options(warn = -1)

####1: get command line parameters if run as a script DO NOT MODIFY THIS SECTION

args <- commandArgs(TRUE)

loop = TRUE

hasBeginDate = FALSE

hasEndDate = FALSE

hasDiseaseName = FALSE

hasUsername = FALSE

hasPassword = FALSE

hasHostname = FALSE

hasDBName = FALSE

hasTable = FALSE

doRateQuery = FALSE

if (length(args) > 0) {

while (loop) {

if (args[1] == "--rate") {

doRateQuery = TRUE

}

if (args[1] == "--begin") {

beginDate = args[2]

hasBeginDate = TRUE

}

if (args[1] == "--end") {

endDate = args[2]

hasEndDate = TRUE

}

if (args[1] == "--disease") {

diseaseName = args[2]

hasDiseaseName = TRUE

}

if (args[1] == "--user") {

USERNAME = args[2]

hasUsername = TRUE

}

if (args[1] == "--password") {

PASSWORD = args[2]

hasPassword = TRUE

}

if (args[1] == "--host") {

HOSTNAME = args[2]

hasHostname = TRUE

}

if (args[1] == "--db") {

DBNAME = args[2]

hasDBName = TRUE

}

if (args[1] == "--table") {

TABLE = args[2]

hasTable = TRUE

}

if (length(args) > 1) {

args = args[2:length(args)]

} else {

loop = FALSE

}

}

}

#### 2: USERS - MODIFY THE FOLLOWING SECTIONS ########

##2.1: change this section to match the database you are attempting to use along with your username/password credentials

if (!hasUsername) { USERNAME <- "ENTER YOUR USERNAME HERE" }

if (!hasPassword) { PASSWORD <- "ENTER YOUR PASSWORD HERE" }

if (!hasHostname) { HOSTNAME <- "USE PROPER SERVER NAME HERE" }

if (!hasDBName) { DBNAME <- "LockerDB" }

if (!hasTable) { TABLE <- "CREATE TABLE NAME HERE" }

##2.2: Include your date range here, using the “YYYYMMDD” format,

#Depending upon the size of your database, you can optimize performance by using a smaller date range, such as a week or two and then concatenate the results for analysis

if (!hasBeginDate || !hasEndDate) {

beginDate = "20151201"

endDate = "20160229"

}

## 2.3: Insert the disease/syndrome name here, and / or definition number

if (!hasDiseaseName) {

diseaseName = "Zika"

}

## 2.4: Set this value to true if you want the RATE query table

# Warning: this could add significant run time to the script

if (!doRateQuery) {

doRateQuery = FALSE

}

######## END USER-MODIFIED SECTION

dates = paste("'", beginDate, "' AND '", endDate, "'", sep = "")

#### 3: Select your columns here - It is currently set to 5 columns

columns <- paste("Diagnosis\_Code,Chief\_Complaint, Earliest\_Date, Facility\_State, Unique\_Visiting\_ID")

##3.1: This step clears any previous results

remove(df.DISEASE\_processed)

remove(df.DISEASE\_processed\_filtered)

####4: This is the actual query from biosense database based upon the Earliest\_Date variable

query.DISEASE\_def <- paste(

"SELECT",

columns,

"FROM", TABLE,

"WHERE",

"Earliest\_Date between ",dates,

";"

)

####5: This step allows you to download data from BioSense database

print(query.DISEASE\_def)

timestamp()

cat("running query for user",USERNAME,"\n")

library(RMySQL)

con <- dbConnect(dbDriver("MySQL"), user = USERNAME, password = PASSWORD, host = HOSTNAME, dbname = DBNAME)

df.disease <- dbGetQuery(con, query.DISEASE\_def)

dbDisconnect(con)

if (nrow(df.disease) == 0) {

timestamp()

cat(paste(diseaseName,"- no results returned\n"))

} else {

#format date fields

result <- tryCatch({

df.disease$Admit\_Date\_Time <- as.Date(df.disease$Admit\_Date\_Time)

}, warning = function(war) {

}, error = function(err) {

}, finally = {

})

result <- tryCatch({

df.disease$Earliest\_Date <- as.Date(df.disease$Earliest\_Date)

}, warning = function(war) {

}, error = function(err) {

}, finally = {

})

result <- tryCatch({

df.disease$Event\_Date\_Time <- as.Date(df.disease$Event\_Date\_Time)

}, warning = function(war) {

}, error = function(err) {

}, finally = {

})

#### 6: Below steps are for aggregating and de-duplicating records

timestamp()

cat("de-duplicating records\n")

df.DISEASE\_processed <- aggregate(df.disease, by=list(df.disease$Unique\_Visiting\_ID), function(x) paste(unique(x),collapse=","))

#### 7: This step is intended to identify and flag strings from chief\_complain text and diagnosis\_code (ICD code)

df.DISEASE\_processed$Group.1 <- NULL

if (nrow(df.DISEASE\_processed) > 0) {

df.DISEASE\_processed$Chief\_Complaint = iconv( df.DISEASE\_processed$Chief\_Complaint, "latin1","UTF-8")

## 7.1: This step converts strings in Chief\_Complaint and Diagnosis\_Code to lower case

df.DISEASE\_processed$Chief\_Complaint = tolower (df.DISEASE\_processed$Chief\_Complaint)

df.DISEASE\_processed$Diagnosis\_Code = tolower (df.DISEASE\_processed$Diagnosis\_Code)

## 7.2: This step parses specific chief complaints: conjunctivitis:[(^conjunctivitis^ ,or, ^red eye^ or ^pink eye) ,AND NOT, ^infectious conjunctivitis^)]

df.DISEASE\_processed$conjunctivitis = 0

df.DISEASE\_processed[(grepl("(conjunctivitis|conjunctiviti|redeye|red eye|pinkeye|pink eye)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(infectious conjunctivitis|infect conjunctivitis)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)),]$conjunctivitis = df.DISEASE\_processed[(grepl("(conjunctivitis|conjunctiviti|redeye|red eye|pinkeye|pink eye)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(infectious conjunctivitis|infect conjunctivitis)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)),]$conjunctivitis + 1

## 7.3: This step parses specific chief complaints: fever

df.DISEASE\_processed$fever = 0

df.DISEASE\_processed[

(grepl("(fev|fver|fv|pyrexia|temp|elev temp|elevated temp|temp elev|hi temp|high temp|temp hi|temp10|temp 10|feeling hot|780|feels hot|feel hot|fuo|febr)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

|

grepl("(^780)", df.DISEASE\_processed$Diagnosis\_Code, perl=TRUE))

& !grepl("(denies fev|shot|afeb|no fev|no temp)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

,]$fever = df.DISEASE\_processed[

(grepl("(fev|fver|fv|pyrexia|temp|elev temp|elevated temp|temp elev|hi temp|high temp|temp hi|temp10|temp 10|feeling hot|780|feels hot|feel hot|fuo|febr)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

|

grepl("(^780)", df.DISEASE\_processed$Diagnosis\_Code, perl=TRUE))

& !grepl("(denies fev|shot|afeb|no fev|no temp)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

,]$fever + 1

## 7.4: This step parses specific chief complaints: headache

df.DISEASE\_processed$headache = 0

df.DISEASE\_processed[

(grepl("hea.{1,3}ac", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(bac|lac|acr|rac|act|fac|mac|jac|heat|injury|bicy)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(ha)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(han|pha|had|hai|hav|has|hac|sha|cha|tha|nhahar|ham|hau|hal|mva|mvc|hag|hab|hap|wha)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(mig)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(migh|migrat)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(h/a)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(ch/ar|tach|ch/as|mva|mvc|injury|gh/ab|gh/an)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$headache = df.DISEASE\_processed[

(grepl("hea.{1,3}ac", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(bac|lac|acr|rac|act|fac|mac|jac|heat|injury|bicy)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(ha)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(han|pha|had|hai|hav|has|hac|sha|cha|tha|nhahar|ham|hau|hal|mva|mvc|hag|hab|hap|wha)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(mig)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(migh|migrat)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(h/a)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(ch/ar|tach|ch/as|mva|mvc|injury|gh/ab|gh/an)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$headache +1

##7.5: This step parses specific chief complaints: rash

df.DISEASE\_processed$rash = 0

df.DISEASE\_processed[

(grepl("(impitago|impetigo|rash) ", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(crash|groin|diaper|vag|geni|pub|peni|test|glut|urin)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(red|bump|spot|herp|folli)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(groin|diaper|vag|geni|peni)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(shing)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(ashing|ushing)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(pox)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(ypox)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(zos)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(bumps)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(groin|diaper|vag|geni|pub|peni|test|glut|urin)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$rash = df.DISEASE\_processed[

(grepl("(impitago|impetigo|rash) ", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(crash|groin|diaper|vag|geni|pub|peni|test|glut|urin)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(red|bump|spot|herp|folli)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(groin|diaper|vag|geni|peni)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(shing)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(ashing|ushing)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(pox)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(ypox)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(zos)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

|

(grepl("(bumps)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) & !grepl("(groin|diaper|vag|geni|pub|peni|test|glut|urin)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$rash +1

## 7.6: This step parses specific chief complaints: arthralgia: (^muscle pain^ ,or, ^body pain^,^joint pain^)

df.DISEASE\_processed$arthralgia = 0

df.DISEASE\_processed[(grepl("(muscl|join|body)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

& grepl("(pain|ach)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

| grepl("(arthralgia)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

,]$arthralgia = df.DISEASE\_processed[(grepl("(muscl|join|body)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

& grepl("(pain|ach)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

| grepl("(arthralgia)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

,]$arthralgia + 1

## 7.7: This step parses specific chief complaints: sum of the flag of above 5 syndromes

df.DISEASE\_processed$Symptom=0

df.DISEASE\_processed$Symptom<-rowSums(df.DISEASE\_processed[, c("conjunctivitis","fever","headache","rash","arthralgia")])

## 7.8: This step parses specific chief complaints: pregnancy by both Chief\_Complaint and Diagnosis\_Code

df.DISEASE\_processed$pregnancy = 0

df.DISEASE\_processed[grepl("(pregnan|matern)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

|grepl("(z3a|z43|o30|z33.1|v22.2|v72.42)", df.DISEASE\_processed$Diagnosis\_Code, perl=TRUE)

,]$pregnancy = df.DISEASE\_processed[grepl("(pregnan|matern)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

|grepl("(z3a|z43|o30|z33.1|v22.2|v72.42)", df.DISEASE\_processed$Diagnosis\_Code, perl=TRUE),]$pregnancy + 1

## 7.9: This step parses specific chief complaints: (Florida’s zika definition no.1) zika/microcephaly

df.DISEASE\_processed$FL1 = 0

df.DISEASE\_processed[grepl("(zika|microcep)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE),]$FL1 = df.DISEASE\_processed[grepl("(zika|microcep)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE),]$FL1 + 1

##7.10: This step parses specific chief complaints: (Florida’s zika definition no.2) guillain barre

df.DISEASE\_processed$FL2 = 0

df.DISEASE\_processed[grepl("(guill|gbs)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) &!grepl("(labor|pregnancy|strep|carrier|hgbss|hgbsc)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE),]$FL2 = df.DISEASE\_processed[grepl("(guill|gbs)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE) &!grepl("(labor|pregnancy|strep|carrier|hgbss|hgbsc)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE),]$FL2 + 1

## 7.11: This step parses keywords related to travel to the following countries (which may need to be expanded in the future): latin america or south america, central america, caribbean, barbados, bolivia, brazil, colombia, puerto rico, costa rica, curacao, dominican republic, ecuador, el salvador, french guiana, guadeloupe, guatemala, guyana, haiti, honduras, jamaica, martinique, mexico, nicaragua, panama, paraguay, saint martin, suriname, virgin island, venezuela, samoa, tonga

df.DISEASE\_processed$travel = 0

df.DISEASE\_processed[

grepl("(travel|latinamerica|southamerica|latin america|south america|central america| centralamerica|caribbean|barbados |bolivia|colombia| puertorico|puerto rico|costa rica|costarica|curacao| dominican|ecuador|salvador|guiana|guadeloupe|guatemala|guyana|haiti|honduras|jamaica|martinique| mexico|nicaragua|panama|paraguay|saint martin|saintmartin|suriname|virgin island|virginisland|venezuela| samoa|tonga)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

&!grepl("(no travel|not travel|denies travel|deny travel|denying travel)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

| (grepl("(brazi)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

&!grepl("(dr brazi|dr[.] brazi|dr[.]brazi)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$travel = df.DISEASE\_processed[

grepl("(travel|latinamerica|southamerica|latin america|south america|central america| centralamerica|caribbean|barbados |bolivia|colombia| puertorico|puerto rico|costa rica|costarica|curacao| dominican|ecuador|salvador|guiana|guadeloupe|guatemala|guyana|haiti|honduras|jamaica|martinique| mexico|nicaragua|panama|paraguay|saint martin|saintmartin|suriname|virgin island|virginisland|venezuela| samoa|tonga)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

&!grepl("(no travel|not travel|denies travel|deny travel|denying travel)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

| (grepl("(brazi)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE)

&!grepl("(dr brazi|dr[.] brazi|dr[.]brazi)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$travel+1

##7.12: This step parses specific chief complaints: Mosquito bite. The syntax below creates a flag variable call “mosquito”

df.DISEASE\_processed$mosquito = 0

df.DISEASE\_processed[

(grepl("(mosquito)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$mosquito = df.DISEASE\_processed[

(grepl ("(mosquito)", df.DISEASE\_processed$Chief\_Complaint, perl=TRUE))

,]$mosquito+1

##7.13: The syntax below uses diagnosis codes (ICD 9 and 10) to flag (Microcephaly) OR (Guillain-Barré) OR (all maternal infectious and parasitic diseases) OR (Other specified mosquito- borne fevers) OR (Other specified viral diseases) OR (Dengue Fever) OR (Chikungunya)

df.DISEASE\_processed$ICDcode = 0

df.DISEASE\_processed[

grepl("(:742[.]1 |q02|:357[.]0 |g61.0|:647[.]63 |o98[.]519|o98.52|o98[.]53|o98[.]511|o98[.]512|o98[.]513|o98[.]519|:066[.]3 |a92[.]08|a92[.]09)", df.DISEASE\_processed$Diagnosis\_Code, perl=TRUE)

,]$ICDcode = df.DISEASE\_processed[

grepl("(:742[.]1 |q02|:357[.]0 |g61.0|:647[.]63 |o98[.]519|o98.52|o98[.]53|o98[.]511|o98[.]512|o98[.]513|o98[.]519|:066[.]3 |a92[.]08|a92[.]09)", df.DISEASE\_processed$Diagnosis\_Code, perl=TRUE)

,]$ICDcode+ 1

#### 8: These steps subset the data flagged by newly created variables, i.e., travel, mosquito pregnancy etc.

df.DISEASE\_processed\_filtered01 = subset(df.DISEASE\_processed, travel==1)

df.DISEASE\_processed\_filtered02 = subset(df.DISEASE\_processed, mosquito==1)

df.DISEASE\_processed\_filtered03 = subset(df.DISEASE\_processed, pregnancy==1)

##8.1: This step subsets the resulting data by definition 1 to 7:

df.DISEASE\_processed\_definition1 = subset(df.DISEASE\_processed, Symptom>=2)

df.DISEASE\_processed\_definition2 = subset(df.DISEASE\_processed, Symptom>=2 & travel==1)

df.DISEASE\_processed\_definition3 = subset(df.DISEASE\_processed, Symptom>=2 & travel==1 & mosquito==1)

df.DISEASE\_processed\_definition4 = subset(df.DISEASE\_processed, ICDcode==1)

df.DISEASE\_processed\_definition5 = subset(df.DISEASE\_processed, FL1==1)

df.DISEASE\_processed\_definition6 = subset(df.DISEASE\_processed, FL2==1)

df.DISEASE\_processed\_definition7 = subset(df.DISEASE\_processed, Symptom>=2 & pregnancy==1)

####9: This step writes the subset results as a .csv

if (nrow(df.DISEASE\_processed\_definition1) > 0) {

timestamp()

cat("writing main output\n")

write.csv(df.DISEASE\_processed\_definition1, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 1", ".csv",sep=""), row.names = FALSE)

write.csv(df.DISEASE\_processed\_definition2, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 2",".csv",sep=""), row.names = FALSE)

write.csv(df.DISEASE\_processed\_definition3, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 3",".csv",sep=""), row.names = FALSE)

write.csv(df.DISEASE\_processed\_definition4, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 4",".csv",sep=""), row.names = FALSE)

write.csv(df.DISEASE\_processed\_definition5, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 5",".csv",sep=""), row.names = FALSE)

write.csv(df.DISEASE\_processed\_definition6, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 6",".csv",sep=""), row.names = FALSE)

write.csv(df.DISEASE\_processed\_definition7, file = paste(diseaseName,"\_", beginDate,"\_", endDate, " def 7",".csv",sep=""), row.names = FALSE)

write.csv(df.disease, file = paste(diseaseName,"\_", beginDate,"\_", endDate, "all",".csv",sep=""), row.names = FALSE)

} else {

timestamp()

cat("no cases identified\n")

}

}

}

timestamp()

cat("finished running")

options(warn = 0)