Analyzed VDI usage at CMU using R.

Sai Sandeep Gollapudi- Midterm Project

BIS 581 Fall 2022 Student name: Sai Sandeep Gollapudi

#load libraries

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

## join the two together just in case

## Data Wrangling

#Using as.POSIXct from date\_manipulation document as the date has hours & minutes also.   
usage$logon\_DTS <- as.POSIXct(usage$logon\_DTS, format= "%m/%d/%y %H:%M")  
usage$logout\_DTS <- as.POSIXct(usage$logout\_DTS, format= "%m/%d/%y %H:%M")  
  
# Taking usage data set and filtering it for only year 2014 & CMUVDI machines   
filterUsersOn14VDI <- usage %>% filter(substring(logon\_DTS, 1, 4)=="2014", substring(comp\_name, 1,6)=="CMUVDI")

#Methodology I used to anlyze the data Explanation - Before starting analysis/ filter, as correcting all the data types required for solving the questions. After joining the data sets, I clicked on the data set and did str to understand the structure of data. As every column in a data set like the logon\_DTS, logout\_DTS are by default in character, we need to correct them. So, Fixed the date and changed format. As we need to time difference for question - top 5 users by the length of time logged in.When we joined the two data sets there were some NAs created in avg\_cpu. So, these were the errors/issues I felt that the reader needs to know about like this bad format of data.

I used str to check if logon\_DTS, logout\_DTS are changed correctly.

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 : POSIXct, format: "2011-04-15 16:36:00" "2011-04-15 16:58:00" ...  
## $ logout\_DTS : POSIXct, format: "2011-04-15 16:40:00" "2011-04-15 17:01:00" ...  
## $ 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" ...

## how many users were on the system total?

total\_system\_users <- as.data.frame(unique(filterUsersOn14VDI$userid)) %>% count()  
# printing the number of users on the system  
paste("there are ",total\_system\_users,"number of users on the system")

## [1] "there are 4331 number of users on the system"

Explanation - I used unique on the data set which had usage data set with filtering for only year 2014 & CMUVDI machines. For using count function I converted the data set into a data frame.

## what is the average number of users per day

#to just have date without time   
filterUsersOn14VDI$year\_without\_time <- as.Date(filterUsersOn14VDI$logon\_DTS)  
#I grouped by year as we need grouping based on Year. Used summarise to aggregate and i used function length and unique to find users per day. i used VDI\_ID as every time a new user logs in a new VDI\_ID is recorded for that user. To calculate the average   
per\_day\_users <- filterUsersOn14VDI %>% group\_by(year\_without\_time) %>% summarise(users\_perday =length(unique(VDI\_ID)))  
# adding all the users per day and dividing by number of days to get the average user per day  
average\_users <- sum(per\_day\_users$users\_perday)/nrow(per\_day\_users)  
# printing the results of average users per day  
paste("Average number of users per day is ",round(average\_users),"users")

## [1] "Average number of users per day is 226 users"

#• what is the highest number of users per day

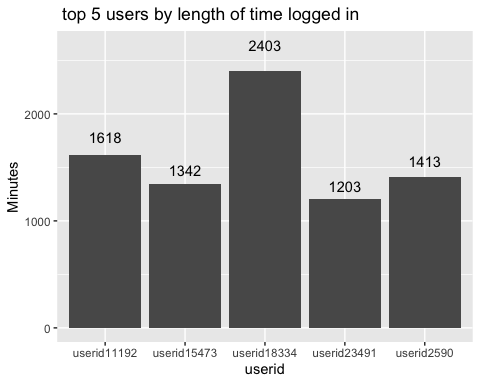
#I used max to find the highest number of users per day  
max(per\_day\_users$users\_perday)

## [1] 669

#Grad students only: #E-H: top 5 users by length of time logged in

# difftime to find difference between logon\_DTS, logout\_DTS with units as minutes  
filterUsersOn14VDI$time\_diff <- difftime(time2= filterUsersOn14VDI$logon\_DTS, time1=filterUsersOn14VDI$logout\_DTS, units = "min" )  
  
#Creating a data frame which has data grouped by userId, as we need grouping based on users. Used summarise to find the maximum time value of that user. Then arranged it based as descending on time difference as we need highest users by length of time logged in.   
  
top\_users <- filterUsersOn14VDI %>% group\_by(userid) %>% summarise(Minutes = max(time\_diff)) %>% arrange(desc(Minutes))  
  
#I used function Head to print the top 5 users  
head(top\_users,5)

## # A tibble: 5 × 2  
## userid Minutes   
## <chr> <drtn>   
## 1 userid18334 2403 mins  
## 2 userid11192 1618 mins  
## 3 userid2590 1413 mins  
## 4 userid15473 1342 mins  
## 5 userid23491 1203 mins



##top 5 applications by number of times run

#Creating a data frame which has data grouped by app\_name, as we need grouping based on app\_name. Used summarise to find the Number of Times Run. Then arranged it based as descending on that Top 5 Applications By Number of Times Run.  
app\_times\_run <- filterUsersOn14VDI %>% group\_by(app\_name) %>% summarise(Number\_run =length(app\_name)) %>%  
arrange(desc(Number\_run))  
#I used function Head to print the top 5 apps  
head(app\_times\_run,5)

## # A tibble: 5 × 2  
## app\_name Number\_run  
## <chr> <int>  
## 1 iexplore 49905  
## 2 WINWORD 42078  
## 3 firefox 26011  
## 4 saplogon 23465  
## 5 devenv 22020

appplot <- head(app\_times\_run,5)  
#used bar graph and added label for number of   
ggplot(data = appplot,mapping = aes(x = app\_name, y = Number\_run,fill = Number\_run ))+geom\_bar(stat = "identity")+geom\_text(mapping = aes(label = Number\_run), position = position\_stack(vjust = 1.1))+ ggtitle(" top 5 applications by number of times run")

