Flag analysis

Marlin Lee, Abe Megahed, Kyllan Wunder
University of Wisconsin Data Science Institute - August, 2022

2022-09-23

```
#load WasteWater_data into the environment
data(WasteWater_data, package = "DSIWastewater")
baseWaste_DF <-  buildWasteAnalysisDF(WasteWater_data)
data(Case_data, package = "DSIWastewater")
Case_DF <- Case_data

Flag_DF <- read.csv("Temp/DHSFlagingMethodOutput.csv")%>%
            mutate(date = as.Date(date))%>%
  select(-X)

date_Flag_DF <- DF_date_vector(Flag_DF, "date",
                names(Flag_DF)[3:68])
```

```
#"case_flag_Cases"                          "case_flag_7DayCases"
#"case_flag_plus_comm.threshold_Cases"      "case_flag_plus_comm.threshold_7DayCases"
#"slope_switch_flag_Cases"                  "slope_switch_flag_7DayCases"
dep_flags <- names(Flag_DF)[9:68]
edgeThresh <- 7
CaseFlag <- "slope_switch_flag_Cases"
DateDistDF <- date_distance_calc(date_Flag_DF, CaseFlag,
                                 dep_flags, edge = edgeThresh)%>%
  select(site, date, dep_flags)%>%
  tidyr::pivot_longer(cols = dep_flags,
                      names_to = c("FlagType","window", "quant"),
                      values_to = "FlagError",
                      names_sep = "_")%>%
  mutate(window = as.numeric(window), quant = as.numeric(quant))
```

<Above has been done before by peter. the next section is where we hope to show value>

```
CaseNumberFlags <- sum(Flag_DF[[CaseFlag]], na.rm = TRUE)

Flag_DF%>%
  group_by(site)%>%
  summarise(across(c(dep_flags, !!sym(CaseFlag)), ~sum(.x, na.rm=TRUE)))%>%
  mutate(across(c(dep_flags), ~(.x-!!sym(CaseFlag))))%>%
  ungroup()%>%
  summarise(across(c(dep_flags), ~sum(abs(.x), na.rm=TRUE)))%>%
  tidyr::pivot_longer(cols = dep_flags,
                      names_to = c("FlagType","window", "quant"),
```

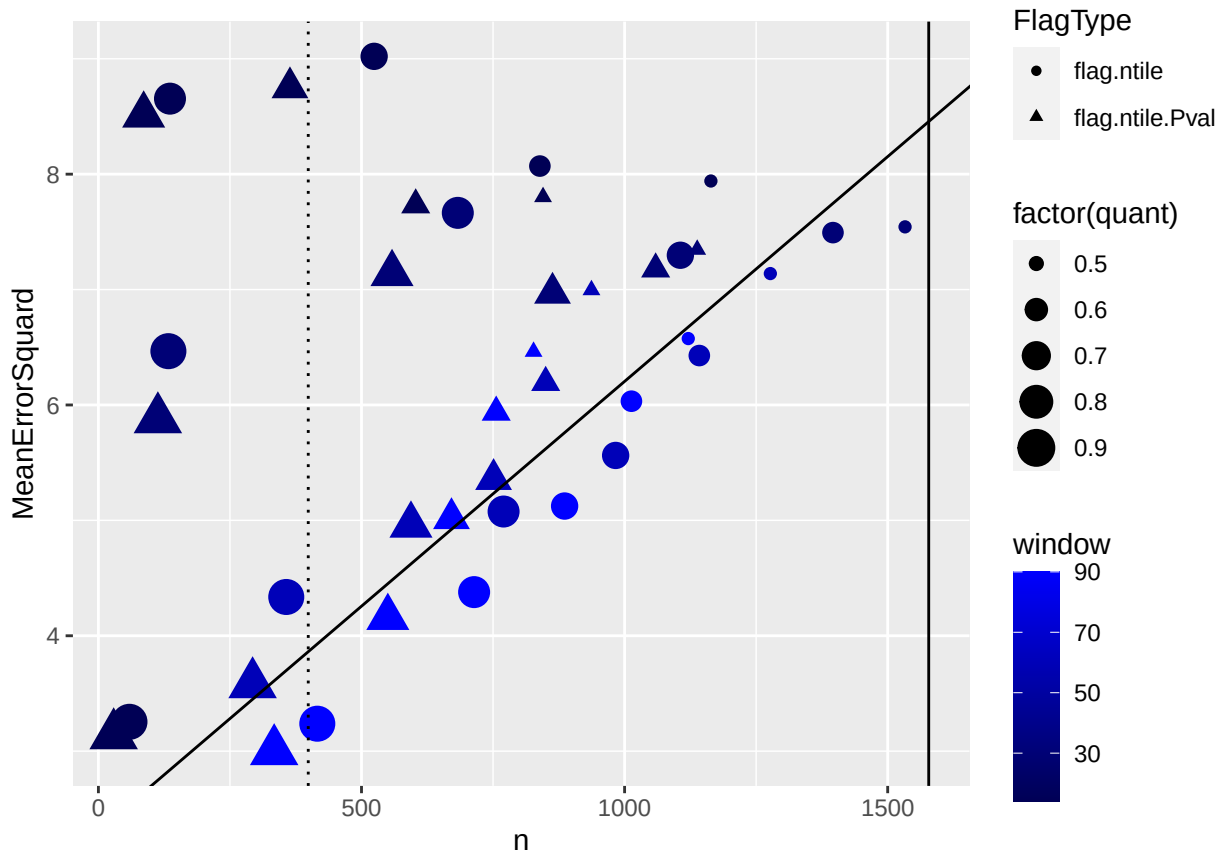
```
                      values_to = "TotalFlagCountDiff",
                      names_sep = "_")%>%
  arrange(TotalFlagCountDiff)
```

```
## # A tibble: 60 x 4
##    FlagType        window quant TotalFlagCountDiff
##    <chr>           <chr>  <chr>              <int>
##  1 flag.ntile.Pval 30     0.5                  968
##  2 flag.ntile.Pval 30     0.6                  983
##  3 flag.ntile      30     0.6                  984
##  4 flag.ntile      30     0.5                 1009
##  5 flag.ntile      60     0.5                 1009
##  6 flag.ntile      60     0.6                 1016
##  7 flag.ntile      90     0.5                 1017
##  8 flag.ntile      30     0.7                 1028
##  9 flag.ntile      90     0.6                 1033
## 10 flag.ntile      60     0.7                 1047
## # ... with 50 more rows
```

```
DistSummary <- DateDistDF%>%
  group_by(window, quant, FlagType)%>%
  summarise(Mean = mean(FlagError, na.rm = TRUE),
            MeanErrorSquard = mean(FlagError^2, na.rm = TRUE),
            Var = var(FlagError, na.rm = TRUE),
            n = sum(!is.na(FlagError)),
            Missed = mean(FlagError == edgeThresh, na.rm = TRUE))

QuantDistSummary <- DistSummary%>%
  filter(FlagType != "cdc.flag")

A <- QuantDistSummary%>%
  ggplot(aes(x = n, y = MeanErrorSquard, color = window,
             size = factor(quant), shape = FlagType))+
  geom_point()+
  geom_abline(slope = 0.0038959, intercept = 2.3082686)+
  geom_vline(xintercept = CaseNumberFlags)+
  scale_colour_gradient(low = "#000055", high = "#0000FF")+
  geom_vline(xintercept = nrow(baseWaste_DF)*CaseNumberFlags/nrow(Case_DF),
             linetype = 3)
A
```




```
#ggplotly(A)

summary(lm(MeanErrorSquard ~ n, data = QuantDistSummary))
```

```
##
## Call:
## lm(formula = MeanErrorSquard ~ n, data = QuantDistSummary)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6667 -1.1793  0.0538  1.0448  3.2855
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.1643851  0.5347058   9.658 8.89e-12 ***
## n           0.0015039  0.0006612   2.275   0.0287 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.605 on 38 degrees of freedom
## Multiple R-squared:  0.1198, Adjusted R-squared:  0.09667
## F-statistic: 5.174 on 1 and 38 DF,  p-value: 0.02866
```

```
summary(lm(MeanErrorSquard ~ n, data = QuantDistSummary[QuantDistSummary$window > 14,]))
```

```
##
## Call:
## lm(formula = MeanErrorSquard ~ n, data = QuantDistSummary[QuantDistSummary$window >
##     14, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7757 -0.7047 -0.1081  0.6502  2.1746
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.9711682  0.4818480   8.242 5.71e-09 ***
## n           0.0024087  0.0005553   4.337 0.000169 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.063 on 28 degrees of freedom
## Multiple R-squared:  0.4019, Adjusted R-squared:  0.3805
## F-statistic: 18.81 on 1 and 28 DF,  p-value: 0.000169
```

```
summary(lm(MeanErrorSquard ~ n, data = QuantDistSummary[QuantDistSummary$window > 30,]))
```

```
##
## Call:
## lm(formula = MeanErrorSquard ~ n, data = QuantDistSummary[QuantDistSummary$window >
##     30, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7118 -0.3907 -0.1225  0.3952  1.0370
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.3082686  0.3749790   6.156 8.21e-06 ***
## n           0.0038959  0.0004634   8.407 1.20e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5636 on 18 degrees of freedom
## Multiple R-squared:  0.797,  Adjusted R-squared:  0.7857
## F-statistic: 70.68 on 1 and 18 DF,  p-value: 1.2e-07
```