Exploring Trends in Marine Water Quality Data

A tutorial in R

Erin Peterson and Alan Pearse, EP Consulting

September 27, 2022

Introduction

This tutorial describes the steps needed to undertake a statistically robust, exploratory analysis of water quality trends in R Statistical Software version 4.2.1 (R Core Team 2022). The tutorial is broken into sections, which cover the following topics:

1. R packages required to run the code in this document;
2. Preliminary data pre-processing protocols;
3. Plots and tables for exploratory analyses;
4. Trend analysis including the van Belle Hughes test, seasonal Kendall test, Sen's slope, and 95% confidence intervals for the slope estimate.

Required packages

You will need the following R packages installed to complete this tutorial:

- `tidyverse`: a collection of R packages used for data wrangling and plotting (Wickham et al. 2019);
- `zoo` for functions that can easily convert between date formats (Zeileis and Grothendieck 2005);
- `ggcorrplot` for a function that produces nice plots of correlation matrices (Kassambara 2019);
- `openxlsx` for functions that interface with Excel worksheets (Schauberger and Walker 2021);
- `lubridate`: Functions to make working with dates easier (Grolemund and Wickham 2011)
- `EnvStats`: Functions for trend analysis (Millard 2013a)

These packages do not come pre-installed with R and will need to be installed separately. This can be done by running the install.packages command.

```
install.packages(c("tidyverse", "zoo", "ggcorrplot", "openxlsx", "lubridate", "EnvStats"))
```

Note that packages only need to be installed the first time they are used. After that, packages can simply be loaded into the R session.

```
## Load packages
library(tidyverse)
library(zoo)
library(ggcorrplot)
library(openxlsx)
library(lubridate)
library(EnvStats)
```

Import and format data

The Healthy Land and Water, Estuarine and Marine water quality dataset is used in this tutorial. It contains 39589 water quality measurements, collected at 182 sites, across 26 different reporting regions. Of this, 18 reporting regions and 129 sites are located within estuarine environments, while 8 reporting regions and 53 sites are in marine environments. Water quality data from these sites were available between January 2001 through to March 2022, though some sites consisted of fewer samples than others. From 2003 to 2014, sampling was typically conducted monthly, after which the sampling frequency was reduced to eight times annually, by excluding sampling in January, April, June and July. In a small number of cases, weather and tides prevented samples from being collected, which resulted in missing data. In other cases, multiple samples were collected within the same month to capture the impact of flood events on water quality. The water quality constituents assessed as part of this analysis included temperature (°C), salinity (g/L), turbidity (NTU), dissolved oxygen saturation (DOSat, %), chlorophyll a (Chl a, μg/L), total nitrogen (TN, mg/L), dissolved inorganic nitrogen (DIN, mg/L NH3 + NOx), total phosphorus (TP, mg/L), and filterable reactive phosphorus (FRP, mg/L).

A number of pre-processing steps are undertaken to make data analyses in R easier. After reading in the data, we create a variable `WaterType`, which indicates whether a site is located in a Bay or an Estuary. The `Date` column must be formatted so that R can recognise its contents as dates instead of unordered strings. This allows some additional columns representing `Month` and `Year` to be created, as well as a modified date column named `RunYear` (a combination of Run and Year). Finally, we remove some unused variables for convenience.

```
## Import data
dat <- read.csv("Example_InputData.csv", fileEncoding="UTF-8-BOM")

## Format data
dat <- dat %>%  mutate(Date = as.Date(Date, "%d/%m/%Y"), ## Convert to Date format
                       WaterType = ifelse( ## Create the WaterType indicator
                           1 - 1 * (Waterway %in% c("Pumicestone Passage",
                                  "Deception Bay", "Bramble Bay",
                                  "Waterloo Bay", "Central Bay",
                                  "Eastern Bay", "Southern Bay",
                                  "Broadwater")), "Estuary","Bay"),
                       Month = month(Date, label=TRUE), ## Create Month
                       Year = year(Date), ## Create Year
                       ## Create a RunYear variable for plotting
                       RunYear = as.yearmon(as.Date(as.yearmon(
                           paste0(Year,"-", Run))))) %>%
    modify_if(is.character, as.factor) %>% ## Convert columns from character to factor
    select(-BasinCode, -Streamme) %>% ## Remove columns
    arrange(SiteCode, Date) ## Sort by SiteCode and then Date
```

It is a good idea to check the dimensions of the data.frame and column names to ensure the data have been imported and formatted properly.

```
## Examine the dimensions of the data.frame
dim(dat)
```

```
## [1] 39589    22
```

```
## Look at column names
names(dat)
```

```
##  [1] "SiteCode"  "Date"      "Basinme"   "MPme"      "Waterway"  "AMTD"
##  [7] "Lat"       "Long"      "Temp"      "Sal"       "Turb"      "DOSat"
```

```
## [13] "Chla"      "TN"        "DIN"       "TP"        "FRP"       "Month"
## [19] "Year"      "Run"       "WaterType" "RunYear"
```

You can also summarise the columns and examine the range of values, as well as the number of missing values (`NA`) to help reveal errors in the data or unexpected formatting issues (e.g. numeric data formatted as character).

```
## Summarise all of the columns in the data.frame
summary(dat)
```

```
##     SiteCode          Date                                    Basinme
##  E00700 :  228   Min.   :2001-01-04   Brisbane River Basin         :5351
##  E00600 :  227   1st Qu.:2006-02-07   Logan River Basin            :6373
##  E00601 :  227   Median :2010-08-12   Maroochy River Basin         :4609
##  E00603 :  227   Mean   :2010-12-13   Noosa River Basin            :2376
##  E00605 :  227   3rd Qu.:2015-05-19   Pine River Basin             :4174
##  E00800 :  227   Max.   :2022-03-11   Queensland Coastal (Offshore):9590
##  (Other):38226                        South Coast Basin            :7116
##                                                                    MPme
##  Brisbane River 1.1km from mouth (EHMP 700)                          :  228
##  Bremer River 0.0km at junction with Brisbane River EHMP MP E00600:  227
##  Bremer River 12.4km from mouth EHMP MP E00605                       :  227
##  Bremer River 2.6km from mouth EHMP MP E00601                        :  227
##  Bremer River 7.1km from mouth EHMP MP E00603                        :  227
##  Moreton Bay grid reference 080820 (088200) (EHMP) site 916          :  227
##  (Other)                                                             :38226
##                 Waterway          AMTD             Lat              Long
##  Lower Brisbane     : 3385   Min.   : 0.000   Min.   :-28.16   Min.   :152.8
##  Logan              : 2687   1st Qu.: 0.000   1st Qu.:-27.74   1st Qu.:153.0
##  Noosa              : 2376   Median : 3.400   Median :-27.53   Median :153.2
##  Caboolture         : 2240   Mean   : 8.806   Mean   :-27.43   Mean   :153.2
##  Nerang             : 2230   3rd Qu.:11.100   3rd Qu.:-27.15   3rd Qu.:153.3
##  Pumicestone Passage: 1987   Max.   :82.000   Max.   :-26.24   Max.   :153.5
##  (Other)            :24684
##       Temp            Sal             Turb            DOSat
##  Min.   :11.40   Min.   : 0.00   Min.   :  0.00   Min.   :  2.90
##  1st Qu.:20.00   1st Qu.:11.40   1st Qu.:  3.10   1st Qu.: 80.30
##  Median :23.60   Median :28.70   Median :  7.00   Median : 92.80
##  Mean   :23.09   Mean   :23.05   Mean   : 19.81   Mean   : 88.48
##  3rd Qu.:26.10   3rd Qu.:34.60   3rd Qu.: 17.00   3rd Qu.: 99.30
##  Max.   :33.90   Max.   :44.00   Max.   :996.00   Max.   :342.10
##  NA's   :191     NA's   :198     NA's   :225      NA's   :260
##       Chla              TN               DIN                TP
##  Min.   :  0.000   Min.   : 0.0050   Min.   : 0.0020   Min.   :0.0010
##  1st Qu.:  1.100   1st Qu.: 0.1800   1st Qu.: 0.0050   1st Qu.:0.0150
##  Median :  2.100   Median : 0.3200   Median : 0.0240   Median :0.0340
##  Mean   :  4.112   Mean   : 0.5129   Mean   : 0.1895   Mean   :0.1225
##  3rd Qu.:  4.300   3rd Qu.: 0.6800   3rd Qu.: 0.2060   3rd Qu.:0.1400
##  Max.   :332.000   Max.   :12.3000   Max.   :11.1000   Max.   :5.3000
##  NA's   :1883      NA's   :610       NA's   :670       NA's   :610
##       FRP             Month            Year           Run
##  Min.   :0.0010   Feb    : 3879   Min.   :2001   Min.   : 1.000
##  1st Qu.:0.0040   Mar    : 3813   1st Qu.:2006   1st Qu.: 3.000
```

```
##  Median :0.0100   Nov    : 3751   Median :2010   Median : 7.000
##  Mean   :0.0823   Oct    : 3749   Mean   :2010   Mean   : 6.738
##  3rd Qu.:0.0750   Dec    : 3727   3rd Qu.:2015   3rd Qu.:10.000
##  Max.   :5.1000   Aug    : 3720   Max.   :2022   Max.   :12.000
##  NA's   :623      (Other):16950
##    WaterType        RunYear
##  Bay    :11577   Min.   :2001
##  Estuary:28012   1st Qu.:2006
##                  Median :2011
##                  Mean   :2011
##                  3rd Qu.:2015
##                  Max.   :2022
##
```

Exploratory Analysis

Summary statistic tables by Estuary, Bay, and Waterway region

The first step to any exploratory analysis is to look at the data to see if there are any obvious errors. However, this can be difficult when there are many variables and sites. In this section, we demonstrate how to produce and save summary statistics tables for water quality variables collected at Estuary sites, Bay sites, and Waterway regions. This is demonstrated here with turbidity.

The first step is to create a "workbook" inside R, where the summary tables for each variable will be stored. Note that this mimics the structure of an Excel file inside R.

```
## Create workbook
out.xlsx.regions <- createWorkbook()

## Create a sheet within the workbook - make sure to change the name
## to reflect the variable you are summarising in the next step
addWorksheet(out.xlsx.regions, "Turbidity")
```

The following code is used to create a summary statistics table containing information about turbidity at sites grouped as estuaries and bays. Since we are interested in summarising tubidity data, the 'Turb' column is selected in the code chunk below. To summarise another water quality variable, simply substitute that column name for 'Turb' in the following R code. It will also be important to add a worksheet for that water quality variable (above) and write to the appropriate worksheet once the data have been summarised (below).

```
## By estuary/bay
watertype.sum <- dat %>%
    select(WaterType, Turb) %>% ## Select WaterType and water quality
                                ## variable column name
  group_by(WaterType) %>% ## Group by WaterType (Bay or Estuary)
  summarise(Min = min(Turb, na.rm = T), ## Calculate summary stats
    `1st Qu` = quantile(Turb, .25, na.rm = T),
    Median = median(Turb, na.rm = T),
    Mean = mean(Turb, na.rm = T),
    `3rd Qu` = quantile(Turb, .75, na.rm = T),
    Max = max(Turb, na.rm = T),
    `Std Dev` = sd(Turb, na.rm = T),
    N = sum(!is.na(Turb)),
```

```
    Missing = sum(is.na(Turb))) %>%
    rename(Group = WaterType) ## Rename column

## Print the table
watertype.sum
```

```
## # A tibble: 2 x 10
##   Group     Min `1st Qu` Median  Mean `3rd Qu`   Max `Std Dev`     N Missing
##   <fct>   <dbl>    <dbl>  <dbl> <dbl>    <dbl> <dbl>     <dbl> <int>   <int>
## 1 Bay         0      1.5    3.6  5.52        7   202      7.83 11497      80
## 2 Estuary     0      4.8    9   25.7        25   996     50.6  27867     145
```

As you can see, a table with 2 rows (Bay, Estuary) has been created containing the summary statistics for turbidity, including the number of turbidity samples (`N`) and number of missing values (`Missing`). Check to ensure the range of values is realistic and that the number of observations and missing values are what you expect. If not, go back and check the data for errors.

A similar process is taken to calculate the summary statistics for turbidity by region.

```
## Summarise by Waterway region and water quality variable
region.sum <- dat %>%
  select(Waterway, Turb) %>%
  group_by(Waterway) %>%
  summarise(Min = min(Turb, na.rm = T),
    `1st Qu` = quantile(Turb, .25, na.rm = T),
    Median = median(Turb, na.rm = T),
    Mean = mean(Turb, na.rm = T),
    `3rd Qu` = quantile(Turb, .75, na.rm = T),
    Max = max(Turb, na.rm = T),
    `Std Dev` = sd(Turb, na.rm = T),
    N = sum(!is.na(Turb)),
    Missing = sum(is.na(Turb))
  ) %>%
    rename(Group = Waterway)

## Print the first few rows of the table
head(region.sum)
```

```
## # A tibble: 6 x 10
##   Group          Min `1st Qu` Median  Mean `3rd Qu`   Max Std De~1     N Missing
##   <fct>        <dbl>    <dbl>  <dbl> <dbl>    <dbl> <dbl>    <dbl> <int>   <int>
## 1 Albert           2     16.8   29   50.6      57   792      72.4   1521       7
## 2 Bramble Bay      0      1.5    3.2  5.13      7    69.9     5.97  1317      33
## 3 Bremer           4     24.6   39.4 58.0      66   844      65.0   1327      11
## 4 Broadwater       0      1.7    4    4.83      6.6  69       5.01  1870       1
## 5 Cabbage Tree     0      6      9.2 12.8      14.2 476      19.5   1031       3
## 6 Caboolture       0      6      9   14.1      14   390      23.2   2230      10
## # ... with abbreviated variable name 1: `Std Dev`
```

Now the two summary statistics tables are combined to form one table for turbidity.

```
## Combine the two tables
output <- bind_rows(watertype.sum, region.sum)

## Look at first 6 rows of the table
head(output)
```

```
## # A tibble: 6 x 10
##   Group         Min `1st Qu` Median  Mean `3rd Qu`   Max `Std Dev`     N Missing
##   <fct>       <dbl>    <dbl>  <dbl> <dbl>    <dbl> <dbl>     <dbl> <int>   <int>
## 1 Bay             0      1.5    3.6  5.52      7   202        7.83 11497      80
## 2 Estuary         0      4.8    9   25.7      25   996       50.6  27867     145
## 3 Albert          2     16.8   29   50.6      57   792       72.4   1521       7
## 4 Bramble Bay     0      1.5    3.2  5.13      7    69.9      5.97  1317      33
## 5 Bremer          4     24.6   39.4 58.0      66   844       65.0   1327      11
## 6 Broadwater      0      1.7    4    4.83      6.6  69        5.01  1870       1
```

The turbidity summary statistics table (`output`) is then stored in the Turbidity sheet we created earlier.

```
## Send table to Turbidity 'sheet' in the workbook. Make sure to
## change the sheet name if you are summarising a different variable
writeData(out.xlsx.regions, sheet = "Turbidity", x = output)
```

This process should be repeated for each water quality variable before exporting the workbook as an `.xlsx` file. If you would like to save the file somewhere other than the current working directory, include the pathname in the output filename.

```
## Save workbook as an Excel file
saveWorkbook(out.xlsx.regions,
             "Region_SumTables.xlsx", ## output filename
             overwrite = TRUE)
```

Summary statistics tables by site

This section provides code for producing a summary statistics table for individual EHMP sites.

Again, we create a "workbook" inside R, with a different name than the first, and add a sheet called Turbidity

```
## Create xlsx file
out.xlsx.sites <- createWorkbook()

## Add a sheet
addWorksheet(out.xlsx.sites, "Turbidity")
```

Then we compute the summary statistics. Note that this code is structurally similar to the code used in the previous section; however, it has been modified to group the data within each unique `SiteCode` when computing the summary statistics. Here, there is also no need to have separate codeblocks for computing summary statistics within bays/estuaries and waterway regions.

```
## Generate summary stats for turbidity by SiteCode
output <- dat %>%
```

```
  select(SiteCode, Turb) %>%  ## Select the SiteCode and water quality
                              ## variable of interest
  group_by(SiteCode) %>%
  summarise(Min = min(Turb, na.rm = T),
    `1st Qu` = quantile(Turb, .25, na.rm = T),
    Median = median(Turb, na.rm = T),
    Mean = mean(Turb, na.rm = T),
    `3rd Qu` = quantile(Turb, .75, na.rm = T),
    Max = max(Turb, na.rm = T),
    `Std Dev` = sd(Turb, na.rm = T),
    N = sum(!is.na(Turb)),
    Missing = sum(is.na(Turb)))

## Look at first few table rows
head(output)
```

```
## # A tibble: 6 x 10
##   SiteCode   Min `1st Qu` Median  Mean `3rd Qu`   Max `Std Dev`     N Missing
##   <fct>    <dbl>    <dbl>  <dbl> <dbl>    <dbl> <dbl>     <dbl> <int>   <int>
## 1 E00100     0        2.8    4    5.46      6     182      12.2    225       0
## 2 E00101     0        3      4    6.34      6     337      22.6    224       0
## 3 E00103     0.1      3.7    5    9.46      6.8   488      33.7    223       1
## 4 E00104     0        5      6.6 11.2       9     370      26.5    222       2
## 5 E00105     1        4      6    6.77      8      69       5.41   224       0
## 6 E00106     0        5      8   10.4      13     117      10.4    224       0
```

```
## Add the table to the worksheet
writeData(out.xlsx.sites, sheet = "Turbidity", x = output)
```

When the tables for all the water quality variables have been created and stored in the workbook, export it as an `.xlsx` file.

```
## Store workbook
saveWorkbook(out.xlsx.sites, "Site_SumTables.xlsx", overwrite = TRUE)
```

Hovmöller plots

Hövmoller plots are used to explore spatio-temporal trends in the data. They are essentially raster plots, where a regular grid is defined based on combinations of a time variable (`RunYear`) and a spatial location (`SiteCode`). Each cell in the grid is coloured by the value of a variable of interest (e.g. turbidity, salinity, etc.). Here, we demonstrate using turbidity.

There can be only one measurement per SiteCode, Run and Year in a Hövmoller plot. Therefore, we average measurements collected in the same sampling Run and year before generating the plots.

```
## Create averaged dataset
avg.dat <- dat %>%
  select(SiteCode, Run, Date, RunYear, Waterway, AMTD, WaterType, Temp:FRP) %>%
  group_by(SiteCode, RunYear, Waterway, AMTD, WaterType) %>%
  summarise_at(vars(Temp:FRP), mean, na.rm = T)
```

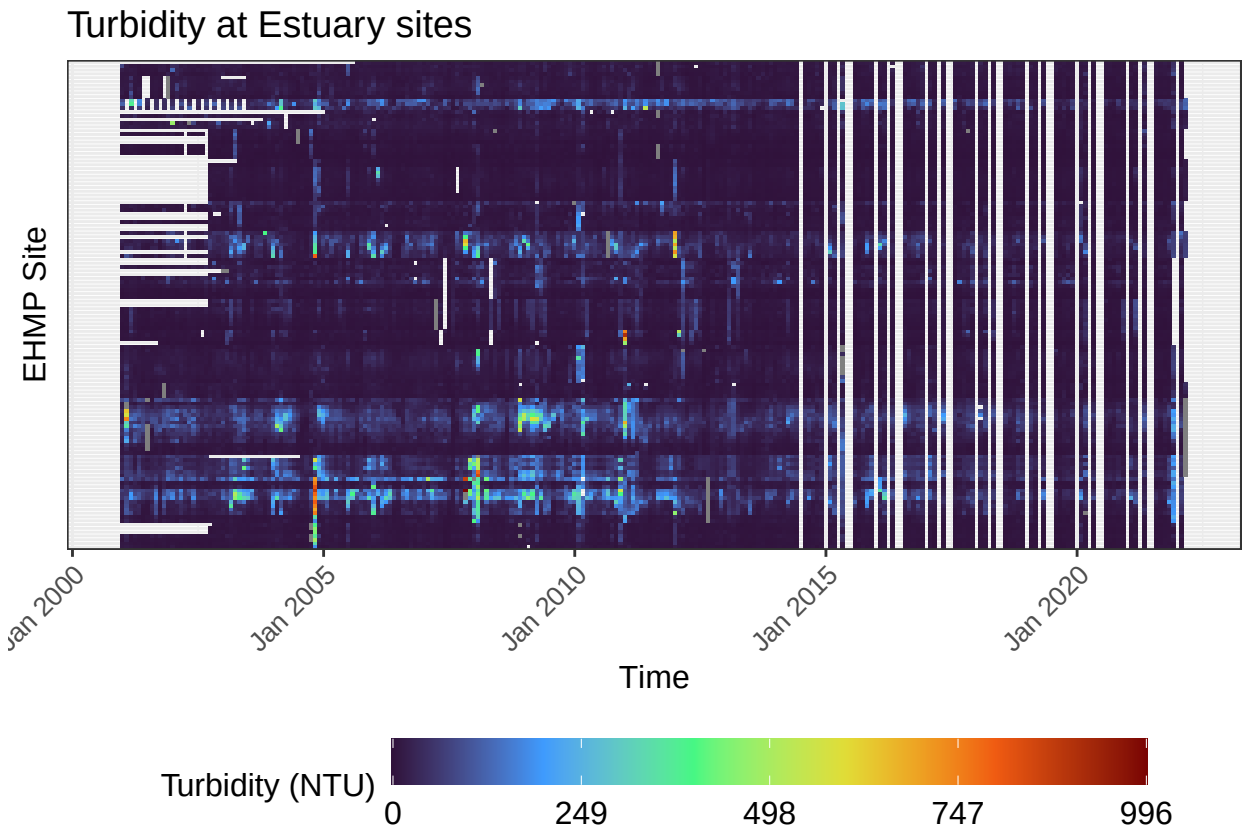
We also define a custom function called `equal_breaks`, which is used to create and label equally spaced breaks

in the scale bar representing continuous data (Massicotte 2022). Note that the arguments in `equal_breaks` (`n`, `s`, `digits`) should be altered when the function is called, rather than altering the custom function itself.

```
## Custom function to create equal breaks in continuous data
equal_breaks <- function(n = 3, s = 0, digits = 3){
 ##   n = number of breaks to create. Default = 3
 ##   s = scaling factor dictating where min and max breaks occur. Default = 0
 ##       creates ticks at the min and max values
 ##   digits = number of decimal places to include in tick labels. Default = 3
 function(x){
    d <- s * diff(range(x)) / (1+2*s)
    round(seq(min(x)+d, max(x)-d, length=n), digits = digits)
  }
}
```

A Hövmoller plot for turbidity at all sites found in estuaries can now be created using the `ggplot2` package (Wickham et al. 2022).

```
## Create a data.frame that contains data from estuaries
est.avg.dat <- filter(avg.dat, WaterType == "Estuary")

## Create the plot
est.hov.plot <- ggplot(data = est.avg.dat, aes(x = RunYear, y = SiteCode, fill = Turb)) +
  geom_tile() +
    labs(x = "Time", y = "EHMP Site", fill = "Turbidity (NTU)",
         title = "Turbidity at Estuary sites") +
  scale_fill_viridis_c(option="turbo", breaks = equal_breaks(n=5, digits = 0)) +
  theme_bw() +
  theme(legend.text = element_text(size = 12),
      legend.title = element_text(size = 12),
      axis.text.x=element_text(angle=45, hjust =1, size = 10),
      axis.title.x=element_text(size = 12),
      axis.title.y = element_text(size =12),
      plot.title = element_text(size = 14),
      axis.text.y=element_blank(),
      axis.ticks.y = element_blank(),
      legend.position = "bottom",
      legend.key.width = unit(2, "cm")
  )
print(est.hov.plot)
```



Notice that missing data are not assigned a colour in the Hövmoller plot. This makes it possible to identify sites that were added after 2001 and also clearly shows that the sampling regime was reduced after 2014.

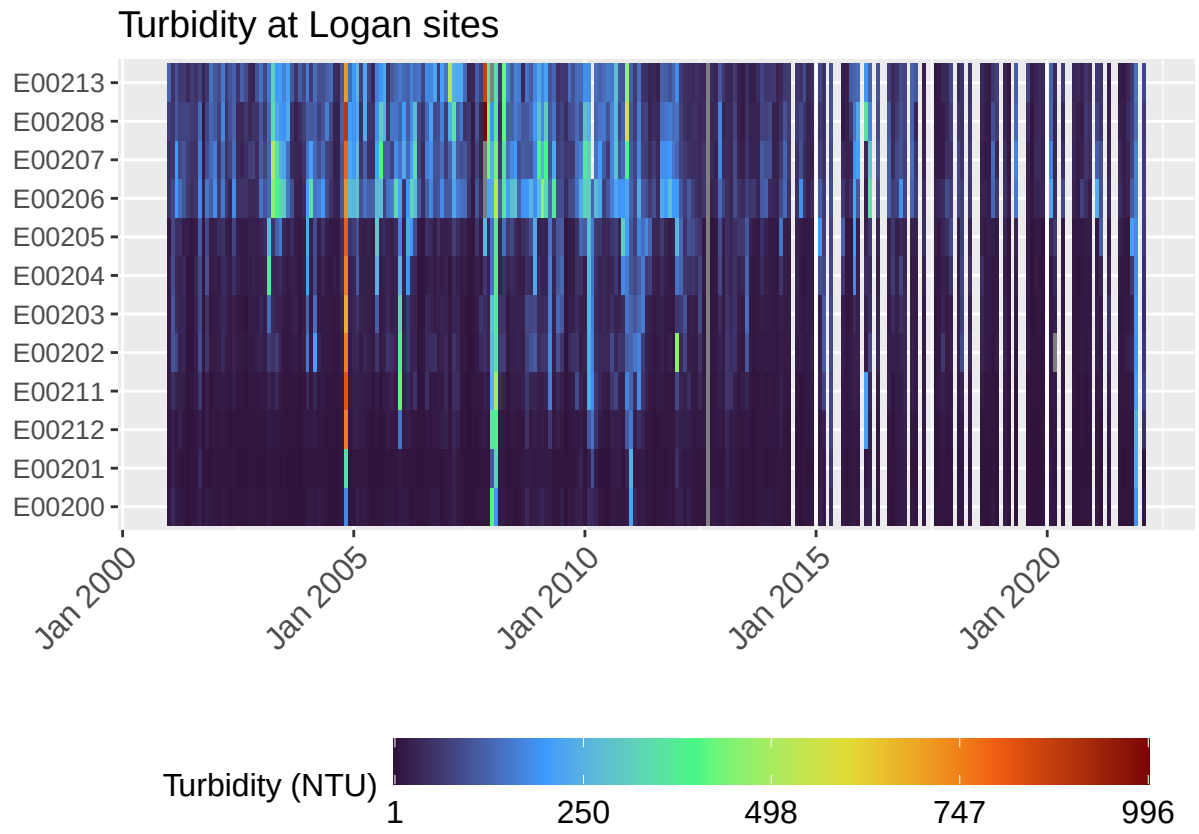
Note that the code `filter(avg.dat, WaterType == "Estuary")` in the previous code block can be changed to `filter(avg.dat, WaterType == "Bay")` to create a data.frame of Bay data. This data.frame can then be used in ggplot to produce a similar plot for Bays.

The Hövmoller plot for turbidity in estuaries clearly suggests that temporal patterns in turbidity at estuaries are similar for some subsets of sites, but different for others. This is not surprising given that subsets of sites in closer proximity to one another are likely subjected to similar rainfall and flow events, that other more distance sites are not affected by. However, it is difficult to interpret these spatio-temporal patterns when large numbers of sites are included in the plot.

A similar approach can be used to create a Hövmoller plot for the sites particular region (e.g. Logan). We also reorder the EHMP sites based on the column `AMTD` to make the plot more interpretable.

```
## Create a data.frame of Logan data
logan.dat <- filter(avg.dat, Waterway == "Logan")

## Drop the SiteCode factor levels not found in Logan and reorder based on AMTD
logan.dat$SiteCode <- droplevels(logan.dat$SiteCode)
logan.dat$SiteCode <- fct_reorder(logan.dat$SiteCode, logan.dat$AMTD)

## Create the plot
logan.hov.plot <- ggplot(data = logan.dat, aes(x = RunYear, y = SiteCode, fill = Turb)) +
  geom_tile() +
```

```
  labs(x = "", y = "", title = "Turbidity at Logan sites", fill = "Turbidity (NTU)") +
  scale_fill_viridis_c(option = "turbo", breaks = equal_breaks(n=5, digits = 0)) +
    theme(legend.text = element_text(size = 12),
          legend.title = element_text(size = 12),
          axis.text.x=element_text(angle=45, hjust =1, size = 12),
          axis.title.x=element_text(size = 12),
          axis.text.y=element_text(size = 10),
          axis.title.y = element_text(size =12),
          plot.title = element_text(size = 14),
        legend.position = "bottom", legend.key.width = unit(2, "cm"))
print(logan.hov.plot)
```



The spatio-temporal patterns in turbidity and it changes in the Logan estuary is more obvious in this plot. Notice that turbidity is much more variable and usually of greater magnitude in the upper estuary (`E00213`, `E00208`, `E00207`, `E00206`) compared to lower in the estuary (`E00212`, `E00201`, `E002112`). However, there are also some major turbidity events that affected all Logan estuary sites during the same sampling Run (e.g. Run 11, 2004 and Run 2, 2008).

Line plots

Line plots are used to show the temporal variability at a single location, or set of locations over time. Next we'll demonstrate how to create three different line plots for

1. A water quality variable at all estuary sites;
2. A water quality variable within one Waterway region; and
3. All water quality variables at a single EHMP site.

To begin, we create a new subset of the original data.frame, `dat` containing water quality data from all of the estuary sites (i.e. no averaging).

```
## Create a data.frame of estuary site data
estuaries.dat <- filter(dat, WaterType == "Estuary")
```

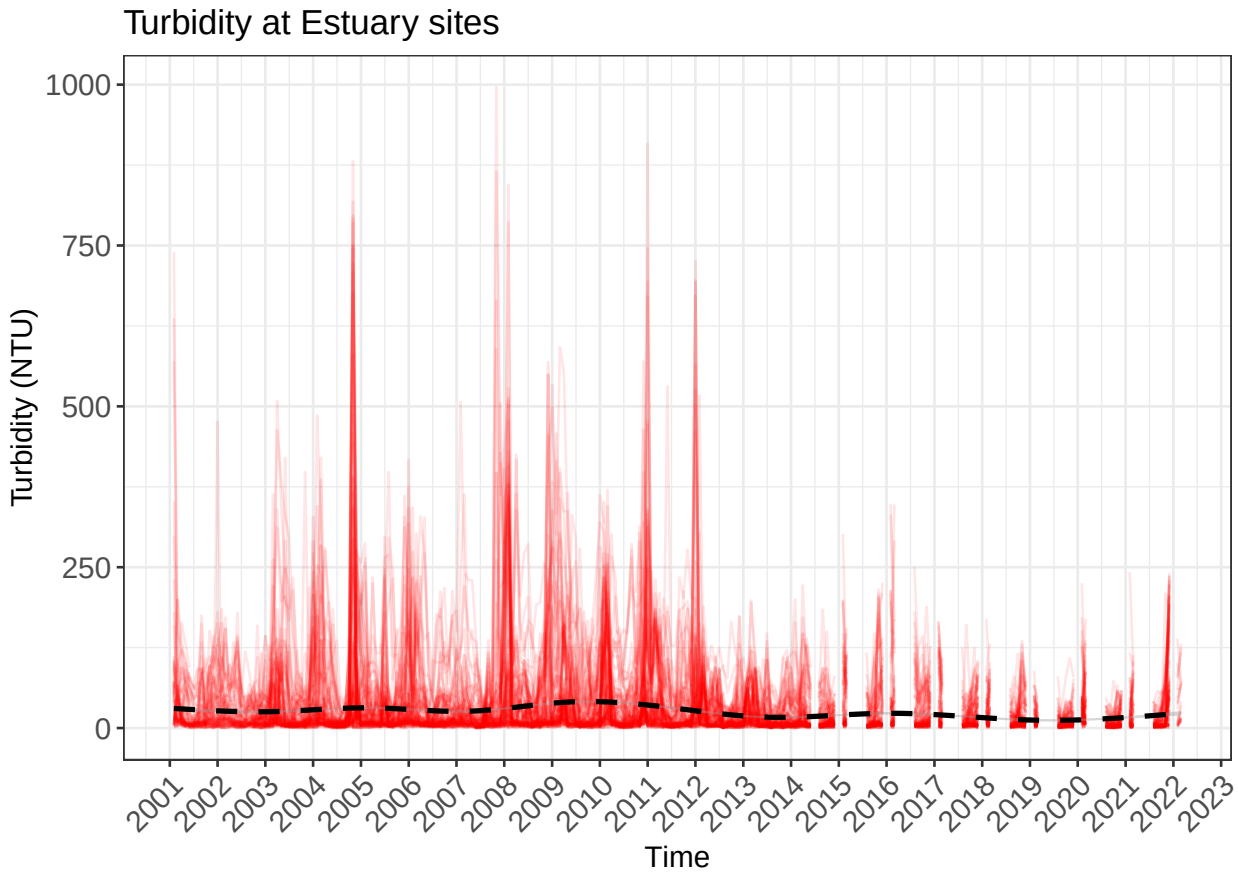
When the A Hövmoller plot was created using `geom_tile`, the function automatically detected when a sampling Run was missing, even when the record was not included in the data.frame. In a line plot, the line simply connects (i.e. interpolates) measurements with the next measurement in time, even if they are months apart. This can be misleading, especially for data collected after 2014 when the EHMP sampling frequency was reduced. Therefore, the data.frame must be expanded to include those missing samples before creating the line plot.

```
## Find first and last sampling date
minDate <- min(estuaries.dat$Date)
maxDate <- max(estuaries.dat$Date)

## Expand the data.frame to include missing RunYear combinations for
## every SiteCode and water quality variable
estuaries.dat<- estuaries.dat %>% complete(Run=seq(1,12, by = 1),
                                           Year = seq(year(minDate),
                                                      year(maxDate), by = 1),
                                           SiteCode = levels(SiteCode))

## Recalculate RunYear column to ensure no missing values
estuaries.dat$RunYear<- as.Date(as.yearmon(paste(estuaries.dat$Year,
                                                 estuaries.dat$Run, sep='-')))

## Remove Runs created before first sampling Run and after last Run
estuaries.dat<- estuaries.dat %>% filter(RunYear >= minDate & RunYear <= maxDate)
```

Next we plot turbidity at all EHMP estuary sites.

```
## Create a line plot of turbity at all estuary sites, with
## RunYear on the x-axis and turbidity on the y-axis
est.line.plot <- ggplot(estuaries.dat, aes(x = RunYear, y = Turb)) +
  geom_line(alpha = 0.1, col = "red", aes(group = SiteCode)) +
  geom_smooth(col = "black", lty = 2) + ## Add a smoothed line to the plot
  theme_bw() +
    scale_x_date(date_breaks = "12 months", date_labels = "%Y") +
  theme(legend.position = "none",
      legend.text = element_text(size = 12),
      legend.title = element_text(size = 12),
      axis.text.x=element_text(angle=45, hjust =1, size = 12),
      axis.title.x=element_text(size = 12),
      axis.text.y=element_text(size = 12),
      axis.title.y = element_text(size =12),
      plot.title = element_text(size = 14),) +
  ggtitle("Turbidity at Estuary sites") +
  xlab("Time") +
  ylab("Turbidity (NTU)")
```

```
print(est.line.plot)
```



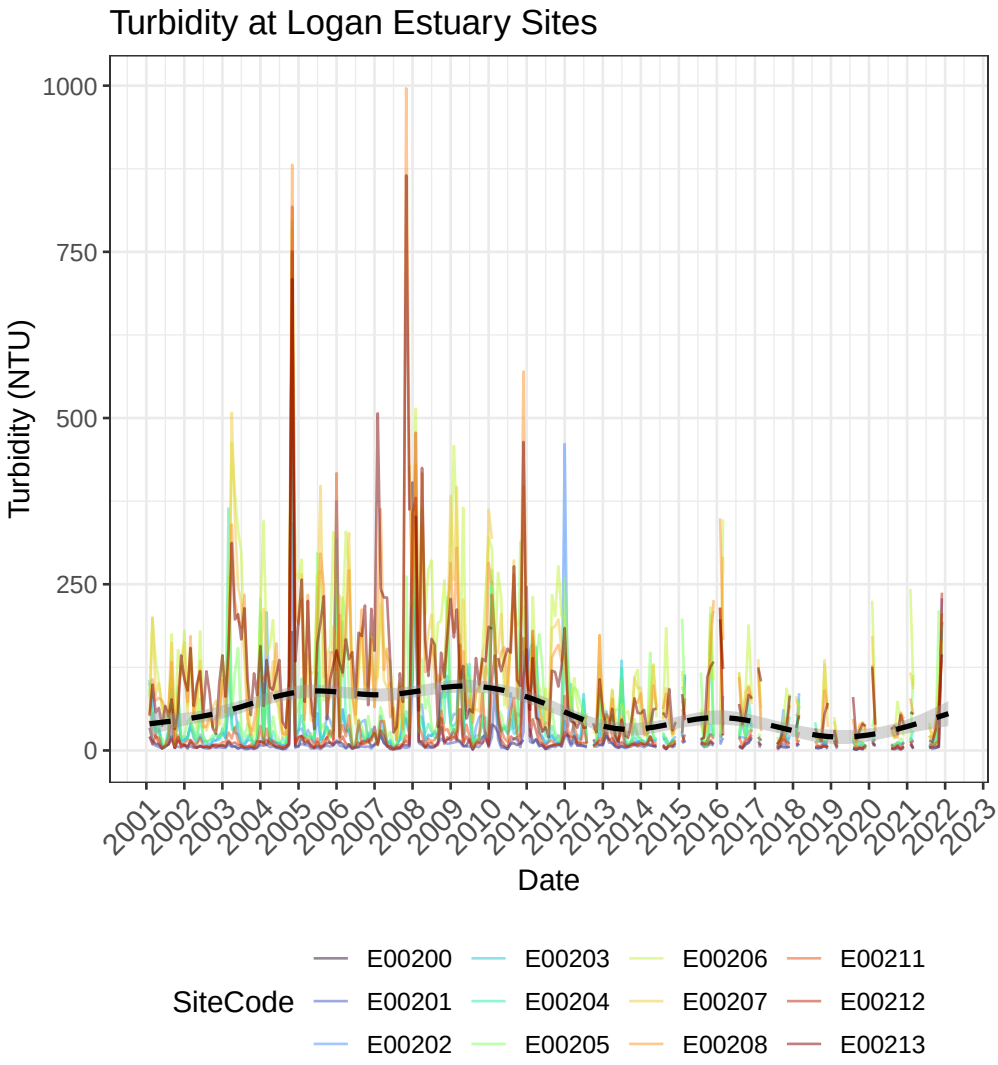
Don't worry if you receive warnings about non-finite values and missing data. We've expanded the data.frame to include these missing data and you can clearly see the gaps in sampling for Runs 1, 4, 6, and 7 after 2014. The plot includes a red line for each SiteCode and a smooth black-dashed line fit to all the turbidity data, which highlights the pattern across all sites. The geom_smooth function controls how the smooth line is fit to the data. Here we use the default method, which is the `gam` function from the package `mgcv` when there are more than 1000 measurements and the `loess` function from the package `stats` otherwise. The help file for the `geom_smooth` function (`help(geom_smooth)`) provides additional information about all of the options that can be used to fit a smooth line to the data.

It's impossible to identify location- or region-specific patterns with so many sites, but the turbidity event observed at Logan estuary sites during Run 11, 2004 is also visible in this line plot.

Next, we take a closer look at turbidity at sites in the Logan Waterway. We start by reformatting the data to make it easier for filtering and plotting.

```
## Filter the dataset to only include turbidity data in the Logan, convert SiteCode
## to factor, and drop SiteCode levels outside of the Logan
logan.dat <- dat %>% filter(Waterway == "Logan")%>%
    mutate(SiteCode = droplevels(as.factor(SiteCode)))

## Find first and last sampling dates in Logan
minDate <- min(logan.dat$Date)
maxDate <- max(logan.dat$Date)
```

```
## Expand the data.frame to include missing RunYear combinations for
## every SiteCode and water quality variable
logan.dat<- logan.dat %>% complete(Run=seq(1,12, by = 1),
                                   Year = seq(year(minDate),
                                              year(maxDate), by = 1),
                                   SiteCode = levels(SiteCode))

## Create a new RunYear column
logan.dat$RunYear<- as.Date(as.yearmon(paste(logan.dat$Year,
                                             logan.dat$Run, sep='-')))

## Remove Runs created before first sampling Run and after last Run
logan.dat<- logan.dat %>% filter(RunYear >= minDate & RunYear <= maxDate)


## Generate a line plot of turbidity at Logan estuary sites
logan.line.plot <- ggplot(logan.dat, aes(x = RunYear, y = Turb, colour = SiteCode)) +
  geom_line(alpha = 0.5) +
  geom_smooth(col = "black", lty = 2) +
  theme_bw() +
  scale_color_viridis_d(option="turbo") +
  scale_x_date(date_breaks = "12 months", date_labels = "%Y") +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 12),
          axis.text.x=element_text(angle=45, hjust =1, size = 12),
          axis.title.x=element_text(size = 12),
          axis.text.y=element_text(size = 10),
          axis.title.y = element_text(size =12),
          plot.title = element_text(size = 14),
          plot.margin = margin(0.5,0.75,0.5,0.75, "in")) +
  ggtitle("Turbidity at Logan Estuary Sites") +
  xlab("Date") +
  ylab("Turbidity (NTU)")
print(logan.line.plot)
```



The line plot contains a separate line for turbidity at each of the 12 EHMP sites in the Logan estuary. The black dotted line is a smooth line fit to the data. The large turbidity events in Run 11, 2004 and Run 2, 2008 are visible here as well. However, the magnitude and variability of turbidity events appears to be greater at most EHMP sites before 2012 than it is between 2013-2022 and the smooth fitted line suggests this is true.

So far, we've looked at turbidity data independently, but visually examining the relationship between water quality variables at a single site can also provide insights into the data. Here we'll use SiteCode E00211 as an example. We already expanded the data to include missing values, but reformatting the data.frame will make plotting multiple variables easier.

```
## Convert the data.frame from wide to long format
logan.long.dat <- logan.dat %>%
    select(SiteCode, Date, RunYear, Waterway, WaterType, AMTD, Temp:FRP) %>%
```

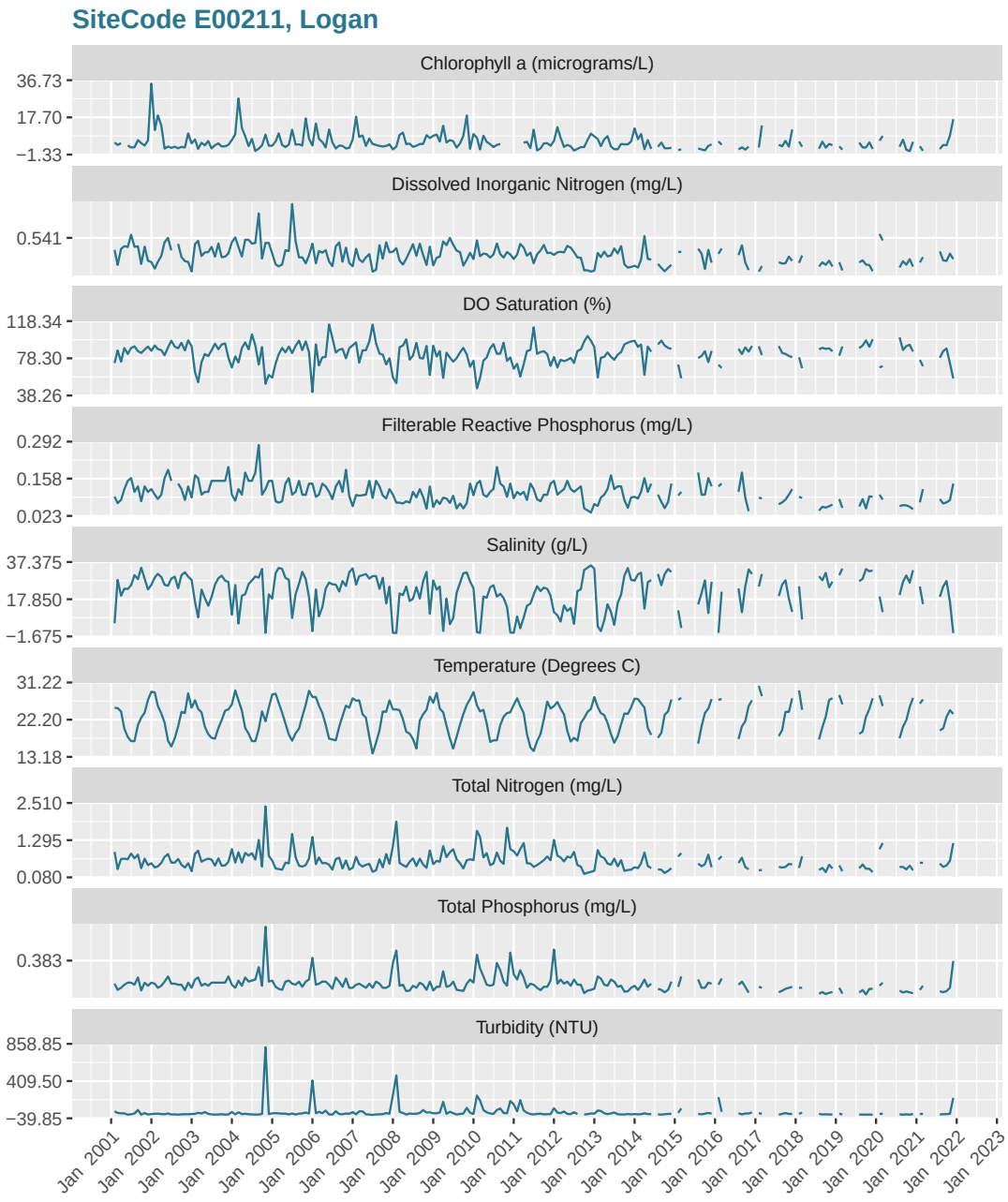
```
    pivot_longer(names_to = "WQ.Var", cols =  Temp:FRP, values_to="Value")

## Create a data.frame of SiteCode E00211 data only
logan.E00211 <- logan.long.dat %>% filter(SiteCode == "E00211")

## Convert Variable to factor format
logan.E00211$WQ.Var<- as.factor(logan.E00211$WQ.Var)


## View the top few rows of the new data.frame
head(logan.E00211)
```

```
## # A tibble: 6 x 8
##   SiteCode Date       RunYear    Waterway WaterType  AMTD WQ.Var Value
##   <chr>    <date>     <date>     <fct>    <fct>     <dbl> <fct>  <dbl>
## 1 E00211   2002-01-10 2002-01-01 Logan    Estuary     7.8 Temp   29
## 2 E00211   2002-01-10 2002-01-01 Logan    Estuary     7.8 Sal    25.4
## 3 E00211   2002-01-10 2002-01-01 Logan    Estuary     7.8 Turb   11
## 4 E00211   2002-01-10 2002-01-01 Logan    Estuary     7.8 DOSat  86.3
## 5 E00211   2002-01-10 2002-01-01 Logan    Estuary     7.8 Chla   35
## 6 E00211   2002-01-10 2002-01-01 Logan    Estuary     7.8 TN      0.54
```

Notice the new long data.frame format, which makes it easier to filter and plot. It contains a row for each unique SiteCode, Run, Year, water quality variable (`WQ.Var`).

Now we can create a plot for EHMP SiteCode `E00211`.

```
## Set SiteCode and Waterway variables:
sc <- logan.E00211$SiteCode[1]
ww <- logan.E00211$Waterway[1]

site.plot <- ggplot(data = logan.E00211,
                    aes(x = RunYear, y = Value, group = WQ.Var)) +
    labs(x = "", y = "", caption = "",
         title =paste0("SiteCode ",sc, ", ", ww)) +
    geom_line(color = "#2A768EFF") +
    scale_x_date(date_breaks = "12 months", date_labels = "%b  %Y") +
    scale_y_continuous(breaks=equal_breaks())+
    theme(axis.text.x=element_text(angle=45, hjust =1),
          legend.position="none",
          plot.title = element_text(color = "#2A768EFF", face="bold"),
          plot.margin = margin(0.75,0.5,0.75,0.5, "in"))+
    facet_wrap(~WQ.Var, scales = "free_y", ncol = 1,
               labeller= as_labeller(c("Temp"="Temperature (Degrees C)",
                                       "Sal"="Salinity (g/L)",
                                       "Turb"="Turbidity (NTU)",
                                       "DOSat"="DO Saturation (%)",
                                       "Chla"= "Chlorophyll a (micrograms/L)",
                                       "TN"= "Total Nitrogen (mg/L)",
                                       "DIN"="Dissolved Inorganic Nitrogen (mg/L)",
                                       "TP" = "Total Phosphorus (mg/L)",
                                       "FRP"= "Filterable Reactive Phosphorus (mg/L)")))
print(site.plot)
```



The turbidity event in Run 11, 2004 is clearly visible in the plot, along with elevated TN and TP concentrations and a decrease in DO saturation and salinity.

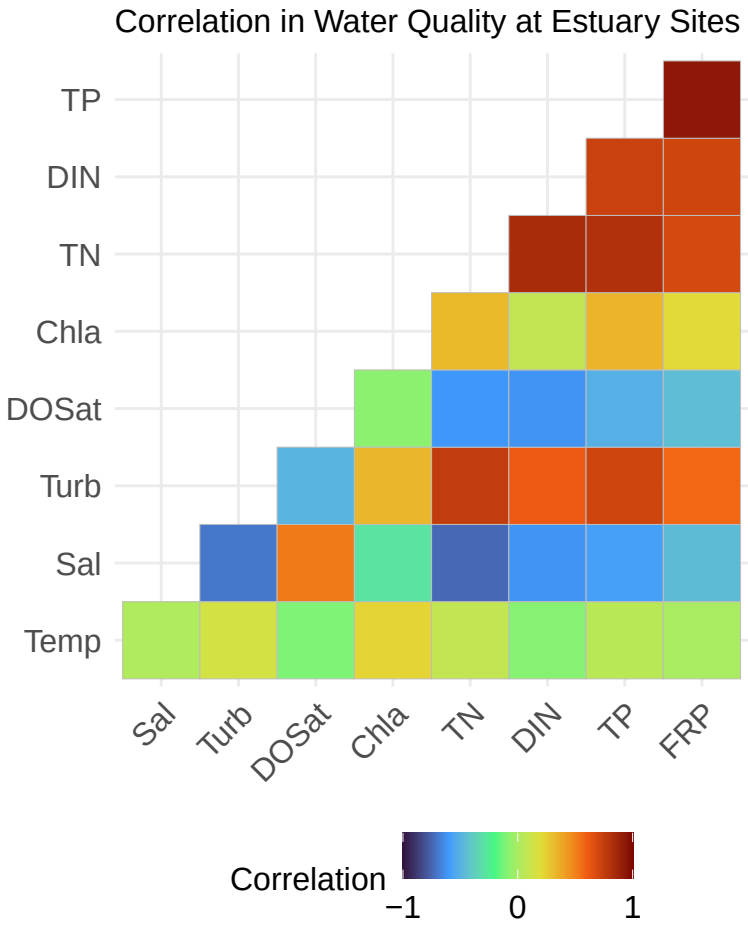
Correlation plots

Correlation plots are used to explore the relationship between pairs of variables. The first step is to generate the correlation matrix for the water quality variables.

```
## Generate a Spearman's Rank Correlation matrix for water quality
## variables at estuary sites
est.wq.cor <- dat %>%
  filter(WaterType == "Estuary") %>%
  select(Temp:FRP) %>%
  cor(method = "spearman", use= "complete.obs")
```

The following code converts the correlation matrix into a plot.

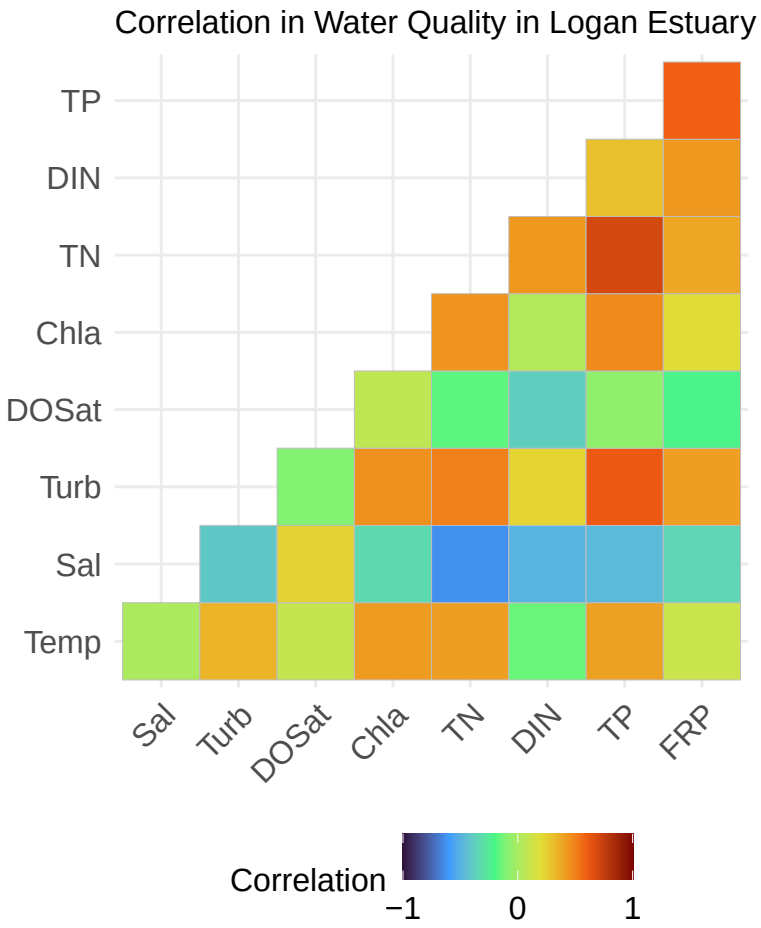
```
est.cor.plot <- ggcorrplot(est.wq.cor, type = "lower")+
    scale_fill_viridis_c(option="turbo",  ## Use the turbo colour scheme
                         breaks = equal_breaks(), ## Set 3 equal breaks
                         limits = c(-1, 1), ## Set legend colour bar limits
                         name = "Correlation") + ## Legend Label
    theme(legend.position = "bottom", ## Set some text parameters
          legend.text = element_text(size = 12),
          legend.title = element_text(size = 12),
          axis.text.x=element_text(angle=45, hjust =1, size = 12),
          axis.text.y=element_text(size = 12),
          plot.title = element_text(size = 12)) +
    ggtitle("Correlation in Water Quality at Estuary Sites") ## Title

print(est.cor.plot)
```



Not surprisingly, there is a relatively strong correlation between the nutrient variables and turbidity. As expected, there is also a negative correlation with salinity and most other water quality variables, with the exception of DO saturation. This suggests that freshwater inputs that decrease salinity are also associated with increases in nutrients and sediments. However, these relationships may differ depending on the the water type (e.g. estuary or bay) or the region. Similar code can be used to produce a correlation plot for sites within the Logan estuary.

```
## Create the correlation matrix based on Logan data only
logan.wq.cor<- dat %>%
  filter(Waterway == "Coomera") %>%
  select(Temp:FRP) %>%
     cor(method = "spearman", use = "complete.obs")


logan.cor.plot <- ggcorrplot(logan.wq.cor, type = "lower") +
    scale_fill_viridis_c(option="turbo",  ## Use the turbo colour scheme
                         breaks = equal_breaks(), ## Set 3 equal breaks
                         limits = c(-1, 1), ## Set legend colour bar limits
                         name = "Correlation") + ## Legend Label
    theme(legend.position = "bottom", ## Set some text parameters
          legend.text = element_text(size = 12),
          legend.title = element_text(size = 12),
          axis.text.x=element_text(angle=45, hjust =1, size = 12),
```

```
          axis.text.y=element_text(size = 12),
          plot.title = element_text(size = 12)) +
    ggtitle("Correlation in Water Quality in Logan Estuary") ## Title

print(logan.cor.plot)
```



The relationships between water quality variables in the Logan estuary differ from those calculated from the full set of estuary data. Notice that the positive relationship between DIN and other nutrients and turbidity is weaker. This is also true of the negative relationship between salinity and the other water quality variables, with the exception of TN.

Trend Analyses

Many methods can be used for trend assessment (Morton and Henderson 2008; Meals et al. 2011; Beck et al. 2022), but here we use the seasonal Kendall trend (SKT) test (Hirsch and Slack 1982) for a number of reasons. First, it is a non-parametric approach, meaning that there are no assumptions about the distribution of the data. It's also a relatively simple method that can be used to account for seasonality in water quality when testing for trend. Missing values can be accommodated without the need for interpolation or imputation before undertaking the trend assessment. In addition, preliminary assessments of the test statistic and diagnostics are relatively simple and there is no need for model selection, which reduces the time it takes to run and interpret large numbers of tests.

It's helpful to have a general understanding about how the SKT test works in practice before implementing the methods using the `kendallSeasonalTrendTest` function in the `EnvStats` package (Millard 2013a). The SKT test differs from the Mann-Kendall trend test (Mann 1945) because it accounts for seasonality in the data (Figure 1, Step 1) when testing for trend (Millard 2013b). This is accomplished by performing a non-seasonal Mann-Kendall test on data *within* season (i.e. sampling run). In other words, trends are calculated for each sampling run independently from other runs, resulting in season-specific estimates of Kendall's τ (i.e. Kendall rank correlation coefficient), a slope parameter estimate, and intercept (Figure 1, Step 2). The SKT test is only appropriate if the sign (-/+) of all the non-zero season-specific trends in Step 2 (Figure 1) are in the same direction and the van Belle Hughes test (van Belle and Hughes 1984) is used to test this assumption (Figure 1, Step 3). If the results of the van Belle Hughes test suggest that there is evidence of heterogeneous trends, the SKT test is not an appropriate method. If there is no evidence of heterogeneous trends, the SKT test can be used to combine all of the season-specific τ, intercept, and slope estimates to obtain information about the overall trends (Figure 1, Step 4).

Although the SKT is a sensible approach for preliminary trend assessment, it has a number of important limitations to keep in mind. First, the SKT test is designed to test for monotonic trends (i.e. constant linear increase or decrease). If the result of the van Belle Hughes test indicate that the data don't meet this assumption, the results will likely be misleading, and an alternative method must be used (Morton and Henderson 2008; Meals et al. 2011; Beck et al. 2022). Second, the user-assigned season is a relatively crude way to describe seasonality and may be insufficient for some variables. For example, here we use sampling run as the "season" (1-12). For some variables, sampling run is likely acting as a surrogate for seasonal climate variability in rainfall and flow. This may be adequate when rainfall and corresponding flood events occur during the same wet months, but it will not describe the effects of out-of-season flood events on water quality. Finally, the SKT test is based on the sign (-/=/+) of the difference between data at each time step and future time steps (e.g. 2017 Run 11 versus 2018-2022 Run 11), rather than the actual values, which leads to a loss of quantitative information about the magnitude of change. Therefore, the SKT should always be evaluated and interpreted in the context of other information, such as exploratory plots and summary statistics. When a positive trend is identified, it is important to check whether the results make sense given the data. When a trend was not detected or heterogeneous trends are identified, the exploratory plots and raw data may provide insights about the reasons why.

A more detailed description of the SKT test is outside the scope of this tutorial. Please see Hirsch and Slack (1982) for additional details about the method. The authors of the `EnvStats` package have also produced a plethora of supporting material, including a companion textbook (Millard et al. 2014) and a package manual in book form (Millard 2013a), both of which include R code. The help files for the `kendallTrendTest` and the `kendallSeasonalTrendTest` functions also provide a surprising level of detail about how the methods are implmented within the `EnvStats` package.

```
## View the help file
help(kendallSeasonalTrendTest)
```

We begin by setting up the data for a long-term trend assessment.

```
## Create a vector of water quality variables to test
wq.vars<- c("Temp","Sal","Turb","DOSat","Chla", "TN","DIN","TP","FRP")

## Convert the data.frame to long format. cols = WQ variables to test,
## WQ.Var = water quality variable names, Value = value of WQ.Var
dat.long <- dat %>%
    pivot_longer(cols = all_of(wq.vars), names_to = "WQ.Var", values_to = "Value") %>%
    select(-Basinme, -Month) %>%
    modify_if(is.character, as.factor)
```

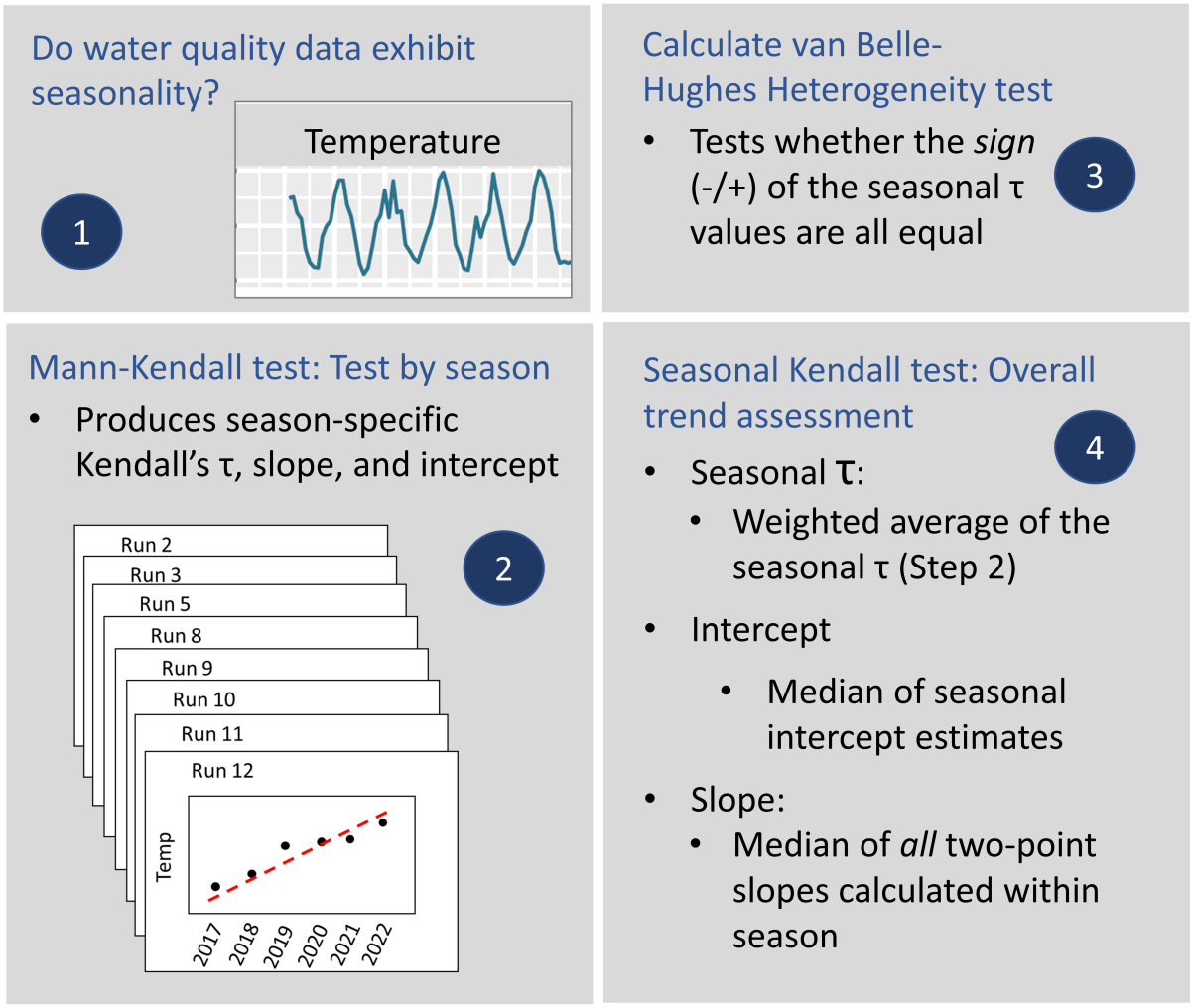


Figure 1: Seasonal Kendall trend test analysis steps

```
## Convert Run number to a factor
dat.long$Run <- as.factor(dat.long$Run)
```

For demonstration purposes, we'll undertake the long-term trend assessment of turbidity at one site in the Logan, E00213.

```
## Filter the data.frame so that it only contains turbidity data from
## SiteCode E00213
lt.E00213 <- filter(dat.long, SiteCode %in% 'E00213' & WQ.Var %in% "Turb")
summary(lt.E00213[,c("SiteCode", "WQ.Var", "Run", "Year", "Value")])
```

```
##     SiteCode       WQ.Var         Run           Year          Value
##  E00213 :223   Turb   :223   2      :22   Min.   :2001   Min.   :  9.1
##  E00100 :  0   Chla   :  0   5      :21   1st Qu.:2005   1st Qu.: 42.8
##  E00101 :  0   DIN    :  0   8      :21   Median :2010   Median : 81.0
##  E00103 :  0   DOSat  :  0   9      :21   Mean   :2010   Mean   :106.4
##  E00104 :  0   FRP    :  0   10     :21   3rd Qu.:2015   3rd Qu.:137.0
##  E00105 :  0   Sal    :  0   11     :21   Max.   :2022   Max.   :865.0
##  (Other):  0   (Other):  0   (Other):96                  NA's   :2
```

The arguments for the `kendallSeasonalTrendTest` make it a flexible test, suitable for many different data scenarios. Here, formula `Value~ Run + Year` and `data = dat.E00213` are used to define the columns and data.frame used in the trend test. `Value` is the name of the column where the turbidity data are stored and `Run + Year` is the user-defined season and year. We use the default values for the arguments `correct`, `conf.level`, and `independent.obs`, but they are explicitly defined here for transparency. Setting `correct = TRUE` indicates that the correction for continuity in computing the z-statistic should be used, which is appropriate when there are "ties" (i.e. no change in value between time steps) in the data. We specifiy that a two-sided 95% confidence interval should be calculated for the slope estimate with `ci.slope = TRUE` and `conf.level = 0.95`. The function can be used to undertake trend assessment for independent or serially correlated data. Here we set independent.obs = `TRUE` because we believe that water quality data collected in the same samplng Run, but different years, are independent of one another.

```
## Run the seasonal Kendall trend test
skt.lt.E00213<- kendallSeasonalTrendTest(Value ~ Run + Year,
                                data = lt.E00213,
                                correct = TRUE,
                                ci.slope = TRUE,
                                conf.level = 0.95,
                                independent.obs = TRUE)

## Examine the results
print(skt.lt.E00213)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Null Hypothesis:                 All 12 values of tau = 0
##
## Alternative Hypothesis:          The seasonal taus are not all equal
##                                  (Chi-Square Heterogeneity Test)
```

```
##                                 At least one seasonal tau != 0
##                                 and all non-zero tau's have the
##                                 same sign (z Trend Test)
##
## Test Name:                      Seasonal Kendall Test for Trend
##                                 (with continuity correction)
##
## Estimated Parameter(s):         tau       =   -0.3084
##                                 slope     =   -3.7375
##                                 intercept = 8963.6758
##
## Estimation Method:              tau:        Weighted Average of
##                                             Seasonal Estimates
##                                 slope:      Hirsch et al.'s
##                                             Modification of
##                                             Thiel/Sen Estimator
##                                 intercept:  Median of
##                                             Seasonal Estimates
##
## Data:                           y      = Value
##                                 season = Run
##                                 year   = Year
##
## Data Source:                    lt.E00213
##
## Sample Sizes:                   1     =  13
##                                 2     =  22
##                                 3     =  20
##                                 4     =  14
##                                 5     =  21
##                                 6     =  14
##                                 7     =  13
##                                 8     =  21
##                                 9     =  20
##                                 10    =  21
##                                 11    =  21
##                                 12    =  21
##                                 Total = 221
##
## Number NA/NaN/Inf's:            2
##
## Test Statistics:                Chi-Square (Het) =  5.484105
##                                 z (Trend)        = -6.340073
##
## Test Statistic Parameter:       df = 11
##
## P-values:                       Chi-Square (Het) = 9.054788e-01
##                                 z (Trend)        = 2.296564e-10
##
## Confidence Interval for:        slope
##
## Confidence Interval Method:     Gilbert's Modification of
##                                 Theil/Sen Method
##
```

```
## Confidence Interval Type:        two-sided
##
## Confidence Level:                95%
##
## Confidence Interval:             LCL = -5.126527
##                                  UCL = -2.500000
```

The output of the `kendallSeasonalTrendTest` contains a lot of information, but the key pieces described in Figure 1 include the:

- `$statistic Chi-Square (Het)`: van Belle Hughes test statistic
- `$p.value Chi-Square (Het)`: p-value for van Belle Hughes test
- `$statistic z (Trend)`: seasonal Kendall test statistic
- `$p-value z (Trend)`: p-value for the seasonal Kendall's test
- `$estimate slope`: Slope parameter estimate
- `$limits LCL and UCL`: lower and upper limits for the two-sided 95% confidence interval for the slope

In this case, the p-value for the van Belle Hughes test is much larger than 0.1 ($\alpha = 0.905$), leading us to conclude that the signs (-/=/+) of the seasonal trends are not heterogeneous and that the SKT test is appropriate (Figure 1, Step 3). Here we've chosen to use $\alpha < 0.1$ as a threshold because it is a moderately conservative value and we want to ensure that the assumptions of the SKT test are not violated. However, choosing a threshold is somewhat subjective and other values could be used. The overall slope estimate is -3.7373, the lower (-5.1265) and upper (-2.50) confidence intervals for the slope do not contain 0, and the p-value for the SKT test is < 0.0001 (Figure 1, Step 4). This suggests that there is a statistically significant, long-term negative trend in turbidity at SiteCode E00213 between 2001-2022, and that the estimated annual trend is -3.7373 NTU/year.

The output of the `kendallSeasonalTrendTest`function also includes information about the Mann-Kendall trend tests for each season (Figure 1, Step 2). Notice that the Kendall S-statistics for each season, `$seasonal.S`, and the seasonal slope estimates found in `$seasonal.estimates` are all relatively large and negative.

To calculate the trend over a different time period, we simply subset the dataset based on `Date`. Here we'll calculate the short-term trend based on the last 5 years of data.

```
## Create a short-term dataset for trend assessment

## Find the maximum RunYear in the data.frame and subtract 5
## years to get the minimum RunYear to include
minRY <- max(dat.long$RunYear)-5

## Create a data.frame of water quality data for the last
## 5 years
dat.short <- dat.long %>% filter(RunYear > minRY)

st.E00213 <- filter(dat.short, SiteCode %in% "E00213" & WQ.Var %in% "Turb")

## Run the seasonal Kendall trend test for short-term trend
skt.st.E00213<- kendallSeasonalTrendTest(Value ~ Run + Year,
                                data = st.E00213,
                                correct = TRUE,
                                ci.slope = TRUE,
                                conf.level = 0.95,
                                independent.obs = TRUE)

## Examine the results
```

```
print(skt.st.E00213)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Null Hypothesis:                 All 8 values of tau = 0
##
## Alternative Hypothesis:          The seasonal taus are not all equal
##                                  (Chi-Square Heterogeneity Test)
##                                  At least one seasonal tau != 0
##                                  and all non-zero tau's have the
##                                  same sign (z Trend Test)
##
## Test Name:                       Seasonal Kendall Test for Trend
##                                  (with continuity correction)
##
## Estimated Parameter(s):          tau       =     0.1367521
##                                  slope     =     2.1666667
##                                  intercept = -3884.0312500
##
## Estimation Method:               tau:        Weighted Average of
##                                              Seasonal Estimates
##                                  slope:      Hirsch et al.'s
##                                              Modification of
##                                              Thiel/Sen Estimator
##                                  intercept:  Median of
##                                              Seasonal Estimates
##
## Data:                            y      = Value
##                                  season = Run
##                                  year   = Year
##
## Data Source:                     st.E00213
##
## Sample Sizes:                    5     =  5
##                                  8     =  5
##                                  9     =  5
##                                  10    =  5
##                                  11    =  5
##                                  12    =  5
##                                  2     =  5
##                                  3     =  4
##                                  Total = 39
##
## Test Statistics:                 Chi-Square (Het) = 2.9510260
##                                  z (Trend)        = 0.8039133
##
## Test Statistic Parameter:        df = 7
##
## P-values:                        Chi-Square (Het) = 0.8894954
##                                  z (Trend)        = 0.4214470
##
## Confidence Interval for:         slope
```

```
##
## Confidence Interval Method:      Gilbert's Modification of
##                                  Theil/Sen Method
##
## Confidence Interval Type:        two-sided
##
## Confidence Level:                95%
##
## Confidence Interval:             LCL = -1.776905
##                                  UCL =  7.792302
```

Again, the p-value for the van Belle Hughes test is > 0.1, which suggests that the assumption of homogeneous trends has not been violated. However, the p-value for the SKT test (0.42) is not significant at $\alpha = 0.05$ and the confidence limits for the slope (-1.77, 7.79) include 0. This indicates that there is no evidence of a statistically significant short-term trend in turbidity at EHMP site E00213 based on the SKT test.

Let's look at another example, this time for the long-term trend in turbidity at EHMP SiteCode E00204.

```
## Filter the data
lt.E00204 <- filter(dat.long, SiteCode %in% "E00204" & WQ.Var %in% "Turb")


## Run the seasonal Kendall trend test
skt.lt.E00204<- kendallSeasonalTrendTest(Value ~ Run + Year,
                                data = lt.E00204,
                                correct = TRUE,
                                ci.slope = TRUE,
                                conf.level = 0.95,
                                independent.obs = TRUE)

## Examine the results
print(skt.lt.E00204)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Null Hypothesis:                 All 12 values of tau = 0
##
## Alternative Hypothesis:          The seasonal taus are not all equal
##                                  (Chi-Square Heterogeneity Test)
##                                  At least one seasonal tau != 0
##                                  and all non-zero tau's have the
##                                  same sign (z Trend Test)
##
## Test Name:                       Seasonal Kendall Test for Trend
##                                  (with continuity correction)
##
## Estimated Parameter(s):          tau       =  -0.05839245
##                                  slope     =  -0.34722222
##                                  intercept = 376.37152778
##
## Estimation Method:               tau:        Weighted Average of
##                                              Seasonal Estimates
```

```
##                                  slope:      Hirsch et al.'s
##                                              Modification of
##                                              Thiel/Sen Estimator
##                                  intercept:  Median of
##                                              Seasonal Estimates
##
## Data:                            y      = Value
##                                  season = Run
##                                  year   = Year
##
## Data Source:                     lt.E00204
##
## Sample Sizes:                    1     =  14
##                                  2     =  22
##                                  3     =  21
##                                  4     =  14
##                                  5     =  21
##                                  6     =  14
##                                  7     =  13
##                                  8     =  21
##                                  9     =  20
##                                  10    =  21
##                                  11    =  21
##                                  12    =  21
##                                  Total = 223
##
## Number NA/NaN/Inf's:             1
##
## Test Statistics:                 Chi-Square (Het) = 24.11895
##                                  z (Trend)        = -1.79650
##
## Test Statistic Parameter:        df = 11
##
## P-values:                        Chi-Square (Het) = 0.01224094
##                                  z (Trend)        = 0.07241500
##
## Confidence Interval for:         slope
##
## Confidence Interval Method:      Gilbert's Modification of
##                                  Theil/Sen Method
##
## Confidence Interval Type:        two-sided
##
## Confidence Level:                95%
##
## Confidence Interval:             LCL = -0.660687672
##                                  UCL =  0.001575915
```

In this case, the p-value for the van Belle Hughes test ($\alpha = 0.012$) is less than 0.1 and so we reject the null hypothesis that the seasonal trends are homogeneous. The results of the SKT test are reported in the output, but should be ignored because the assumptions of the test have been violated and no conclusions about the trend can be made.

Statistics are easy to interpret when they all support the same conclusion, as was the case in the previous

three examples. Next we'll focus on FRP at EHMP site E01609, where the results and conclusions are more nuanced.

```
## Filter the data
lt.E01609 <- filter(dat.long, SiteCode %in% "E01609" & WQ.Var %in% "FRP")

## Run the seasonal Kendall trend test
skt.lt.E01609<- kendallSeasonalTrendTest(Value ~ Run + Year,
                                data = lt.E01609,
                                correct = TRUE,
                                ci.slope = TRUE,
                                conf.level = 0.95,
                                independent.obs = TRUE)

## Examine the results
print(skt.lt.E01609)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Null Hypothesis:                 All 12 values of tau = 0
##
## Alternative Hypothesis:          The seasonal taus are not all equal
##                                  (Chi-Square Heterogeneity Test)
##                                  At least one seasonal tau != 0
##                                  and all non-zero tau's have the
##                                  same sign (z Trend Test)
##
## Test Name:                       Seasonal Kendall Test for Trend
##                                  (with continuity correction)
##
## Estimated Parameter(s):          tau       = 0.1741711
##                                  slope     = 0.0000000
##                                  intercept = 0.0010000
##
## Estimation Method:               tau:        Weighted Average of
##                                              Seasonal Estimates
##                                  slope:      Hirsch et al.'s
##                                              Modification of
##                                              Thiel/Sen Estimator
##                                  intercept:  Median of
##                                              Seasonal Estimates
##
## Data:                            y      = Value
##                                  season = Run
##                                  year   = Year
##
## Data Source:                     lt.E01609
##
## Sample Sizes:                    1     =  13
##                                  2     =  21
##                                  3     =  21
##                                  4     =  13
```

```
##                                  5     =  19
##                                  6     =  12
##                                  7     =  13
##                                  8     =  21
##                                  9     =  20
##                                  10    =  21
##                                  11    =  21
##                                  12    =  20
##                                  Total = 215
##
## Number NA/NaN/Inf's:             10
##
## Test Statistics:                 Chi-Square (Het) =      NaN
##                                  z (Trend)        = 4.843208
##
## Test Statistic Parameter:        df = 11
##
## P-values:                        Chi-Square (Het) =          NaN
##                                  z (Trend)        = 1.277597e-06
##
## Confidence Interval for:         slope
##
## Confidence Interval Method:      Gilbert's Modification of
##                                  Theil/Sen Method
##
## Confidence Interval Type:        two-sided
##
## Confidence Level:                95%
##
## Confidence Interval:             LCL = 0
##                                  UCL = 0
```

The first thing to notice is that the the van Belle Hughes test is returning an `NAN`. Why would this occur? The first step is go back and look at the data.

```
## Summary statistics for FRP data @ site E01609
summary(lt.E01609$Value)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## 0.001000 0.001000 0.001000 0.001586 0.001000 0.009000       10
```

```
## Summary statistics for FRP @ all estuary sites
summary(estuaries.dat$FRP)
```

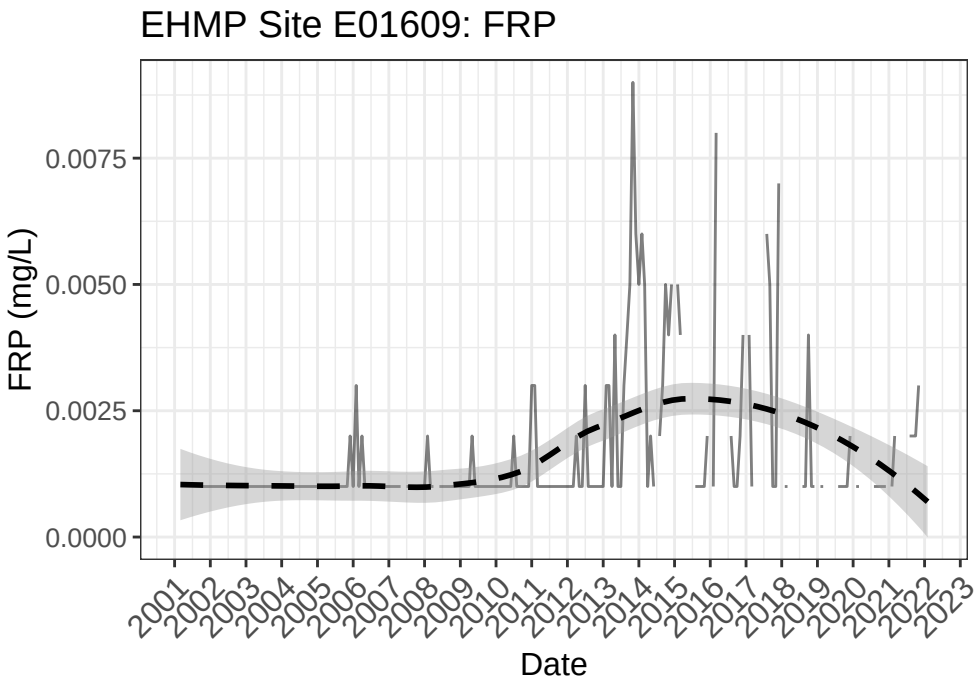
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.001   0.006   0.024   0.112   0.140   5.100   18855
```

The summary statistics for FRP (i.e. Value) at site E01609 show that the majority of the FRP values are 0.0001, which is the value assigned when FRP was below the detection limit. In other words, there doesn't appear to be a lot of variability in the data. These values are also quite small compared to FRP measured at all estuary sites (0.001-5.1).

Next we create a line plot to see how FRP varies over time. Notice that the R code used to create the FRP line plot is the same as that used to create the line plot of turbidity at Logan estuary sites above. The

only differences are that we've changed the axis labels and title, and we've removed the `colour=SiteCode` argument in `aes()` because we're only plotting data from a single site.

```
## Create a data.frame for SiteCode E01609. Recall that estuaries.dat
## was previously expanded to include missing sampling runs, which is
## important for line plots
lt.E01609.exp <- estuaries.dat %>% filter(SiteCode %in% "E01609")

## Generate a line plot of FRP.
frp.line.plot <- ggplot(lt.E01609.exp, aes(x = RunYear, y = FRP)) +
  geom_line(alpha = 0.5) +
  geom_smooth(col = "black", lty = 2) +
  theme_bw() +
  scale_color_viridis_d(option="turbo") +
  scale_x_date(date_breaks = "12 months", date_labels = "%Y") +
  theme(legend.position = "bottom",
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 12),
          axis.text.x=element_text(angle=45, hjust =1, size = 12),
          axis.title.x=element_text(size = 12),
          axis.text.y=element_text(size = 10),
          axis.title.y = element_text(size =12),
          plot.title = element_text(size = 14),
          plot.margin = margin(0.5,0.75,0.5,0.75, "in")) +
  ggtitle("EHMP Site E01609: FRP") +
  xlab("Date") +
  ylab("FRP (mg/L)")
print(frp.line.plot)
```



The line plot clearly shows that there was no variability in FRP values before 2006 and also that there are other periods between 2006 and 2022 with no variability. However, the SKT test is based on within season Mann-Kendall test results and it is difficult to compare seasonal variability (i.e. within Run) in this plot. Therefore, we create a simple summary statistics table to look at variability in FRP by season.

```
## Create summary statistics table by sampling Run
frp.run.sum <- lt.E01609 %>% group_by(Run) %>%
    mutate(Min = min(Value, na.rm = TRUE),
           Med= median(Value, na.rm=TRUE),
           Max = max(Value, na.rm=TRUE),
           Var = var(Value, na.rm = TRUE)) %>%
    distinct(SiteCode, Run, Min, Med, Max, Var)

frp.run.sum
```

```
## # A tibble: 12 x 6
## # Groups:   Run [12]
##    SiteCode Run     Min   Med   Max         Var
##    <fct>    <fct> <dbl> <dbl> <dbl>       <dbl>
##  1 E01609   1     0.001 0.001 0.005 0.00000144
##  2 E01609   2     0.001 0.001 0.006 0.00000229
##  3 E01609   3     0.001 0.001 0.008 0.00000326
##  4 E01609   4     0.001 0.001 0.002 0.000000141
##  5 E01609   5     0.001 0.001 0.007 0.00000225
##  6 E01609   6     0.001 0.001 0.001 0
##  7 E01609   7     0.001 0.001 0.003 0.000000359
```

```
##  8 E01609   8     0.001 0.001 0.006 0.00000136
##  9 E01609   9     0.001 0.001 0.005 0.00000132
## 10 E01609   10    0.001 0.001 0.005 0.00000176
## 11 E01609   11    0.001 0.001 0.009 0.00000343
## 12 E01609   12    0.001 0.001 0.007 0.00000352
```

Now we can clearly see that the variance in FRP values during Run 6 is 0 and there is little variability in other Runs. The `kendallSeasonalTrendTest` will return a `NaN` value if the variance is 0 in at least one season. This is purely an artifact in the way the test is implemented in the `EnvStats` package. Adding a minute amount of noise to one observation (1/1000000 of the minimum) tricks the function into returning a value for the van Belle Hughes test staatistic and p-value, with almost no influence on the overall results.

```
## Define noise as 1/1M of the minimum FRP value at site E01609
noise <- min(lt.E01609$Value, na.rm = TRUE)/1000000

## Add noise to one randomly selected observation in Run 6
ind <- lt.E01609$Run %in% "6"
rn <- sample(1:sum(ind), 1, replace = F)
r <- which(ind)
lt.E01609$Value[r[rn]] <-  lt.E01609$Value[r[rn]]+noise
rm(ind, rn, r)

## Re-run the seasonal Kendall trend test
skt.lt.E01609<- kendallSeasonalTrendTest(Value ~ Run + Year,
                                data = lt.E01609,
                                correct = TRUE,
                                ci.slope = TRUE,
                                conf.level = 0.95,
                                independent.obs = TRUE)

## Examine the results
print(skt.lt.E01609)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Null Hypothesis:                 All 12 values of tau = 0
##
## Alternative Hypothesis:          The seasonal taus are not all equal
##                                  (Chi-Square Heterogeneity Test)
##                                  At least one seasonal tau != 0
##                                  and all non-zero tau's have the
##                                  same sign (z Trend Test)
##
## Test Name:                       Seasonal Kendall Test for Trend
##                                  (with continuity correction)
##
## Estimated Parameter(s):          tau       = 0.1716341
##                                  slope     = 0.0000000
##                                  intercept = 0.0010000
##
## Estimation Method:               tau:        Weighted Average of
```

```
##                                          Seasonal Estimates
##                                 slope:   Hirsch et al.'s
##                                          Modification of
##                                          Thiel/Sen Estimator
##                                 intercept: Median of
##                                          Seasonal Estimates
##
## Data:                           y      = Value
##                                 season = Run
##                                 year   = Year
##
## Data Source:                    lt.E01609
##
## Sample Sizes:                   1     =  13
##                                 2     =  21
##                                 3     =  21
##                                 4     =  13
##                                 5     =  19
##                                 6     =  12
##                                 7     =  13
##                                 8     =  21
##                                 9     =  20
##                                 10    =  21
##                                 11    =  21
##                                 12    =  20
##                                 Total = 215
##
## Number NA/NaN/Inf's:            10
##
## Test Statistics:                Chi-Square (Het) = 7.036434
##                                 z (Trend)        = 4.777164
##
## Test Statistic Parameter:       df = 11
##
## P-values:                       Chi-Square (Het) = 7.961254e-01
##                                 z (Trend)        = 1.777844e-06
##
## Confidence Interval for:        slope
##
## Confidence Interval Method:     Gilbert's Modification of
##                                 Theil/Sen Method
##
## Confidence Interval Type:       two-sided
##
## Confidence Level:               95%
##
## Confidence Interval:            LCL = 0
##                                 UCL = 0
```

Now a result is returned for the van Belle Hughes test and the p-value is not significant, so we can interpret the results of the SKT test. However, the p-value for the SKT test is highly significant ($\alpha < 0.0001$), but the slope estimate is 0. Although this may at first seem counterintuitive, it reflects the fact that the SKT test statistic and slope estimate are calculated very differently. As mentioned previously, the Mann-Kendall test

applied to data from each season is based on the sign (-/=/+) of the differences in values, while the slope estimate is the median of all the two-point slopes. The difference between the SKT test and slope estimate can be large when the majority of the slopes are equal to zero, as is the case for FRP at site E01609. In cases such as this, the slope estimate and associated confidence intervals provide a more reliable estimate of trend because they are based on numerical values and account for the magnitude of the difference in water quality. Therefore, we would conclude that there is no evidence of a trend in FRP at EHMP site E01609.

Now that we've explored some examples of SKT test results for individual sites, the next step is to automatically calculate SKT test results for multiple sites and water quality variables. In this example, we'll use all water quality variables collected at EHMP sites in the Logan estuary.

First, we set up the data.frame, `output`, which will hold the results of the trend asssment.

```
## Create data.frame of static EHMP site location information
ind<- !duplicated(dat$SiteCode)
site.info <- dat[ind,] %>% select(SiteCode, Waterway, MPme, AMTD,
                                  Lat, Long)

## Create long-term dataset for all water quality variables collected
## at Logan estuary sites
input.dat <- dat.long %>% filter(Waterway %in% "Logan")

## For each unique EHMP SiteCode and WQ.Var combination, store the
## minimum and maximum sampling date, and the number of non-missing
## observations (N) used in the trend assessment
lt.samp.info<- input.dat %>% filter(!is.na(Value)) %>% ## Remove missing values
    group_by(SiteCode, WQ.Var) %>%
    mutate(minDate = min(Date), ## create columns for min and max date sampled
           maxDate = max(Date)) %>%
    group_by(SiteCode, WQ.Var, minDate, maxDate) %>%
    summarise(N=n()) %>%  ## N = # observations used in trend assessment
    distinct() ## remove duplicate rows

## Set up data.frame to hold trend test results
output <- right_join(site.info, lt.samp.info, by = "SiteCode") %>%
    mutate(CSq.Het = NA, ## van Belle Hughes test statistic
           CSq.P = NA,   ## p-value for van Belle Hughes test
           kst.ts = NA,  ## SKT test statistic
           kst.p = NA,   ## p-value for SKT test
           slope = NA,   ## slope estimate
           LCI = NA,     ## lower confidence interval for slope
           UCI = NA)     ## upper confidence interval for slope

## Look at data.frame format and contents
head(output)
```

```
##   SiteCode Waterway                                                MPme AMTD
## 1   E00200    Logan Logan River 0.0km at Lagoon Island (EHMP) site 200    0
## 2   E00200    Logan Logan River 0.0km at Lagoon Island (EHMP) site 200    0
## 3   E00200    Logan Logan River 0.0km at Lagoon Island (EHMP) site 200    0
## 4   E00200    Logan Logan River 0.0km at Lagoon Island (EHMP) site 200    0
## 5   E00200    Logan Logan River 0.0km at Lagoon Island (EHMP) site 200    0
## 6   E00200    Logan Logan River 0.0km at Lagoon Island (EHMP) site 200    0
##         Lat     Long WQ.Var    minDate    maxDate   N CSq.Het CSq.P kst.ts
## 1 -27.70098 153.3243   Chla 2001-01-10 2022-02-04 215      NA    NA     NA
```

```
## 2 -27.70098 153.3243    DIN 2001-01-10 2022-02-04 222      NA    NA     NA
## 3 -27.70098 153.3243  DOSat 2001-01-10 2022-02-04 224      NA    NA     NA
## 4 -27.70098 153.3243    FRP 2001-01-10 2022-02-04 223      NA    NA     NA
## 5 -27.70098 153.3243    Sal 2001-01-10 2022-02-04 224      NA    NA     NA
## 6 -27.70098 153.3243   Temp 2001-01-10 2022-02-04 224      NA    NA     NA
##   kst.p slope LCI UCI
## 1    NA    NA  NA  NA
## 2    NA    NA  NA  NA
## 3    NA    NA  NA  NA
## 4    NA    NA  NA  NA
## 5    NA    NA  NA  NA
## 6    NA    NA  NA  NA
```

The example code is written to calculate trends for all water quality variables and EHMP sites found in `input.dat`. In this example, it includes the long-term trends for 9 water quality variables at 12 Logan estuary sites. To change that, subset `dat.long` based on `Date`, `SiteCode`, `Waterway` or `WQ.Var`. Some examples include:

```
## Start with full dataset in long format
dat.long <- dat %>%
    pivot_longer(cols = all_of(wq.vars), names_to = "WQ.Var",
                 values_to = "Value") %>%
    select(-Basinme, -Month) %>%
    modify_if(is.character, as.factor)

## Convert Run number to a factor
dat.long$Run <- as.factor(dat.long$Run)

## ------------------------------------------------------------------
## Long-term trend for all EHMP sites and water quality variables
## ------------------------------------------------------------------
input.dat<- dat.long

## ------------------------------------------------------------------
## Short-term trend (last 5 years) for all EHMP sites and water
## quality variables
## ------------------------------------------------------------------
yrs <- 5
minRY <- max(dat.long$RunYear)-yrs
input.dat <- dat.long %>% filter(RunYear > minRY)

## ------------------------------------------------------------------
## Long-term trend for all sites and 2 water quality variables
## ------------------------------------------------------------------
input.dat <- filter(dat.long, WQ.Var %in% c("Turb","FRP"))

## ------------------------------------------------------------------
## Short-term trend for sites in one reporting region and all water
## quality variables
## ------------------------------------------------------------------
yrs <- 5
minRY <- max(dat.long$RunYear)-yrs
input.dat <- filter(dat.long, RunYear > minRY &
```

```
                                        Waterway %in% "Coomera")

## ------------------------------------------------------------------
## Long-term trend for a single EHMP site and 1 variable
## ------------------------------------------------------------------
input.dat <- filter(dat.long, SiteCode %in% "E00100" &
                                    WQ.Var %in% "DIN")
```

Once the input and output data.frames are set up, the following R code can be used to generate a SKT test for all unique site/water quality variable combinations.

```
## For each unique SiteCode/WQ.Var combination
for(i in 1:nrow(output)) {
    tmp <- filter(input.dat, SiteCode %in% output$SiteCode[i] &
                             WQ.Var %in% output$WQ.Var[i])

    ## Check the variance of WQ values within each run. Remove runs
    ## with variance == 0 to avoid NaNs in Chi-square test for
    ## heterogeneity.
    test.var<- tmp %>% group_by(Run) %>%
        mutate(Var = var(Value, na.rm = TRUE), min = min(Value, na.rm = TRUE),
               max = max(Value, na.rm = TRUE))%>%
        distinct(Run, min, max, Var)

    ind<- test.var$Var == 0 | is.na(test.var$Var)

    ## If a sampling Run has variance == 0 (i.e. all values are the
    ## same), add a small value to one randomly selected observation
    noise <- min(tmp$Value, na.rm = TRUE)/1000000

    if(sum(ind) == 1) {
        ind2<- tmp$Run %in% test.var$Run[ind]
        rn <- sample(1:sum(ind2), 1, replace = F)
        r <- which(ind2)
        tmp$Value[r[rn]] <-  tmp$Value[r[rn]]+noise
        print(paste("SiteCode =", tmp$SiteCode[1], "& Run =", test.var$Run[ind],
                    ":", noise, "added to one observation b/c no seasonal variation"))
        rm(ind2, rn, r)

    } else if(sum(ind)>1) {
        zero.var<- which(ind)
        for(j in 1:length(zero.var)) {
               ind2 <- tmp$Run %in% test.var$Run[zero.var[j]]
               rn <- sample(1:sum(ind2), 1, replace = F)
               r <- which(ind2)
               tmp$Value[r[rn]] <-  tmp$Value[r[rn]]+noise
               print(paste("SiteCode =", tmp$SiteCode[1], "& Run =", zero.var[j],":",
                           noise, "added to one observation b/c no seasonal variation"))
               rm(ind2, rn, r)
        }
    }
```

```
    ## Run seasonal Kendall trend test and save output to output
    kst.i <- kendallSeasonalTrendTest(Value ~ Run + Year,
                                      data = tmp)
    output$CSq.Het[i] <- kst.i$statistic[["Chi-Square (Het)"]]
    output$CSq.P[i] <- kst.i$p.value[["Chi-Square (Het)"]]
    output$kst.ts[i]<- kst.i$statistic[["z (Trend)"]]
    output$kst.p[i] <- kst.i$p.value[["z (Trend)"]]
    output$slope[i]<- kst.i$estimate[["slope"]]
    output$LCI[i]<- kst.i$interval$limits["LCL"]
    output$UCI[i]<- kst.i$interval$limits["UCL"]

    rm(tmp, kst.i)
}

## Look at the output to ensure errors have not occurred
summary(output)
```

Once the program finishes, summarise the results. If there are missing values in the output columns (`CSq.Het`, `CSq.P`, `kst.ts`, `kst.p`, `slope`, `LCI`, `UCI`), then an error has occured. If not, then write the results to a .csv file and make sure to give it a meaningful name.

```
## Write the results to a csv.file.
write.csv(output, "LT_Trends_Logan.csv", row.names = FALSE)
```

Keep in mind that trend assessment is an iterative process. The results in the output table should be carefully examined to ensure they make sense, given what we know about water quality processes. Revisiting the exploratory plots and tables may provide insights about the results, or help identify errors in the input data. Once you're confident in the results, they can be used to create summary tables and maps of short- and long-term trend assessments by EHMP site.

References

Beck MW, de Valpine P, Murphy R, et al (2022) Multi-scale trend analysis of water quality using error propagation of generalized additive models. Science of The Total Environment 802:149927

Grolemund G, Wickham H (2011) Dates and times made easy with lubridate. Journal of Statistical Software 40:1–25

Hirsch RM, Slack JR (1982) Techniques of trend analysis for monthly water quality data. Water Resources Research 18:107–121

Kassambara A (2019) ggcorrplot: Visualization of a correlation matrix using 'ggplot2'. https://CRAN.R-project.org/package=ggcorrplot; R package version 0.1.3

Mann HB (1945) Nonparametric tests against trend. Econometrica 13:245–259

Massicotte P (2022) ggpmthemes: Personal themes for ggplot2. https://github.com/PMassicotte/ggpmthemes; R package version 0.0.2

Meals D, Spooner J, Dressing S, Harcum J (2011) Statistical analysis for monotonic trends, Tech Notes 6. Developed for the US Environmental Protection Agency by Tetra Tech, Inc

Millard S (2013a) EnvStats: An R package for environmental statistics. Springer, New York

Millard S (2013b) Chapter 7: Hypothesis tests. In: EnvStats: An R package for environmental statistics. Springer, pp 149–173

Millard S, Dixon P, Neerchal NK (2014) Environmental statistics with R. CRC Press, Boca Ratan, Florida

Morton R, Henderson B (2008) Estimation of nonlinear trends in water quality: An improved approach using generalized additive models. Water Resources Research 44:W07420

R Core Team (2022) R: A language and environment for statistical computing. R Foundation for Statistical

Computing, Vienna, Austria
Schauberger P, Walker A (2021) openxlsx: Read, write and edit xlsx files. https://CRAN.R-project.org/package=openxlsx; R package version 4.2.5
van Belle G, Hughes J (1984) Nonparametric tests for trend in water quality. Water Resources Research 20:127–136
Wickham H, Averick M, Bryan J, et al (2019) Welcome to the tidyverse. Journal of Open Source Software 4:1686. https://doi.org/10.21105/joss.01686
Wickham H, Navarro D, Pedersen TL (2022) ggplot2: Elegant graphics for data analysis, 3rd edition. Springer-Verlag New York
Zeileis A, Grothendieck G (2005) zoo: S3 infrastructure for regular and irregular time series. Journal of Statistical Software 14:1–27. https://doi.org/10.18637/jss.v014.i06