Outliers

Marlin

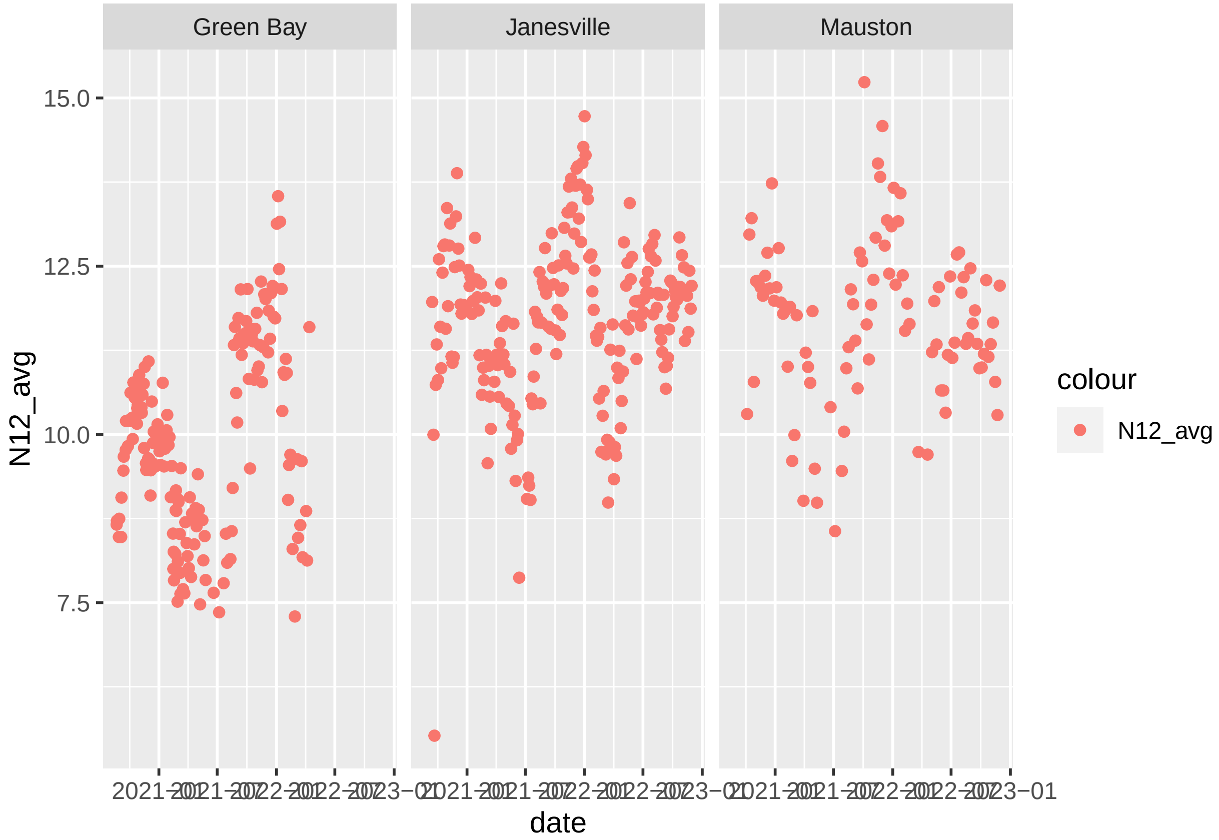
2023-06-15

This package contains two very different methods to remove outliers from the Time series data. The first method only uses the adjacent data points making it very functional for procedural use. The other method uses an estimated trend to get a more robust understanding of outliers. This is generally going to be less effective for production but should give a more accurate baseline for research. This vignette will have four sections, data, adjacent outlier detection, trend outlier detection, and finally comparing them.

Data

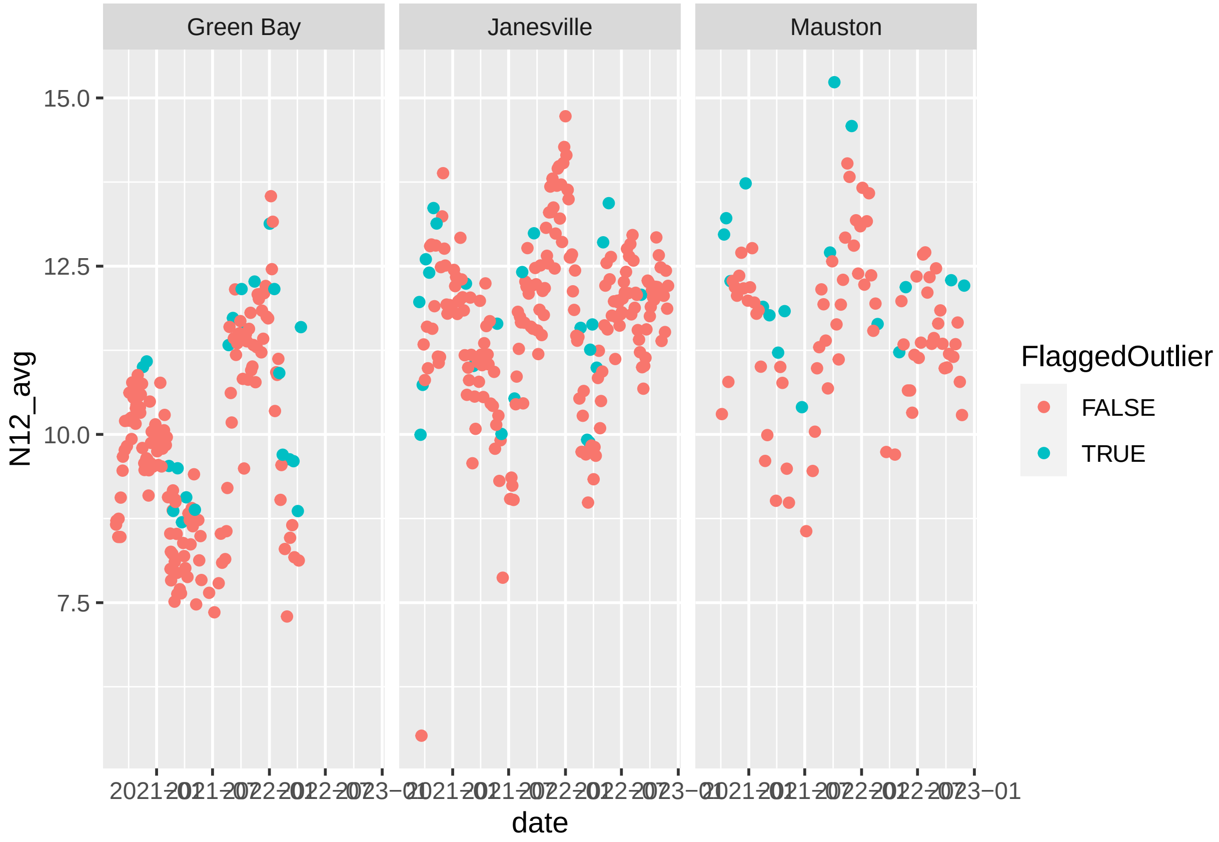
This vignette seeks to cover different methods for finding time series outliers.

```
library(DSIWastewater)
library(dplyr)
library(ggplot2)

data("Example_data", package = "DSIWastewater")

smoothing_df <- Example_data%>%
  select(site, date, N1, N2)%>%
  filter(N1 != 0, N2 != 0)%>%
  mutate(N1 = log(N1), N2 = log(N2), N12_avg = (N1 + N2) / 2)

base_plot <- smoothing_df%>%
  ggplot(aes(x = date))+
  geom_point(aes(y = N12_avg, color = "N12_avg"))+
  facet_wrap(~site)

base_plot
```



## Adjacent outliers

```
df_data <- computeJumps(smoothing_df, N1Quant = "N1", N2Quant = "N2")
ranked_data <- rankJumps(df_data)
ranked_quantile_data <- computeRankQuantiles(ranked_data)
classied_data <- flagOutliers(ranked_quantile_data, 9)%>%
  select(site, date, N12_avg, FlaggedOutlier)

classied_data%>%
  ggplot(aes(x = date))+
  geom_point(aes(y = N12_avg, color = FlaggedOutlier))+
  facet_wrap(~site)
```

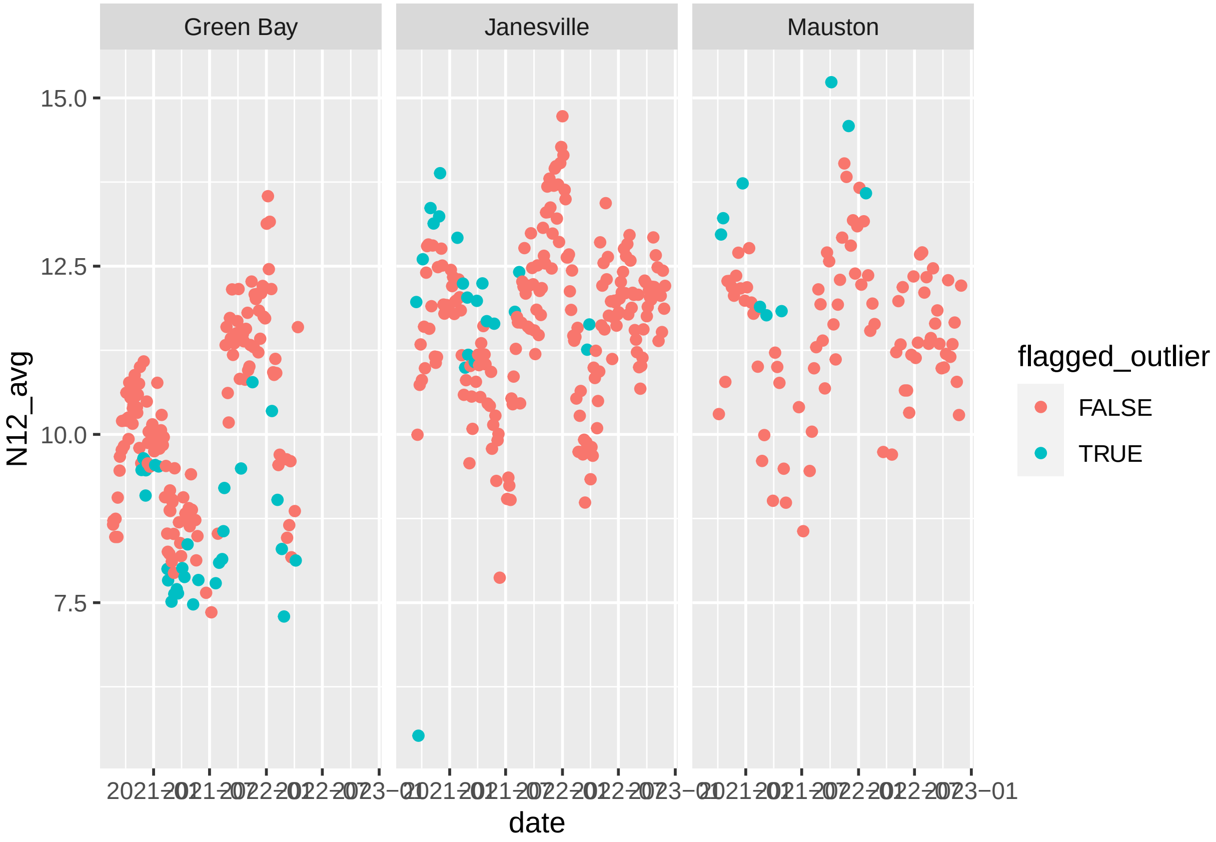


```
#result_df <- removeOutliers(classied_data, Messure = N12_avg)
```

Trend outlier

Unlike the adjacency method The trend method does not come fully formed. It can use any trend method you give it.

```
df_data <- loessSmoothMod(smoothing_df, "N12_avg", "N12_avg_loess", Filter = NULL)
classied_data_trend <- df_data%>%
  group_by(site)%>%
  Flag_From_Trend( N12_avg, N12_avg_loess)%>%
  select(site, date, N12_avg, flagged_outlier)

classied_data_trend%>%
  ggplot(aes(x = date))+
  geom_point(aes(y = N12_avg, color = flagged_outlier))+
  facet_wrap(~site)
```



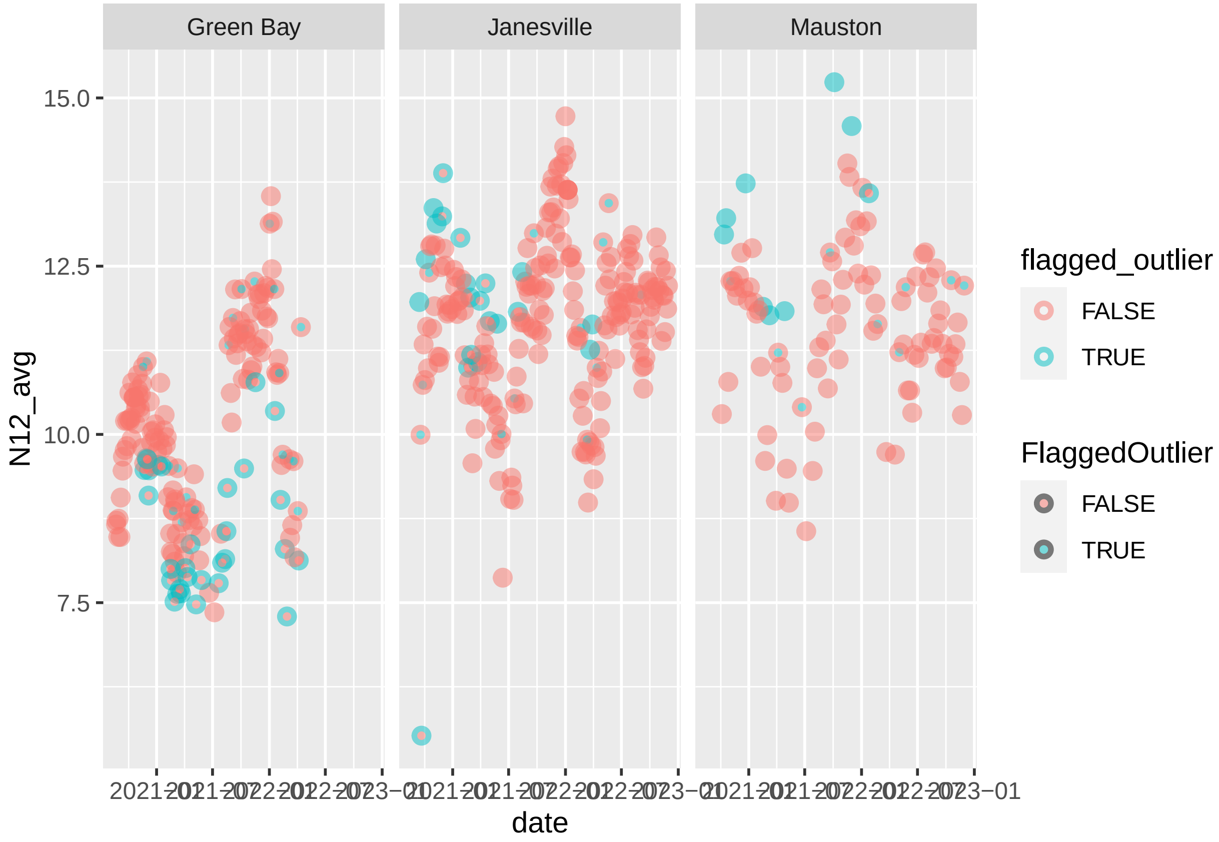
compare

These methods work in very different ways so it makes sense to see how they compare to each other.

```
library(dplyr)
full_df <- full_join(classied_data, classied_data_trend)
```

```
## Joining with ‘by = join_by(site, date, N12_avg)‘
```

```
full_df%>%
  ggplot(aes(x = date))+
  geom_point(aes(y = N12_avg, color = flagged_outlier, fill = FlaggedOutlier),
             shape = 21, size = 1.5, alpha = .5, stroke = 1.5)+
  facet_wrap(~site)
```



We used these functions in some analysis TODO