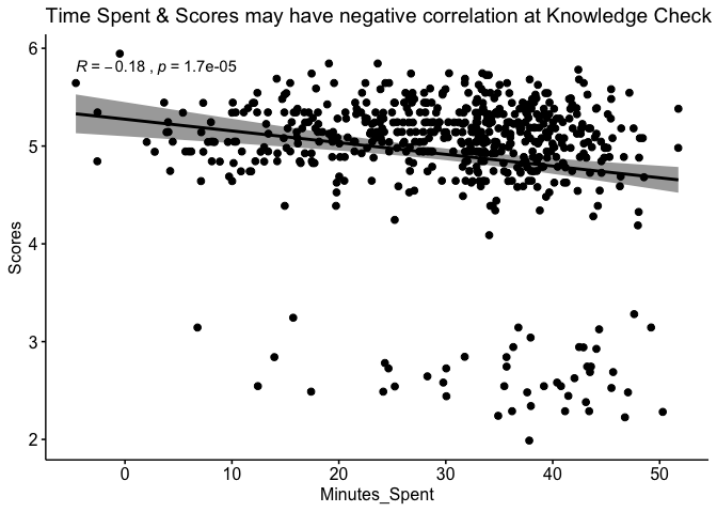
```
# Combine all the scores for Knowledge Check 1
level2_1 <- dat.kc1[, c('Level 2_Mechanics', 'Level 2_Mechanics_Time Spend')]
level2_2 <- dat.kc1[, c('Level 2_Momentum', 'Level 2_Momentum_Time Spend')]
level2_3 <- dat.kc1[, c('Level 2_Gravity', 'Level 2_Gravity_Time Spend')]
dat.kc.reshape2 <- rbindlist(l = list(level2_1, level2_2, level2_3), fill = FALSE)
dat.kc.reshape2 <- setnames(dat.kc.reshape2,
                            old = c('Level 2_Mechanics','Level 2_Mechanics_Time Spend'),
                            new = c('Scores','Time_Spent'))

# Draw the plot with regression line
ggscatter(dat.kc.reshape2, x = "Time_Spent", y = "Scores", add = "reg.line",
          conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
          xlab = "Minutes_Spent", ylab = "Scores", title = "Time Spent & Scores have no corr
```




```
# Combine all the scores for Knowledge Check 2
level5_1 <- dat.kc2[, c('Level 5_Mechanics', 'Level 5_Mechanics_Time Spend')]
level5_2 <- dat.kc2[, c('Level 5_Momentum', 'Level 5_Momentum_Time Spend')]
level5_3 <- dat.kc2[, c('Level 5_Gravity', 'Level 5_Gravity_Time Spend')]
level5_4 <- dat.kc2[, c('Level 5_Electricity', 'Level 5_Electricity_Time Spend')]
level5_5 <- dat.kc2[, c('Level 5_Magnetism', 'Level 5_Magnetism_Time Spend')]
level5_6 <- dat.kc2[, c('Level 5_Relativity', 'Level 5_Relativity_Time Spend')]
dat.kc.reshape5 <- rbindlist(l = list(level5_1, level5_2, level5_3, level5_4, level5_5, leve
dat.kc.reshape5 <- setnames(dat.kc.reshape5,
                            old = c('Level 5_Mechanics','Level 5_Mechanics_Time Spend'),
                            new = c('Scores','Time_Spent'))

# Draw the plot with regression line
ggscatter(dat.kc.reshape5, x = "Time_Spent", y = "Scores", add = "reg.line",
          conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Minutes_Spent",
          title = "Time Spent & Scores may have negative correlation at Knowledge Check 2")
```



Summary of Scores

```
# Get the information of each score item
hw1 <- summarize.hw.class(dat.grades, "Homework Grade_Mechanics")
hw2 <- summarize.hw.class(dat.grades, "Homework Grade_Momentum")
hw3 <- summarize.hw.class(dat.grades, "Homework Grade_Gravity")
hw4 <- summarize.hw.class(dat.grades, "Homework Grade_Electricity")
hw5 <- summarize.hw.class(dat.grades, "Homework Grade_Magnetism")
hw6 <- summarize.hw.class(dat.grades, "Homework Grade_Relativity")
mid_score <- summarize.hw.class(dat.grades, "Midterm")
final_score <- summarize.hw.class(dat.grades, "Final Exam")
total_score <- summarize.hw.class(dat.grades, "Total Score")

# Combine the results into one table and round the numeric variables
summary_grades <- rbindlist(l = list(hw1, hw2, hw3, hw4, hw5, hw6, mid_score, final_score, t
```

```
summary_grades <- summary_grades[, lapply(X = .SD, FUN = "round.numerics", digits = 2)]

# Display the table through the datatable function in DT
datatable(data = summary_grades, rownames = FALSE)
```

Show 10 entries Search:

score_item	student_count	score_mean	score_sd
Homework Grade_Mechanics	100	88.25	7.43
Homework Grade_Momentum	100	92.76	6.05
Homework Grade_Gravity	100	82.41	6.58
Homework Grade_Electricity	100	87.32	5.83
Homework Grade_Magnetism	100	88	5.89
Homework Grade_Relativity	100	87.72	6.08
Midterm	100	84.82	7.51
Final Exam	100	84.85	8.04
Total Score	100	86	3.94

Showing 1 to 9 of 9 entries Previous 1 Next

Correlations with Outcomes

```
# Add the new outcome variable of "Homework Average" (we've had the other three outcomes in
dat <- dat[, Homework_Avg:= lapply(X = .SD, FUN = mean), by = id.name, .SDcols = homework.to

# Add the new predictor variable of "Total Minutes Spent"
dat <- dat[, Min_Spent_Total:= lapply(X = .SD, FUN = sum), by = id.name, .SDcols = total.min

# Add the new predictor variable of "Average Score on Level 5 Check"
dat <- dat[, Score_Level5_Avg:= lapply(X = .SD, FUN = mean), by = id.name, .SDcols = score.k
```

```
# Compute the correlation matrix for the outcomes and inputs
outcomes_inputs <- dat[, c("Level", "Min_Spent_Total", "Score_Level5_Avg",
                           "Homework_Avg", "Midterm", "Final Exam", "Total Score")]
correlations <- data.table(cor(outcomes_inputs, use = "complete.obs"))

# Round the results, add a heading column, and then display by datatable function
correlations <- correlations[, lapply(X = .SD, FUN = "round.numerics", digits = 2)]
correlations <- data.table(Outcomes = c("Level", "Min_Spent_Total", "Score_Level5_Avg", "Hom
                                        "Midterm", "Final Exam", "Total Score"), correlation
datatable(correlations, rownames = FALSE)
```

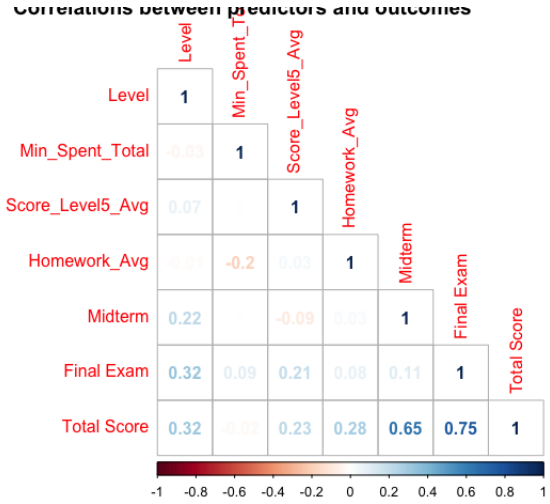
Show 10 entries Search:

Outcomes	Level	Min_Spent_Total	Score_Level5_Avg	Homework_Avg	Midterm	Final Exam	Total Score
Level	1	-0.03	0.07	-0.01	0.22	0.32	0.32
Min_Spent_Total	-0.03	1	0	-0.2	0	0.09	-0.02
Score_Level5_Avg	0.07	0	1	0.03	-0.09	0.21	0.23
Homework_Avg	-0.01	-0.2	0.03	1	0.03	0.08	0.28
Midterm	0.22	0	-0.09	0.03	1	0.11	0.65
Final Exam	0.32	0.09	0.21	0.08	0.11	1	0.75
Total Score	0.32	-0.02	0.23	0.28	0.65	0.75	1

Showing 1 to 7 of 7 entries Previous 1 Next

Qualitative and Quantitative Conclusions

```r
# Present the results in a clearer way by drawing a correlation plot
corrpl <- cor(outcomes_inputs, use = "complete.obs")
corrplot(corrpl, method = "number", type = "lower", title = "Correlations between predictors
```



Comment: To evaluation the effectiveness of the digital system, two aspects shall be inspected: the time spent and the scores got on the system:

1) Based on the correlations between the Min_Spent_Total and the outcomes (Homework_Avg, Midterm, Final Exam, Total Score), we could tell that no better scores related to the time spent in the digital system. Rather, the time spent even has a minor negative correlation with the homework performance.

2) From the scores perspective, we could see that the scores got on the system at Level 5 has minor correlations to the Final Exam (0.21), and to the Total Score (0.23). Yet these relationships are not as strong as the correlations between the Prior Knowledge Level and the Final/ Total Score (0.32).

In conclusion, based on the given dataset, the digital system that the students used **does not have strong positive impact** on the students' performance (homework, mid/final exam, and total score), and spending more time on the system might negatively impact students' homework grade. Therefore, I recommend that the school should consider improve the design of the system or change to another one.