MSDS650 Week 2 EDA Assignment - Nathan Worsham

For this assignment I was interested in using a data set that I had created at work which was login information for the website agents.pinnacol.com using Splunk to output the raw data as a table to csv file. The data collected is from June 22, 2015 through October 26, 2015 (127 days of data). It included (all date and time data represents when the session started or was attempted), time, the hour of the day (military time), the day of the week, the email address, http status code (200 successful or 401 failed), IP address of the client, and geo information (location, longitude, and latitude) for the ip address (if applicable, as some IPs only get a location of the US). I spent the majority of my time concentrating on correlations about trends surrounding login times and days along with successful versus failed attempts. I really wanted to do a second page using with the longitude and latitude data to do mapping but I had already invested too much time as it was on the previously mentioned.

I spoke with my boss about getting the ”okay” to use this data, the only stipulation was to anonymize the data, so using a function I found on r-bloggers I was able to hash the values like so:

```
setwd("/Users/worshamn/Dropbox/Documents/Regis/MSDS650/week2/")
agents_online= read.csv("agents0622-1026.csv")
#anonymize function - pulled from http://www.r-bloggers.com/anonymising-data/
anonymize <- function(data, cols_to_anon, algo = "sha256")
{
  if(!require(digest)) stop("digest package is required")
  to_anon <- subset(data, select = cols_to_anon)
  unname(apply(to_anon, 1, digest, algo = algo))
}
agents_online$email <- anonymize(agents_online, "email")
summary(agents_online)
```

To get the ”General Stats” section I used the following code, though a couple of stats (success and failure rate, averages per day) I just hand calculated using the values R provided:

```
#total events
length(agents_online$X_time)
#total successful and failed logins
table(agents_online$status)
#total count of unique login ids that have logged in successfully
length(unique(agents_online[agents_online$status==200,"email"]))
#high and low count of successful logins by user
tail(sort(table(agents_online_success$email)))
successCounts <- as.data.frame(table(agents_online_success$email))
table(successCounts$Freq)
#topuser's failed login count
topuser <- agents_online_fail[which(agents_online_fail$email==
  "eba108585da5161006a186ca04755736de71ea84b6382b8cb764d5b88b3d3bbb"),]
length(topuser$status)
```

For the ”Login Stats” section I simply used the summary command:

```
#summary hours and days successful logins
agents_online_success <- agents_online[agents_online$status==200,]
summary(agents_online_success$date_hour)
summary(agents_online_success$date_wday)
#summary hours failed logins
agents_online_fail <- agents_online[agents_online$status==401,]
summary(agents_online_fail$date_hour)
summary(agents_online_fail$date_wday)
```

The parts of this assignment that was the most difficult was getting R to graph exactly what I wanted which was to graph success and failed stacked on top of each other in a by hour view and a by day of the week view and control the colors it used for the values. The day of the week view proved very time consuming because by default R puts categorical data in alphabetical order which makes days of the week end up in strange order. Here is the code for the success versus failure by hour view, I first was working with `qplot` but have learned that while `qplot` (which is just a wrapper for `ggplot`) is easier to use, `ggplot` is much more customizable:

```
#hoursVsStatus
hoursVsStatus <- table(agents_online$date_hour,agents_online$status,
      dnn=c("Success")); hoursVsStatus
hoursVsStatusDF <- data.frame(hoursVsStatus)
names(hoursVsStatusDF) <- c("Hour", "Status", "Freq")
#hoursVsStatusPlot <- qplot(factor(hoursVsStatusDF$status),
#    x=hoursVsStatusDF$Hour, y=hoursVsStatusDF$Freq,
#    data=hoursVsStatusDF, geom="bar",
#    fill=factor(hoursVsStatusDF$Status),
#    stat="identity")
hoursVsStatusPlot <- ggplot(hoursVsStatusDF, aes(x=Hour, y=Freq,
      fill=factor(hoursVsStatusDF$Status))) + geom_bar(stat="identity")
hoursVsStatusPlot + scale_fill_manual(values = c("200" = "#83CAFF",
      "401" = "#FF90B5"), name = 'Status', labels=c("Success","Fail")) +
      ylab('Frequency') + xlab('HourOfDay') + theme(legend.position = 'top')
```

And finally the code for the success versus failure by day of week graph:

```
#daysVsStatus
daysVsStatus <- table(agents_online$date_wday,agents_online$status,
      dnn=c("Success"))
daysVsStatus
daysVsStatusDF <- data.frame(daysVsStatus)
names(daysVsStatusDF) <- c("Day", "Status", "Freq")
daysVsStatusDF$Day2 <- factor(daysVsStatusDF$Day,
      levels = c(2,9,6,13,7,14,5,12,1,8,3,10,4,11))
daysVsStatusPlot <- ggplot(daysVsStatusDF, aes(daysVsStatusDF$Day,
      y=Freq, fill=factor(daysVsStatusDF$Status))) + geom_bar(stat="identity")
daysVsStatusPlot + scale_fill_manual(values = c("200" = "#83CAFF",
      "401" = "#FF90B5"), name = 'Status', labels=c("Success","Fail")) +
      ylab('Frequency') +   xlab('HourOfDay') + scale_x_discrete(limits=
      c("monday","tuesday","wednesday","thursday","friday","saturday",
"sunday")) + theme(legend.position = ’top’)
```