

# Manual para el manejo del script para analizar los datos de los sensores de movimiento

13 de febrero de 2020, 15:00 horas

## Antes de utilizarlo

Para utilizar el script es necesario que tenga instalado el programa R (version 3.6.1 o superiores) y R Studio (version 1.2.1335 o superiores).

Para ejecutar el script se debe tener los siguientes tres archivos:

- Manual\_UsoScriptSensores.Rmd Este es el archivo principal donde se ejecuta las líneas de código descritas a continuación.
- datos\_brutos. Está es una tabla que tiene todos los datos de los sensores.
- sensores\_movimiento\_FUNS\_V2.R. Están todas las funciones que utilizan en el script analisis\_sensores.R
- validacion.csv En este documento está la validación del comportamiento de las vacas.

Recuerde copiar y pegar estos archivos en su carpeta de trabajo.

## Cargar librerías y funciones

Las librerías que se utilizan son las mostradas a continuación y también se cargará las funciones creadas que están en el archivo Sensores\_movimiento\_FUNS\_V2.R.

```
library(tidyverse)
library(plyr)
library(data.table)
library(dplyr)
library(pracma)
library(RMySQL)

rm(list=ls())

source('Sensores_movimiento_FUNS_V2.R')
```

## Abrir la base de datos

En esta sección se abrirá la base de datos de los sensores. Es importante mencionar que la base de datos está en formato sql, por lo tanto asegúrese de guardar esta base de datos en algún repositorio para luego conectarla. En este caso, la base de datos está guardada en el disco local, local host, y la base de datos se llama sensores\_ganaderia que contiene la tabla datos\_brutos. Para comprobar que está correctamente conectada se hace una consulta la cantidad de datos que tiene y el resultado debe ser 13524712.

```
mydb = dbConnect(MySQL(), user='localhost', password='', dbname='sensores_ganaderia')
total_registros = dbGetQuery(mydb, 'SELECT COUNT(*) AS total_registros FROM datos_brutos')
total_registros

## total_registros
## 1 13524712
```

## Ingresar datos de entrada

Los datos de entrada son el código de la vaca, la fecha de inicio y fecha final de la toma de datos de los sensores. Una vez fijados estos parametros se filtra la base de datos y la salida es un data frame compuesto por las columnas Hora, X, Y, Z, Finca y Vaca. En este ejemplo, la fecha de inicio es 2019-09-05 y la fecha final es 2019-09-07 y el codigo de la vaca es 7301.

```
base_filtros = dbGetQuery(mydb,
"SELECT Fecha, Hora, X, Y, Z, Finca, Vaca
  FROM datos_brutos
 WHERE DATE(Fecha) BETWEEN '2019-09-05' AND '2019-09-13' AND Vaca = 7301")

head(base_filtros)
```

```
##      Fecha      Hora      X      Y      Z      Finca Vaca
## 1 2019-09-05 11:04:17 -0.98 -0.11 0.14 Agrosavia 7301
## 2 2019-09-05 11:04:17 -0.99 -0.11 0.13 Agrosavia 7301
## 3 2019-09-05 11:04:17 -0.99 -0.11 0.14 Agrosavia 7301
## 4 2019-09-05 11:04:17 -0.99 -0.10 0.13 Agrosavia 7301
## 5 2019-09-05 11:04:17 -0.98 -0.11 0.14 Agrosavia 7301
## 6 2019-09-05 11:04:17 -0.99 -0.10 0.14 Agrosavia 7301
```

## Calcular la actividad por cada registro

Aqui se estimará la actividad que está haciendo la vaca. Hay tres variables para variar, Fac\_R1, Fac\_R2 y Fac\_C0. En este ejemplo tiene los valores 0.95, 1.05 y -0.25, respectivamente.

```
estimar_actividades <- estimar_actividad(base_filtros, Fac_R1 =0.95, Fac_R2 = 1.05,
                                         Fac_C0 = -0.25)

head(estimar_actividades )
```

```
##      Fecha      Hora      X      Y      Z      Finca Vaca Cuadrado actividad
## 1 2019-09-05 11:04:17 -0.98 -0.11 0.14 Agrosavia 7301    0.9921    Reposo
## 2 2019-09-05 11:04:17 -0.99 -0.11 0.13 Agrosavia 7301    1.0091    Reposo
## 3 2019-09-05 11:04:17 -0.99 -0.11 0.14 Agrosavia 7301    1.0118    Reposo
## 4 2019-09-05 11:04:17 -0.99 -0.10 0.13 Agrosavia 7301    1.0070    Reposo
## 5 2019-09-05 11:04:17 -0.98 -0.11 0.14 Agrosavia 7301    0.9921    Reposo
## 6 2019-09-05 11:04:17 -0.99 -0.10 0.14 Agrosavia 7301    1.0097    Reposo
```

## Determinar la actividad en un determinado tiempo

Para determinar la actividad de la vaca, se debe primero fijar un periodo de tiempo, 10, 20, 30, o 60 minutos, La actividad está definida como la actividad que tuvo mayor frecuencia en ese periodo de tiempo.

Recuerde cambiar el parametro número de vaca en la funcion subset para tomar los datos de la vaca 7301 de la validación. En este caso la vaca es 7301.

```
actividades_agrupadas <- agrupar_actividades(registros=estimar_actividades ,tiempo='10mins')
#Leer las validacion
validacion <- read.csv("validacion.csv", header = T)
#Borrar los valores nulos
validacion <- validacion[!apply(is.na(validacion) | validacion == "", 1, all),]
validacion$completTime <- as.POSIXct(as.character(validacion$completTime))
#Tomar los datos de la vaca 7301
validacion <- subset(validacion, Vaca = 7301)
actividades_agrupadas <- rbind(actividades_agrupadas, validacion)
```

## **Graficar las actividades**

Se grafican las actividades diarias de la vaca. Estas imagenes son guardadas en el directorio de trabajo.