Dashboards for Clicker Data

INFO 4100 Learning Analytics

Haowen Bao, Ada Chai, Shuhui Zhu

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?
  + Student Dashboard
  + Students will use it for understanding their own performance in this course. They share similar backgrounds, in that they are all Cornell students who are interested in learning computer science.
* Why? What is the goal? What questions to answer?
  + The goal is to help students understand their performance on quizzes and clicker questions. It can give them an overview of their own status in the course. Questions the dashboard will answer include:
    - How am I doing on average in this course?
    - What does my attendance rate look like over time?
    - How well do I perform on clicker questions per session?
    - How does my quiz performance compare to the class?
* What? What data to show and what is its structure?
  + Temperature scores
  + Quizzes and clicker questions performance
  + It will take the form of summary statistics in information boxes at the top, as well as graphs.
* How? How will visualizations support the goal?
  + It will allow students to visualize their performance compared to other students in the course, through average quiz and clicker scores.

*Planning for the teacher dashboard*

* For whom? Who will use it and what is their background?
  + Instructor Dashboard
  + The instructor will use this dashboard. Their background will be in computer science and their specific course knowledge. They will use this dashboard to understand how the class is doing in their course.
* Why? What is the goal? What questions to answer?
  + Goals include getting an overall sense of students’ performance in the course. They will understand students’ course performance in both quizzes and clicker questions. Questions the graphs will answer include:
  + How does performance differ between undergrad and grad students?
  + How are students feeling in the course? How does this affect attendance?
  + How does quiz score affect student temperature?
  + How do students’ self-estimated skill sets compare to actual quiz scores?
* What? What data to show and what is its structure?
  + student overall quiz and clicker score performance
  + student attendance
  + student temperature
  + It will take the form of summary statistics in information boxes at the top, as well as graphs.
* How? How will visualizations support the goal?
  + It will allow the instructor to understand the following:
  + Class attendance
  + Quiz average scores, low and high
  + student temperature scores in relation to performance
  + clicker score vs. quiz score

**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.

[See below for team sketch]

A picture containing diagram

Description automatically generated

# 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)  
student\_dash = tabItem(  
 tabName = "student",  
 h2("Student Dashboard"),  
   
 # Dynamic infoBoxes  
 fluidRow(  
 infoBoxOutput("inst.info1", width = 6),  
 infoBoxOutput("inst.info2", width = 6),  
 infoBoxOutput("inst.info3", width = 6),  
 infoBoxOutput("inst.info4", width = 6)  
 ),  
 # Any visualization  
 fluidRow(  
 box(  
 title = "Average Class Quiz Performance vs. Personal Average Quiz Score",  
 plotOutput("inst.plot1", height = 250)  
 ),  
 box(  
 title = "Clicker Temperature Score vs. Quiz Score",  
 plotOutput("inst.plot2", height = 250)  
 ),  
 box(  
 title = "Attendance Over Time",  
 plotOutput("inst.plot3", height = 250)  
 ),  
 box(  
 title = "Clicker Completion Accuracy by Session",  
 plotOutput("inst.plot4", height = 250)  
 )  
 )  
)  
  
# Another empty tab  
instructor\_dash = tabItem(  
 tabName = "instructor",  
 h2("Instructor Dashboard"),  
   
 # Dynamic infoBoxes  
 fluidRow(  
 infoBoxOutput("inst.info5", width = 6),  
 infoBoxOutput("inst.info6", width = 6),  
 infoBoxOutput("inst.info7", width = 6),  
 infoBoxOutput("inst.info8", width = 6)  
 ),  
   
 fluidRow(  
 box(  
 title = 'Average Quiz Score by Program',  
 plotOutput('inst.plot5', height = 250)  
 ),  
 box(  
 title = 'Attendance vs Temperature',  
 plotOutput('inst.plot6', height = 250)  
 )  
 ),  
   
   
 fluidRow(  
 box(  
 title = "Student Self-estimated Skillset Score vs. Student Quiz Score",  
 plotOutput("inst.plot7", height = 250)  
 ),  
 box(  
 title = "Temperature vs Quiz Score",  
 plotOutput("inst.plot8", 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 #######  
#######################################  
  
# cleaning up student experience, omitting NA  
StudentExperience <- experience %>%  
 group\_by(STUDENT\_KEY) %>%  
 na.omit()  
  
#quiz mean  
QuizMean <- quiz %>%  
 group\_by(QUIZ\_NUMBER) %>%  
 na.omit() %>%  
 summarise(QuizMean = mean(QUIZ\_SCORE))  
  
# cleaning up student & quiz information, 1 student 1 quiz, replacing NA with 0, combining mean quiz score with the cleaned up data set  
StudentQuizScore <- quiz %>%  
 group\_by(STUDENT\_KEY, QUIZ\_NUMBER) %>%  
 mutate(QUIZ\_SCORE = replace\_na(QUIZ\_SCORE, 0)) %>%  
 summarise(QuizScore = QUIZ\_SCORE[1],  
 ClickerT = AVG\_T\_CLICKER) %>%  
 na.omit() %>% left\_join(QuizMean) %>%  
 left\_join(StudentExperience)

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

## Joining, by = "QUIZ\_NUMBER"

## Joining, by = "STUDENT\_KEY"

# cleaning up student experience, omitting NA  
StudentExperience <- experience %>%  
 group\_by(STUDENT\_KEY) %>%  
 na.omit()  
  
# summary of student average temperature clicker each session  
SessionAvgT <- quiz %>%  
 filter(COMPLETED\_Q\_CLICKER > 0) %>%  
 group\_by(SESSION\_NUMBER) %>%  
 summarise(AveT = mean(AVG\_T\_CLICKER))  
  
# summary of student accuracy clicker score each session, out of clicker questions they answered  
SessionClickerAccuracy <- quiz %>%  
 filter(TOTAL\_COMPLETED\_CLICKER > 0 ) %>%  
 group\_by(SESSION\_NUMBER) %>%  
 summarise(ClickerAccuracy = CORRECT\_Q\_CLICKER / TOTAL\_COMPLETED\_CLICKER )

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

# join students' progress  
quiz\_prog = left\_join(quiz, experience[c('STUDENT\_KEY','PROG')], by = c('STUDENT\_KEY' = 'STUDENT\_KEY'))  
quiz\_prog$QUIZ\_SCORE\_NA = replace\_na(quiz\_prog$QUIZ\_SCORE, 0)  
  
# summary of student average temperature clicker each session  
SessionAvgT <- quiz %>%  
 filter(COMPLETED\_Q\_CLICKER > 0) %>%  
 group\_by(SESSION\_NUMBER) %>%  
 summarise(AveT = mean(AVG\_T\_CLICKER))  
  
# join students' program  
quiz\_prog = left\_join(quiz, experience[c('STUDENT\_KEY','PROG')], by = c('STUDENT\_KEY' = 'STUDENT\_KEY'))  
quiz\_prog$QUIZ\_SCORE\_NA = replace\_na(quiz\_prog$QUIZ\_SCORE, 0)  
  
# attendance and temperature  
SessionAvgATT <- quiz %>%  
 group\_by(SESSION\_NUMBER) %>%  
 summarise(AveT = mean(ATTENDED))  
SessionAvgATT$ATT\_T <- 'Attendance'  
SessionAvgT$ATT\_T <- 'Temperature'  
SessionAvgATT\_T<-rbind(SessionAvgT, SessionAvgATT)  
  
# student 1 attendance over time  
student1\_quiz = quiz %>% filter(STUDENT\_KEY == 1)  
student1\_quiz = student1\_quiz[order(student1\_quiz$SESSION\_NUMBER),]  
student1\_cumattendance = student1\_quiz %>% summarize(n\_classes = 1:nrow(student1\_quiz), n\_attended = cumsum(ATTENDED), attendance\_rate = n\_attended/n\_classes)  
  
# student 1 average quiz score  
student1\_avrgQuiz = student1\_quiz %>% na.omit() %>% summarize(session\_quiz\_score = QUIZ\_SCORE / 20 )  
  
# student 1 average clicker score   
student1\_avrgClicker = student1\_quiz %>% na.omit() %>% group\_by(SESSION\_NUMBER) %>% summarize(session\_n = SESSION\_NUMBER, clicker\_score = CORRECT\_Q\_CLICKER / COMPLETED\_Q\_CLICKER) %>% na.omit()  
# student1\_avrgClicker[is.na(student1\_avrgClicker$clicker\_score), "clicker\_score"] = 0  
  
  
#######################################  
#######################################

# 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 #######  
#######################################  
   
 # student dashboard info boxes   
   
 output$inst.info1 = renderInfoBox({  
 infoBox("Student ID",  
 1,   
 icon = icon("address-card"), color = "purple")  
 })  
   
 output$inst.info2 = renderInfoBox({  
 infoBox("Average Attendance",  
 paste0(round(100 \* mean(student1\_cumattendance$attendance\_rate)), "%"),  
 icon = icon("clipboard"), color = "yellow")  
 })  
   
 output$inst.info3 = renderInfoBox({  
 infoBox("Average Quiz Score",  
 paste0(round(100 \* mean(student1\_avrgQuiz$session\_quiz\_score)), "%"),  
 icon = icon("book"), color = "blue")  
 })  
   
 output$inst.info4 = renderInfoBox({  
 infoBox("Average Clicker Score",  
 paste0(round(100 \* mean(student1\_avrgClicker$clicker\_score)), "%"),  
 icon = icon("lightbulb"), color = "red")  
 })  
   
 # instructor dashboard info boxes  
   
 output$inst.info5 = renderInfoBox({  
 infoBox("Class Size",  
 length(unique(quiz$STUDENT\_KEY)),  
 icon = icon("address-card"), color = "purple")  
 })  
  
 output$inst.info6 = renderInfoBox({  
 infoBox("Attendance",  
 paste0(round(100 \* mean(quiz$ATTENDED)), "%"),  
 icon = icon("clipboard"), color = "yellow")  
 })  
  
 output$inst.info7 = renderInfoBox({  
 infoBox("Average Quiz Score",  
 paste0(round(mean(QuizMean$QuizMean/20)\*100, 0) , "%"),  
 icon = icon("book"), color = "blue")  
 })  
  
 output$inst.info8 = renderInfoBox({  
 infoBox("Average Clicker Score",  
 paste0(round(100 \* mean(SessionClickerAccuracy$ClickerAccuracy)), "%"),  
 icon = icon("lightbulb"), color = "red")  
 })  
  
   
 # plot1  
 output$inst.plot1 = renderPlot({  
 ggplot(data = StudentQuizScore[StudentQuizScore$STUDENT\_KEY == 1,], aes(x = as.factor(QUIZ\_NUMBER), group = 1)) +  
 geom\_col(aes(y = QuizScore), color = "black") + geom\_line(aes(y = QuizMean), color = "blue") +   
 labs(x = "Quiz Number", y = "Quiz Score") +   
 theme\_classic()   
 })  
   
 # plot2  
 output$inst.plot2 = renderPlot({  
 ggplot(data = StudentQuizScore[StudentQuizScore$STUDENT\_KEY == 1,], aes(x = as.factor(QUIZ\_NUMBER), group = 1)) +  
 geom\_line(aes(y = QuizScore), color = "black") + geom\_line(aes(y = ClickerT), color = "blue") +   
 labs(x = "Quiz Number", y = "Quiz Score/Clicker Temperature Score") +   
 theme\_classic()   
 })  
  
 #plot 3   
 output$inst.plot3 = renderPlot({  
 ggplot(data = student1\_cumattendance, aes(x = as.factor(student1\_cumattendance$n\_classes), group = 1)) +  
 geom\_line(aes(y = student1\_cumattendance$attendance\_rate), color = "black") +  
 labs(x = "Number of Sessions", y = "Cumulative Attendance Rate") + ylim(0,1) +   
 theme\_classic()  
 })  
  
# #plot 4   
 output$inst.plot4 = renderPlot({  
 ggplot(data = student1\_avrgClicker, aes(x = as.factor(student1\_avrgClicker$session\_n), group = 1)) +  
 geom\_line(aes(y = student1\_avrgClicker$clicker\_score), color = "black") +  
 labs(x = "Session Number", y = "Clicker Percentage") +  
 theme\_classic()   
})  
  
   
 # plot5  
 output$inst.plot5 = renderPlot({ggplot(quiz\_prog, aes(x=as.factor(QUIZ\_NUMBER), y=QUIZ\_SCORE\_NA, fill=PROG)) +   
 geom\_boxplot()})  
   
 # plot6  
 output$inst.plot6 = renderPlot({  
 ggplot(SessionAvgATT\_T, aes(x = factor(SESSION\_NUMBER), y = AveT, colour = ATT\_T, group = ATT\_T)) +  
 geom\_line()  
 })  
 # plot7  
 output$inst.plot7 = renderPlot({  
 ggplot(data = na.omit(StudentQuizScore), aes(x = SKILL\_SURVEY\_SCORE, y = QuizScore)) +  
 geom\_point() +   
 geom\_smooth(method = "lm") +  
 labs(x = "Skill Survey Score", y = "Quiz Score") +   
 theme\_classic()   
 })  
 # plot8  
 output$inst.plot8 = renderPlot({  
 ggplot(data = na.omit(StudentQuizScore), aes(x = ClickerT, y = QuizScore)) +  
 geom\_point() +   
 geom\_smooth(method = "lm") +  
 labs(x = "Temperature Score", 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("Student View", tabName = "student", icon = icon("th")),  
 menuItem("Instructor View", tabName = "instructor", icon = icon("dashboard"))  
 )  
)  
  
# Here we set up the body of the dashboard  
dbody = dashboardBody(  
 tabItems(  
 instructor\_dash,  
 student\_dash  
 )  
)  
  
# 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:

Student Dashboard Screenshots:

Graphical user interface, application

Description automatically generated

Graphical user interface, chart, application

Description automatically generated

Instructor Dashboard Screenshots:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**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?
  + We like the summary metrics at the top, the “average class quiz performance vs personal average quiz score” graph, and the clicker accuracy graph.
* What would you change or add to it if you had more time?
  + We would add a legend to the “average class quiz performance vs personal average quiz score” graph so that it’s more clear that the line represents class quiz averages and the bars represent personal averages. We might also add convert the clicker temperature score to be out of 20, as it might be a bit misleading at the moment compared to the quiz score. It might also be nice to vary the types of graphs in the student dashboard, because quite a few feature line graphs whereas it might be more interesting to view other types as well.
* What was the biggest challenge you faced? How did you address it?
  + One of the biggest challenges was tackling the attendance over time graph; it took some time to figure out how to cumulatively sum together attendance. However, eventually we figured it out by calculating the number of classes, and averaging the attendance rate for each number of classes attended.
  + Also, filtering out NA values and when to filter out 0 values was challenging for some calculations. For some graphs, we decided to take out values that would have led to “0/0” calculations so computations could be computed.

*Reflection for the teacher dashboard*

* What do you like about it?
  + We like the use of color and variety of graphs represented, as well as the summary statistics.
* What would you change or add to it if you had more time?
  + We might clarify the x and y labels for some of the graphs so it’s represented less like R code and more audience-friendly. We might add legends to some of these graphs as well so it’s easier to differentiate between what is being represented.
* What was the biggest challenge you faced? How did you address it?
  + It was also challenging to filter out data and recombine it for calculations here. Ultimately for most graphs we were able to filter out NA data; however, for some it was difficult to filter out so it may still be represented in the graph, e.g. “Average Quiz Score by Program” graph.

# 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.