#####################################################################

#Yuhao Wang 03/27/2018 data cleanning process USDKRW

#####################################################################

#import data and necessary functions

#function index0: gives new index for matrix

#function slincing: take only price out of raw data

#####################################################################

USD = read.csv("C:/Users/dell/Desktop/821project/USDKRW\_wyh.csv")

slicing <- function(x)

{

x <- x[-c(3,5)]

return(x)

}

index0 <- function(x)

{

rownames(x) <- 1:dim(x)[1]

return(x)

}

#####################################################################

#reindex the matrix and separtate the data

#first calculate the index number and then add the index to the data

#name different data from data1 to data8

#reorganize the data after that using rbind function

#####################################################################

head(USD)

nas <- seq(rownum,1)

USDreverse <- USD[nas,]

rownum <- dim(USD)[1]

rownames(USDreverse) <- 1:nrow(USDreverse)

rownum

l = dim(USDKRWreverse)[1]

data1 <- data.frame(index0(USDreverse[1613:l,1:6]))

data2 <- data.frame(index0(USDreverse[245:l,8:13]))

data3 <- data.frame(index0(USDreverse[489:l,15:20]))

data4 <- data.frame(index0(USDreverse[529:l,22:27]))

data5 <- data.frame(index0(USDreverse[261:l,29:34]))

data6 <- data.frame(index0(USDreverse[1:2908,36:41]))

data7 <- data.frame(index0(USDreverse[261:l,43:48]))

data8 <- data.frame(index0(USDreverse[265:l,50:55]))

names(data8) = names(data7) = names(data6) = names(data5) = names(data4) = names(data3) = names(data2) = names(data1)

collection <- rbind(data1, data2, data3, data4, data5, data6, data7, data8)

#####################################################################

#we should convert the data in to posixct format so that we can operate

#notice that the date of posixct data starts at 0017 rather than 2017

#so we need to adjust the data accordingly by adding enough number

#####################################################################

collection$Date <- as.POSIXct(collection$Date, origin = "11-00-2017 00:00:00 GMT", format = "%m/%d/%Y %H:%M", tz = "GMT")

collection$Date <- collection$Date+31556926\*2000+52000

#####################################################################

#Check to see if ask and bid prices mathes, we define a for loop

#####################################################################

len = length(collection$Date)

count = 0

for (i in len-1)

{

if((collection[i,3]==collection[i+1,5])==FALSE){

print("the result does not match", i)

count = count+1

}else{

print("the result matches")

}

}

#####################################################################

#Data aggregation: we need to add the price within the same minute

#using for loop.

#for here we use mean as to get the price

#####################################################################

collection <- slicing(collection)

USD\_minute = vector()

row1 = collection[1,2:4]

time = 1

for (i in 2:len){

if (collection$Date[i-1]==collection$Date[i]){

row1 <- row1+collection[i,2:4]

time <- time+1

}else{

row1 <- row1/time

Date = collection$Date[i-1]

USD\_minute <- rbind(USD\_minute,cbind(Date,row1))

row1 <- collection[i-1,2:4]

time = 1

}

}

USD\_minute <- index0(USD\_minute)

#####################################################################

#Creating a time new time grid

#Reorganize the data and build up a new data file

#####################################################################

day1 = seq.POSIXt(from = as.POSIXct("11/22/2017 20:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("11/23/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day2 = seq.POSIXt(from = as.POSIXct("11/23/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("11/24/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day3 = seq.POSIXt(from = as.POSIXct("11/26/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("11/27/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day4 = seq.POSIXt(from = as.POSIXct("11/27/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("11/28/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day5 = seq.POSIXt(from = as.POSIXct("11/28/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("11/29/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day6 = seq.POSIXt(from = as.POSIXct("11/29/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("11/30/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day7 = seq.POSIXt(from = as.POSIXct("11/30/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("12/01/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day8 = seq.POSIXt(from = as.POSIXct("12/03/2017 19:00",format = "%m/%d/%Y %H:%M"),to = as.POSIXct("12/04/2017 01:29",format = "%m/%d/%Y %H:%M"),by = "min")

day = c(day1,day2,day3,day4,day5,day6,day7,day8)

daylen = length(day)

USD\_timegrid= data.frame(Date = day, lastprice = rep(0,daylen), bid = rep(0,daylen), ask = rep(0,daylen))

daylength = list(length(day1),length(day2),length(day3),length(day4),length(day5),length(day6),length(day7),length(day8))

count1 = 0

for (i in 1:length(USD\_timegrid$Date)){

for (j in 1:length(USD\_minute$Date)){

if(USD\_timegrid$Date[i] != USD\_minute$Date[j]){

#USD\_timegrid[i,2:4] = 0.5\*(USD\_minute[i,2:4]+USD\_timegrid[i-1,2:4])

}else{

USD\_timegrid[i,2:4] <- USD\_minute[j,2:4]

}

}

}

#####################################################################

#We now need to fill the missing data; here we use average of previous

#day and later day

#####################################################################

USD\_timegrid[3060,2:4] = USD\_timegrid[3059,2:4]

for (i in 2:length(USD\_timegrid$Date)-1)

{

if(USD\_timegrid[i,2] == 0){

if(USD\_timegrid[i+1,2] != 0 && USD\_timegrid[i-1,2] != 0){

USD\_timegrid[i,2:4] = 0.5\*(USD\_timegrid[i-1,2:4]+USD\_timegrid[i+1,2:4])

}

else

{

ran <- runif(3, -3, 3 )

if(USD\_timegrid[i+1,2] == 0 && USD\_timegrid[i-1,2] != 0){

USD\_timegrid[i,2:4] = USD\_timegrid[i-1,2:4] + ran

}

else

{

USD\_timegrid[i,2:4] = USD\_timegrid[i+1,2:4]

}

}

}

}

#####################################################################

#Here we calculate the return

#First we get the mid price

#derive return based on mid price

#####################################################################

USD\_return = data.frame(Date = day[-1], Trade\_return = rep(1,daylen-1), Quote\_return = rep(1,daylen-1))

for (i in 1:daylen-1){

USD\_return[i,2] = (USD\_timegrid[i+1,2]-USD\_timegrid[i,2])/USD\_timegrid[i,2]

midP = (USD\_timegrid[i,3]+USD\_timegrid[i,4])/2

midN = (USD\_timegrid[i+1,3]+USD\_timegrid[i+1,4])/2

USD\_return[i,3] = (midN-midP)/midP

}

#####################################################################