R Notebook

BIS 581 #load libraries

#load in the data  
VDI <- read.csv("vdi\_serverlogs.csv", header=TRUE, stringsAsFactors = FALSE)  
  
apps <- read.csv("vdi\_statsapps.csv", header=TRUE, stringsAsFactors = FALSE)

#join the two together  
apps$VDI\_ID <- as.integer(apps$VDI\_ID)

## Warning: NAs introduced by coercion

usage <- VDI %>% inner\_join(apps)

## Joining with `by = join\_by(VDI\_ID)`

## Warning in inner\_join(., apps): Each row in `x` is expected to match at most 1 row in `y`.  
## ℹ Row 8 of `x` matches multiple rows.  
## ℹ If multiple matches are expected, set `multiple = "all"` to silence this  
## warning.

#clean up  
str(usage)

## 'data.frame': 3182089 obs. of 13 variables:  
## $ VDI\_ID : int 32 35 38 39 39 39 39 39 39 39 ...  
## $ comp\_name : chr "CMULABA55" "CMULABA34" "CMULIB11" "CMULABA63" ...  
## $ userid : chr "userid1" "userid4" "userid6" "userid7" ...  
## $ logon\_DTS : chr "4/15/11 16:36" "4/15/11 16:58" "4/15/11 18:19" "4/15/11 20:07" ...  
## $ logout\_DTS : chr "4/15/11 16:40" "4/15/11 17:01" "4/15/11 18:35" "4/16/11 0:15" ...  
## $ connection\_server: chr "" "" "" "" ...  
## $ remote\_ip : chr "" "" "" "" ...  
## $ remote\_od : chr "" "" "" "" ...  
## $ avg\_cpu : int NA NA NA NA NA NA NA NA NA NA ...  
## $ app\_id : chr "16" "17" "18" "19" ...  
## $ app\_name : chr "WINWORD" "iexplore" "iexplore" "scrnsave.scr" ...  
## $ start : chr "2011-04-15 16:37:00" "2011-04-15 17:01:00" "2011-04-15 18:20:00" "2011-04-15 20:20:00" ...  
## $ stop : chr "2011-04-15 16:38:00" "2011-04-15 17:01:00" "1900-01-01 00:00:00" "2011-04-15 20:39:00" ...

summary(usage)

## VDI\_ID comp\_name userid logon\_DTS   
## Min. : 32 Length:3182089 Length:3182089 Length:3182089   
## 1st Qu.:227370 Class :character Class :character Class :character   
## Median :460688 Mode :character Mode :character Mode :character   
## Mean :445041   
## 3rd Qu.:666553   
## Max. :853952   
##   
## logout\_DTS connection\_server remote\_ip remote\_od   
## Length:3182089 Length:3182089 Length:3182089 Length:3182089   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## avg\_cpu app\_id app\_name start   
## Min. : -9752.0 Length:3182089 Length:3182089 Length:3182089   
## 1st Qu.: 4.0 Class :character Class :character Class :character   
## Median : 8.0 Mode :character Mode :character Mode :character   
## Mean : 47.3   
## 3rd Qu.: 13.0   
## Max. :743932.0   
## NA's :38664   
## stop   
## Length:3182089   
## Class :character   
## Mode :character   
##   
##   
##   
##

Filter it by 2015 and CMUVDI only

#The variable logon\_DTS is a chr and we need to convert this into the correct data type as a POSIXct which is a Date/Time format.  
usage$logon\_DTS <- as.POSIXct(usage$logon\_DTS, format = "%m/%d/%y %H:%M")

#Filter the data to contain only data from 2015.  
usage\_2015 <- usage %>%  
#Do this by filtering the logons by the year 2015 and VDI, and capture the output.  
 filter(year(logon\_DTS) == 2015)

#search for any computer names containing CMUVDI as the first 6 letters within the 2015 dataset and send it to a new dataset.  
usage\_2015\_cmu <- usage\_2015[grep("^CMUVDI", usage\_2015$comp\_name),]

how many users were on the system total?

#count number of rows and save it to a new output.  
num\_users\_2015 <- nrow(usage\_2015\_cmu)  
#print this output back.  
cat("The total number of cmu users on the system in 2015 was:", num\_users\_2015)

## The total number of cmu users on the system in 2015 was: 156171

What is the average number of users per day

#save the 2015 usage to a new file  
Users\_per\_day <- usage\_2015\_cmu %>%  
#create a column called date and format it.  
 mutate(date = format(as.Date(logon\_DTS), "%Y-%m-%d")) %>%  
#group the rows by date   
 group\_by(date) %>%  
#count num of users for each date  
 summarise(num\_users =n())  
#calc mean(avg) of of the num of users per day, round this becuase you cant have half a user.  
avg\_users <- round(mean(Users\_per\_day$num\_users))  
#print this back  
cat("The average number of users per day in 2015 was:", avg\_users)

## The average number of users per day in 2015 was: 700

str(Users\_per\_day)

## tibble [223 × 2] (S3: tbl\_df/tbl/data.frame)  
## $ date : chr [1:223] "2015-01-07" "2015-01-08" "2015-01-09" "2015-01-10" ...  
## $ num\_users: int [1:223] 20 70 129 143 168 236 230 297 157 100 ...

head(Users\_per\_day,20)

## # A tibble: 20 × 2  
## date num\_users  
## <chr> <int>  
## 1 2015-01-07 20  
## 2 2015-01-08 70  
## 3 2015-01-09 129  
## 4 2015-01-10 143  
## 5 2015-01-11 168  
## 6 2015-01-12 236  
## 7 2015-01-13 230  
## 8 2015-01-14 297  
## 9 2015-01-15 157  
## 10 2015-01-16 100  
## 11 2015-01-17 330  
## 12 2015-01-18 346  
## 13 2015-01-19 641  
## 14 2015-01-20 631  
## 15 2015-01-21 857  
## 16 2015-01-22 886  
## 17 2015-01-23 773  
## 18 2015-01-24 821  
## 19 2015-01-25 826  
## 20 2015-01-26 1226

What is the highest number of users per day, which day did that occur

#get the max num of users per day  
max\_users\_per\_day <- max(Users\_per\_day$num\_users)   
#find date where this occured  
date\_of\_max\_users <- Users\_per\_day$date[which.max(Users\_per\_day$num\_users)]  
#print this back to screen   
cat("The Highest number of users per day is:", max\_users\_per\_day, "This occured on:", date\_of\_max\_users)

## The Highest number of users per day is: 3637 This occured on: 2015-04-28

Plot average CPU usage per month

#save data to a new DS  
usage\_2015\_month <- usage\_2015\_cmu %>%   
#create new variable month to show just the month number  
 mutate(month = format(as.Date(logon\_DTS), "%m"))  
#save data to a new DS  
avg\_cpu\_usage\_month <- usage\_2015\_month %>%  
 group\_by(month) %>%  
#compute avg cpu usage per month  
 summarise(avg\_cpu = mean(avg\_cpu))  
  
ggplot(data = avg\_cpu\_usage\_month, mapping = aes(x =month, y =avg\_cpu))+  
 geom\_col(fill = "gold") +  
 geom\_text(aes(label = round(avg\_cpu,)), vjust = -0.5) +  
 ggtitle("Average CPU usage per month in 2015")+  
 xlab("Month") +  
 ylab("Average CPU usage")

