# Self-regulated Learning

INFO 5200 Learning Analytics Homework

*[[Kimberly Williamson, khw44]]*

In this homework, you will learn how to mine a clickstream dataset for correlational patterns. You are given the SRL survey responses and the most recent export of the edx clickstream log for this course.

The SRL responses are recoded to range from 0 (low) to 4 (high) for each strategy index.

Learning Objectives:

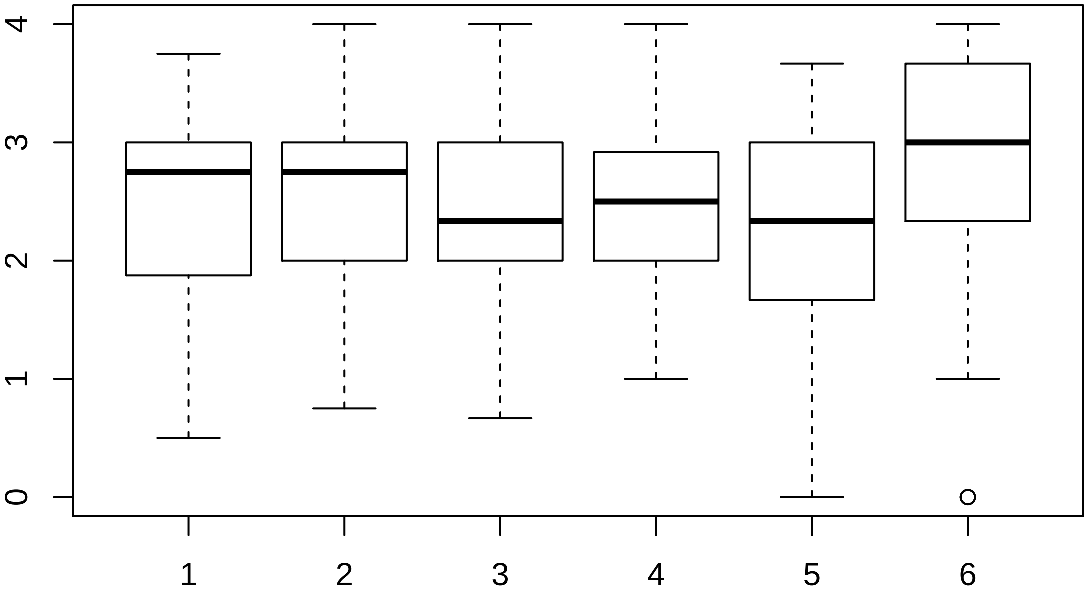
1. Exploring response distributions of survey data
2. Merging survey with behavioral data
3. Engineering features that could represent SRL strategies
4. Checking if any behavioral features predict survey responses using a linear model

## Part 1: Explore responses and merge data

**Question 1:** Plot a boxplot for the responses for each SRL strategy in the dataset (6 in total).

```
# Import cleaned up SRL data
srl = read.csv("info5200.srl.responses.csv")

####################################################
####### BEGIN INPUT: Plot the distribution #######
####################################################
boxplot(srl$goal_setting, srl$strategic_planning, srl$selfevaluation, srl$task_strategies,
        srl$elaboration, srl$help_seeking)
```



```
####################################################
####################################################
```

**Question 2:** Now merge the clickstream data with the SRL survey data using the `survey_id` identifier. Note that not everyone in the clickstream data has survey records. You only need to keep students who have survey data. Thus, you should end up with all clickstream records for students who filled out the survey: 29 students and 65,407 rows total.

```
# Import edx clickstream log
cl = readRDS("info5200.srl.clickstream.rds")

###############################################
####### BEGIN INPUT: Merge Datasets    ########
###############################################
srlmerge = merge(srl, cl, by="survey_id")

###############################################
###############################################
```

## Part 2: Engineering Features

Once again you get to practice feature engineering. By now you know what the clickstream data looks like thanks to the prediction project. Unlike in the project, however, you do not need to worry this time about the timing. Your goal is to come up with behavioral indicators of self-regulated learning. You know what the survey responses measure (check out the actual questions) and you know what the clickstream records represent (check out the corresponding pages in the course).

**Question 3:** Before coding up variables, describe in words ONE OR MORE (sequences/patterns of) actions that you think might be indicative of EACH SRL strategy. It will be easier for some than for others. At the end, you can describe some additional "general" activity features.

```
###############################################
####### BEGIN INPUT: Plan features ###########
###############################################

# goal setting
# 1. Clicks on the home page (9/course/)
# 2.

# strategic planning
# 1. Syllabus Clicks (2d815b2e787344838a1509c7a5861d2d)
# 2.

# self-evaluation
# 1. Progress page clicks (/courses/course-v1:Cornellx+Info5200+fall2019/progress)
# 2.

# task strategies
# 1. Submission Clicks (openassessmentblock.create_submission)
# 2.

# elaboration
# 1. Notes (notes)
# 2.

# help seeking
# 1. Discussion Forum clicks (edx.forum.thread.viewed)
# 2.

# general features
# 1. Following sequence blocks (edx.ui.lms.sequence.next_selected)
# 2.

###############################################
###############################################
```

**Question 4:** For each student, engineer the features you described above using the clickstream data. Combine all features into a dataset that has 1 row per student (29 rows here) with all features you created and the SRL data.

```
##################################################
####### BEGIN INPUT: Feature engineering #########
##################################################

# goal setting features
gs = srlmerge %>% group_by(survey_id) %>% filter(grepl("9/course/", event_type)) %>%
  summarise(gs=n())

# strategic planning features
sp = srlmerge %>% group_by(survey_id) %>%
  filter(grepl("2d815b2e787344838a1509c7a5861d2d", event_type)) %>% summarise(sp=n())

# self-evaluation features
se = srlmerge %>% group_by(survey_id) %>%
  filter(grepl("progress", event_type)) %>% summarise(se=n())

# task strategies features
ts = srlmerge %>% group_by(survey_id) %>%
  filter(grepl("openassessmentblock.create_submission", event_type)) %>% summarise(ts=n())

# elaboration features
el = srlmerge %>% group_by(survey_id) %>%
  filter(grepl("student_notes", event_type)) %>% summarise(el=n())

# help seeking features
hs = srlmerge %>% group_by(survey_id) %>%
  filter(grepl("edx.forum.thread.viewed", event_type)) %>% summarise(hs=n())

# general features
ge = srlmerge %>% group_by(survey_id) %>%
  filter(grepl("edx.ui.lms.sequence.next_selected", event_type)) %>% summarise(ge=n())

# combine into one dataset
dat = srlmerge %>%
    select(survey_id, goal_setting:help_seeking) %>%
    unique %>%
    left_join(gs) %>%
    left_join(sp) %>%
    left_join(se) %>%
    left_join(ts) %>%
    left_join(el) %>%
    left_join(hs) %>%
    left_join(ge)
```

```
## Joining, by = "survey_id"
## Joining, by = "survey_id"
## Joining, by = "survey_id"
## Joining, by = "survey_id"
## Joining, by = "survey_id"
## Joining, by = "survey_id"
## Joining, by = "survey_id"
```

```
dat[is.na(dat)] = 0
##############################################
##############################################
```

## Part 3: Explore SRL Association

There are many options for how to check if the behavioral features are associated with SRL strategies. We will use a very simple method: linear regression predicting each SRL index (self-report) with the relevant behavioral features. (Feel free to try out more complex ideas.)

**Question 5:** For each SRL index (goal_setting, strategic_planning, etc.) as the outcome, fit TWO linear regression models (`lm()`): one that only has the relevant features you describe above, and another that has all of the features.

For example, if you created two features specifically to indicate help seeking (H1 and H2), then you fit one model `help_seeking ~ H1 + H2` and a second model `help_seeking ~ H1 + H2 + all_other_features`. For each one, you should output the `summary()` of the `lm()` object. **Do not use the survey_id and the other SRL measures as predictors in the model!** So if you predict help_seeking then do not have goal_setting etc. in the model.

```
###################################################
####### BEGIN INPUT: Fit Regression Models #########
###################################################

# goal setting
summary(lm(goal_setting ~ gs, data = dat))
```

```
##
## Call:
## lm(formula = goal_setting ~ gs, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5153 -0.7661  0.2252  0.5275  1.2332
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.5505598  0.3491295   7.305  3.2e-08 ***
## gs          -0.0003791  0.0034440  -0.110    0.913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.78 on 31 degrees of freedom
## Multiple R-squared:  0.0003908,  Adjusted R-squared:  -0.03185
## F-statistic: 0.01212 on 1 and 31 DF,  p-value: 0.9131
```

```
summary(lm(goal_setting ~ gs+sp+se+ts+el+hs+ge, data = dat))
```

```
##
```

```
## Call:
## lm(formula = goal_setting ~ gs + sp + se + ts + el + hs + ge,
##     data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7564 -0.4441  0.1819  0.3997  1.3461
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.428483   0.575481   5.958 3.21e-06 ***
## gs           0.001181   0.005084   0.232   0.8182
## sp           0.005365   0.021346   0.251   0.8036
## se           0.004515   0.019949   0.226   0.8228
## ts          -0.178421   0.079509  -2.244   0.0339 *
## el           0.087462   0.080184   1.091   0.2858
## hs          -0.008560   0.006446  -1.328   0.1962
## ge           0.008458   0.010667   0.793   0.4353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7667 on 25 degrees of freedom
## Multiple R-squared:  0.221,  Adjusted R-squared:  0.002903
## F-statistic: 1.013 on 7 and 25 DF,  p-value: 0.4459
```

```
# strategic planning
summary(lm(strategic_planning ~ sp, data = dat))
```

```
##
## Call:
## lm(formula = strategic_planning ~ sp, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10789 -0.44186  0.03207  0.41818  1.32213
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.90180    0.25173  11.527 9.69e-13 ***
## sp          -0.01400    0.01225  -1.143    0.262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6405 on 31 degrees of freedom
## Multiple R-squared:  0.04041,    Adjusted R-squared:  0.009453
## F-statistic: 1.305 on 1 and 31 DF,  p-value: 0.262
```

```
summary(lm(strategic_planning ~ gs+sp+se+ts+el+hs+ge, data = dat))
```

```
##
## Call:
## lm(formula = strategic_planning ~ gs + sp + se + ts + el + hs +
##     ge, data = dat)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.14940 -0.37096  0.01173  0.40630  1.04266
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.954301   0.464695   6.358 1.18e-06 ***
## gs          -0.003136   0.004106  -0.764   0.4521
## sp          -0.015732   0.017237  -0.913   0.3701
## se          -0.003578   0.016108  -0.222   0.8260
## ts          -0.003572   0.064202  -0.056   0.9561
## el          -0.024243   0.064748  -0.374   0.7112
## hs           0.002137   0.005205   0.410   0.6850
## ge           0.019654   0.008613   2.282   0.0313 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6191 on 25 degrees of freedom
## Multiple R-squared:  0.2769, Adjusted R-squared:  0.07449
## F-statistic: 1.368 on 7 and 25 DF,  p-value: 0.2619
```

```
# self-evaluation
summary(lm(selfevaluation ~ se, data = dat))
```

```
##
## Call:
## lm(formula = selfevaluation ~ se, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45693 -0.61307 -0.08694  0.61028  1.59195
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.48748    0.21985  11.315 1.55e-12 ***
## se          -0.00611    0.01836  -0.333    0.742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.868 on 31 degrees of freedom
## Multiple R-squared:  0.00356,    Adjusted R-squared:  -0.02858
## F-statistic: 0.1108 on 1 and 31 DF,  p-value: 0.7415
```

```
summary(lm(selfevaluation ~ gs+sp+se+ts+el+hs+ge, data = dat))
```

```
##
## Call:
## lm(formula = selfevaluation ~ gs + sp + se + ts + el + hs + ge,
##     data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5564 -0.3478  0.1351  0.3975  1.3684
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.326243   0.592347   3.927 0.000597 ***
## gs          -0.003731   0.005233  -0.713 0.482474    
## sp          -0.006966   0.021972  -0.317 0.753841    
## se           0.025694   0.020533   1.251 0.222392    
## ts           0.022569   0.081839   0.276 0.784992    
## el          -0.041917   0.082534  -0.508 0.615994    
## hs          -0.008930   0.006635  -1.346 0.190397    
## ge           0.026862   0.010979   2.447 0.021795 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.7892 on 25 degrees of freedom
## Multiple R-squared:  0.3357, Adjusted R-squared:  0.1497 
## F-statistic: 1.805 on 7 and 25 DF,  p-value: 0.1307
```

```r
# task strategies
summary(lm(task_strategies ~ ts, data = dat))
```

```
## 
## Call:
## lm(formula = task_strategies ~ ts, data = dat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.1914 -0.5247  0.1420  0.4753  1.1669 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.32512    0.42070   5.527 4.74e-06 ***
## ts           0.02495    0.05631   0.443    0.661    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.6309 on 31 degrees of freedom
## Multiple R-squared:  0.006292,   Adjusted R-squared:  -0.02576 
## F-statistic: 0.1963 on 1 and 31 DF,  p-value: 0.6608
```

```r
summary(lm(task_strategies ~ gs+sp+se+ts+el+hs+ge, data = dat))
```

```
## 
## Call:
## lm(formula = task_strategies ~ gs + sp + se + ts + el + hs + 
##     ge, data = dat)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.01701 -0.52167  0.00439  0.42259  1.23729 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.312496   0.469313   4.927  4.5e-05 ***
## gs           0.001243   0.004146   0.300   0.7669    
## sp          -0.020007   0.017408  -1.149   0.2613    
## se           0.004556   0.016269   0.280   0.7817    
## ts           0.007300   0.064840   0.113   0.9113    
## el           0.028474   0.065391   0.435   0.6670  
## hs          -0.001743   0.005257  -0.332   0.7430  
## ge           0.020140   0.008699   2.315   0.0291 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.6253 on 25 degrees of freedom
## Multiple R-squared:  0.2127, Adjusted R-squared:  -0.007715 
## F-statistic: 0.965 on 7 and 25 DF,  p-value: 0.4773
```

```
# elaboration
summary(lm(elaboration ~ el, data = dat))
```

```
## 
## Call:
## lm(formula = elaboration ~ el, data = dat)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.36446 -0.69780  0.02111  0.68778  1.35444 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.31222    0.18626  12.414 1.45e-13 ***
## el           0.05224    0.06995   0.747    0.461    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.8038 on 31 degrees of freedom
## Multiple R-squared:  0.01768,    Adjusted R-squared:  -0.01401 
## F-statistic: 0.5578 on 1 and 31 DF,  p-value: 0.4608
```

```
summary(lm(elaboration ~ gs+sp+se+ts+el+hs+ge, data = dat))
```

```
## 
## Call:
## lm(formula = elaboration ~ gs + sp + se + ts + el + hs + ge, 
##     data = dat)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.51427 -0.46785  0.02443  0.40771  1.26149 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)   
## (Intercept)  1.756483   0.585141   3.002  0.00601 **
## gs           0.004648   0.005170   0.899  0.37723   
## sp          -0.034502   0.021705  -1.590  0.12449   
## se          -0.020820   0.020284  -1.026  0.31450   
## ts           0.077320   0.080843   0.956  0.34802   
## el           0.017718   0.081530   0.217  0.82973   
## hs           0.005060   0.006554   0.772  0.44736   
## ge           0.020333   0.010846   1.875  0.07255 . 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7796 on 25 degrees of freedom
## Multiple R-squared:  0.2548, Adjusted R-squared:  0.04615
## F-statistic: 1.221 on 7 and 25 DF,  p-value: 0.3282
```

```r
# help seeking
summary(lm(help_seeking ~ hs, data = dat))
```

```
##
## Call:
## lm(formula = help_seeking ~ hs, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29532 -0.32715 -0.02612  0.66208  1.00079
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.993821   0.191366  15.644 2.92e-16 ***
## hs          0.005384   0.005831   0.923    0.363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.874 on 31 degrees of freedom
## Multiple R-squared:  0.02677,    Adjusted R-squared:  -0.004625
## F-statistic: 0.8527 on 1 and 31 DF,  p-value: 0.3629
```

```r
summary(lm(help_seeking ~ gs+sp+se+ts+el+hs+ge, data = dat))
```

```
##
## Call:
## lm(formula = help_seeking ~ gs + sp + se + ts + el + hs + ge,
##     data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92847 -0.31187 -0.04408  0.50763  1.38173
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.3103886  0.6336539   5.224 2.09e-05 ***
## gs           0.0081456  0.0055983   1.455    0.158
## sp          -0.0001842  0.0235043  -0.008    0.994
## se          -0.0263560  0.0219654  -1.200    0.241
## ts          -0.1444105  0.0875458  -1.650    0.112
## el           0.0620177  0.0882894   0.702    0.489
## hs           0.0013241  0.0070975   0.187    0.854
## ge           0.0100627  0.0117450   0.857    0.400
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8442 on 25 degrees of freedom
## Multiple R-squared:  0.2676, Adjusted R-squared:  0.06255
## F-statistic: 1.305 on 7 and 25 DF,  p-value: 0.2887
```

```
###############################################
###############################################
```

**Question 6:** Describe what you found. Which SRL strategies, if any, were you able to predict with which features? Were there any surprises? Which self-reported SRL strategy were you able to predict best with all features (look at the `Multiple R-squared`)?

I was not able to accurately predict SRL strategies from any of the features I engineered. Looking at the boxplots, most of the responses were 2 or 3 and I suspect the lack of variation in the responses made it hard to predict. Using all features, task strategy had the best predictions with the lowest multiple R-squared. # Self-reflection (ungraded)

**Briefly summarize your experience on this homework. What was easy, what was hard, what did you learn?**

This homework did feel easier and more straight forward compared to the past homeworks.

## Submit Homework

This is the end of the homework. Please **Knit a PDF report** that shows both the R code and R output and upload it on the EdX platform. Alternatively, you can Knit it as a "doc", open it in Word, and save that as a PDF.

**Important:** Be sure that all your code is visible. If the line is too long, it gets cut off. If that happens, organize your code on several lines.