Dashboards for Clicker Data

INFO 4100 Learning Analytics

[[Caitlin Lau, CJL236; Destiny Malloy, DAM359; Esuvat Bomani,EKB72 ;Asma Khan, AK2355 ; Jeremy Clerc, KC2267; Trevor Warner, TW448]]

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.

This is why you (each student but ideally as a team) should come to OH during the first week and get feedback on your plan with sketches.

**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.

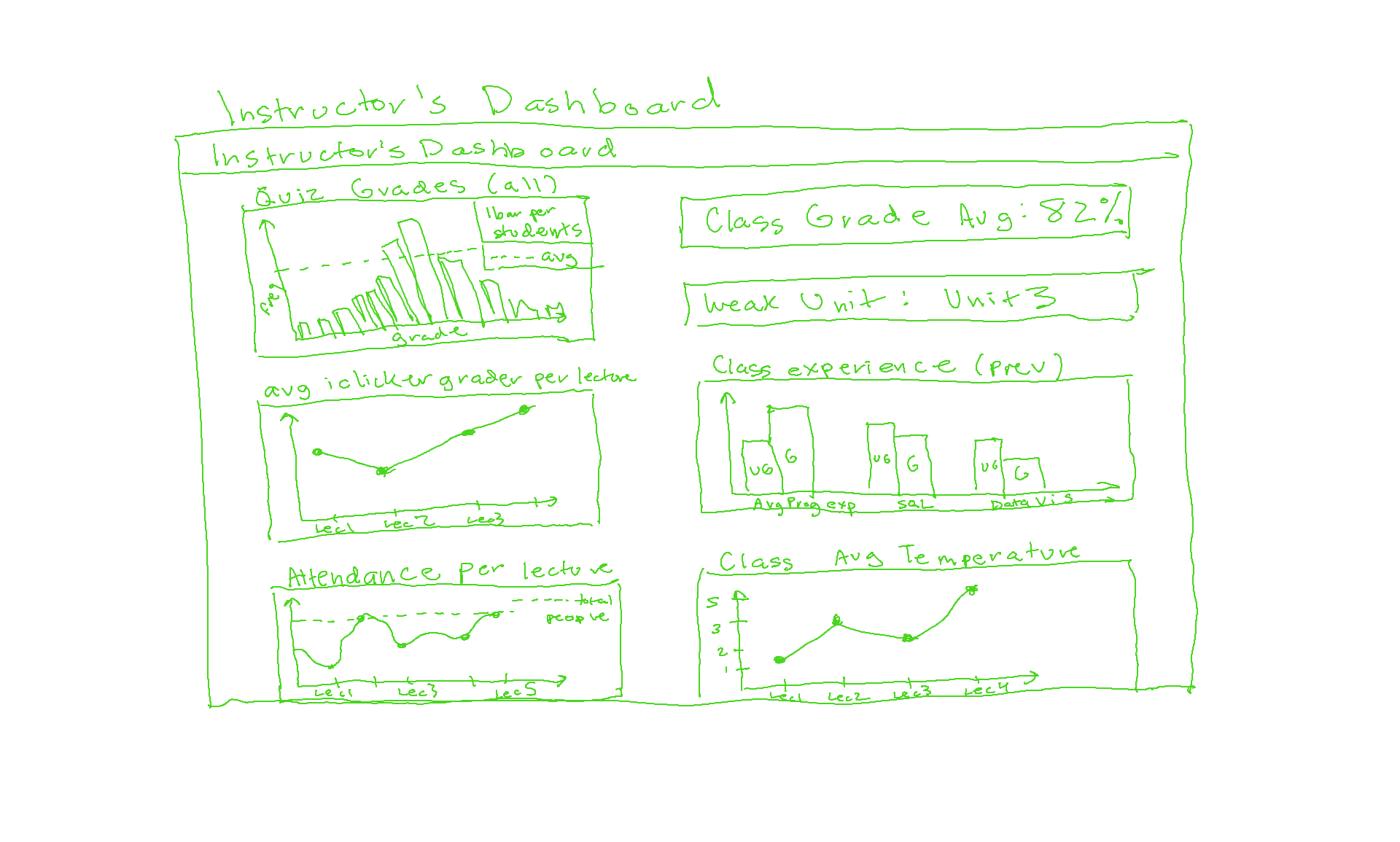
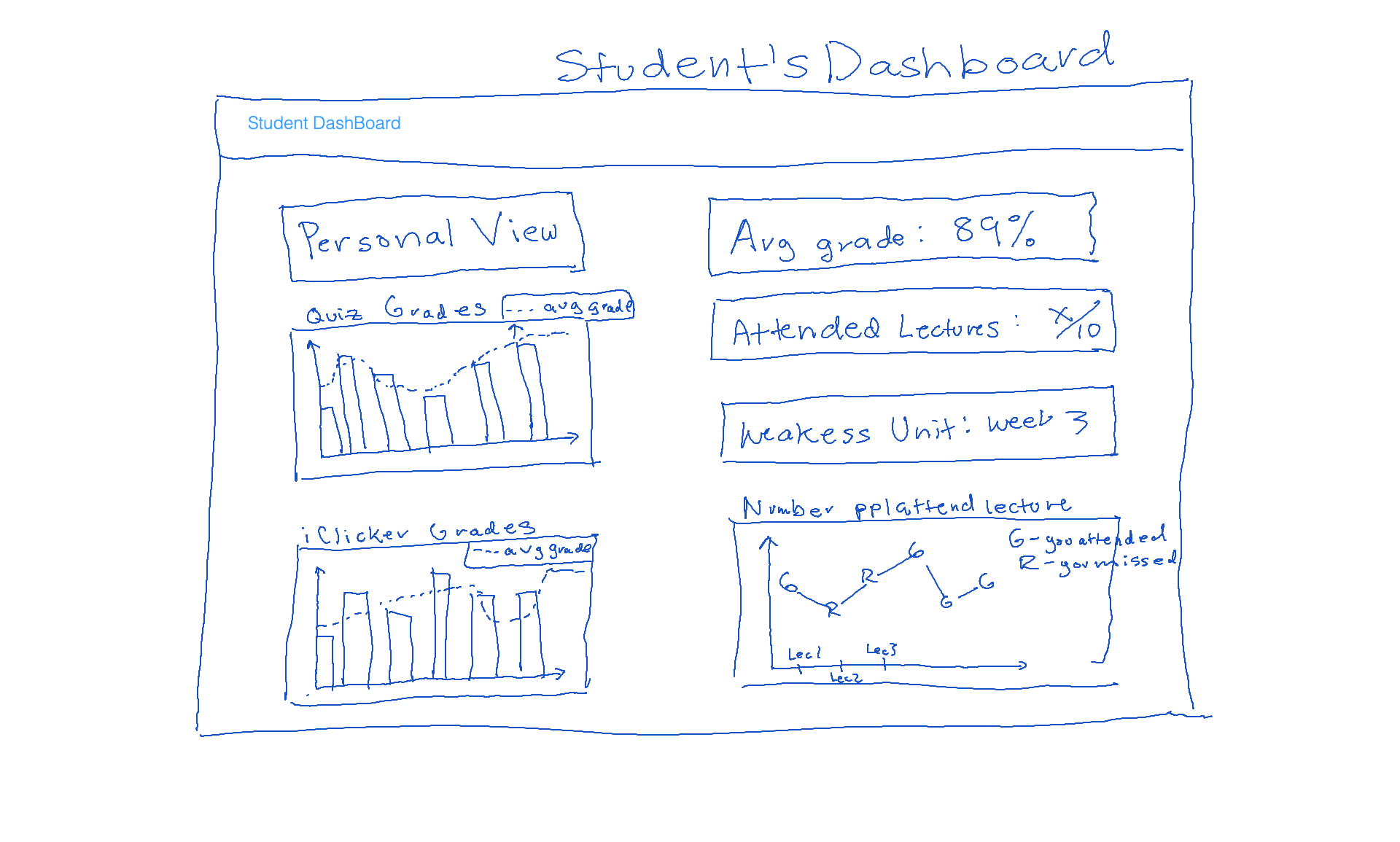
*Planning for the student dashboard*

* For whom? Who will use it and what is their background?
  + [A college (undergrad and grad) student who is interested in their grades in a CS business intelligence course]
  + [They come from a variety of backgrounds in programming experience, varied SQL experience, and data visualization experience]
* Why? What is the goal? What questions to answer?
  + [They want to track their progress in the course in order to maintain their grades or to see how well they need to do in the rest of the course, Identify weaknesses in their skills, provide a more interactive experience with videos and quizzes, ]
  + [Their goal to understand their standing in class. The questions to answers is what topics should they focus on more, How can I improve my time management for studying for this course.]
* What? What data to show and what is its structure?
  + [Show previous grades that the student has received. Their grades as compared to the rest of the class. Upcoming deadlines. ]
  + [It’s structures would be numeric means for previous grades and bar graphs and line graphs as visual representations. ]
* How? How will visualizations support the goal?
  + [For students the visualization will help them know how they are doing compared to other students at every single point in the semester rather than just exams. To know which topics they need to focus more on.]
  + [Gives them a broader perspective of the big picture of the course]

*Planning for the teacher dashboard*

* For whom? Who will use it and what is their background?
  + [College CS professor who is interested in tracking the progress of their student]
  + [CS department manager to oversee how well the CS professor is teaching their students]
* Why? What is the goal? What questions to answer?
  + [To identify when which students need intervention, A question would be: What topics the overall students need more help on?, ]
  + [The goal is to gauge the overall effectiveness of the course. Why: To see how much the students are participating in the course. To answer the question of gauging participation with performance. ]
* What? What data to show and what is its structure?
  + [The data to show is the grades that people have been getting in the form of a distribution rather than numeric values. Data to track the performance and participation in the course. The average temperature variable ( how students feel about how they are doing in the class)]
  + [Experience level of the students in the course prior to the course (programming, SQL, datavis). We will show its structure through histograms, scatterplots, and line graphs. ]
* How? How will visualizations support the goal?
  + [Save them a lot of time on figuring out how people are performing and how the course material is being received. ]
  + [Helps keep information organized for the instructor.]

**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 OH to get credit for this question.

# 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)  
#inserted this directly into dashboard() code below  
#instructor\_dash =   
  
# Another empty tab  
student\_dash = tabItem(  
 tabName = "student",  
 h2("Student Dashboard")  
   
)  
#######################################  
#######################################

# 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 #######  
#######################################  
library(dplyr)  
  
combined\_allyears = full\_join(x = quiz, y = select(experience, !"YEAR"), by = "STUDENT\_KEY", copy = F, keep = F)  
  
#instructor dashboard  
#2018  
combined\_2018 = combined\_allyears %>% filter(combined\_allyears$YEAR == 2018 & !is.na(combined\_allyears$QUIZ\_NUMBER))  
  
#quiz grades frequency  
#plot  
inst\_quiz\_score\_hist = ggplot(combined\_2018, aes(x = QUIZ\_SCORE, fill = PROG)) +  
 geom\_histogram(binwidth=0.5) +  
 geom\_vline(aes(xintercept=mean(QUIZ\_SCORE)),  
 color="black", linetype="dashed", size=1.5) +  
 scale\_x\_continuous(breaks = 1:20)  
  
#avg iclick grade per lecture  
#data  
inst\_iclick\_data = combined\_2018 %>% select(CORRECT\_Q\_CLICKER, PROG, SESSION\_NUMBER) %>%  
 group\_by(SESSION\_NUMBER) %>%  
 summarise(  
 avg\_iclk\_grade = mean(CORRECT\_Q\_CLICKER)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

#plot  
inst\_iclick\_plot = ggplot(inst\_iclick\_data, aes(x = SESSION\_NUMBER, y = avg\_iclk\_grade)) +  
 geom\_line(color = "blue") +  
 geom\_point() +  
 scale\_x\_continuous(breaks = 1:23)  
  
#attendance per lecture  
#data  
inst\_attendance\_data = combined\_2018 %>% select(SESSION\_NUMBER, ATTENDED) %>%  
 group\_by(SESSION\_NUMBER) %>%  
 summarise(  
 sum\_attended = sum(ATTENDED)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

#plot  
inst\_attendance\_plot = ggplot(inst\_attendance\_data, aes(x = SESSION\_NUMBER, y = sum\_attended)) +  
 geom\_line(color = "blue") +  
 geom\_point() +  
 scale\_x\_continuous(breaks = 1:23)  
  
#class experience  
#data  
inst\_classexp\_data = combined\_2018 %>% select(14:22) %>%  
 filter(!is.na(DATABASE\_SCORE)) %>%  
 group\_by(PROG) %>%  
 summarise(  
 database = mean(DATABASE\_SCORE),  
 sql = mean(SQL\_SCORE),  
 programming = mean(PROGRAMING\_SCORE),  
 stored = mean(STORED\_PROC\_SCORE),  
 etl = mean(ETL\_SCORE),  
 data\_vis = mean(DATA\_VIS\_SCORE),  
 requirement = mean(REQUIREMENT\_GATHER\_SCORE),  
 skill\_survey = mean(SKILL\_SURVEY\_SCORE)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

inst\_classexp\_data\_2 <- pivot\_longer(inst\_classexp\_data, cols=c('database', 'sql', 'programming', 'stored',  
 'etl', 'data\_vis', 'requirement', 'skill\_survey'),  
 names\_to='variable', values\_to="value")  
  
  
#plot  
inst\_classexp\_bar = ggplot(inst\_classexp\_data\_2, aes(x = variable, y = value, fill = PROG)) +  
 geom\_bar(stat = "identity", position = 'dodge')  
  
  
#2017  
combined\_2017 = filter(quiz, (quiz$YEAR == 2017 & !is.na(quiz$QUIZ\_NUMBER)))  
#2016  
combined\_2016 = filter(quiz, (quiz$YEAR == 2016 & !is.na(quiz$QUIZ\_NUMBER)))  
  
  
  
  
#student dashboard  
single\_student = combined\_2018 %>% filter(STUDENT\_KEY == 103)  
  
  
# student's score per quiz  
ss\_quiz\_scores <- single\_student %>%  
 group\_by(QUIZ\_NUMBER) %>%  
 summarise(  
 score = QUIZ\_SCORE  
 ) %>%  
 arrange(QUIZ\_NUMBER)

## `summarise()` regrouping output by 'QUIZ\_NUMBER' (override with `.groups` argument)

ss\_quiz\_scores <- distinct(ss\_quiz\_scores)  
  
ss\_quiz\_scores\_bar = ggplot(ss\_quiz\_scores, aes(x = QUIZ\_NUMBER, y = score, color='blue')) +  
 geom\_bar(stat = "identity", position = 'dodge', color='black')  
  
#temperature clicker scores  
single\_student$Lecture = single\_student$SESSION\_NUMBER  
single\_student$Temperature = ifelse(single\_student$COMPLETED\_T\_CLICKER == 0, 'N/A', single\_student$AVG\_T\_CLICKER)  
  
ss\_temp\_scores <- single\_student %>%  
 summarise(  
 Lecture = single\_student$SESSION\_NUMBER,  
 Temperature = ifelse(single\_student$COMPLETED\_T\_CLICKER == 0, 'N/A', single\_student$AVG\_T\_CLICKER)  
 ) %>%  
 arrange(Lecture)  
  
  
#they have attended ... lectures so far  
#lecture\_attendance <- combined\_2018 %>%  
# group\_by(SESSION\_NUMBER) %>%  
# summarise(  
 # total\_attendees = sum(ATTENDED),  
 # single\_student\_attended = ifelse( nrow(subset(single\_student, ATTENDED== 1 && single\_student$Lecture==lecture\_attendance$SESSION\_NUMBER)) == 1, 1, 0)  
# )  
  
lectures\_attended <- single\_student %>%  
 group\_by(STUDENT\_KEY) %>%  
 summarise(  
 n = sum(ATTENDED)  
 )

## `summarise()` ungrouping output (override with `.groups` argument)

#weakest unit (lowest grade paired with lowest temperature scores)  
single\_student$t\_by\_q\_score = ifelse(single\_student$COMPLETED\_T\_CLICKER == 0,   
 single\_student$QUIZ\_SCORE,   
 single\_student$QUIZ\_SCORE \* single\_student$AVG\_T\_CLICKER)  
  
   
#how many people attended each lecture  
# HOW ARE WE DEALING WITH MISSING LECTURES??  
attendance\_per\_lecture <- combined\_allyears %>%  
 group\_by(SESSION\_NUMBER, YEAR) %>%  
 summarise(  
 n\_attended = sum(ATTENDED),  
 # student1\_attended = nrow(subset(all\_data, STUDENT\_KEY==1 && SESSION\_NUMBER==))  
 )

## `summarise()` regrouping output by 'SESSION\_NUMBER' (override with `.groups` argument)

#######################################  
#######################################

# 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 #######  
#######################################  
   
 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({  
 inst\_quiz\_score\_hist  
 })  
   
 output$inst.plot2 = renderPlot({  
 inst\_iclick\_plot  
 })  
 output$inst.plot3 = renderPlot({  
 inst\_attendance\_plot  
 })  
 output$inst.plot4 = renderPlot({  
 inst\_classexp\_bar  
 })  
   
   
   
 output$student.plot1 = renderPlot({  
 ss\_quiz\_scores\_bar  
 })  
 output$student.plot2 = renderTable({  
 ss\_temp\_scores  
 })  
#######################################  
#######################################  
   
}

# Part 5: Produce Dashboard and Reflect

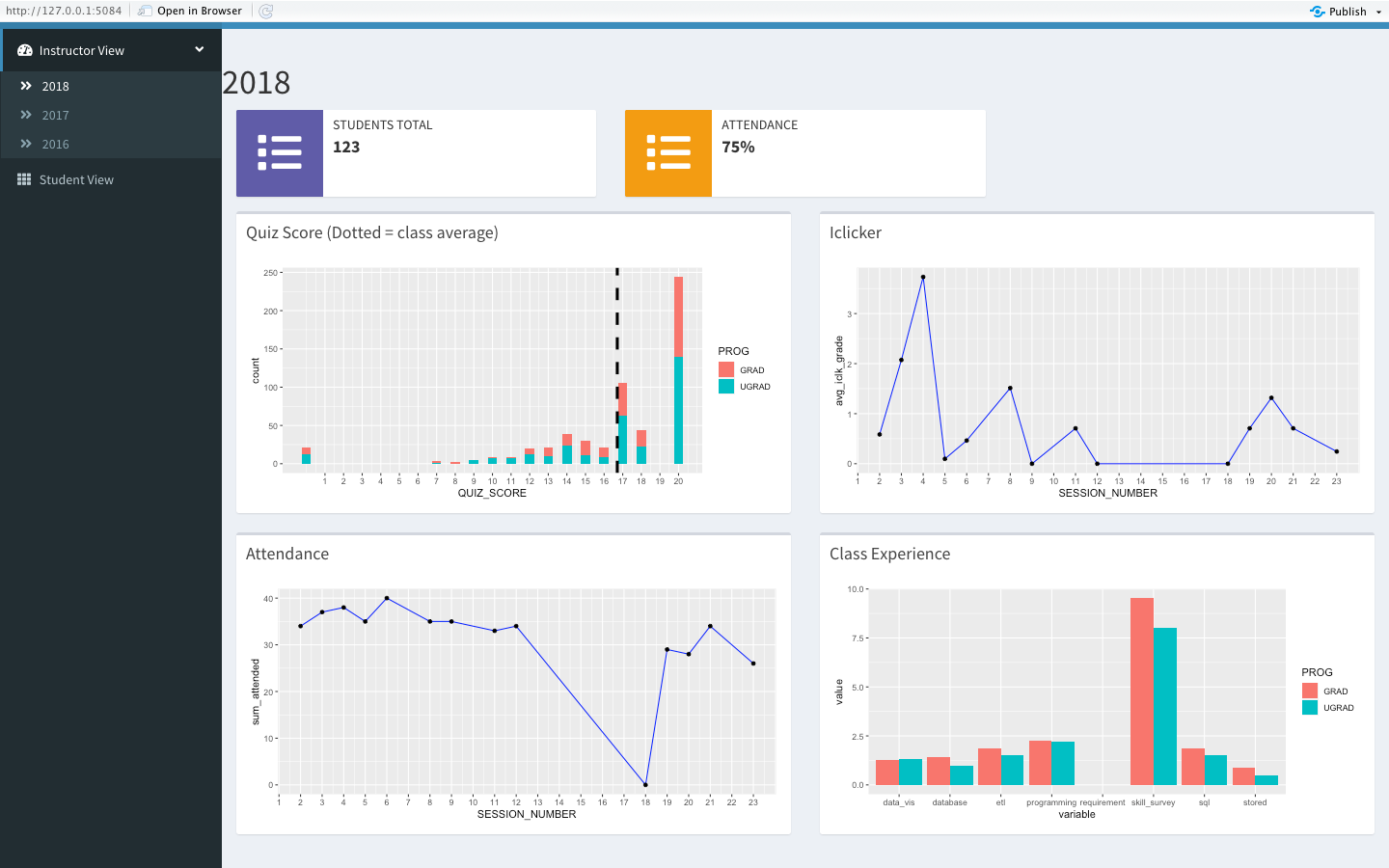
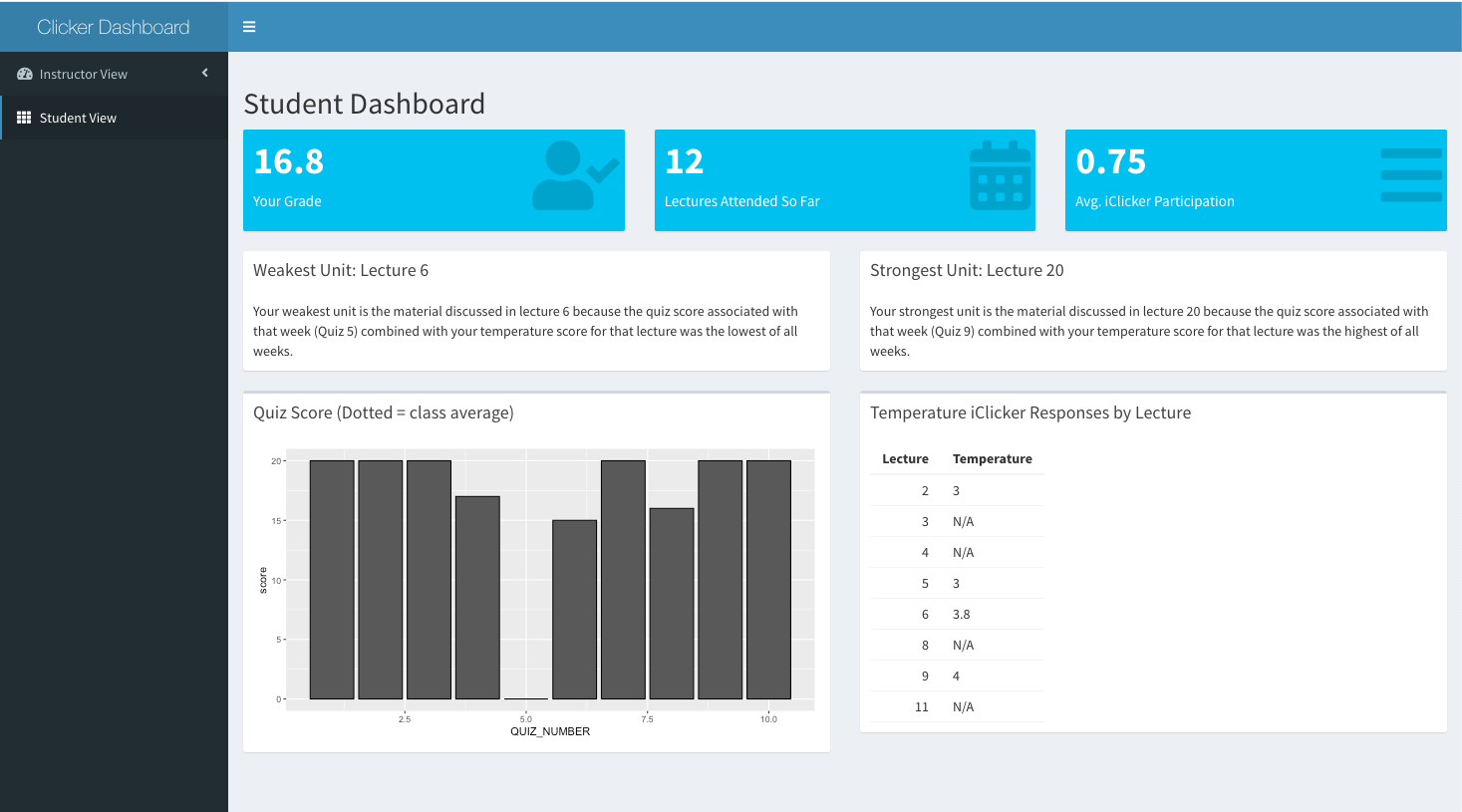
You should be able to simply run the code below **as is** to see your dashboard.

**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.

student\_dash = tabItem(  
 tabName = "student",  
 h2("Student Dashboard"),  
 fluidRow(  
 #column(width = 12,  
 valueBox(  
 mean(ss\_quiz\_scores$score),  
 'Your Grade',  
 icon= icon('user-check')),  
 valueBox(  
 sum(single\_student$ATTENDED),   
 "Lectures Attended So Far",   
 icon= icon('calendar')),  
 valueBox(  
 mean(single\_student$TOTAL\_COMPLETED\_CLICKER/single\_student$TOTAL\_POSSIBLE\_CLICKER),   
 "Avg. iClicker Participation",   
 icon = icon('bars'))  
 # )  
 ),  
 fluidRow(  
 box(title = "Weakest Unit: Lecture 6", width = 6, solidHeader = TRUE,  
 "Your weakest unit is the material discussed in lecture 6 because the quiz   
 score associated with that week (Quiz 5) combined with your temperature score   
 for that lecture was the lowest of all weeks."),  
 box(title = "Strongest Unit: Lecture 20", width = 6, solidHeader = TRUE,  
 "Your strongest unit is the material discussed in lecture 20 because the quiz   
 score associated with that week (Quiz 9) combined with your temperature score  
 for that lecture was the highest of all weeks.")  
 ),  
 fluidRow(  
 #column( width = 12,  
 box(  
 title = "Quiz Score (Dotted = class average)",  
 plotOutput("student.plot1", height = 300)  
 ),  
 box(  
 title = 'Temperature iClicker Responses by Lecture',  
 tableOutput("student.plot2"),  
 style = "height:300px; overflow-y: scroll;overflow-x: scroll;")  
 )  
)

#######################################  
### 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"),  
 menuSubItem("2018", tabName = "2018"),  
 menuSubItem("2017", tabName = "2017"),  
 menuSubItem("2016", tabName = "2016")  
 ),  
 menuItem("Student View", tabName = "student", icon = icon("th"))  
 )  
)  
  
# Here we set up the body of the dashboard  
dbody = dashboardBody(  
 tabItems(  
 student\_dash,  
 tabItem(tabName = "instructor",  
 h2("Instructor Dashboard")  
 ),  
 tabItem(tabName = "2018",  
 # Dynamic infoBoxes  
 fluidRow(  
 h1("2018"),  
 infoBoxOutput("inst.info1"),  
 infoBoxOutput("inst.info2")  
 ),  
 # Any visualization  
 fluidRow(  
 box(  
 title = "Quiz Score (Dotted = class average)",  
 plotOutput("inst.plot1", height = 250)  
 ),  
 box(  
 title = "Iclicker",  
 plotOutput("inst.plot2", height = 250)  
 ),  
 box(  
 title = "Attendance",  
 plotOutput("inst.plot3", height = 250)  
 ),  
 box(  
 title = "Class Experience",  
 plotOutput("inst.plot4", height = 250)  
 )  
 )  
 ),  
 tabItem(tabName = "2017",  
 fluidRow(  
 h1("2017")  
 )),  
 tabItem(tabName = "2016",  
 fluidRow(  
 h1("2016")  
 ))  
))  
  
  
# 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:

Dashboard Screenshot  

**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?

*Reflection for the student dashboard*

* What do you like about it?
  + There are a lot of details that we could show the student in a simple way using value boxes.
* What would you change or add to it if you had more time?
  + We would add more styling elements (changing colors and positioning of elements to make the dashboard look more energizing and fun)
* What was the biggest challenge you faced? How did you address it?
  + The biggest challenge we faced was determining how to show data for lectures that were not represented in the dataset. We chose to skip over those lectures and only show data for what we have. If we had not done this, the dashboard chart and table would look very empty.

*Reflection for the teacher dashboard*

* What do you like about it?
  + The design is very simple and the charts are complex but still easy to understand.
* What would you change or add to it if you had more time?
  + I would want to make it interactive so I could examine each student individually.
* What was the biggest challenge you faced? How did you address it?
  + It was very difficult trying to think of what things could be made interactive to show data from multiple years (or just one year) by selection, so instead we made the years into different tabs in the sidebar.

# 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.