Project 2

St. Clair / Frank Yang / Il Shan Ng

Due: by 3pm Friday, Oct. 21

Moodle User Data

Our course management system, Moodle, records data on every "click" that is made on the site. Carly Born, Academic Technologist in ITS, is able to pull this data from the moodle database. A typical data set for one term will have over a million "clicks" (rows) in it. The variables recorded for each click during fall term 2012 are:

variable	description	sample data
`log_id`	id number assigned by Moodle for the log entry	`47753`, `61292`
`category`	equal to term	`3,3` (for all fall 2012 entries)
`courseid`	id number assigned by Moodle for each course	`7133`, `7734` (unique id for each course)
`userid`	id number assigned by Moodle for each user	`28578`, `2390` (unique id for each Moodle user)
`roleid`	id number assigned by Moodle for each role in the system	`teacher`, `student`
`action`	action performed, usually an action verb that describes best what the action was	`view`, `submit`, `add post`, `upload`, ...
`module`	Moodle module upon or within which the action was performed	`course`, `forum`, `quiz`, ...
`url`	URL of the specific module in question, containing the instance id number of that specific module	`view.php?id=7133`, `view.php?id=363`
`time`	unix timecode date/time stamp for the action	`1344235886`, `1347126996` (can be converted in readable date/time). Most clicks occur during the term, but some activity is seen before and after the term.

The data and code book (for module, roleid) is in the Common Material in the **Courses** folder. The `.txt` file for the data is tab delimited (not comma or space) so can read it in to R with `read_delim` command from the `readr` package:

```
> library(readr)
Warning: package 'readr' was built under R version 3.4.3
> moodle <- read_delim("moodle12-13logresults.txt", delim="\t")
```

You can change the unix time (seconds since 1/1/1970) to a date and time using `as.POSIXct`:

```
> library(dplyr)
Warning: package 'dplyr' was built under R version 3.4.3
> moodle <- moodle %>% mutate(timestamp = as.POSIXct(time, origin = "1970-01-01"))
Warning: package 'bindrcpp' was built under R version 3.4.3
```

To do:

You can consider this project an implementation of the day 4 visualization activity. I would like you to create **three visualization** that illustrate something interesting about this data. Along with the graph, provide a brief written description of what the graph is displaying along with the process you undertook to create the graph. If needed, you can use Shiny graphics too.

As a reminder, here are a few questions one *might* have about this data:

- How do teachers use moodle differently than students?
- How does student (or teacher) usage evolve over the course of a term?
- Do students use moodle differently for courses that have more content vs. courses that have less content?

You *do not* need to create graphs to answer these questions - they are just suggestions!

With whatever graphs you decide you create, I am fairly certain that you will need to transform or restrucure the data in some way. For example, you may want to first find total actions taken in a day (or week) to look at trends over time. Or you may want to compute the average number of `quiz` modules targeted per student over the course of the term (or week). Or the average number of times (per term, per week) a teacher does an `upload` action.

Note: Make sure that your graph nicely formatted and contains useful labels and title(s) to provide context.

Visualization 1:

First, we extract some basic time information from the `timestamp` variable using `lubridate()`.

```
> library(lubridate)
Warning: package 'lubridate' was built under R version 3.4.3
>
> moodle2 <- moodle %>%
+   mutate(year = year(timestamp),                    # extract year
+          month = month(timestamp, label = TRUE),    # extract month
+          day = weekdays(timestamp),                 # extract day of week
+          hour = hour(timestamp))                    # extract hour of day
```

For the first plot, we intend to look at how hourly trends may differ between days of the week. In other words, we want to graph the total number of clicks made in each hour, for each of the seven days of the week. Because we will focus only on the clicks made during the Fall 2012 term, we will first have to filter out all other clicks made outside of this term. From the 2012 Academic Calender, we determined that the Fall 2012 term started on September 10, 2012, and ended on November 14, 2012.

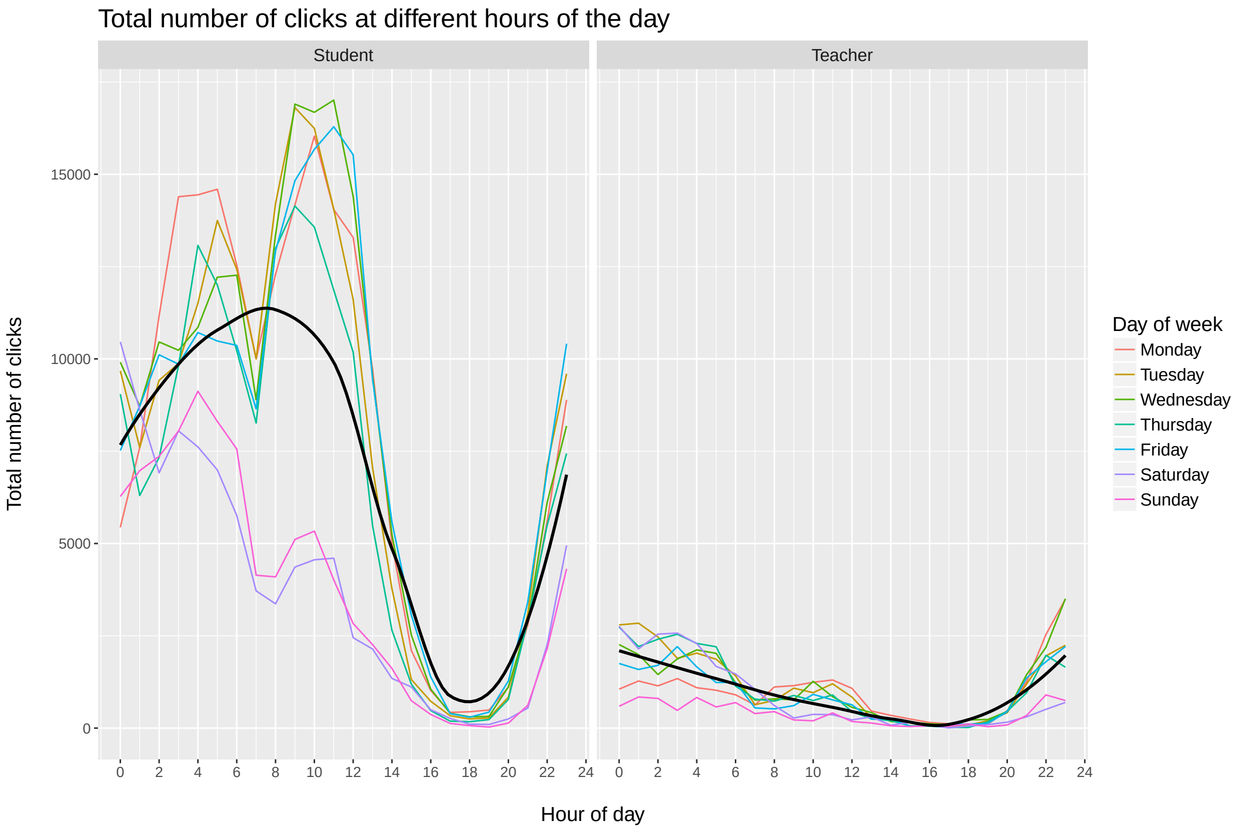
```
> moodle3 <- moodle2 %>%
+   filter(timestamp >= ymd("2012-09-10"), timestamp <= ymd("2012-11-19"))
```

Next, we need to aggregate the total number of clicks made within each hour of the day, for each day of the week. We also intend to separate the clicks by whether they were made by students or teachers. According to the codebook, the level `3` under the `roleid` variables represents teacher, and the level `5` represents student. To avoid cluttering our graph, we remove all clicks made by other agents.

```
> library(ggplot2)
>
> moodle3 <- moodle3 %>%
+   filter(roleid == 3 | roleid == 5) %>%     # filter only clicks by students and teachers
+   mutate(
+     day = factor(day,                        # set factor levels for day and roleid variables
+                  levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")),
+     roleid = as.factor(ifelse(roleid == 3, "Teacher", "Student"))
+     ) %>%
+   group_by(roleid, day, hour) %>%
+   summarise(total_clicks = n())            # calculate total number of clicks for each category
```

The following chunk of code plots the intended graph. The black line represents the mean trend across all seven days of the week.

```
> ggplot(moodle3, aes(x = hour, y = total_clicks)) +
+     geom_line(aes(color = day)) +
+     stat_smooth(method = "loess", color = "black", se = FALSE) +   # add mean trend line
+     facet_wrap( ~ roleid) +
+     scale_x_continuous(breaks = seq(0, 24, 2)) +
+     labs(x = "\nHour of day", y = "Total number of clicks\n",
+         title = "Total number of clicks at different hours of the day") +
+     scale_color_discrete(name = "Day of week") +  # relabel legend title
+     theme(axis.title = element_text(size = 14),   # adjust label sizes
+           axis.text = element_text(size = 10),
+           plot.title = element_text(size = 18),
+           legend.title = element_text(size = 14),
+           legend.text = element_text(size = 12),
+           strip.text.x = element_text(size = 12))
```



The above graph yields four main conclusions:

- Moodle activity due to students is, on average, lowest in the early morning (4 to 5am) and highest at night (8 to 10pm). This is in contrast to the moodle activity due to teachers, which on average shows a symmetric distribution with peak activity sometime at noon and low activity at night and in the early morning.

- On Fridays and Saturdays, moodle activity due to students decreases past afternoon (2 to 3pm). This is probably due to the students' tendency to not do work in the latter half of the day on Fridays and Saturdays. Otherwise, the number of clicks shows a rather consistent distrbution across the other five days of the week.
- Moodle activity due to teachers is lowest during the weekend. This is unlike the moodle activity for students, which shows a large number of clicks even on Sunday.
- The obvious drops in students' moodle activity at noon and 6pm coincide with lunch and dinner. This suggests that many students do not use moodle while having lunch or dinner. The same cannot be said for breakfast, since the low activity early in the morning could be a consequence of students waking up late.

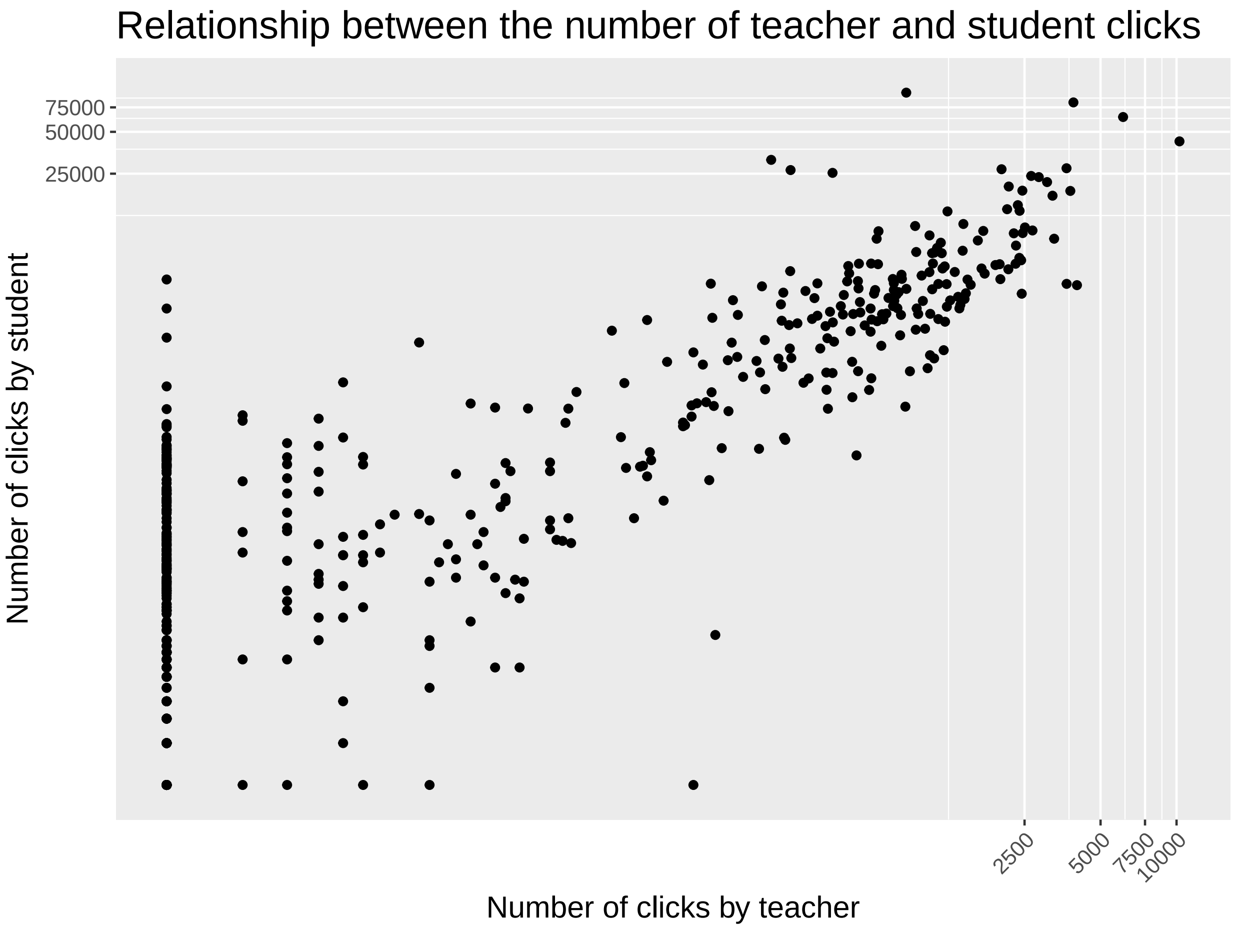
Visualization 2:

For the second visualization, we intend to investigate whether courses that have higher teacher activity also have higher student activity. The following chunk of code calculates the total number of teacher and student clicks for each course. It then converts the data frame into a wide format for plotting.

```
> library(tidyr)
Warning: package 'tidyr' was built under R version 3.4.3
>
> moodle4 <- moodle2 %>%
+   mutate(roleid = as.factor(ifelse(roleid == 3, "Teacher", "Student"))) %>%
+   group_by(courseid, roleid) %>%
+   summarise(total_clicks = n()) %>%    # calculate total number of student and teacher clicks for eac
+   spread(key = "roleid", value = "total_clicks", fill = 0)
```

The code below creates a scatter plot of the number of student clicks against the number of teacher clicks. Each glpyh represents a course from Fall 2012 that had moodle click data. Because the marginal distributions were highly right skewed, we applied a log_2 transformation on both axes to better visualize the majority of the data points.

```
> moodle4 %>%
+   mutate(Student = ifelse(Student == 0, 1, Student),        # replace 0 values with 1
+          Teacher = ifelse(Teacher == 0, 1, Teacher)) %>%    # to avoid errors when taking logs
+   ggplot(aes(x = Teacher, y = Student)) +
+     geom_point() +
+     coord_trans(x = "log2", y = "log2") +         # apply log transformations on both axes
+     labs(x = "Number of clicks by teacher", y = "Number of clicks by student",
+          title = "Relationship between the number of teacher and student clicks") +
+     theme(axis.title = element_text(size = 14),   # adjust label sizes
+           axis.text = element_text(size = 10),
+           axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
+           plot.title = element_text(size = 18))
```

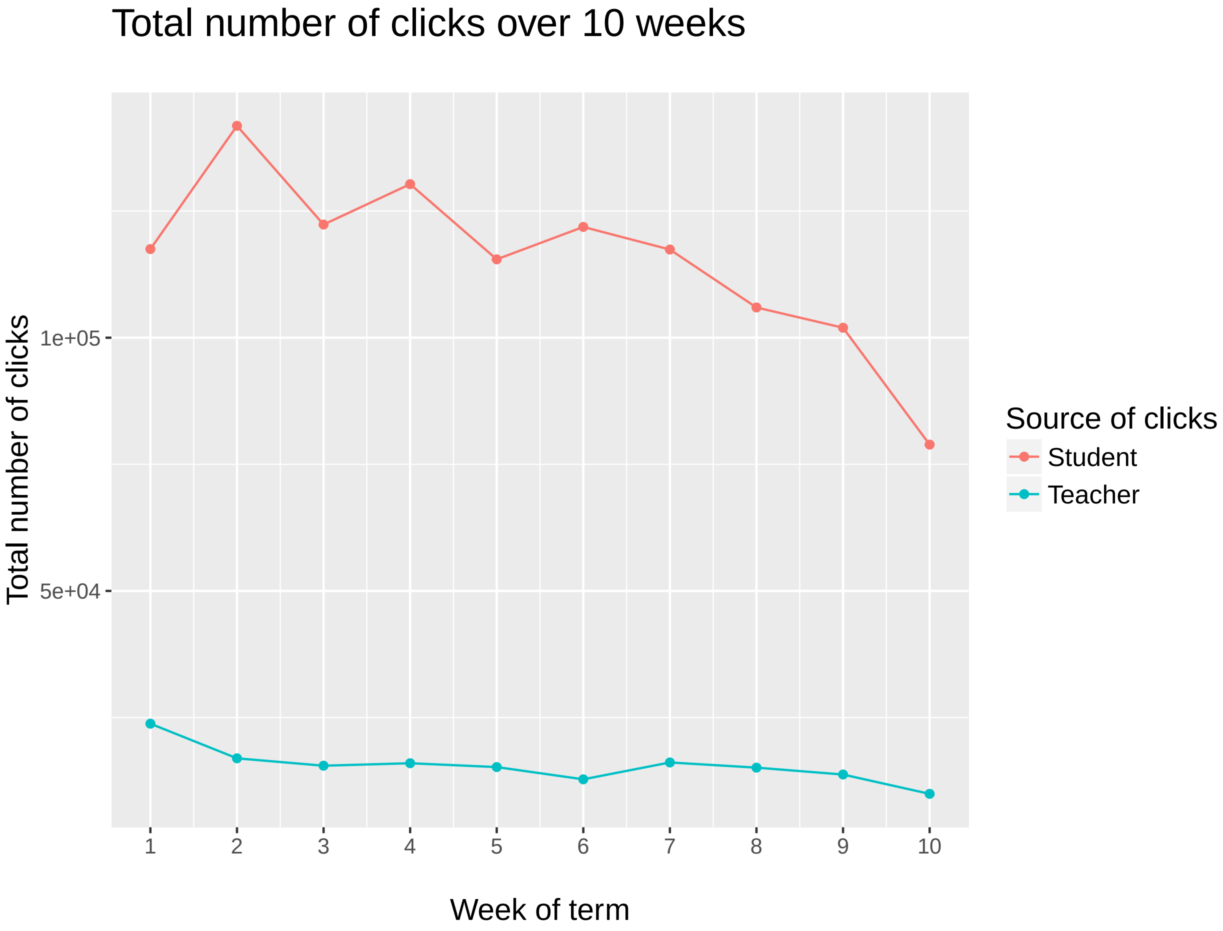


The graph above shows a positive linear relationship betwen the number of teacher clicks and the number of student clicks. This suggests that courses that require more moodle activity by the teacher to maintain also induces a higher number of moodle clicks from students. If the former can reliably serve as a proxy for the amount of course content, then this graph is some evidence that courses that have more content tend to garner higher moodle activity from students.

Visualization 3:

For the third visualization, we want to investigate how teacher and student's number of clicks change over time in a term. To do so, we decide to look at the number of clicks at week level rather than date level so that we can control the effect of weekends. The following code chunk first filters out the "term days" (that is bettwen "2012-09-10" and "2012-11-18"). We didn't use ""2012-11-19" because that is a Monday and will create week `11` consisting of only 1 day. Then the code creates a `week` variable to record which week the `timestemp` is in and aggregate the number of clicks on the week level. After that, we use a simple line graph to show the trend of the number of clicks and over time, grouped by the source of clicks.

```
> moodle2 %>%
+   filter(timestamp >= ymd("2012-09-10"), timestamp <= ymd("2012-11-18")) %>%
+   mutate(roleid = as.factor(ifelse(roleid == 3, "Teacher", "Student")),
+          week = interval(ymd("2012-09-10"),timestamp)%/%dweeks(1) + 1) %>%
+   group_by(roleid, week) %>%
+   summarize(clicks = n()) %>%
+   ggplot(aes(x = week, y = clicks, color = roleid, group = roleid)) + geom_point() + geom_line() +
+   scale_x_continuous(breaks = seq(1,10,1)) +
+   scale_color_discrete(name = "Source of clicks") +
+   labs(x = "\nWeek of term", y = "Total number of clicks",
+        title = "Total number of clicks over 10 weeks\n") +
+        theme(axis.title = element_text(size = 14),    # adjust label sizes
+              axis.text = element_text(size = 10),
+              plot.title = element_text(size = 18),
+              legend.title = element_text(size = 14),
+              legend.text = element_text(size = 12))
```



From the above graph we can see that students' total number of clicks reaches its peak at the second week while the teachers' total number of clicks is highest in the first week. This makes sense since teachers usually need to do a large amount of set-up at the beginning. Moodle page for a course usually contains lots of navigation information and therefore students might want to check it more often in the beginning of the week – so the total number of clicks often decreases over the term. We see two small peaks for the total number of clicks for students in week 4 and week 6 — this might be caused by the fact that midterms are usually during those terms and students more often check the moodle for review information.