Sensativity Function

Kriddie

2023-01-18

Sensativity function

This R markdown uses precipitation and discharge data to estimate the sensativity function within a paramo watershed

##read in data Q / water level solar radiation humidity preciptiation

data sources are from la virgen de papallacta weather station and water level measurments collected within the gavilan catchment in CC NP

```
Q_df <- read.csv(here::here("Prelims_SensativityAnalysis/Stn01_Q.csv"))
Solar_df <-  read.csv(here::here("Prelims_SensativityAnalysis/Radiación_Solar_Horario.csv"))
Precip_df <- read.csv(here::here("Prelims_SensativityAnalysis/Precipitación_Horario.csv"))
Humidity_df <- read.csv(here::here("Prelims_SensativityAnalysis/Humedad_del_aire_Horario.csv"))
```

format data

1. Datetime format
2. clean Q data - fix units
3. down sample Q to hourly

Units conundrum

Q units in the publication is mm/sec. The data collected is waterlevel (m) and Q (m^3/s). The fact that a 1d representation of discharge is not the norm

Two options I can think of - (1) convert waterlevel data to rate by treating it like precipitation data, and calculate rate of hourly change in water level (2) Use Q, but treat it as a 1d rate, so 1 m^3 is and increase in depth by 1m.

Based on figures of discharge in the paper, and also the fact that wl doesn't account for velocity so, I am going with option 2. But I am unclear about this

merge data

```
#DateTime
Q_df$DateTime <- as.POSIXct(Q_df$DateTime, format="%Y-%m-%d %H:%M:%S", tz="UTC")
Solar_df$DateTime <- as.POSIXct(Solar_df$DateTime, format="%Y-%m-%d %H:%M:%S", tz="UTC")
Precip_df$DateTime <- as.POSIXct(Precip_df$DateTime, format="%Y-%m-%d %H:%M:%S", tz="UTC")
Humidity_df$DateTime <- as.POSIXct(Humidity_df$DateTime, format="%Y-%m-%d %H:%M:%S", tz="UTC")
```

```
#clean Q df
Q_df <- Q_df[c("DateTime","WL_m","Q_m3s")]
#select time range
Q_df <-Q_df%>%filter(DateTime > as.POSIXct("2021-06-01 00:00:00", tz="UTC") & DateTime < as.POSIXct("202

#the units need to be in mm/sec. so what am I supposed to do with that?? right now discharge is in m^3 f
#I think that's what I will do for this analysis, but I am very unclear about this. So we multiply by 10
Q_df$Q_mm_d <- Q_df$Q_m3s * 1000 * 60 * 60
#let convert wl to mm, just for fun
Q_df$WL_mm <- Q_df$WL_m * 1000
Q_df$WL_m <- NULL

#ok, not I downsize to 1 hour, but we can do that by left join, it's a little easier that way

df <- right_join(Q_df,Precip_df, by="DateTime")
df <- full_join(df,Solar_df, by="DateTime")
df <- full_join(df,Humidity_df, by="DateTime")

#remove all NA
```

filter data

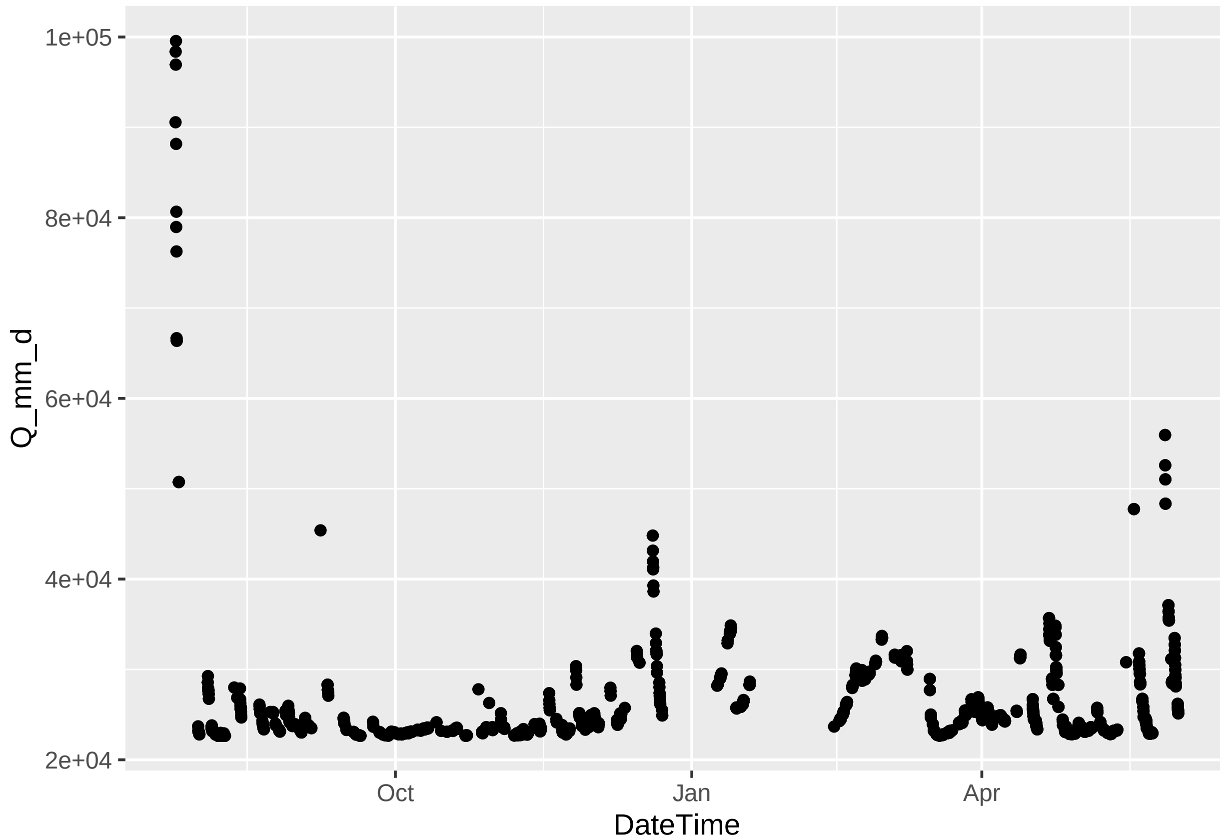
now we select Q data that was collected at night when humidity is high and no precipitation

1. humidity >= 90%
2. solar radiation < 1W/,2
3. precipitation == 0 (in Kirchner (2009) they use a time period with no preceding precipitation 6 hr before and 2 hr after)

```
df_filter <- df
#make loop to see if precipitation occured 6 hours before or 2 hours later
df_filter$precipt_filter <- NA

for(i in 1:nrow(df_filter)) {
  if ( i > 7 ) {

   df_filter$precipt_filter[i]  <- sum(df_filter$Precipt_mm[(i-6):(i+2)])

    } else {
      df_filter$precipt_filter[i] = NA
      }
}

#remove all NA
df_filter <- df_filter%>%drop_na()

#filter for P = 0 and E = 0
df_filter <- df_filter%>%filter(Humidity_. > 90 & Solar_Wm2 < 1 & precipt_filter == 0 )
df_filter <- df_filter[c("DateTime","Q_mm_d")]
```

#plot filtered data

```r
df_filter <- df_filter%>%filter(Q_mm_d < 300000)
df_filter <- df_filter%>%filter(DateTime < as.POSIXct("2022-01-20 00:00:00",tz="UTC") |
                                  DateTime > as.POSIXct("2022-02-12 00:00:00",tz="UTC"))
ggplot(df_filter,
        aes(x=DateTime, y=Q_mm_d)) +
  geom_point()
```



##recession plots low recession rate (-dQ/dt) as a function of discharge (Q)

estimate the rate of flow recession as the difference in discharge between two successive hours, -dQ/dt = (Qt(t-Dt) - Qt)/Dt, and plot this as a function of the average discharge over the two hours, (Q(t-Dt) + Qt)/2.

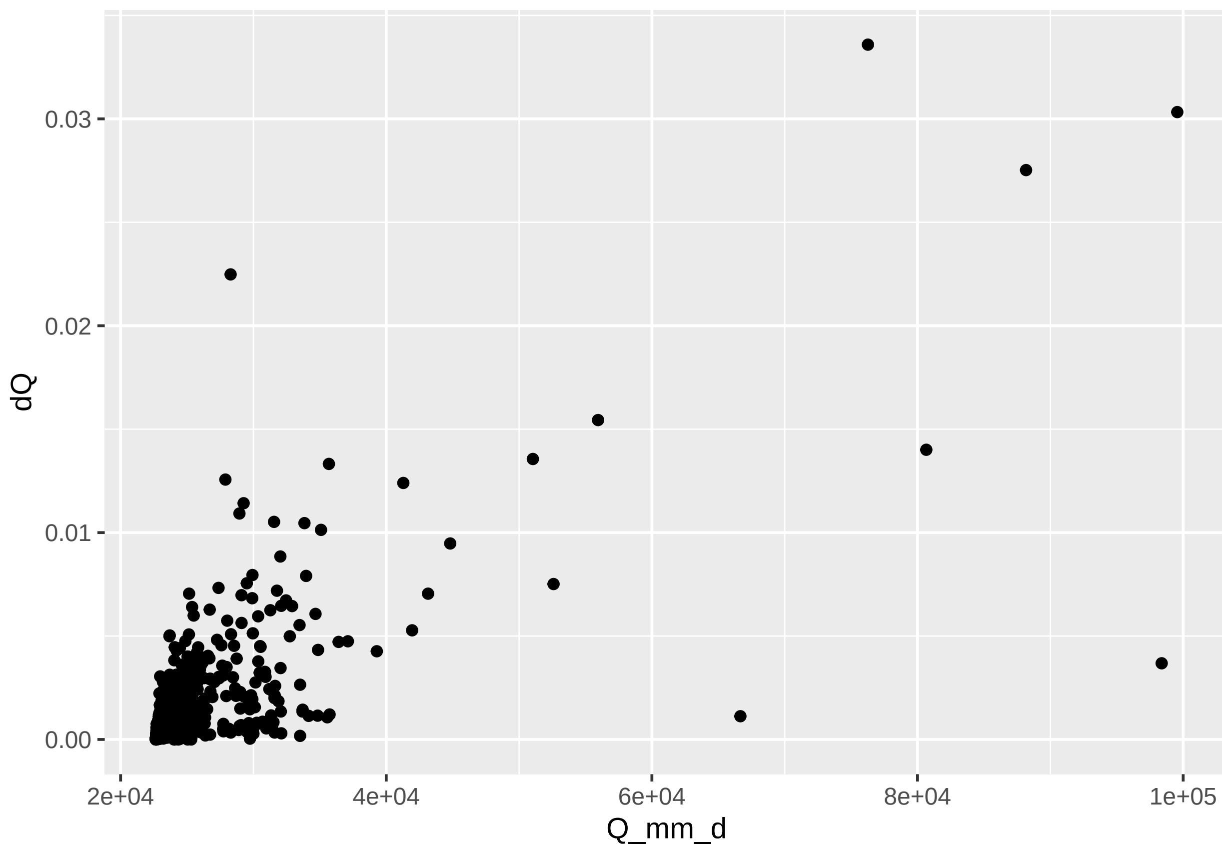
```r
#calculate flow recession rate

df_filter_2 <- df_filter%>%drop_na()


df_filter_2$dQ <- NA
df_filter_2$dQ_1 <- NA



for(i in 1:(nrow(df_filter_2)-1)) {
  if ( as.numeric(difftime(df_filter_2$DateTime[i+1], df_filter_2$DateTime[i], units="hours")) == 1 ) {
```

```r
      df_filter_2$dQ[i] <-( (df_filter_2$Q_mm_d[i] - df_filter_2$Q_mm_d[i+1])/1 ) /   (df_filter_2$Q_mm_

    df_filter_2$dQ_1[i] <-(df_filter_2$Q_mm_d[i] - df_filter_2$Q_mm_d[i+1])/1


    } else {
      df_filter_2$dQ[i] = NA
      }
}
```

```r
#calculate flow recession rate

df_filter_2$dQ_log <- log10(df_filter_2$dQ)
```

```
## Warning: NaNs produced
```

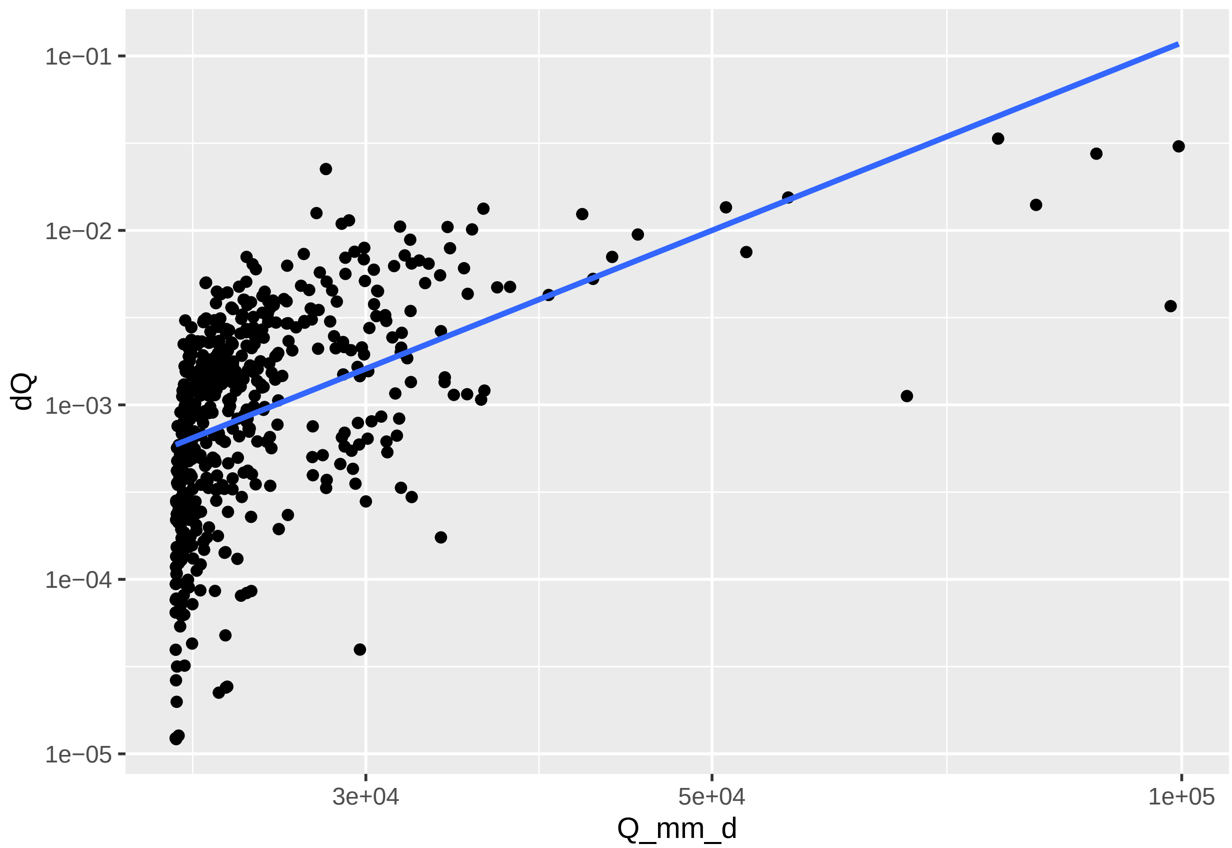
```r
ggplot(df_filter_2%>%drop_na(),
        aes(x=Q_mm_d, y=dQ)) +
  geom_point()
```




```r
#x and y axis in log scale

ggplot(df_filter_2%>%drop_na()%>%filter(dQ_log > -5 ),
        aes(x=Q_mm_d, y=dQ)) +
```

```
  geom_point() +
 scale_y_log10() +
  scale_x_log10() +
     geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
df_filter_2$dQ_log <- log(df_filter_2$dQ)
```

```
## Warning in log(df_filter_2$dQ): NaNs produced
```

```
df_filter_2$Q_log <- log(df_filter_2$Q)

model_df <- df_filter_2[,c("DateTime","dQ_log","Q_log")]

model_df <- model_df%>%drop_na()%>%filter(dQ_log != "Inf")


model <- lm(df_filter_2$dQ ~ df_filter_2$Q_mm_d)

#model_2 <- lm(model_df$Q_log ~ model_df$dQ_log + I(model_df$dQ_log^2))

print(model)
```

```
##
## Call:
## lm(formula = df_filter_2$dQ ~ df_filter_2$Q_mm_d)
##
## Coefficients:
##        (Intercept)  df_filter_2$Q_mm_d
##         -3.050e-03           1.545e-07
```

ln(-dQ/dt) = -3.050e-03 + 1.545e-07 * ln(Q)