Extracting data from SQL Server will and simulating inserts in SQL Server table will primarily simulate the near real-time data. If you have followed the previous post, you will notice that I am using same T-SQL table and query to extract real-time data.

First, we will create a sample table in SQL Server and populate it with some sample data:

CREATE DATABASE Test;

GO

USE Test;

GO

CREATE TABLE dbo.LiveStatsFromSQLServer

(ID INT IDENTITY(1,1)

,Num tinyint NOT NULL)

And populate it with some sample data:

-- Insert some test data

INSERT INTO dbo.LiveStatsFromSQLServer(num)

SELECT ABS(CHECKSUM(NewId())) % 14

GO 10

Now, that we have SQL foundations set up, let’s focus on R code.

First we set the environment variable and the RODBC library:

library(RODBC)

# create env for storing the variables/data frames between the functions

assign("getREnvironment", new.env(), envir = .GlobalEnv)

We will generate a function for extracting data from SQL Server and storing it in environment data.frame variable:

# Function to read data from SQL Server

getSQLServerData <- function()

{

#extract environment settings for storing data

getREnvironment <- get("getREnvironment", envir = .GlobalEnv, mode = "environment")

#get the SQL Server data

con <- odbcDriverConnect('driver={SQL Server};

server=TOMAZK\\MSSQLSERVER2017;

database=test;trusted\_connection=true')

db\_df <- sqlQuery(con, 'SELECT

TOP 20 id

,num

FROM LiveStatsFromSQLServer ORDER BY id DESC')

close(con)

#overwrite existing data with new data

df\_overwrite <- db\_df

getREnvironment$db\_df <- data.frame(df\_overwrite)

try(assign("getREnvironment", getREnvironment, envir = .GlobalEnv))

invisible() #do not print the results

}

Once we have this function registered, we can now create a small for loop that will update the plot with newly fetched data from SQL Server:

# Plot graph

n=1000 #nof iterations

windowQuery=20 # syncronised with TOP clause in SELECT statement

for (i in 1:(n-windowQuery)) {

flush.console()

getSQLServerData()

getREnvironment <- get("getREnvironment", envir = .GlobalEnv, mode = "environment")

data <- getREnvironment$db\_df

plot( data$id, data$num, type='l',main='Realtime data from SQL Server')

Sys.sleep(0.5)

}

Once we run the complete R code, we need to trigger and run also the new inserts in SQL Server Management studio:

-- Do some inserts to mimic the data stream

INSERT INTO dbo.LiveStatsFromSQLServer(num)

SELECT ABS(CHECKSUM(NewId())) % 14

WAITFOR DELAY '00:00:00.500'

GO 100

Once we do this, we can observe the realtime data from SQL Server being plotted in R environment (R Studio).

