Dashboards for Clicker Data

INFO 4100 Learning Analytics

[[Eva Ruse, ewr43]]

This project is about developing a learning analytics dashboard based on clicker data. You will work as a team to learn how to make a dashboard using R Shiny (official page with several tutorials: <https://shiny.rstudio.com/tutorial/>).

**Learning Objectives**

1. Understand the structure of clicker data
2. Create multiple different visualizations
3. Design and implement an instructor and student dashboard
4. Critically evaluate your own dashboard design

You are given aggregated clicker records for a CS course taught at Cornell. There are two datasets: the experience dataset and the quiz dataset.

**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 a dashboard using her clicker data: this is a dashboard for the instructor which offers an overview of the class characteristics, engagement, and performance; and it 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 the Blackboard LMS) 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 |
| 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 / Sketching

Go through the planning / sketching process described in the reading about dashboards. While some dashboards are certainly better than others, there is not one correct solution here. However, spending enough time to make a concrete plan is essential for the success of your project. Everything you do to make the dashboards will be easier if you have a clear plan, especially because you will be splitting up the work and everyone needs to know what they should work on.

**Question 1:** You will make a student dashboard and a teacher dashboard. Carefully consider the implications of this for design and content. To plan, answer the following prompts once for the student dashboard and then for the teacher dashboard. The more concrete you are here the easier it will be later. Focus on the concrete ideas that you will implement in the next steps. You can iterate on this step and modify your responses as your ideas for the dashboard become clearer. You should explore the dataset in R for 5-10 minutes to get a good sense of what the dataset has to offer.

*Planning for the student dashboard* \* For whom? Who will use it and what is their background? + Individual students in this course + They have taken this course with the skills survey and in-class clicker quizzes \* Why? What is the goal? What questions to answer? + To keep students informed of their attendance and quiz entries + How many \* What? What data to show and what is its structure? + Attendance rate as the average of “attended” + percentage of clicker questions answered + percentage of correct clicker questions answered each session + quiz score frequency + quiz score achieved each session \* How? How will visualizations support the goal? + info box for attendance rate + info box for percentage of clicker questions answered + scatter plot for percentage of correct clicker questions answered trending + histogram for quiz score frequency + scatter plot for quiz score trending

*Planning for the teacher dashboard*

* For whom? Who will use it and what is their background?
  + This is for students
  + These students are taking a CS class in Business Intelligence
* Why? What is the goal? What questions to answer?
  + The goal is to provide the student with insight into how they are doing in the class
  + Some questions could be how the student compare to other students, what their potential grade outcome is.
* What? What data to show and what is its structure?
  + We can show averages; average attendance, average quiz score.
  + [add your answer here]
* How? How will visualizations support the goal?
  + We can show a histogram of the quiz scores for a student, scatterplot to show how attendance has changed over the course of the semester

*Planning for the teacher dashboard* \* For whom? Who will use it and what is their background? + For teachers + This teacher teaches business intelligence \* Why? What is the goal? What questions to answer? + To compare students across different years + To see if students similarly struggle with certain things \* What? What data to show and what is its structure? + Compare undergraduate and graduate performance + show attendance by session + show quiz score by skill survey score + show quiz score by clicker temperature + Compare quiz scores over multiple years \* How? How will visualizations support the goal? + Scatterplots + boxplots + line graphs + These visualizations will help the instructors gather an overall understanding of students by year just by glancing at the plot.

**Question 2:** Based on your plan above, make a sketch of what the dashboard would look like. See this week’s readings for examples. Be detailed about what kinds of data points and visualizations you want to see in different parts of the page. Consider the user experience and how you should position more general information compared to more specific information, and where you may need some additional explanation to help the viewer understand a graphic, for example. In your sketch, it is useful to give labels to different objects, because in the steps below you can split up work between team members and the labels will help you connect the UI with the data objects. Show your sketches in section to get feedback from the teaching team.

# Part 2: Dashboard Wire-frame Implementation

This is where you generate the dashboard layout. You are given a very basic 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>. 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.

Your team can split up the tasks. Some work on creating the UI (this part), while others work on pre-processing the data and creating the statistics and visualizations that will populate the UI (next part).

**Question 3:** Create the layout for the dashboard tabs. You can have as many “tabs” as you like. Each tab is the content displayed when the user clicks on one of the menu items (so it is the page content). Here you are just specifying the wire frame i.e. **what goes where on the pages**, not what goes into it.

#######################################  
####### BEGIN INPUT: Question 3 #######  
#######################################  
# Example of a tab (i.e. page)  
instructor\_dash = tabItem(  
 tabName = "instructor",  
 h2("Instructor Dashboard"),  
 selectInput(inputId="years",   
 label="Select Year:",   
 choices=c("All Years","2018", "2017", "2016"), width=100),  
   
 # Dynamic infoBoxes  
 fluidRow(  
 infoBoxOutput("instructor.attendance"),  
 infoBoxOutput("instructor.quiz\_score"),  
 infoBoxOutput("instructor.skill\_survey\_score")  
 ),  
 # Any visualization  
  
 fluidRow(  
 box(  
 title = "Attendance",  
 plotOutput("instructor.attendance\_per\_session", height = 250)  
 ),  
 box(  
 title = "Quiz Score By Temperature Clicker Session",  
 plotOutput("instructor.quiz\_score\_temperature", height = 250)  
 ),  
 box(  
 title="Quiz Scores by Skill Survey Score",  
 plotOutput("instructor.quiz\_score\_skill", height = 250)  
   
 ),  
 box(  
 title="Undergraduate vs. Graduate Quiz Scores",  
 plotOutput("instructor.ug\_v\_g", height=250)  
   
 )  
)  
)  
  
  
# Another empty tab  
student\_dash = tabItem(  
 tabName = "student",  
 h2("Student Dashboard"),  
 # Dynamic infoBoxes  
 fluidRow(  
 infoBoxOutput("student.attendance"),  
 infoBoxOutput("student.q\_answered")  
 ),  
 # Visualization  
 fluidRow(  
 box(  
 title = "Correct Quiz Rate Trend",  
 plotOutput("student.correct\_q\_rate\_trend", height = 250)  
 ),  
 box(  
 title = "Quiz Score Frequency",  
 plotOutput("student.quiz\_score\_frequency", height = 250)  
 ),  
 box(  
 title = "Quiz Score Trend",  
 plotOutput("student.quiz\_score\_trend", height = 250)  
 ),  
 ),  
)  
  
#######################################  
#######################################

# Part 3: 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 instructor dashboard should show information for all students. The student dashboard is typically focused on an individual student. You can either pick a student (at random or intentionally) 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.

Use the space below to be ready for your information visualizations in the dashboards.

#######################################  
####### BEGIN INPUT #######  
#######################################  
quiz[is.na(quiz)] <- 0  
comb<-left\_join(quiz, experience, by="STUDENT\_KEY")  
  
  
# data processing for student dashboard  
#correct\_q\_clicker/completed\_q\_clicker for a student for a session  
quiz <- quiz %>%  
 mutate(CLICKER\_QUIZ\_PERCENT = ifelse(COMPLETED\_Q\_CLICKER == 0, 0, CORRECT\_Q\_CLICKER/COMPLETED\_Q\_CLICKER))  
#average correct\_q\_clicker/completed\_q\_clicker for a session for a year  
quiz <- quiz %>%  
 group\_by(SESSION\_NUMBER, YEAR) %>%  
 mutate(AVG\_CLICKER\_QUIZ\_PERCENT = mean(CLICKER\_QUIZ\_PERCENT)) %>%   
 ungroup  
#student attendance view  
quiz <- quiz %>%  
 group\_by(YEAR, STUDENT\_KEY) %>%  
 mutate(ATTENDANCE = mean(ATTENDED)) %>%   
 ungroup  
#student completed clicker question view  
quiz <- quiz %>%  
 group\_by(YEAR, STUDENT\_KEY) %>%  
 mutate(CLICKER\_Q\_ANSWERED = sum(TOTAL\_COMPLETED\_CLICKER)/sum(TOTAL\_POSSIBLE\_CLICKER)) %>%   
 ungroup  
  
#  
student\_sub\_quiz <- subset(quiz, STUDENT\_KEY==1)  
   
#pre-processing test for dynamic attendance by quiz score  
quiz\_sum <- quiz %>%  
 group\_by(YEAR,SESSION\_NUMBER) %>%  
 summarise(  
 avg\_attendance=mean(ATTENDED),  
 avg\_quiz\_score=mean(QUIZ\_SCORE)  
 )

## `summarise()` has grouped output by 'YEAR'. You can override using the `.groups` argument.

quiz\_sum\_all\_years=data.frame(quiz\_sum)  
quiz\_sum\_all\_years$YEAR="All Years"  
quiz\_sum$YEAR=as.character(quiz\_sum$YEAR)  
qa=rbind(quiz\_sum,quiz\_sum\_all\_years)  
  
#pre-processing for average temp clicker by quiz score  
student\_averages <-quiz %>%   
 group\_by(STUDENT\_KEY) %>%   
 summarise(   
 avg\_quiz\_score=mean(QUIZ\_SCORE),  
 avg\_clicker=mean(AVG\_T\_CLICKER),  
 YEAR=YEAR  
 )

## `summarise()` has grouped output by 'STUDENT\_KEY'. You can override using the `.groups` argument.

skill\_survey\_score=data.frame(experience$SKILL\_SURVEY\_SCORE)  
skill\_survey\_score$STUDENT\_KEY=experience$STUDENT\_KEY  
skill\_survey\_score$PROG=experience$PROG  
  
student\_averages=left\_join(skill\_survey\_score, student\_averages, by="STUDENT\_KEY")  
student\_averages\_all\_years = data.frame(student\_averages)  
student\_averages\_all\_years$YEAR="All Years"  
student\_averages$YEAR=as.character(student\_averages$YEAR)  
student\_averages\_db=rbind(student\_averages,student\_averages\_all\_years)  
  
colors <- c("clicker quiz percent"="darkblue", "average clicker quiz percent"="red")  
  
#######################################  
#######################################

# Part 4: Prepare All Data Visualizations

This is where you create the content for the wire frames you created above. 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 4:** For each of the pieces of content you planned for in the wire frames above, generate the relevant 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.

server = function(input, output) {  
   
#######################################  
####### BEGIN INPUT: Question 4 #######  
#######################################  
 # INSTRUCTOR  
 reactive\_year <- reactive({  
 input$years  
 })  
   
 observe({  
   
 # data processing by selected year  
 if(identical(reactive\_year(), "All Years")){  
 sub\_quiz <- quiz  
 sub\_experience <- experience  
 }else if(identical(reactive\_year(), "2018")){  
 sub\_quiz <- subset(quiz, YEAR==2018)  
 sub\_experience <- subset(experience, YEAR==2018)  
 }else if(identical(reactive\_year(), "2017")){  
 sub\_quiz <- subset(quiz, YEAR==2017)  
 sub\_experience <- subset(experience, YEAR==2017)  
 }else{  
 sub\_quiz <- subset(quiz, YEAR==2016)  
 sub\_experience <- subset(experience, YEAR==2016)  
 }  
   
 sub\_quiz$ATTENDED <- ifelse(is.na(sub\_quiz$ATTENDED), 0, sub\_quiz$ATTENDED)  
 sub\_quiz$QUIZ\_SCORE <- ifelse(is.na(sub\_quiz$QUIZ\_SCORE), 0, sub\_quiz$QUIZ\_SCORE)  
 sub\_experience <- sub\_experience %>% drop\_na(SKILL\_SURVEY\_SCORE)  
   
   
 # filling outputs  
 output$instructor.attendance = renderInfoBox({  
 infoBox("Attendance",  
 paste0(round(mean(sub\_quiz$ATTENDED)\*100, 1), "%"),  
 icon = icon("list"), color = "purple")  
 })  
   
 output$instructor.quiz\_score = renderInfoBox({  
 infoBox("Average Quiz Score",  
 paste0(round(mean(sub\_quiz$QUIZ\_SCORE), 2), "/20"),  
 icon = icon("list"), color = "yellow")  
 })  
   
 output$instructor.skill\_survey\_score = renderInfoBox({  
 infoBox("Average Skill Survey Score",  
 paste0(round(mean(sub\_experience$SKILL\_SURVEY\_SCORE),2),"/35"),  
 icon = icon("list"), color = "blue")  
 })  
   
   
  
 # end of observe   
 })  
   
 output$instructor.attendance\_per\_session = renderPlot({  
 ggplot(data=subset(qa, YEAR==input$years), aes(x=SESSION\_NUMBER, y=avg\_attendance)) +  
 geom\_line()+  
 geom\_point()+  
 labs(x = "Session Number", y = "Average Attendance")+  
 theme\_classic()  
 })  
  
   
 output$instructor.quiz\_score\_temperature = renderPlot({  
 ggplot(data=subset(student\_averages\_db, YEAR==input$years), aes(x=avg\_clicker, y=avg\_quiz\_score)) +  
 geom\_point()+  
 labs(x = "Average Clicker Temperature", y = "Average Quiz Score")+  
 theme\_classic()  
  
 })  
 output$instructor.quiz\_score\_skill = renderPlot({  
 ggplot(data=subset(student\_averages\_db, YEAR==input$years), aes(x=experience.SKILL\_SURVEY\_SCORE, y=avg\_quiz\_score)) +  
 geom\_point()+  
 labs(x = "Skill Survey Score", y = "Average Quiz Score")+  
 theme\_classic()  
 })  
   
 output$"instructor.ug\_v\_g" = renderPlot({  
 ggplot(data=subset(student\_averages\_db, YEAR==input$years), aes(x=PROG, y=avg\_quiz\_score)) +   
 geom\_boxplot()+  
 labs(x = "Program", y = "Average Quiz Score")+  
 theme\_classic()  
 })  
   
 # student dashboard   
 output$student.attendance = renderInfoBox({  
 infoBox("Attendance",  
 paste0(round(mean(student\_sub\_quiz$ATTENDANCE)\*100), "%"),  
 icon = icon("list"), color = "purple")  
 })  
   
 output$student.q\_answered = renderInfoBox({  
 infoBox("Clicker Questions Answered",  
 paste0(round(mean(student\_sub\_quiz$CLICKER\_Q\_ANSWERED)\*100), "%"),  
 icon = icon("list"), color = "yellow")  
 })  
 output$student.correct\_q\_rate\_trend= renderPlot({  
 ggplot(student\_sub\_quiz, aes(x = SESSION\_NUMBER))+  
 geom\_line(aes(y=CLICKER\_QUIZ\_PERCENT,color="clicker quiz percent"))+  
 geom\_line(aes(y=AVG\_CLICKER\_QUIZ\_PERCENT,color="average clicker quiz percent"))+  
 labs(x = "Session Number", y = "Clicker Quiz Percentage", color="Legend") +# change axis labels  
 theme\_classic()+  
 scale\_color\_manual(values=colors)  
})  
  
 output$student.quiz\_score\_frequency= renderPlot({  
 ggplot(student\_sub\_quiz, aes(QUIZ\_SCORE)) +  
 geom\_histogram(fill="darkblue")+  
 labs(x = "Quiz Score", y = "Frequency")+  
 theme\_classic()  
 })  
 output$student.quiz\_score\_trend= renderPlot({  
 ggplot(student\_sub\_quiz, aes(x=SESSION\_NUMBER, y=QUIZ\_SCORE)) +  
 geom\_line(color="darkblue")+  
 geom\_point(color="darkblue")+  
 labs(x = "Session Number", y = "Quiz Score")+  
 theme\_classic()  
 })  
#######################################  
#######################################  
}

# Part 5: Produce Dashboard and Reflect

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

**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("th"))  
 )  
)  
  
# Here we set up the body of the dashboard  
dbody = dashboardBody(  
 tabItems(  
 student\_dash,  
 instructor\_dash  
 )  
)  
# observe({  
# updateSelectInput(session, "years", choices = experience$YEAR)  
#})  
# Combining header, sidebar, and body  
ui = dashboardPage(dhead, dside, dbody)  
  
# Generating a local instance of your dashboard  
shinyApp(ui, server)

**Question 5:** Add screenshots of your group’s dahsboards below using this syntax or simply add them to the Word document after knitting:

**Question 6:** Evaluate your group dashboard from the perspective of the instructor (teacher dashboard) and from the perspective of the student (student dashboard). What do you like about it, what would you change or add to it if you had more time?

Student Dashboard:

Graphical user interface, application

Description automatically generated

Instructor Dashboard: All years

Graphical user interface, application

Description automatically generated

Instructor Dashboard: 2016

Graphical user interface, application

Description automatically generated

Instructor Dashboard: 2017

Graphical user interface, application

Description automatically generated

Instructor Dashboard: 2018

Graphical user interface, application

Description automatically generated

*Reflection for the student dashboard* \* What do you like about it? + The info boxes stating my current attendance and clicker data are straightforward and clearly readable. + The plots show me my performance trends and distributions. A comparison with the course average performance (represented by different colors in the same plot) was helpful. + I like that I can see how I’m doing in comparison with the rest of class. I also like that I can see the progress and quick stats about how I’m doing in the class. \* What would you change or add to it if you had more time? + If we had more time, we would work on a selection of study id so that the dashboard is not hard coded with just one student’s data. + If we had more time we would add a class average in the score histogram as well. \* What was the biggest challenge you faced? How did you address it? + We faced challenges adding different layers to a ggplot, especially when we were trying to create color keys. After some research online, we made it work by adding legend inside the ggplot labs.

*Reflection for the teacher dashboard* \* What do you like about it? + The year selection input give the instructor the flexibility to check data from all years or a specific year. \* What would you change or add to it if you had more time? + We could create more graphs using more of the available data, for example, incorporating each of the experience survey skillsets beyond using the overall score. + If we had more time we would add more comparability, perhaps by adding a functionality to see 2 years plotted against each other in each plot. \* What was the biggest challenge you faced? How did you address it? + It took us a while to feature out how to make our data responsive to the year selection input. To solve this problem, we used reactive() and observe(), and we adjusted our data pre-processing section.

# Submit Project

Please **Knit a Docx 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.