

COMPASS_SynopticCB_PW_SO4_Cl_202408

Zoe Read, Stephanie J. Wilson

2025-09-03

Contents

0.1	Run Information	2
0.2	Assess Standards	6
0.3	Assess Blanks	6
0.4	Assess Duplicates	7
0.5	Calculate mmol/L concentrations & salinity, add dilutions	9
0.6	Assess Analytical Spikes	10
0.7	Check if samples within the range of the standard curve	11
0.8	Check to see if samples run match metadata & merge info	11
0.9	Visualize Data by Plot	12
0.10	Export Processed Data	12

##Add Required Packages

0.1 Run Information

```
##### Run information - PLEASE CHANGE
Date_Run = "2025-06-16" #Date that instrument was run
Run_by = "Zoe Read" #Instrument user
Script_run_by = "Zoe Read" #Code user
run_notes = "All S04 blanks were zero" #any notes from the run
samples <- c("GCW", "GWI", "MSM", "SWH") #whatever identifies your samples within the same names
samples_pattern <- paste(samples, collapse = "|")
#samples_pattern <- "GCW" #use this instead of the line above if you have only one site code

##### File Names - PLEASE CHANGE
#file path and name for raw summary data file
raw_file_name_cl = "Raw Data/COMPASS_Synoptic_CB_MonMon_202408_Cl.txt"
raw_file_name_so4 = "Raw Data/COMPASS_Synoptic_CB_MonMon_202408_S04.txt"

#file path and name of processed data file
processed_file_name = "Processed Data/COMPASS_SynopticCB_PW_Processed_Cl_S04_202408.csv"

##### Log Files - PLEASE CHECK
#downloaded metadata csv - downloaded from Google drive as csv for this year
Raw_Metadata = "Raw Data/COMPASS_SynopticCB_PW_SampleLog_2024.csv"

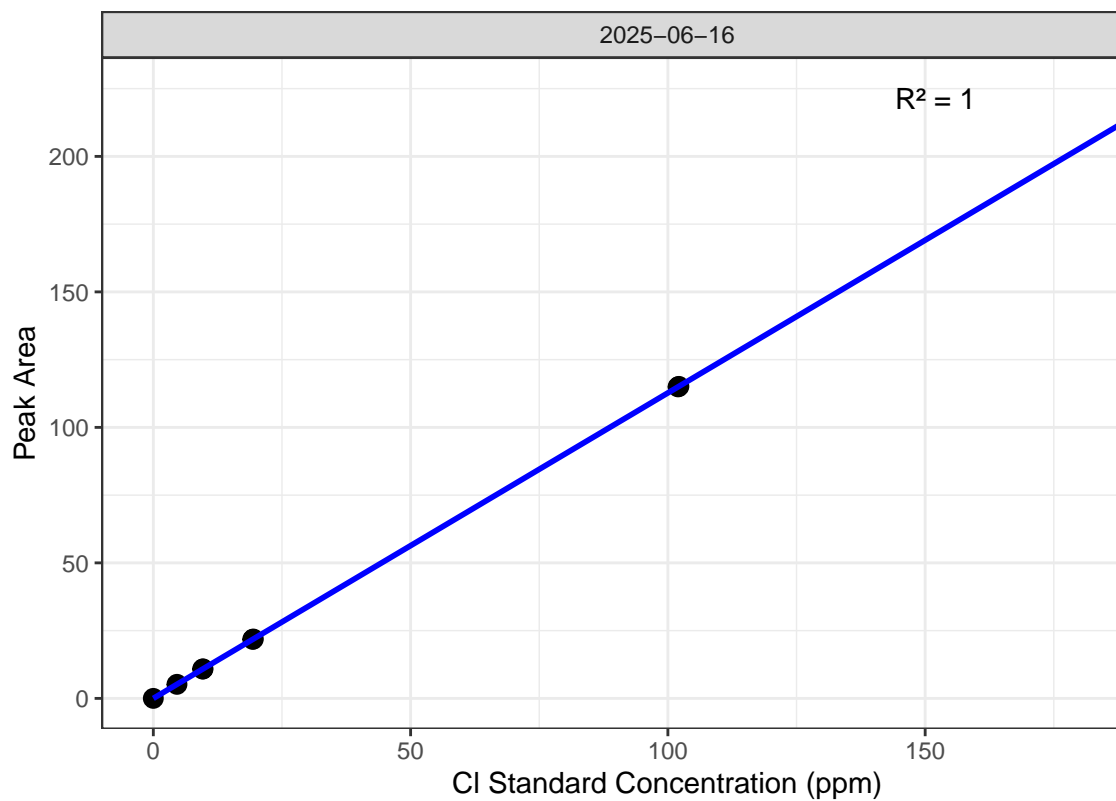
#qaqc log file path for this year
Log_path = "Raw Data/COMPASS_Synoptic_Cl_S04_QAQClog_2024.csv"
```

##Set Up Code - constants and QAQC cutoffs

##Read in metadata and create similar sample IDs for matching to samples

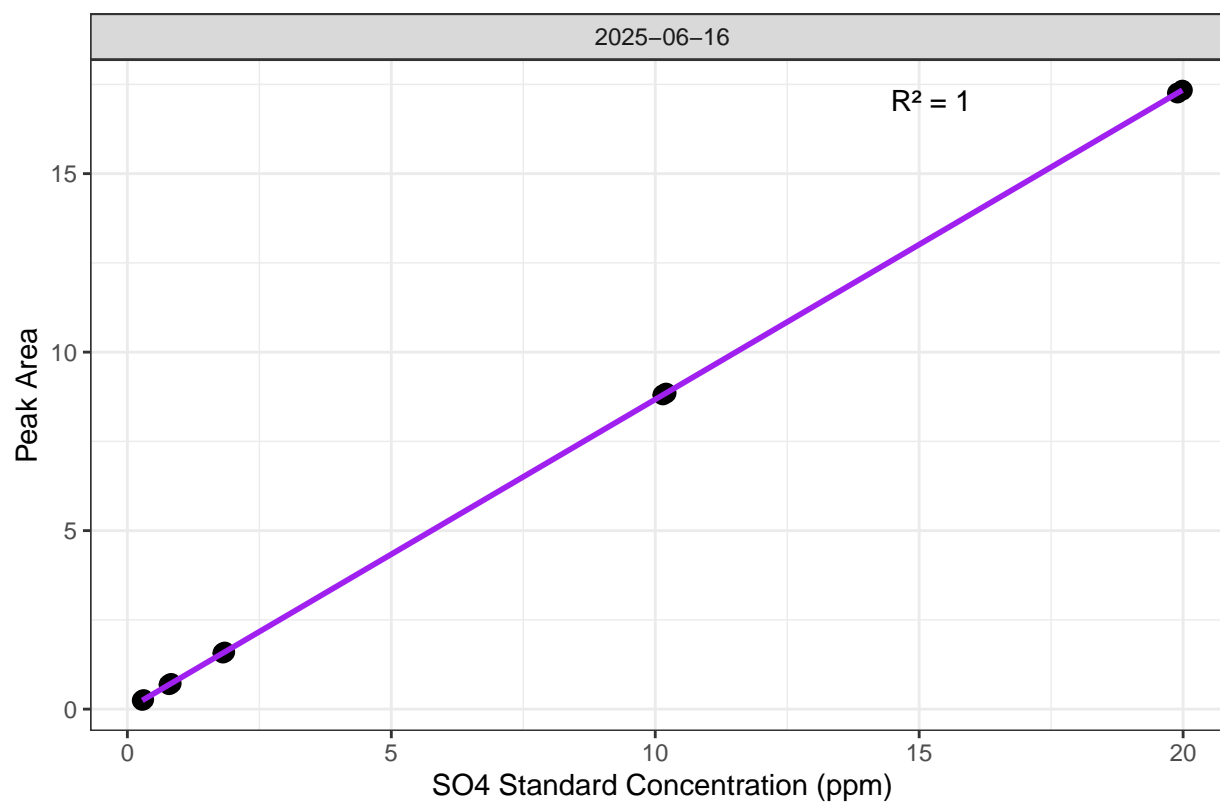
##Import Sample Data

Chloride Std Curve

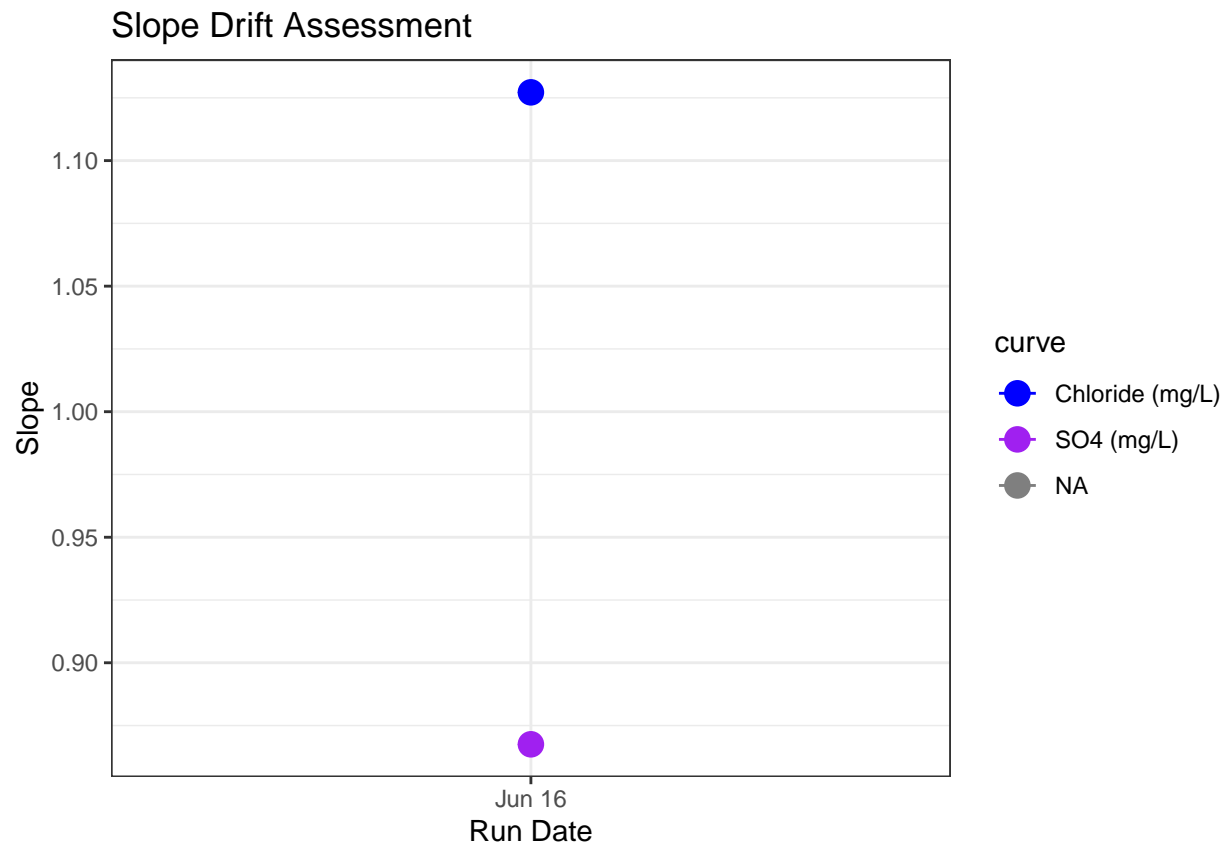


Assess Check Standards

Sulfate Std Curve



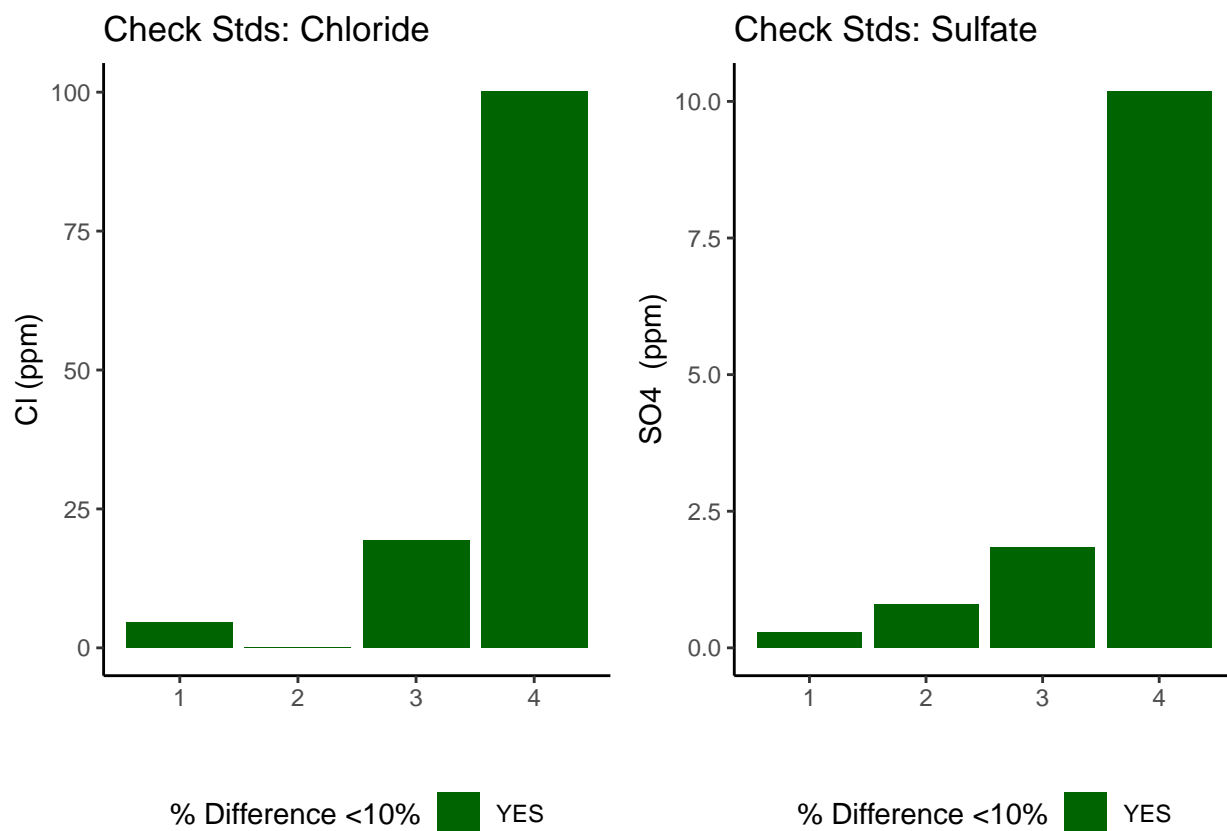
```
## [1] "QAQC log file exists and has been read into the code."
```



```
## [1] "Cl Curve r2 GOOD"
```

```
## [1] "SO4 Curve r2 GOOD"
```

0.2 Assess Standards



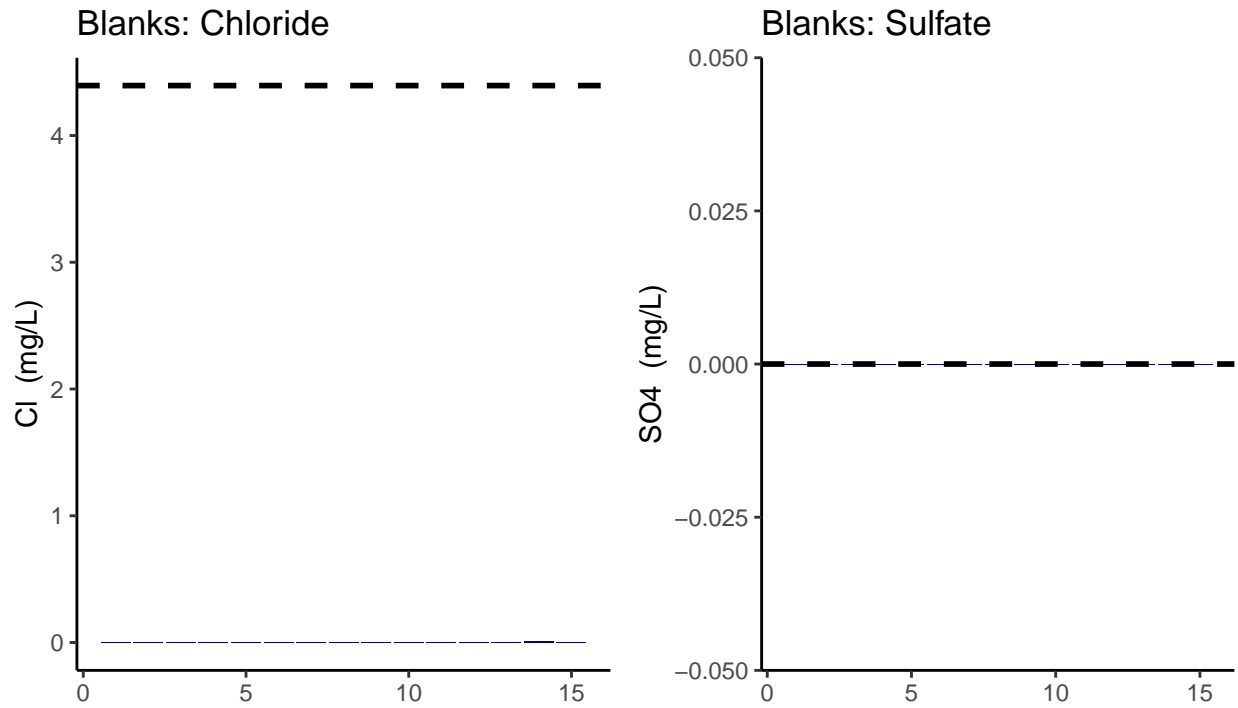
```
## [1] ">80% of Chloride Check Standards are within range of expected concentration - PROCEED"
```

```
## [1] ">80% of Sulfate Check Standards are within range of expected concentration - PROCEED"
```

0.3 Assess Blanks

```
## [1] ">80% of Chloride Blank concentrations are lower 25% quartile of samples"
```

```
## [1] ">80% of Sulfate Blank concentrations are lower 25% quartile of samples"
```



```

Blank Conc <25% Quartile Samples YE
Blank Conc <25% Quartile Samples

## Chloride blanks mean ppm:
## [1] 0.0008533333
## Sulfate blanks mean ppm:
## [1] 0

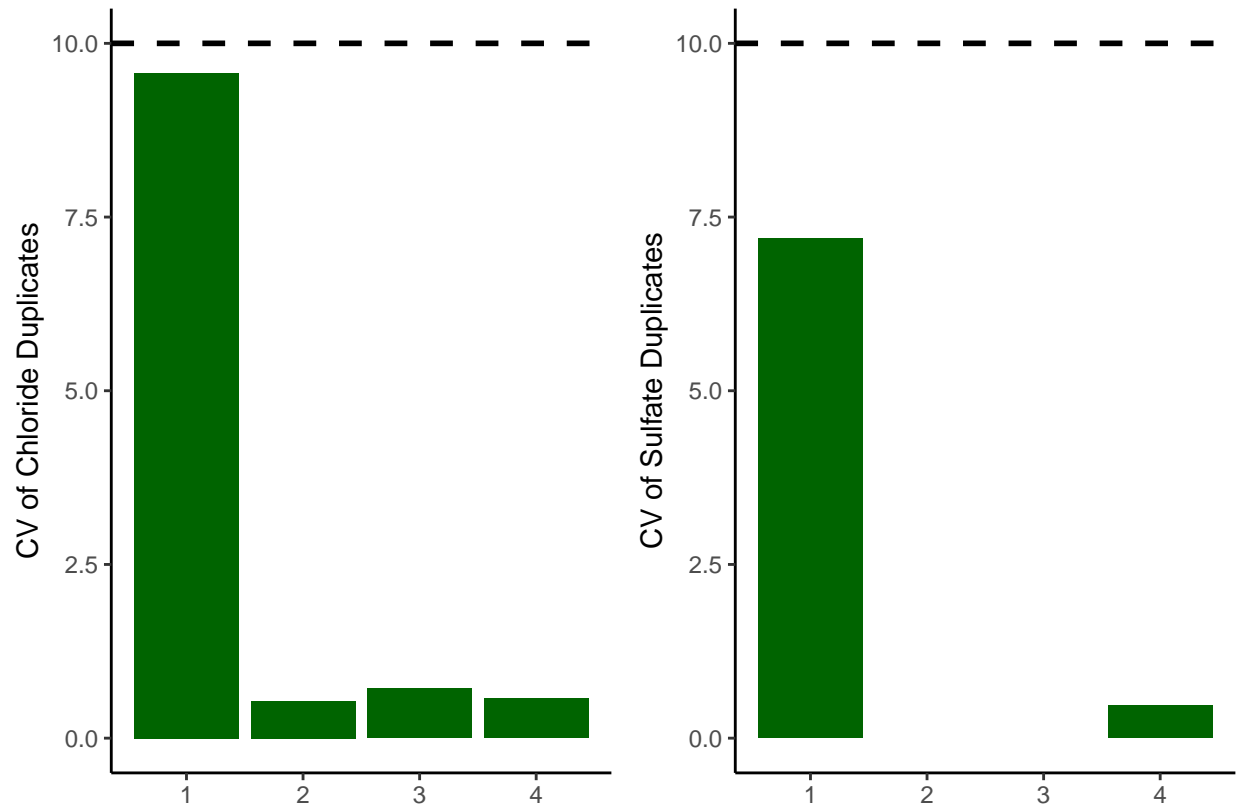
```

0.4 Assess Duplicates

```

## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_bar()').

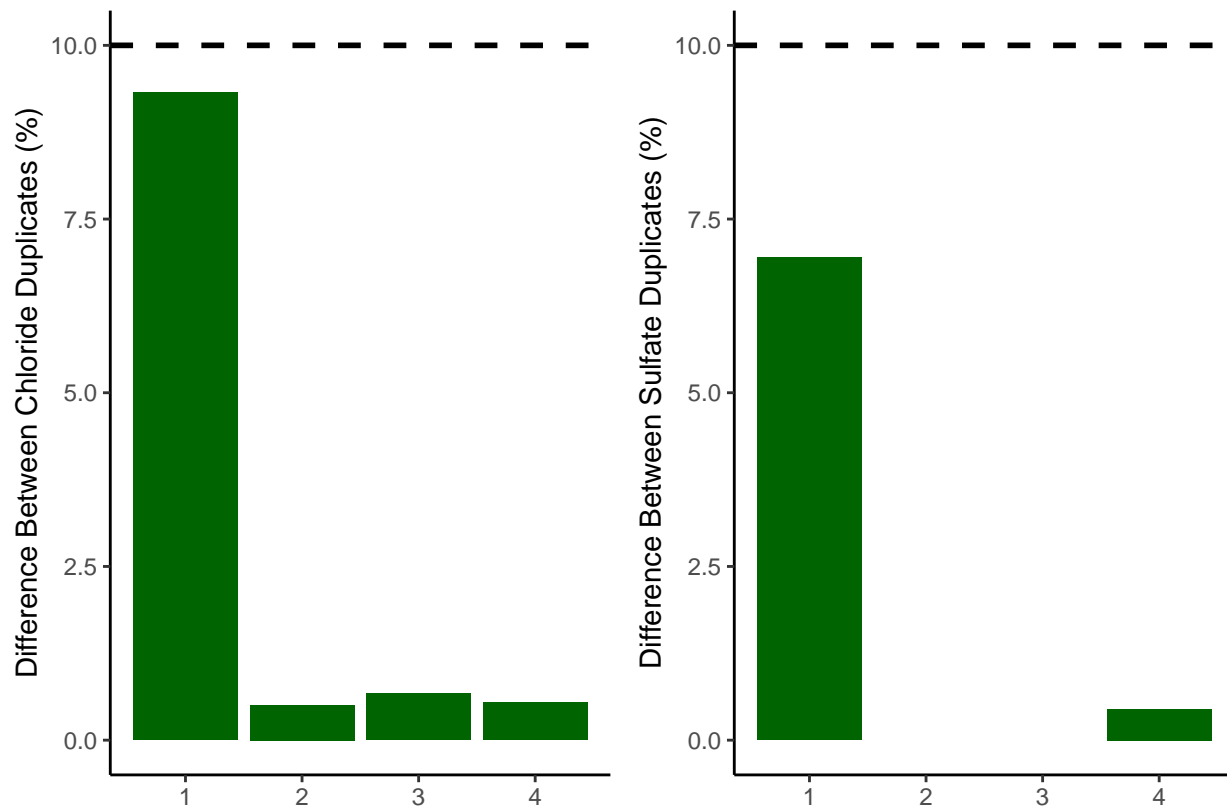
```



```
## [1] ">80% of Chloride Duplicates have a CV <10%"
```

```
## [1] ">80% of Sulfate Duplicates have a CV <10%"
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## ('geom_bar()').
```

```
## [1] ">80% of Chloride Duplicates have a percent difference <10% - PROCEED"
```

```
## [1] ">80% of Sulfate Duplicates have a percent difference <10% - PROCEED"
```

0.5 Calculate mmol/L concentrations & salinity, add dilutions

```
# Convert ppm to mmol/L
all_dat$S04_Conc_mM <- (all_dat$S04_ppm / s_mw)
all_dat$Cl_Conc_mM <- (all_dat$Cl_ppm / cl_mw)

# Calculate Salinity
# calculated using the Knudsen equation
# Salinity = 0.03 + 1.8050 * Chlorinity
# Ref: A Practical Handbook of Seawater Analysis by Strickland & Parsons (P. 11)
# = ((1.807 * Cl_ppm) + 0.026) / 1000
all_dat$salinity <- ((1.8070 * all_dat$Cl_ppm) + 0.026) / 1000

#Need to determine dilution factors for your samples
#for Steph / COMPASS this depends on the site so...
all_dat$Dilution <- 1
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "MSM") & str_detect(all_dat$sample_ID, "UP"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "MSM") & str_detect(all_dat$sample_ID, "TR"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "MSM") & str_detect(all_dat$sample_ID, "WC"),
```

```

all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "MSM") & str_detect(all_dat$sample_ID, "SW"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GCW") & str_detect(all_dat$sample_ID, "UP"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GCW") & str_detect(all_dat$sample_ID, "TR"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GCW") & str_detect(all_dat$sample_ID, "WC"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GCW") & str_detect(all_dat$sample_ID, "SW"),

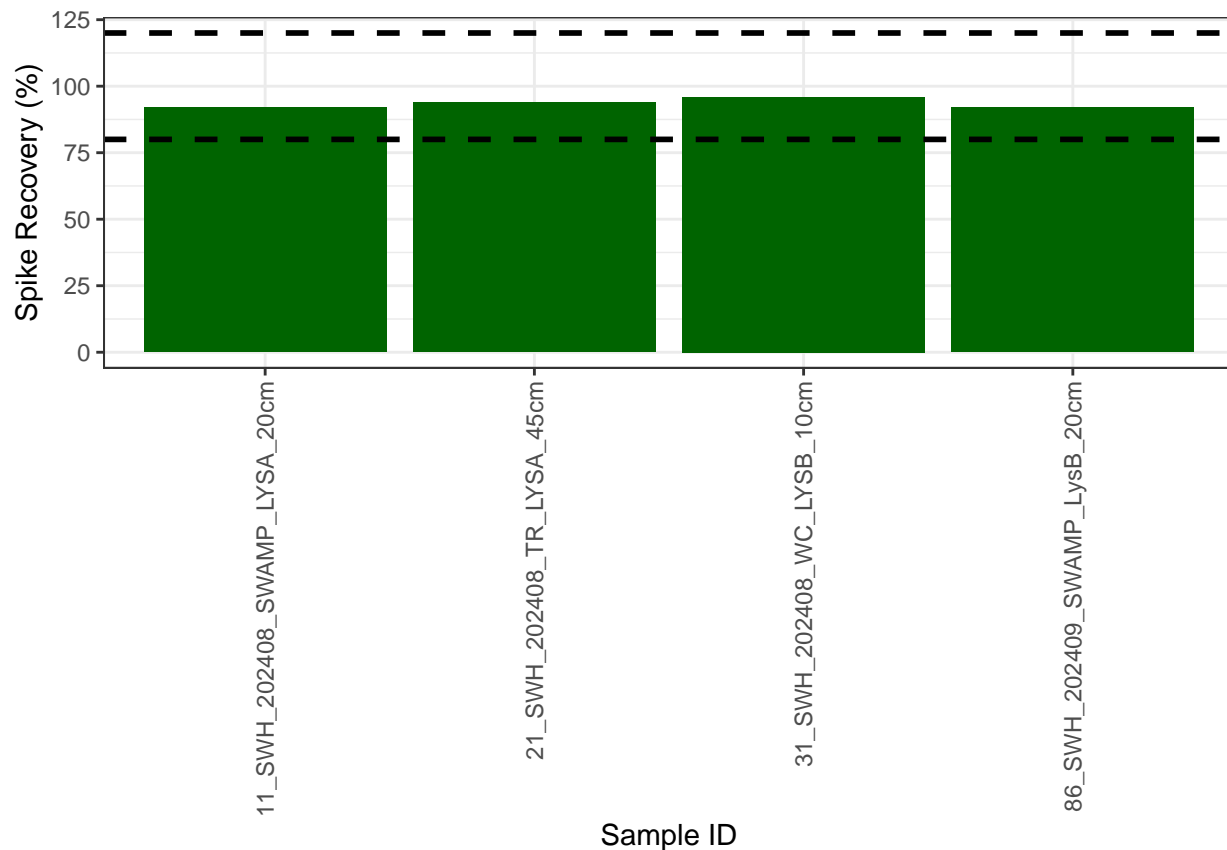
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GWI") & str_detect(all_dat$sample_ID, "UP"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GWI") & str_detect(all_dat$sample_ID, "TR"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GWI") & str_detect(all_dat$sample_ID, "WC"),
all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "GWI") & str_detect(all_dat$sample_ID, "SW"),

all_dat$Dilution <- ifelse(str_detect(all_dat$sample_ID, "SWH"), 50, all_dat$Dilution)

head(all_dat)

```

0.6 Assess Analytical Spikes



```

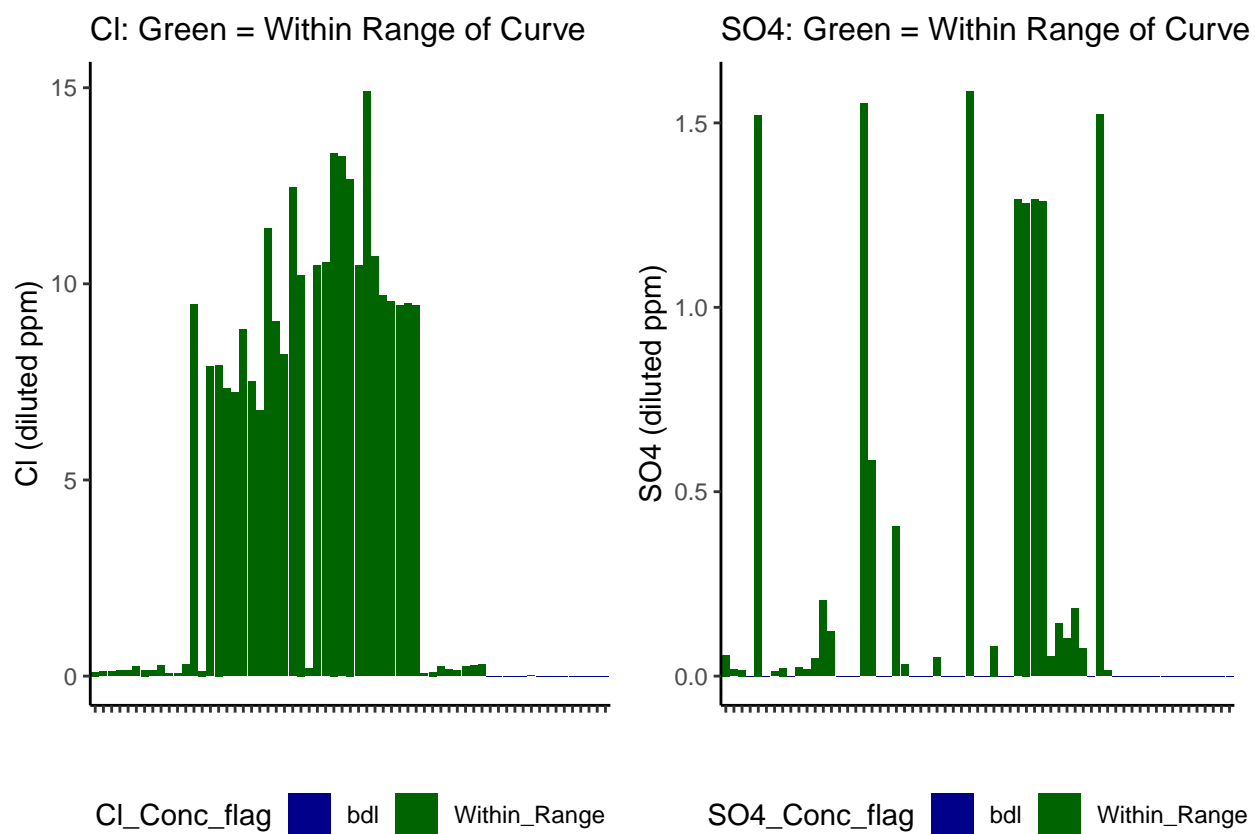
## # A tibble: 1 x 4
##   S04_spks_flag no_rows Total Percent
##   <chr>         <int> <int>   <dbl>
## 1 Yes           4     4    100

```

```
## [1] ">80% of S04 spikes have a recovery between the high and low cutoff - PROCEED"
```

0.7 Check if samples within the range of the standard curve

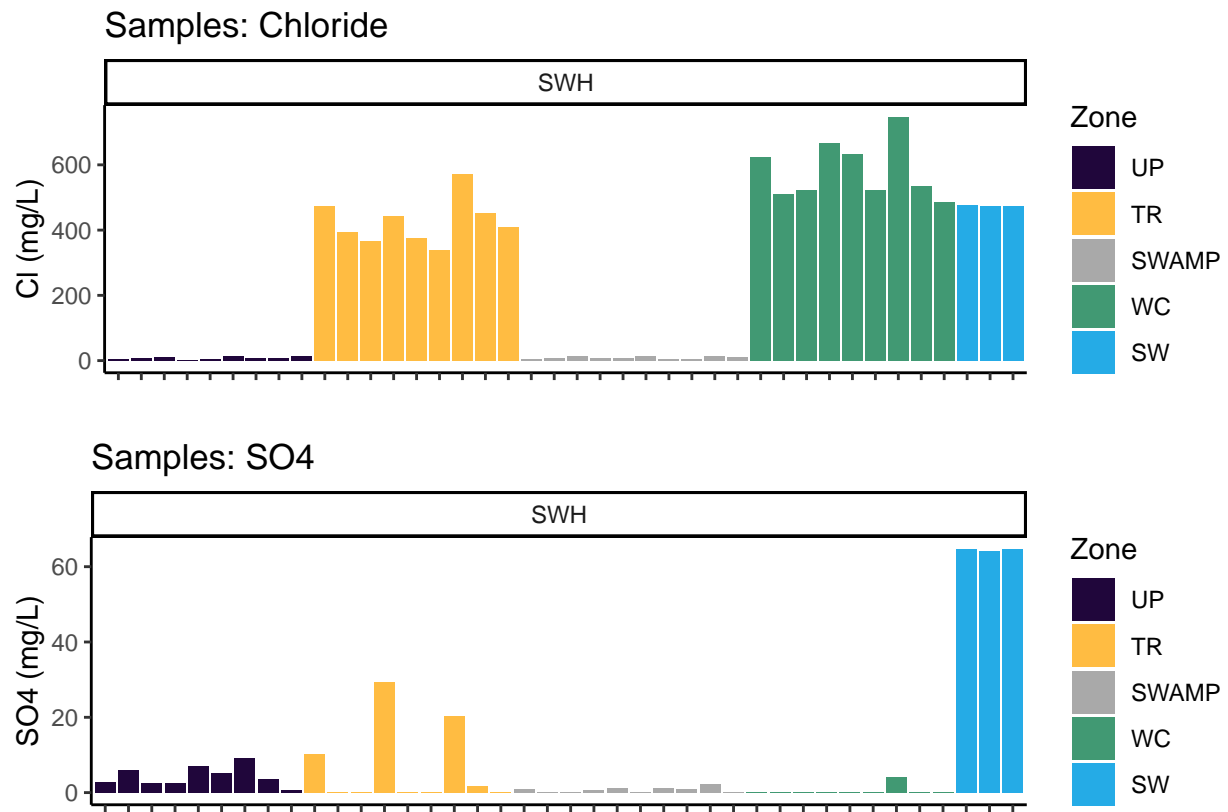
Sample Flagging



0.8 Check to see if samples run match metadata & merge info

All sample IDs are present in metadata.

0.9 Visualize Data by Plot



0.10 Export Processed Data

#end