

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

Este manual está dividido en dos partes, la primera está enfocado para los datos que provienen de los sensores de movimiento y la segunda para los datos de los sensores timpanicos.

1.Sensores de movimiento

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 lineas de codigo 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 librerias y funciones

Las librerias que se utilizan son las mostradas a continuación y tambien 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 datos de los sensores. Es importante mencionar que la base de datos está en formato sql, por lo tanto aseguresé de guardar esta base de datos en algun 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
```

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 Vaca = '42-3' OR Vaca = '7301' OR Vaca = '12504' OR Vaca = '3395' OR
Vaca = '13502' OR Vaca ='E30'")

head(base_filtros)
```

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 por cada registro
estimar_actividades <- estimar_actividad(base_filtros, Fac_R1 =0.95, Fac_R2 = 1.05,
                                         Fac_C0 = -0.25)

#head(estimar_actividades )
```

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.

```
#Estimar actividades en un rango de tiempo

system.time(actividades_agrupadas <- agrupar_actividades_V1(registros=estimar_actividades,
                                                             tiempo='10mins'))

#actividades_agrupadas
```

Graficar las actividades

En esta parte se grafica los datos de los sensores.

```
grafica_validacion_sensor_V1(actividades_agrupadas)
#head(actividades_agrupadas)
```

Guardar Informacion

Se guarda la consultada realizada

```
write.csv(actividades_agrupadas,paste0("Validacion","_SensoresMovimiento",".csv"), row.names = F)
```

2.Sensores de temperatura timpanica

Esta parte necesita los mismos requerimientos que la parte 1.

Cargar librerias

Adicionamos estas librerias a las anteriores/

```
library(purrr)
library(PerformanceAnalytics)
rm(list=ls())
```

Lectura de datos

En esta sección se leerá las bases de datos de clima y de temperatura timpanica.

```
load("Data/Produccion_Clima_Agrosavia.RData")
load("Data/Produccion_Clima_TempTimAgrosavia.RData")
```

Funciones

Estas son las funciones que se utilizarán en la ejecución del código.

```
estad.basic <- function(x)
{
  est <- cbind(mean(x, na.rm = TRUE),var(x, na.rm = TRUE),
               sd(x, na.rm = TRUE),t(quantile(x, na.rm = TRUE)))
  colnames(est) <- c("media","var","desv.est","min","Q1","Q2","Q3","max")
  return(round(est ,2))
}

cantidadDatosTimpanicos <- function (x)
{
  index <- which(is.na(x))
  len <- length(x[index])
  return(len)
}
```

Resumen estadístico

En esta parte es el resumen estadístico luego se guardará en un archivo.

```
resumen <- produccion_agrosavia %>% group_by(Vaca) %>%
  dplyr::summarize (total_NA = cantidadDatosTimpanicos(Temp_Timpani),
                   total = length(Temp_Timpani) )
resumen$totalDatos <- resumen$total - resumen$total_NA
total <- sum(resumen$totalDatos)
total

est.total_agrosavia <- apply(produccion_agrosavia[,c(2,3,4,5,6,7)] ,2,estad.basic)
rownames(est.total_agrosavia)<- c("media","var","desv.est","min","Q1","Q2","Q3","max")
colnames(est.total_agrosavia) <- c("Temperatura Timpanica Celsius", "Temperatura Celsius",
```

```
"Humedad Relativa %", "Precipitacion mm", "Radiacion Solar cal/cm2", "Velocidad del viento mps")
write.csv(est.total_agrosavia, "Data/Resumen_Agrosavia_Estad.csv")
```

Graficos

Se realizo las graficos para conocer las correlaciones entre la temperatura timpanica y las ambientales.

```
jpeg('Graficos/Articulo/Timpanico_Clima_agrosavia.jpeg', width = 600, height = 600)
agrosavia <- produccion_agrosavia[,c(2,3,4,5,6,7)]
chart.Correlation(agrosavia, histogram=TRUE, pch=19)
dev.off()

produccion_agrosavia$FechaHora_2 <- as.POSIXct(produccion_agrosavia$FechaHora)
graficar <- produccion_agrosavia[,c(3,4,5,6,7,11)]
colnames(graficar) <- c("Temp (C)", "HR (0 - 100)", "Pre (mm)", "SR (cal/cm2)",
                      "V_V(mps)", "Dia")
gr <- gather(graficar, "Variable", "Valor", -c(Dia))
#gr$Dia <- as.Date(gr$Dia)

bp <- ggplot(gr, aes(x=Dia, y=Valor, group = 1)) +
  geom_line() + theme_bw() + theme(axis.title.x = element_blank(),
                                   axis.title.y = element_blank()) +
  ggtitle("Finca 4") + theme(plot.title = element_text(hjust = 0.5))

grafica <- bp + facet_grid(Variable~ ., scales="free")
grafica

nombre <- paste0("Graficos/Articulo/", "Agrosavia_2", ".jpeg")
ggsave(nombre, units = "cm", width = 10, height = 11)
```

Regresiones Lineales

En esta parte se realizará las regresiones correspondientes a las temperaturas timpanicas y de clima.

```
index <- which (colnames(produccion_agrosavia) %in%
               c("Temp_Timpani", "Temperatura", "Humedad Relativa",
                 "Precipitacion" , "SR","VientoVelocidad" ))
modelo_lineal <- produccion_agrosavia[, index ]
LinearModel<- lm(Temp_Timpani ~., data=modelo_lineal)
summary(LinearModel)

sink("Graficos/Temperatura Timpanica/Regresion Lineal/RegresionLineal_Agrosavia.txt")
print(summary(LinearModel))
sink()
```