Dashboards for Clicker Data

INFO 5200 Learning Analytics: Week 11 Homework

[[Cole Walsh, 4399966]]

In this homework, you will learn how to make a dashboard using R Shiny for clicker data. Here is the official page with several tutorials (https://shiny.rstudio.com/tutorial/). You are given aggregated clicker records for a CS course taught at Cornell. There are two datasets, the experience dataset and the quiz dataset.

Learning Objectives

1. Understand the structure of clicker data
2. Create multiple different visualizations
3. Design and implement an instructor dashboard and a student dashboard

Scenario

You are approached by a college instructor who uses iClickers in her CS class on Business Intelligence. She would like to gain insights about her students and how they are engaging/performing in order to better help them in class. She would also like to better support students by giving them feedback at scale about where they stand and perhaps how they compare to others in the class.

You offer to build a prototype of two dashboards using her clicker data: one is a dashboard for the instructor which offers an overview of the class characteristics, engagement, and performance; the other is a dashboard for students which offers a specific student an overview of their engagement and performance (and how it compares to others).

Data

The **experience dataset** contains one record per student who completed the CS course between 2016-2018. There are two sources to this dataset: Faculty Center and a Skills Survey (administered via Blackboard) where students self reported their skill level for various skills the first week of class. This data has been de-identified. Name, netid, emplid, major have all been removed and replaced with a unique numeric identifier. Note that not all students completed the skills survey, they will have null values for the survey result fields.

Attribute Name	Data Type	Definition
student_key	numeric Unique key	Assigned as part of de-identification process. Uniquely identifies student records for this data set only.
year	numeric	Four digit year student was enrolled in BI Class.
prog	character Values (GRAD, UGRAD)	Indicates whether the student was a graduate or undergraduate student when they were enrolled in BI course.
database_score	numeric (0-5)	Self reported experience level with database technology prior to taking course. 0= no experience, 5= expertise
sql_score	numeric (0-5)	Self reported experience level with SQL prior to taking course. 0= no experience, 5=expertise
programing_score	numeric (0-5)	Self reported experience level with Any Programing language prior to taking course. 0=no experience, 5=expertise
stored_proc_score	numeric (0-5)	Self reported experience level with stored procedure languages prior to taking course. 0=no experience, 5=expertise
etl_score	numeric (0-5)	Self reported experience level with Extract Transform Load (ETL) development prior to taking course. 0=no experience, 5=expertise

Attribute Name	Data Type	Definition
data_vis_score	numeric (0-5)	Self reported experience level using data visualization tools prior to taking course. 0=no experience, 5=expertise
requirement_gather_score	numeric (0-5)	Self reported experience level gathering customer requirements prior to taking course. 0=no experience, 5=expertise
skill_survey_score	numeric	Sum of the self reported skill level scores.

The **quiz dataset** contains one record per student per class session held where iClickers were used. Sources used in the creation of this data set include: iClicker session xml files, Blackboard gradebook (for quiz scores), and the Blackboard class schedule (used to map iClicker session to related quiz scores). Note that in some cases there are multiple iClicker sessions / lectures associated with a single quiz. This dataset may be joined to the experience dataset by the student_key field.

Attribute Name	Data Type	Definition
Acad_date_key	numeric	Date key in the form of YYYYMMDD indicating the date the class session was held.
student_key	numeric	Unique identifier for students who took BI class 2016-2018. This key is the primary key for the experience_data file.
year	numeric	Four digit year class session was held.
session_number	numeric	Identifies the session number for a particular semester. Session number is assigned by iClicker.
quiz_number	numeric	There are 10 quizzes throughout the BI course. This attribute indicates which quiz is associated with the iClicker session(s).
attended	numeric (0,1)	Binary indicating whether the student attended that particular class session / lecture. 0=no, 1=yes.
total_possible_clicker	numeric	The total number of iClicker questions asked that session.
total_completed_clicker	numeric	The number of iClicker questions answered by student that session.
completed_q_clicker	numeric	The number of completed Quiz iClicker questions
correct_q_clicker	numeric	How many correct Quiz answers by student that session.
completed_t_clicker	number	How many Temperature questions answered by student that session. Temperature questions are 0-5, 0= bad, 5=great. There is no correct answer to Temperature questions, they are used to guage how students are feeling about a particular subject, assignment, etc.
avg_t_clicker	number	The average temperature answer by student for that session. An average of 1 or 2 would be generally negative, while 4 or 5 would be generally positive responses.
quiz_score	numeric	Quiz score out of 20 points possible.

Part 1: Planning

Go through the planning process described in the reading for this week. While some dashboards are certainly better than others, there is not one correct solution here.

Question 1: Briefly answer the following prompts for both the teacher and student dashboards. The more concrete you are here the easier it will be later. Try to focus on a few important points/questions that you will implement in the next step. You can iterate on this step and modify your responses as your ideas for the dashboard become clearer.

- Why? What is the goal? What questions to answer?

  - Instructor Dashboard
    - What are the characteristics/skills of students in the class? What is the overall student performance on quizzes? Are clicker questions related to quiz performance? Are individual students at risk? How do student's feel (temperature) about a particular subject? Attendance/# of clicker questions attempted
  - Student Dashboard
    - How am I doing on quizzes relative to other students? Performance over time compared to other students on clicker questions. Do I need to brush up on a particular skill before/while taking this course?
- For whom? Who will use it and what is their background?
  - Instructor Dashboard
    - Instructors of the course who have a background in CS/business/technical expertise
  - Student Dashboard
    - Students taking the course who have a motivation for taking CS/business courses and are attending Cornell University
- What? What data to show and what is its structure?
  - Instructor Dashboard
    - Student's background/prior preparation — self-reported, ordinal from lower to higher preparation
    - Temperature scores — how students feel about a particular subject
    - Student performance, clicker questions answered and % of clicker questions correct for each session, and 10 quiz scores associated with at least 1 session
  - Student Dashboard
    - Similar as instructors, but I don't think temperature scores are useful here and can probably only have a negative effect
- How? How will visualizations support the goal?
  - Instructor Dashboard
    - Number of students in the class
    - Class average attendance
    - Distribution of avg quiz scores (overall and for each quiz), histogram
    - Mean/Median of student responses to skills survey (flipped coords) — do skills stick out as students having little preparation?
    - Correlations of clicker performance with quiz performance — could also show scatter plots, one point per student, could break this down further and examine correlations/scatter plots of session performance with quiz performance
  - Student Dashboard
    - Average attendance
    - Personal average quiz score
    - Performance (as a dot) relative to other students (as boxplots) on quizzes
    - Performance (as a dot) relative to other students (as boxplots) over time on clicker questions
    - Self-reported skills (as a dot) relative to other students in the class (as boxplots)

Part 2: Data Pre-processing

Get the data ready for use in the dashboard. Before the next stage, you want to have the data ready in the right format for simple computations and plotting. To do this effectively, you need to know by now what you want to display in each dashboard. However, this is also an iterative process. Once you have completed a first iteration of the design, you can come back to this step and add further pre-processing for more visualizations you like to add. This step is also an opportunity to better understand the structure of the datasets.

The student dashboard is typically for focused on an individual student. You can either pick a student (at random or intentionally) up here and use them as the "reference student" for the student dashboard. Or, a bit more ambitious but also more rewarding to try out, you can create an interactive dashboard in which you select the student and then the dashboard updates to show the information for that student. I would recommend you start with the simpler version and get that to work before you try to make it dynamic.

Question 2: Process the data in whatever way you need for it to be ready for your information visualizations in the dashboards.

```r
#########################################
####### BEGIN INPUT: Question 2 #######
#########################################

# Dataframe of 1 row for each student for each quiz

Student <- quiz %>%
  group_by(STUDENT_KEY, QUIZ_NUMBER) %>%
  mutate(QUIZ_SCORE = replace_na(QUIZ_SCORE, 0)) %>%
  summarize(quiz.score = QUIZ_SCORE[1]) %>%
  na.omit()

# Dataframe of average temperature for each session, using only students who completed at
# least one temperature question

Session.Temperature <- quiz %>%
  filter(COMPLETED_T_CLICKER > 0) %>%
  group_by(SESSION_NUMBER) %>%
  summarize(avg.temperature = mean(AVG_T_CLICKER))

# Dataframe of average clicker correctness for each session, using only students who
# completed at least one clicke question

Session.Performance <- quiz %>%
  filter(COMPLETED_Q_CLICKER > 0) %>%
  group_by(SESSION_NUMBER) %>%
  summarize(avg.correct = mean(CORRECT_Q_CLICKER),
            clicker.quiz.cor = cor(CORRECT_Q_CLICKER/COMPLETED_Q_CLICKER, QUIZ_SCORE,
                                   use = 'pairwise.complete.obs'))

# Dataframe with average scores for each self-reported skill

Subject.cols <- c("DATABASE_SCORE", "SQL_SCORE", "PROGRAMING_SCORE", "STORED_PROC_SCORE",
                  "ETL_SCORE", "DATA_VIS_SCORE", "REQUIREMENT_GATHER_SCORE")

Experience <- data.frame(avg.experience = sapply(experience[, Subject.cols],
                                                 function(x) mean(x, na.rm = TRUE)))
Experience$Question <- rownames(Experience)

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

Part 3: Dashboard Wire frame Implementation

This is where you generate the dashboard layout. You are given a wire frame example for the dashboard below. For more information on how R Shiny Dashboards work, look at https://rstudio.github.io/shinydashboard/get_started.html and https://rstudio.github.io/shinydashboard/structure.html.

Note: You can add different types of content into a `fuidRow()`. In the starter code there are 2 rows of content: the first has two little info boxes; the second has two larger viz boxes. You can add more rows and change what is in them as you wish. Follow the naming convention, e.g. `inst.info1` is the first info box for instructors.

Question 3: Create the layout for the instructor dashboard tab. Here you are just specifying the wire frame i.e. **what goes where on the page**.

```r
# Instructor Dashboard Tab
instructor_tab = tabItem(
    tabName = "instructor",
    h2("Instructor Dashboard"),

#######################################
####### BEGIN INPUT: Question 3 #######
#######################################
    # Dynamic infoBoxes
    fluidRow(
      infoBoxOutput("inst.info1"),
      infoBoxOutput("inst.info2")
    ),
    # Any visualization
    fluidRow(
        box(
            title = "How are my students doing on quizzes?",
            plotOutput("inst.plot1", height = 250)
        ),
        box(
            title = "How do my students feel about sessions?",
            plotOutput("inst.plot2", height = 350)
        )
    ),
    fluidRow(
        box(
            title = "What are my students' preparation?",
            plotOutput("inst.plot3", height = 250)
        ),
        box(
            title = "Are my clicker questions related to the quizzes associated with a
            session?",
            plotOutput("inst.plot4", height = 250)
        )
    ),
    fluidRow(
        box(
            title = "How difficult are my quizzes?",
            plotOutput("inst.plot5", height = 250)
        ),
        box(
            selectInput("quiz", "Quiz Number:", choices = sort(unique(quiz$QUIZ_NUMBER)))
        )
    )
#######################################
#######################################

)
```

Question 4: Create the layout for the student dashboard tab. Again, you are just specifying the wire frame i.e. **what goes where on the page**.

```
# Student Dashboard
student_tab = tabItem(
    tabName = "student",
    h2("Student Dashboard"),

#######################################
####### BEGIN INPUT: Question 4 #######
#######################################
    # Dynamic infoBoxes
    fluidRow(
      infoBoxOutput("stud.info1"),
      infoBoxOutput("stud.info2")
    ),
    # Any visualization
    fluidRow(
        box(
            title = "How am I doing on quizzes relative to other students?",
            plotOutput("stud.plot1", height = 250)
        ),
        box(
            title = "How does my background and preparation compare to other students?",
            plotOutput("stud.plot2", height = 350)
        )
    ),
    fluidRow(
        box(
            title = "How does my performance on clicker questions compare to other
            students?",
            plotOutput("stud.plot3", height = 250)
        )
    )
#######################################
#######################################

)
```

Part 4: Prepare All Data Visualizations

This is where you create the content for the wire frames you created in part 3. Again, you can refer to the examples and documentation in https://rstudio.github.io/shinydashboard/get_started.html and https:

//rstudio.github.io/shinydashboard/structure.html for guidance. You can also find many examples online just by searching with Google.

Question 5: For each of the pieces of content you planned for in the wire frames above, generate that content. You need to assign them all to the `output` variable by referencing the name of the wire frame element you chose above like this `output$name.of.element`.

```
# Instructor Dashboard Tab
server = function(input, output) {

#######################################
####### BEGIN INPUT: Question 5 #######
#######################################
    quiz
    #### Student Dashboard ####
    output$stud.info1 = renderInfoBox({
        infoBox("Attendance",
                paste0(round(100 * mean(quiz[quiz$STUDENT_KEY == input$student,
                                             'ATTENDED'])), "%"),
                icon = icon("list"), color = "purple")
    })

    output$stud.info2 = renderInfoBox({
        infoBox("Average Quiz Score",
                paste0(5 * round(Student %>%
                               filter(STUDENT_KEY == input$student) %>%
                               summarize(mean(quiz.score)) %>%
                               pull()), '%'),
                icon = icon("list"), color = "yellow")
    })
    output$stud.plot1 = renderPlot({
        ggplot(Student, aes(x = as.factor(QUIZ_NUMBER), y = quiz.score)) +
          geom_boxplot(outlier.shape = NA) +
          geom_point(data = Student[Student$STUDENT_KEY == input$student,],
                     aes(x = as.factor(QUIZ_NUMBER), y = quiz.score, color = 'red', size = 20)) +
          theme_classic() +
          theme(legend.position = 'none') +
          labs(x = 'Quiz Number', y = 'Quiz Score')
    })
    output$stud.plot2 = renderPlot({
        ggplot(melt(experience[, c('STUDENT_KEY', Subject.cols)], id.vars = 'STUDENT_KEY'),
               aes(x = variable, y = value)) +
          geom_boxplot(outlier.shape = NA) +
          geom_point(data = melt(experience[experience$STUDENT_KEY == input$student,
                                            Subject.cols]), aes(x = variable, y = value,
                                                                color = 'red', size = 20)) +
          theme_classic() +
          theme(legend.position = 'none') +
          labs(x = 'Subject', y = 'Reported Experience') +
          scale_x_discrete(labels = c('Database', 'SQL', 'Programming', 'Stored Procedure',
                                      'ETL', 'Data Visuals', 'Requirement Gathering'))
    })
    output$stud.plot3 = renderPlot({
        ggplot(quiz, aes(x = as.factor(SESSION_NUMBER), y = CORRECT_Q_CLICKER)) +
          geom_boxplot() +
          geom_point(data = quiz[quiz$STUDENT_KEY == input$student,],
                     aes(x = SESSION_NUMBER, y = CORRECT_Q_CLICKER, color = 'red', size = 20)) +
        theme_classic() +
        theme(legend.position = 'none') +
        labs(x = 'Session Number', y = 'Number of clicker questions correct')

    })


    #### Instructor Dashboard ####
    output$inst.info1 = renderInfoBox({
        infoBox("Students total",
                length(unique(quiz$STUDENT_KEY)),
                icon = icon("list"), color = "purple")
    })
    output$inst.info2 = renderInfoBox({
        infoBox("Attendance",
                paste0(round(100 * mean(quiz$ATTENDED)), "%"),
                icon = icon("list"), color = "yellow")
    })
    output$inst.plot1 = renderPlot({
        ggplot(Student %>%
                 group_by(STUDENT_KEY) %>%
                 summarize(avg.quiz.score = mean(quiz.score)), aes(x = avg.quiz.score)) +
        geom_histogram() +
        theme_classic() +
        labs(x = 'Average Quiz Score', y = 'Number of Students')
    })
    output$inst.plot2 = renderPlot({
        ggplot(Session.Temperature, aes(x = reorder(SESSION_NUMBER, avg.temperature),
                                        y = avg.temperature)) +
        geom_point() +
        coord_flip() +
        theme_classic() +
        labs(x = 'Session', y = 'Average Temperature Score')
    })
    output$inst.plot3 = renderPlot({
        ggplot(Experience, aes(x = Question, y = avg.experience)) +
        geom_point() +
        theme_classic() +
        labs(x = 'Subject', y = 'Average Reported Experience') +
        scale_x_discrete(labels = c('Database', 'SQL', 'Programming', 'Stored Procedure',
                                    'ETL', 'Data Visuals', 'Requirement Gathering'))
    })
    output$inst.plot4 = renderPlot({
        ggplot(Session.Performance, aes(x = reorder(SESSION_NUMBER, clicker.quiz.cor),
                                        y = clicker.quiz.cor)) +
        geom_point() +
        coord_flip() +
        theme_classic() +
        labs(x = 'Session', y = 'Correlation of Average Clicker Score with Quiz Score')
    })
    output$inst.plot5 <- renderPlot({
      ggplot(Student[Student$QUIZ_NUMBER == input$quiz,], aes(x = quiz.score)) +
        geom_histogram() +
        theme_classic() +
        labs(x = paste('Score on Quiz ', input$quiz), y = 'Number of Students')
   })
#######################################
#######################################

}
```

Part 5: Produce Dashboard and Reflect

You should be able to simply run the code below **as is** to see your dashboard prototype.

Note: Unfortunately, you cannot knit this part into a pdf. So I added `eval=FALSE` to let the knitting run smoothly and you can submit your PDF.

```
#######################################
### This code creates the dashboard ###
#######################################

# Here we set up the Header of the dashboard
dhead = dashboardHeader(title = "Clicker Dashboard")

# Here set up the sidebar which has links to two pages
dside = dashboardSidebar(sidebarMenu(
    menuItem("Instructor View", tabName = "instructor", icon = icon("dashboard")),
    menuItem("Student View", tabName = "student", icon = icon("dashboard")),
    selectInput("student", "Student ID:", choices = sort(unique(quiz$STUDENT_KEY)))
))

# Here we set up the body of the dashboard
dbody = dashboardBody(
    tabItems(instructor_tab, student_tab)
)

# Combining header, sidebar, and body
ui = dashboardPage(dhead, dside, dbody)

# Generating a local instance of your dashboard
shinyApp(ui, server)
```

Question 6: Evaluate your own prototype from the perspective of the instructor (instructor dashboard) and then from the perspective of the student (student dashboard). What do you like, what would you change, what else would you like to see?

- **Instructor dashboard evaluation:** I think I covered most of the things the instructor was interested in, particularly the performance of the class as a whole on quizzes and individual quizzes. Instructors can also evaluate the quality of their quizzes as they relate to clicker questions and their students' background. I liked that I allowed for some interaction on the art of the instructor to look at particular quizzes that they are interested in. This layout doesn't really allow instructors to immediately see what students or quizzes are outliers. By accessing the student dashboard, one can obtain some student by student information, but this isn't readily avaliable to instructors. With more time, I would've made this more interactive and 'prettify' it a little more. It seems bland to me at the moment and although it makes information readily apparent, it doesn't highlight trends and outliers that I might think would be useful information.

- **Student dashboard evaluation:** First off, this could augment the instructor dashboard, since information is available for any student from a dropdown menu. Students are able to immediately see what percentage of classes they've attended and their average quiz score — perhaps, the single most important piece of information to a student. They can also see their performance on quizzes and clickers over time as compared to other students, which is useful to know where you stand in a course. Similarly, students can compare their self-reported background skills to others. Thus, if they are lacking in a particular skill, they can see that they are not alone in that regard. I didn't provide a lot of information here, partly because I think that students are most interested in how they're doing on quizzes compared to the class. Similar to my issues with the instructor dashboard, I didn't leave a lot of time to add interactivity and 'prettify' the dashboard. At the student level, this isn't as concerning to me, since I believe the data students should see should not be as fine-grained as that available to an instructor, but if I had more time I would consider adding more interactive plots and making this look nicer.

Self-reflection

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

First time building interactive plots and dashboards, so took a while, but I feel like this is the most useful homework we've done since I think this will be super beneficial to me in the near future where my research requires conveying information to instructors, as well as long-term where I can see myself using dashboards again. Fun and useful!

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.