CVEN 6345

Water Quality Modeling/Monitoring

Semester Project

Part 3 Report

Aalok Sharma Kafle

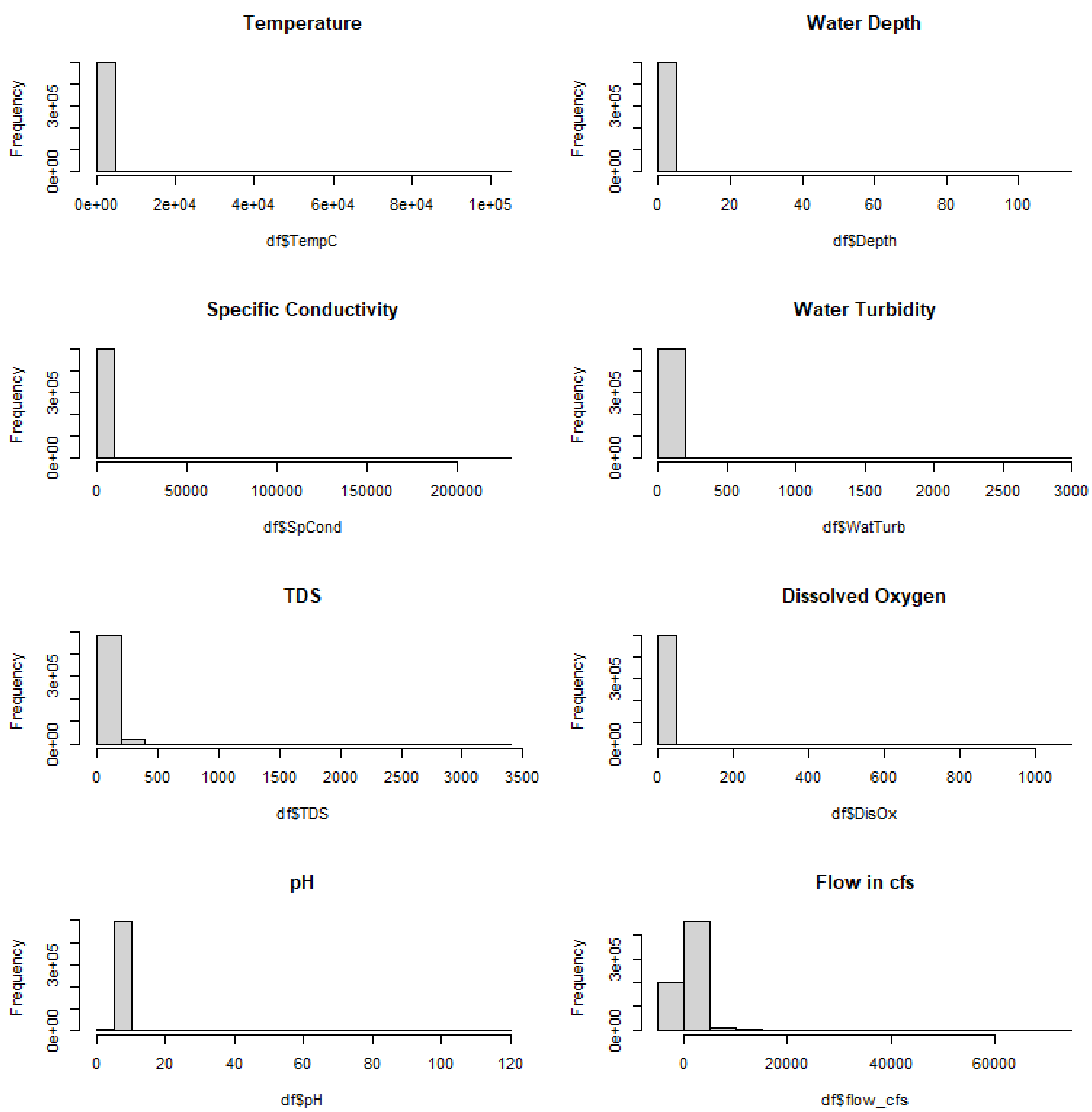
Data Filtration

Time series data can be prone to errors and anomalies due to various reasons, such as equipment malfunction or environmental factors. Therefore, it is crucial to filter out illogical data points from the dataset before analyzing it. During the initial exploration of the water quality data, we observed that various text strings were attached to the parameter values, such as '23.4AQI' for Temperature. To ensure that the data was properly formatted for analysis, we created a custom function in R to filter out these text instances and extract only the numeric values from each cell. For instance, the function was designed to extract only '23.4' as a numeric value from the '23.4AQI' string. By implementing this function, we were able to transform the water quality data into a more usable and reliable format, facilitating subsequent analyses.

Next for each parameter, i.e., temperature, water depth, specific conductivity, water turbidity, TDS, DO, pH, and stream discharge, several methods were used to identify and remove erroneous data. For instance, data points that fall outside the expected range of values for each parameter were flagged and removed. For this, summary statistics of each parameter were extracted from R and sudden spikes or drops in the data that are not consistent with the surrounding data points were identified and removed. By filtering out illogical data from the time series dataset, we can ensure that the remaining data is comparably more accurate and reliable for further analysis and modeling.

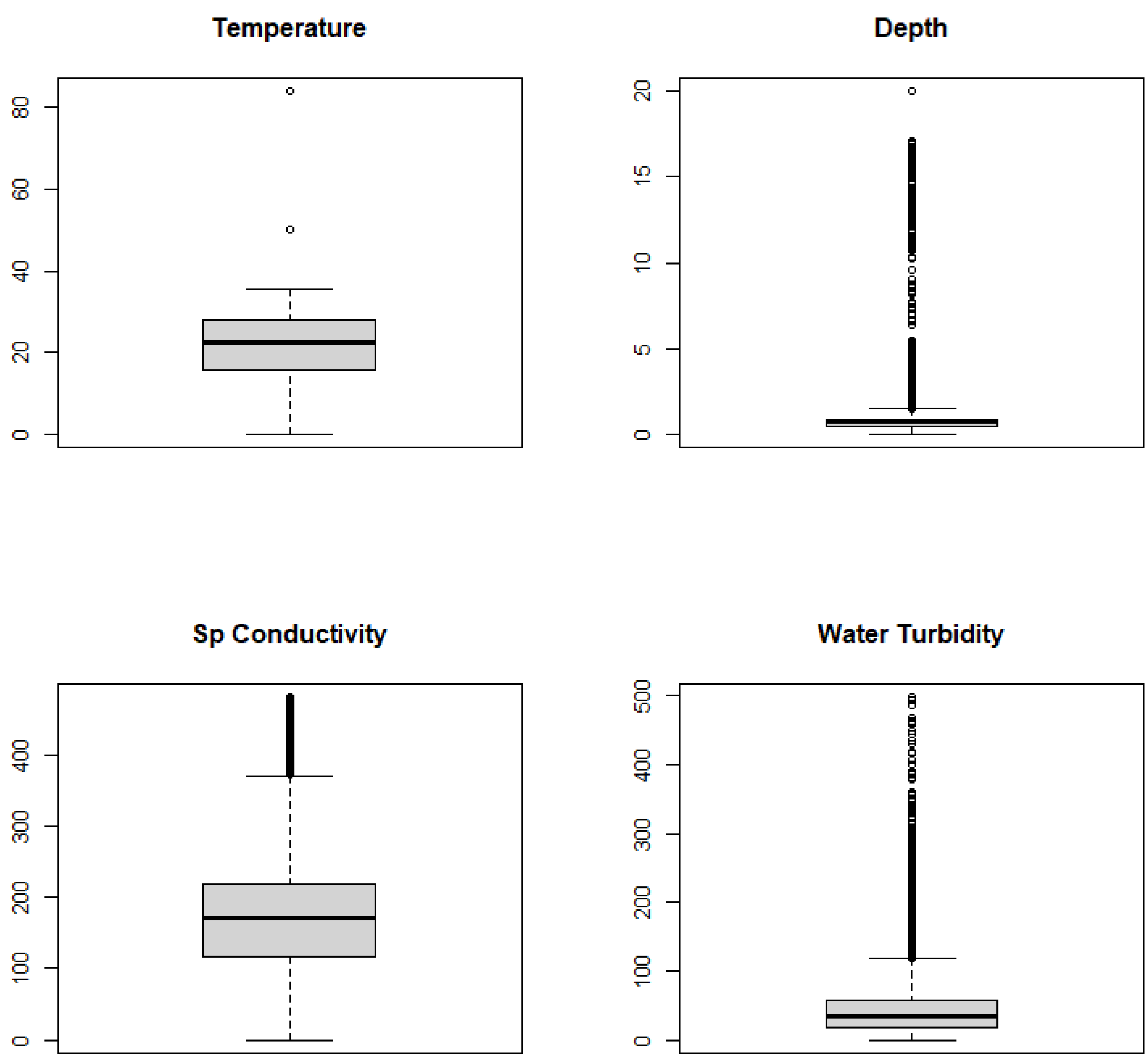
The visualization of histograms for each parameter, presented in Figure 1, proved to be a valuable tool for identifying missing data and detecting any illogical values that may have been present in the dataset. Specifically, values such as '50000' or '100000' were identified as outliers and deemed to be erroneous. To remove such data points from the dataset and maintain the accuracy of the analysis, we applied appropriate filters as tabulated in Table 1 and converted such values to NA.

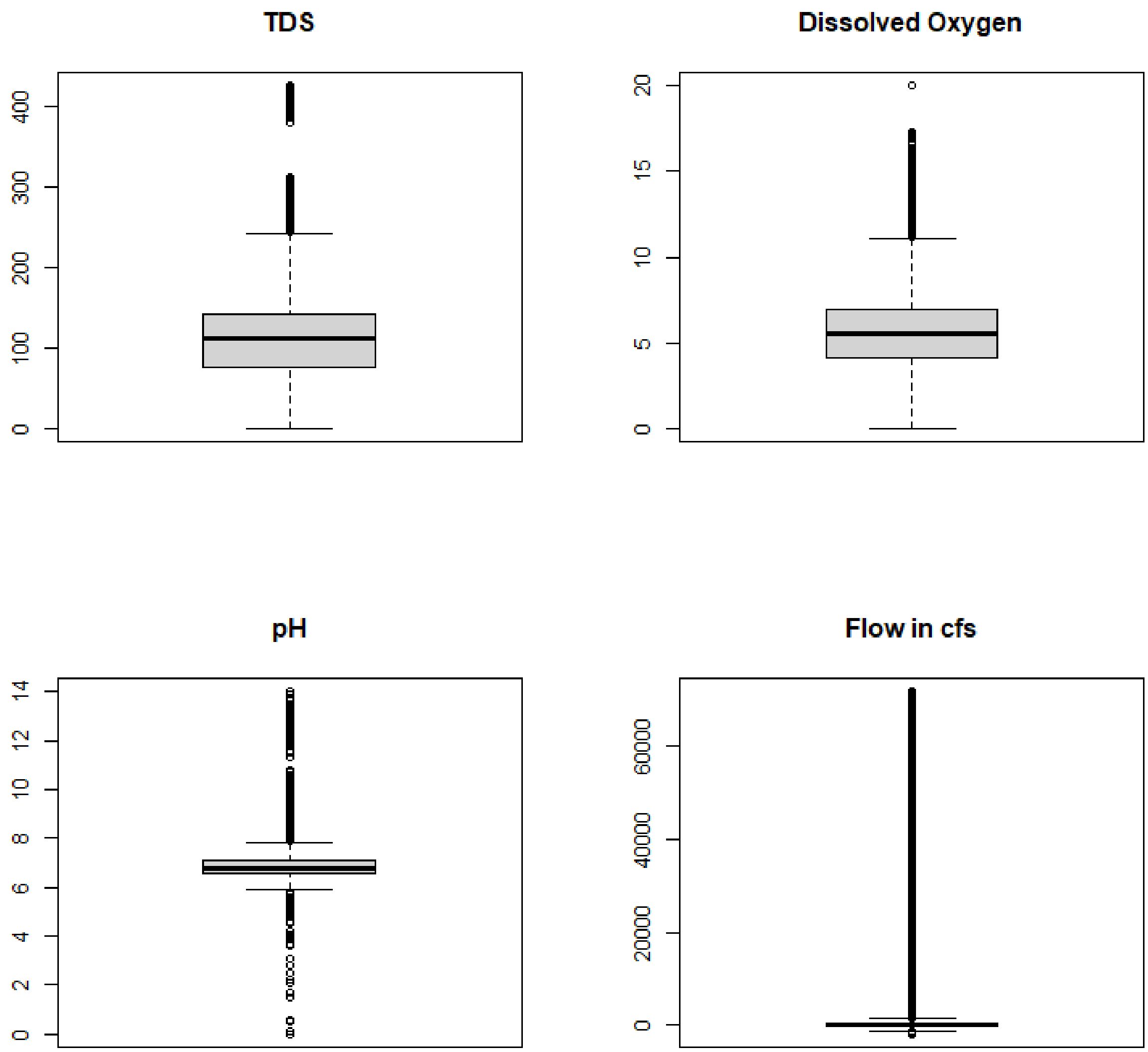
Parameter	Removed values
Temperature	Greater than 10000
Water Depth	Greater than 10000
Specific Conductivity	Greater than 10000
Water Turbidity	Greater than 2500
TDS	Greater than 2500
Dissolved Oxygen	Greater than 10000
pH	Greater than 14
Flow in cfs	None



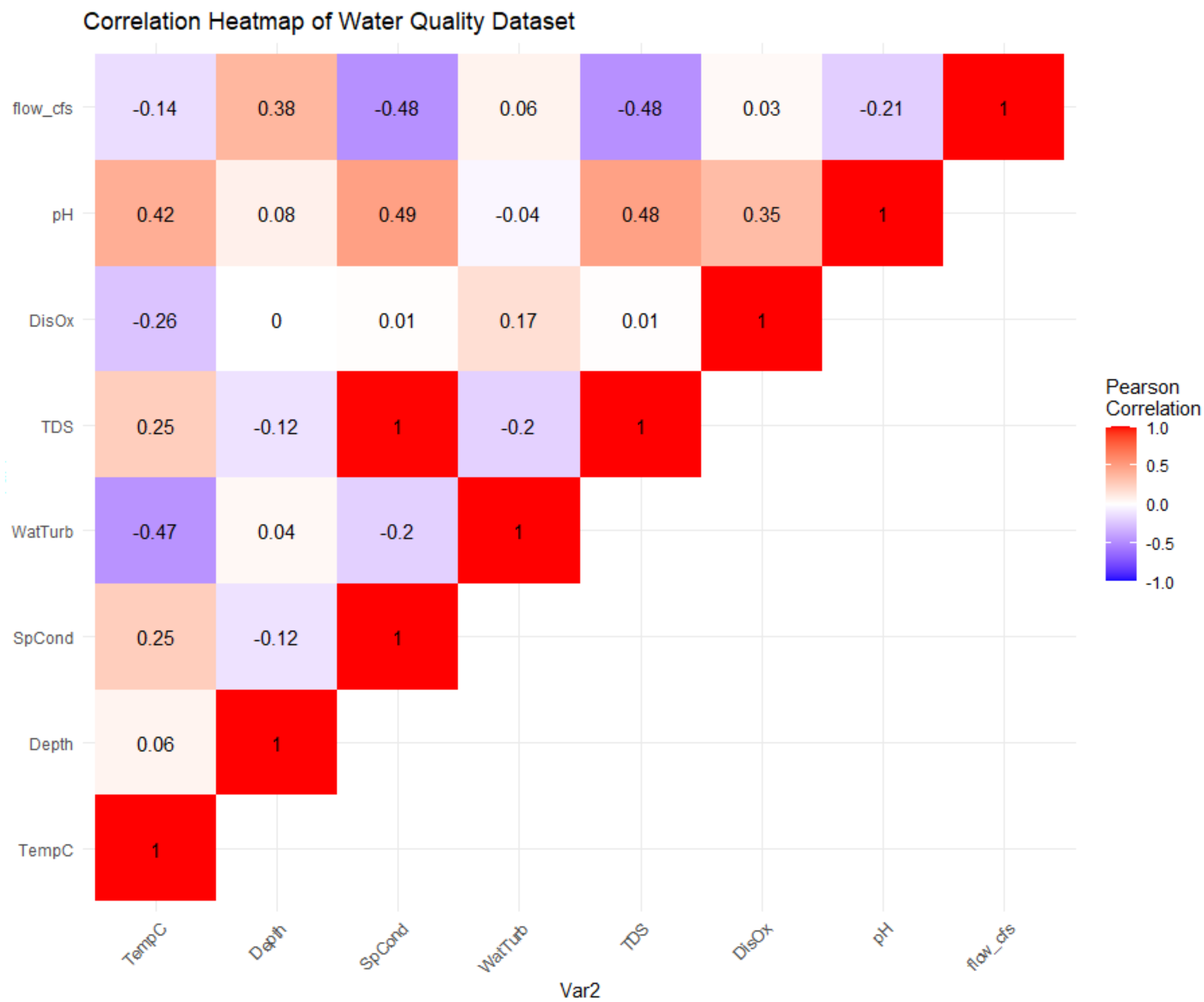
Summary Statistics

In order to gain a deeper understanding of the range of each parameter, a histogram was plotted for each one. as seen in Figures 2 and 3. By examining the histograms, we were able to determine the frequency distribution of values within each parameter range and identify any outliers or unusual patterns.




TDS
0
100
200
300
400
Dissolved Oxygen
0
5
10
15
20
pH
0
2
4
6
8
10
12
14
Flow in cfs
0
20000
40000
60000

Relationship between Water Quality Parameters



Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
pH	-0.21
Dissolved Oxygen	0.03
TDS	-0.48
Water Turbidity	0.06
Specific Conductivity	-0.48
Temperature	-0.14

The negative correlation coefficients for pH, Specific Conductivity, and Temperature suggest that as the flow rate increases, these parameters decrease. On the other hand, the negative correlation coefficient for TDS indicates that as the flow rate increases, the TDS levels decrease. Meanwhile, the positive correlation coefficients for Dissolved Oxygen and Water Turbidity suggest that as the flow rate increases, the DO levels and Water Turbidity also increase, although to a very small extent.

Overall, these results provide insights into the relationships between water quality parameters and flow rate. Further analysis is necessary to determine the causality of these relationships, and how they may impact the overall health of the water body.

Data Aggregation

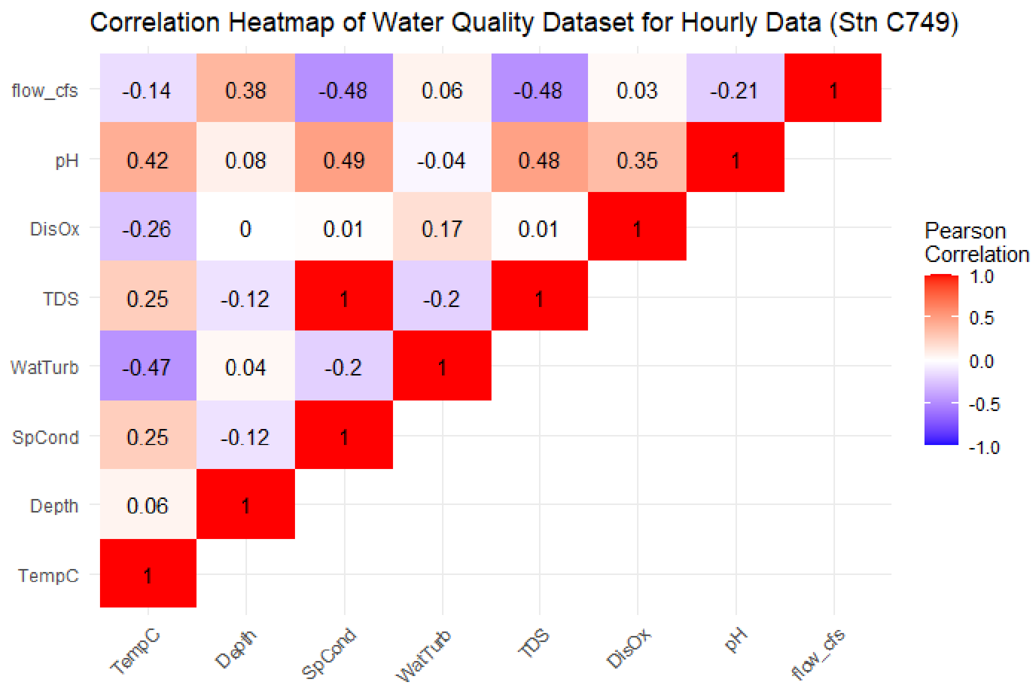
To generate 15-minute, hourly, daily mean, daily max, and daily min data, the initial step involved creating an xts object in R. The data was then aggregated to an hourly and daily format using suitable aggregation functions.

To obtain hourly mean, daily mean, daily max, and daily min from a dataset in R using xts object, we can first create an xts object from the dataset. Then, we can use the "aggregate()" function with appropriate aggregation functions like "mean()", "max()", and "min()" to aggregate the data into hourly and daily intervals. The resulting output will contain the hourly mean, daily mean, daily max, and daily min values. The code is presented below:

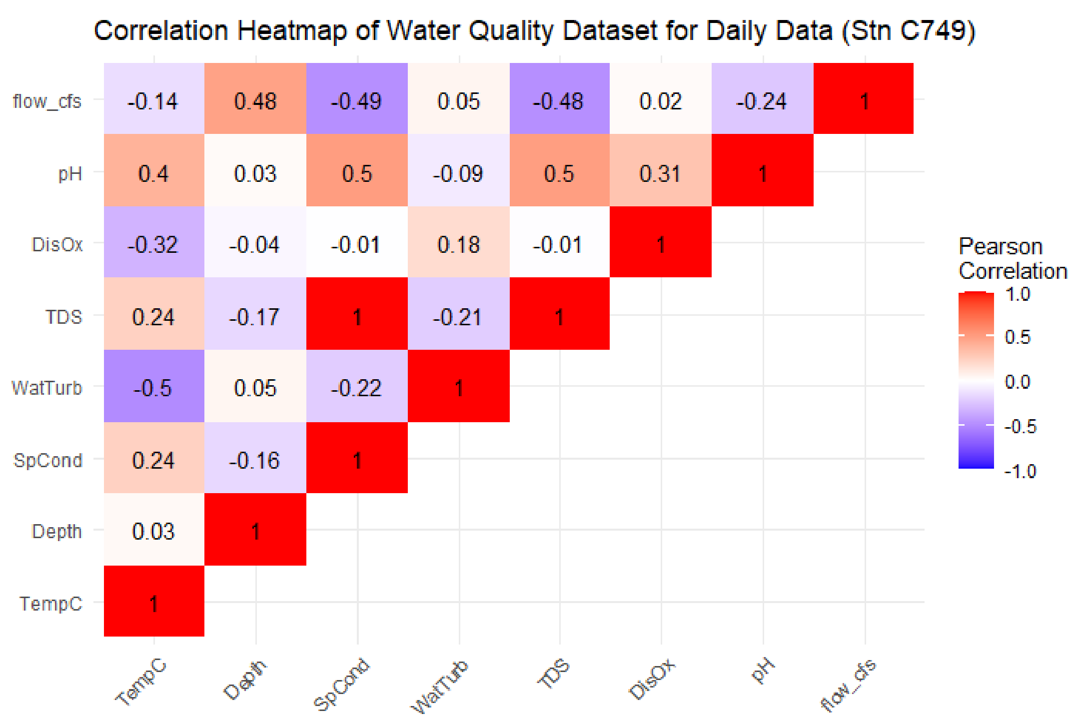
```
#Time series conversion
df <- merged_df[, c("datetime","TempC","Depth","SpCond","WatTurb","TDS","DisOx","pH","flow_cfs")]

df <- df[complete.cases(df$datetime), ] #Remove rows without flow

df$datetime <- as.POSIXct(df$datetime, format = "%d/%m/%Y %H:%M") #Arrange date format in df

# Convert to xts object with 15-minute intervals
xts_data <- xts(df[, 2:9], order.by = df$datetime)
#Line 2 to 9 includes all the parameters

# Convert to hourly data
hourly_data <- aggregate(xts_data, as.POSIXct(cut(index(xts_data), breaks="hour")), mean)

# Convert to daily data
daily_data <- aggregate(xts_data, as.Date(index(xts_data)), mean)

daily_min <- aggregate(xts_data, as.Date(index(xts_data)), min)

daily_max <- aggregate(xts_data, as.Date(index(xts_data)), max)
```

The correlation among the parameters was subsequently assessed for all four aggregation methods, and their respective outcomes are enumerated and graphed below:

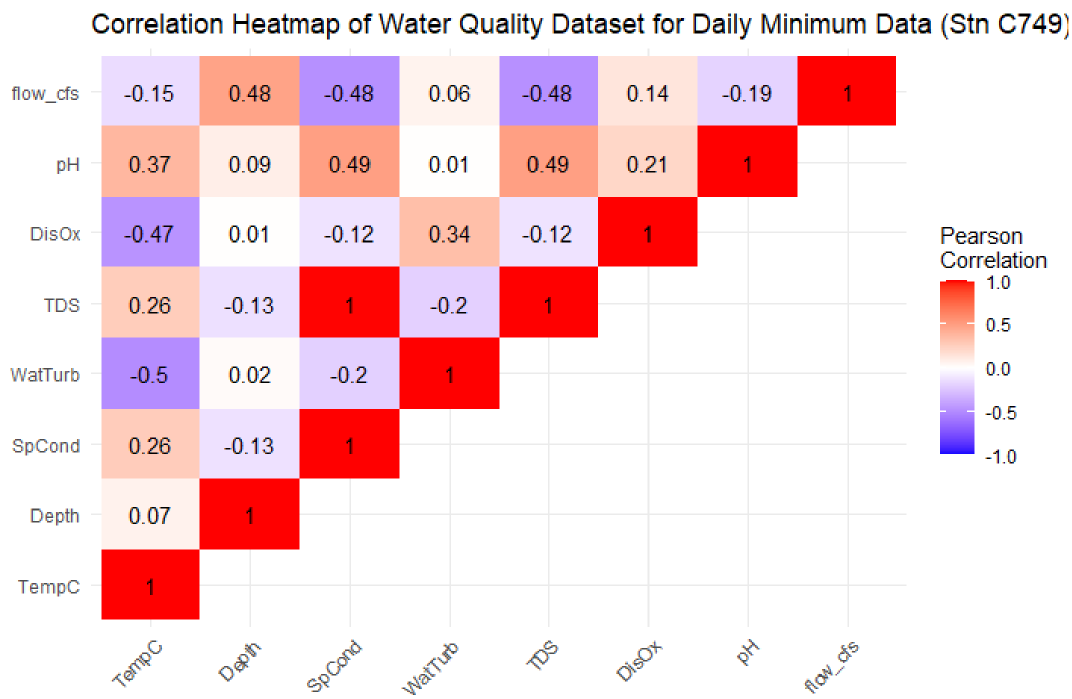
Hourly Aggregation



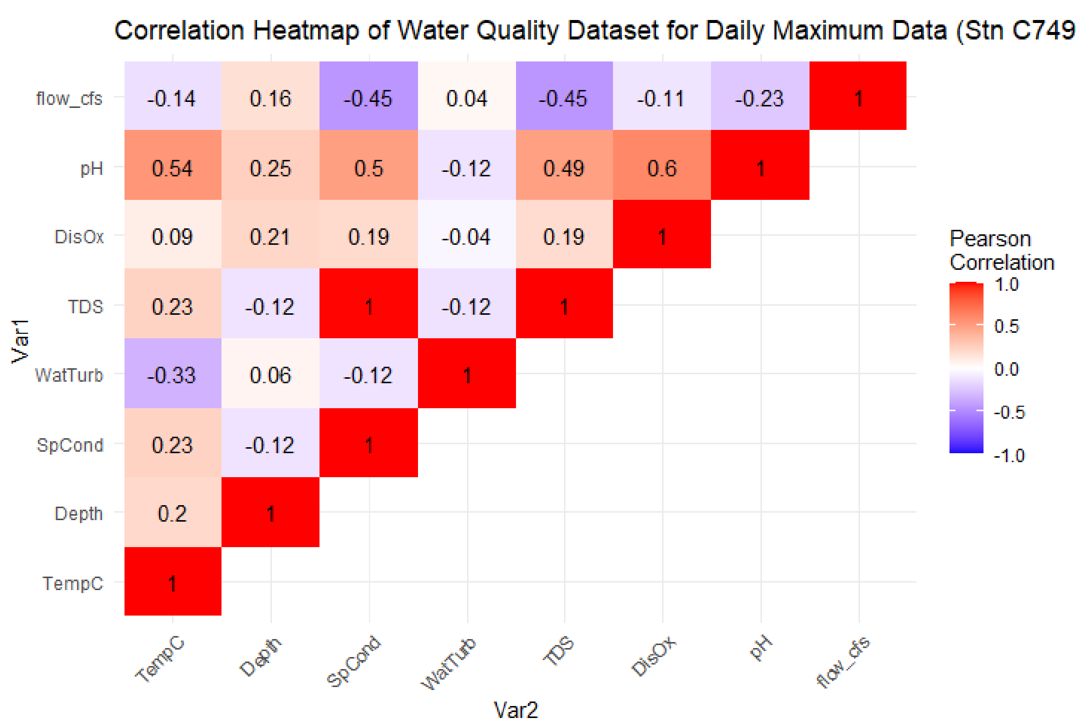
Daily Average Aggregation



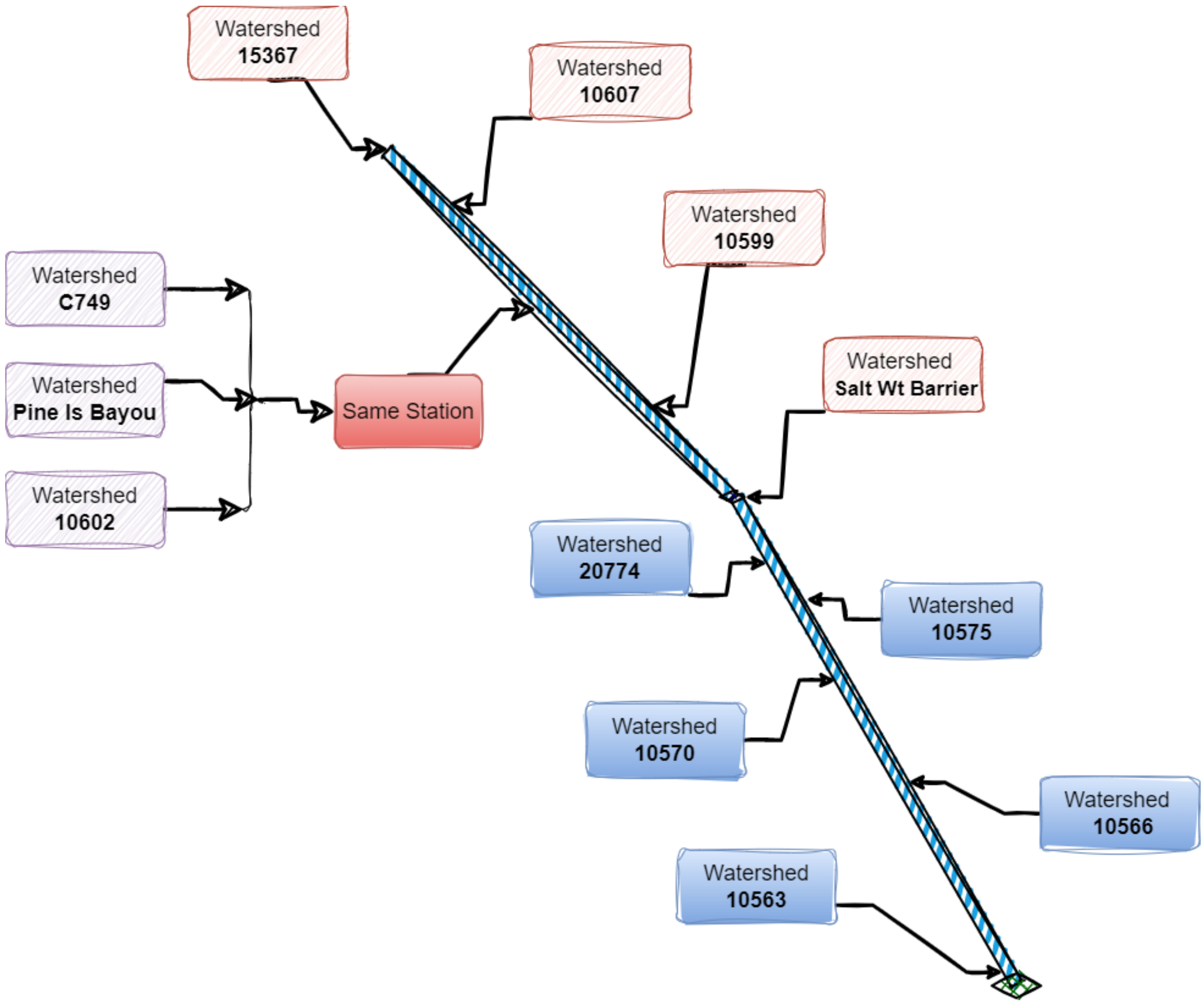
Daily Minimum Aggregation



Daily Maximum Aggregation

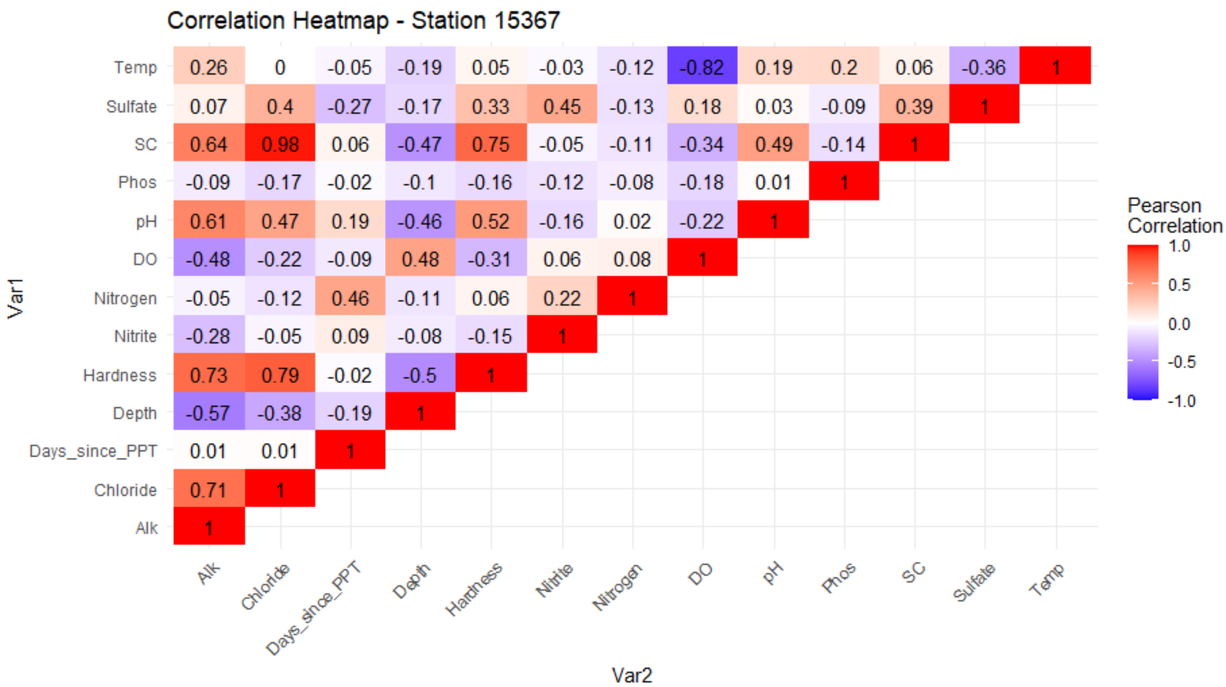


Order of Subbasins



Station 15367 (Segment 607)
Pine Island Bayou Subbasin

Relationship between Water Quality Parameters

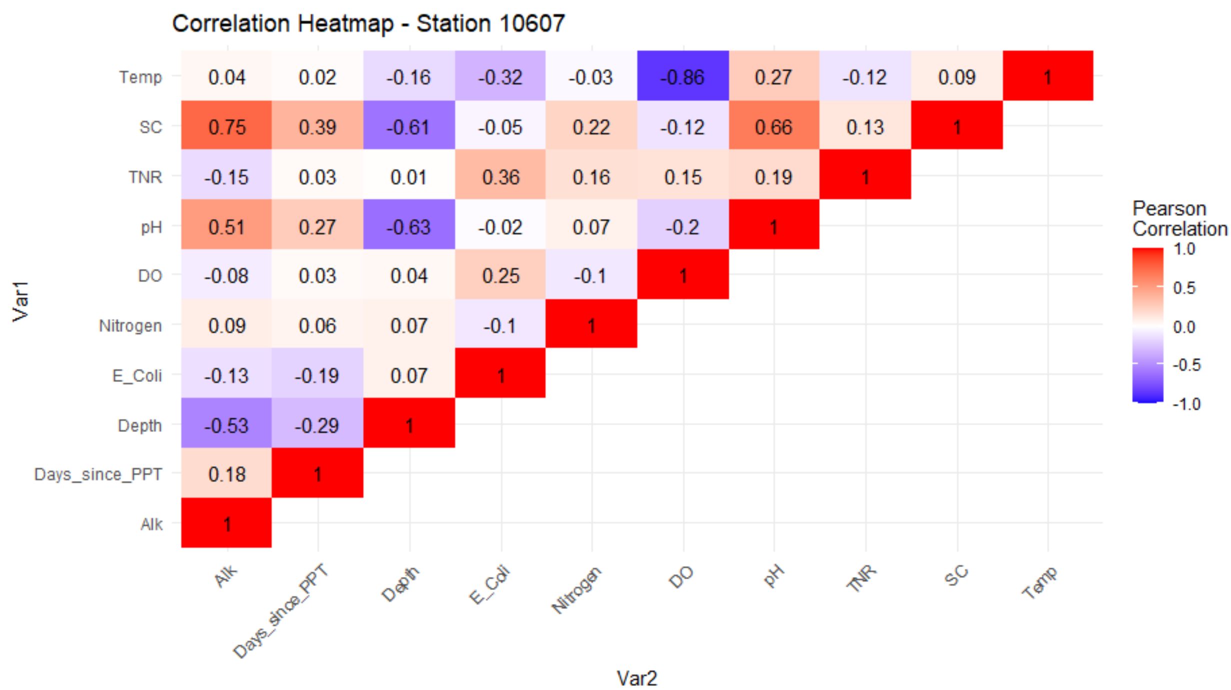


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	-0.567
Chloride	-0.377
Days Since PPT	-0.189
Hardness	-0.5
Nitrite	-0.084
Nitrogen	-0.111
DO	0.481
pH	-0.46
Phos	-0.101
SC	-0.474
Sulfate	-0.168
Temp	-0.187

Station 10607 (Segment 607)
Pine Island Bayou Subbasin

Relationship between Water Quality Parameters

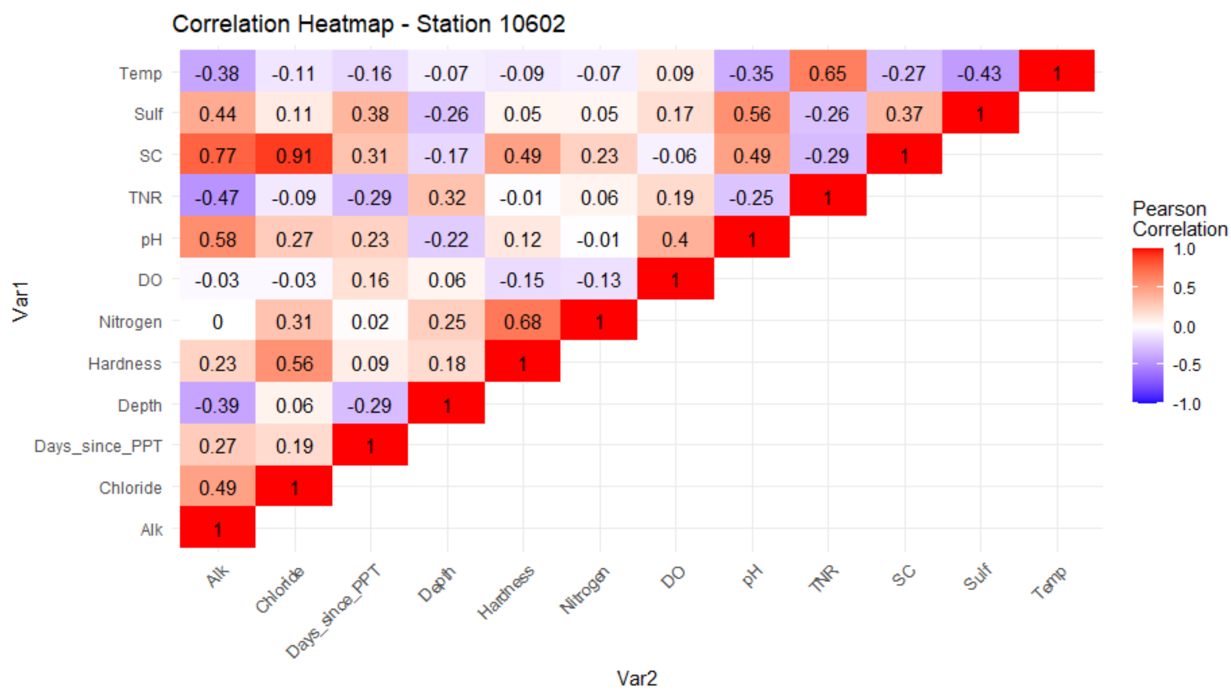


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	-0.533
Days Since PPT	-0.294
E. Coli	0.068
Nitrogen	0.071
DO	0.039
pH	-0.628
TNR	0.006
SC	-0.607
Temp	-0.158

Station 10602 (Segment 607)
Pine Island Bayou Subbasin

Relationship between Water Quality Parameters

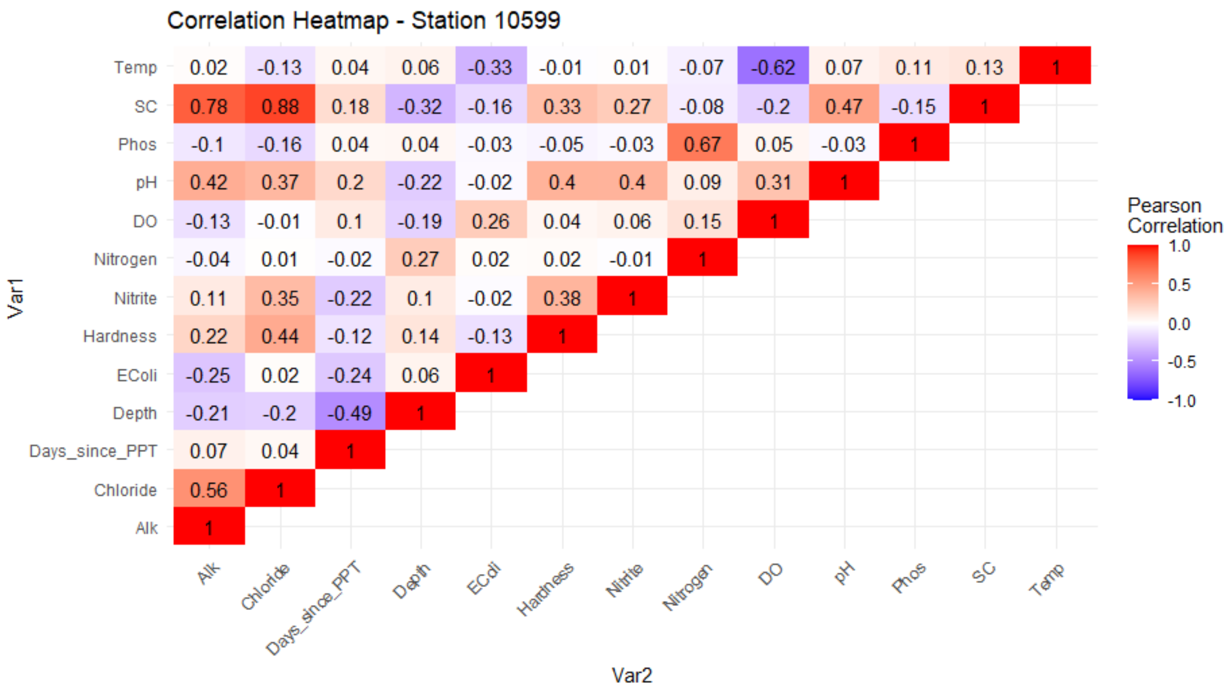


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	0.266
Chloride	0.187
Depth	-0.286
Hardness	0.088
Nitrogen	0.019
DO	0.157
pH	0.235
TNR	-0.288
SC	0.307
Sulf	0.38
Temp	-0.165

Station 10599 (Segment 607)
Pine Island Bayou Subbasin

Relationship between Water Quality Parameters

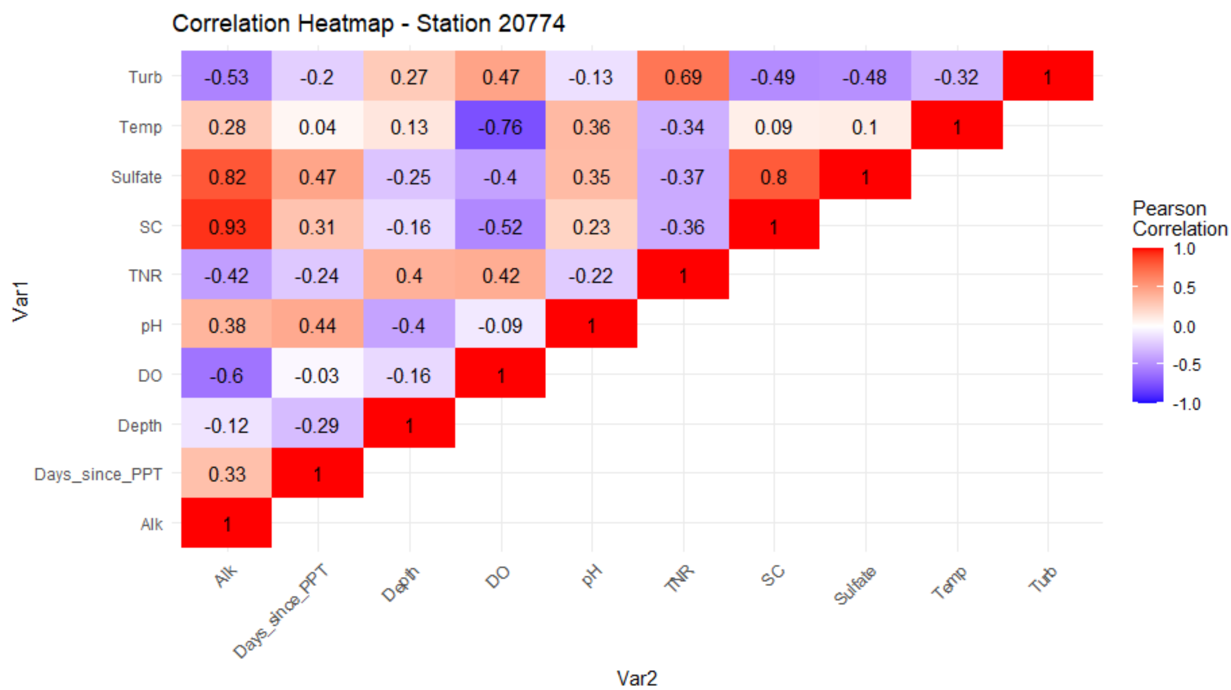


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	0.072
Chloride	0.038
Depth	-0.495
E. Coli	-0.236
Hardness	-0.12
Nitrite	-0.221
Nitrogen	-0.021
DO	0.105
pH	0.199
Phos	0.038
SC	0.176
Temp	0.044

Station 20774 (Segment 601)

Relationship between Water Quality Parameters

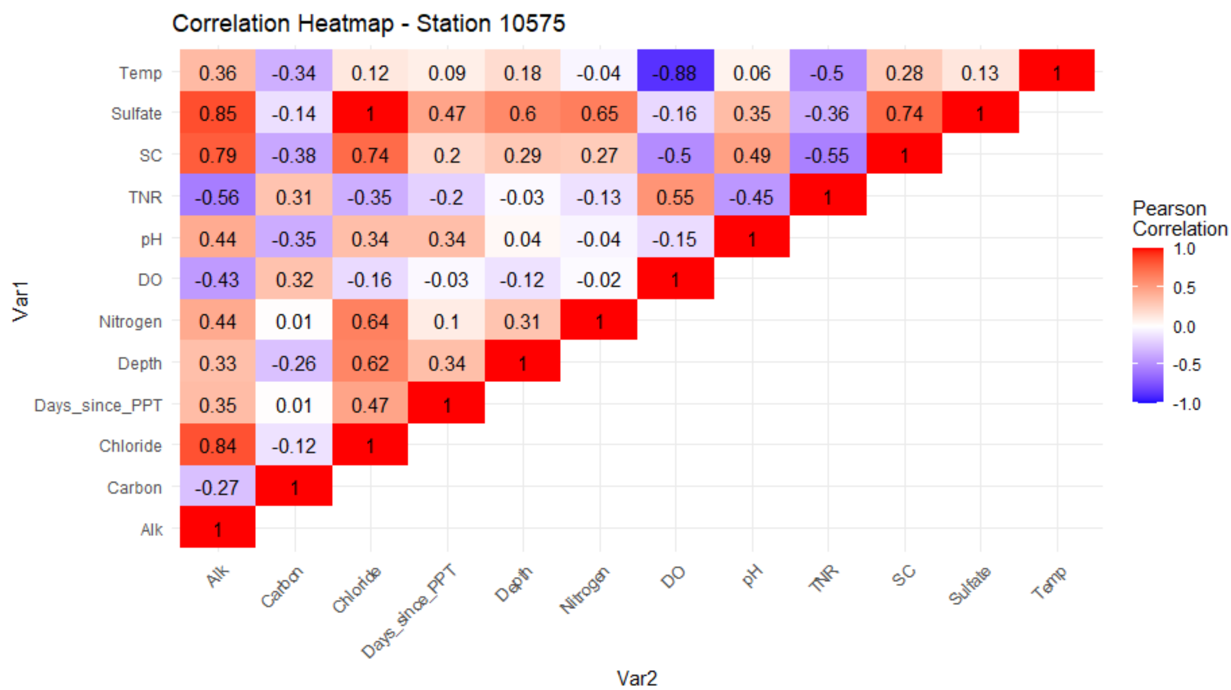


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	-0.12
Days Since PPT	-0.288
DO	-0.161
pH	-0.396
TNR	0.402
SC	-0.156
Sulfate	-0.251
Temp	0.129
Turb	0.271

Station 10575 (Segment 601)

Relationship between Water Quality Parameters

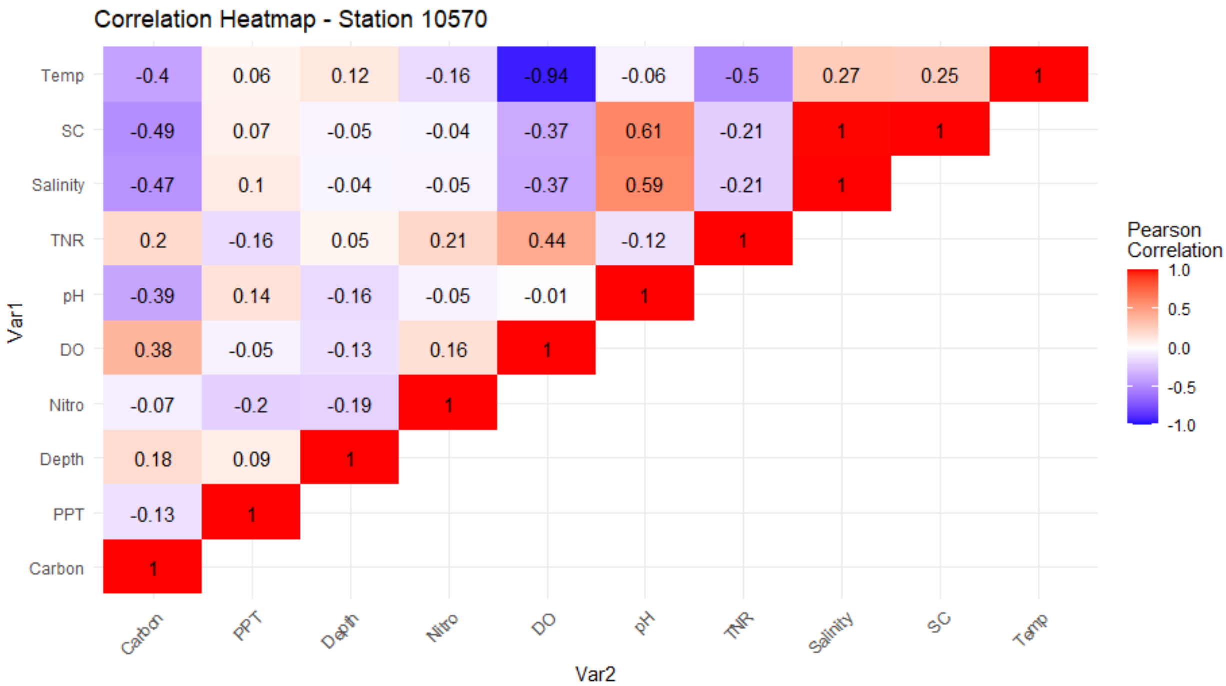


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	0.332
Carbon	-0.264
Chloride	0.616
Days Since PPT	0.336
Nitrogen	0.307
DO	-0.123
pH	0.036
TNR	-0.025
SC	0.289
Sulfate	0.6
Temp	0.183

Station 10570 (Segment 601)

Relationship between Water Quality Parameters

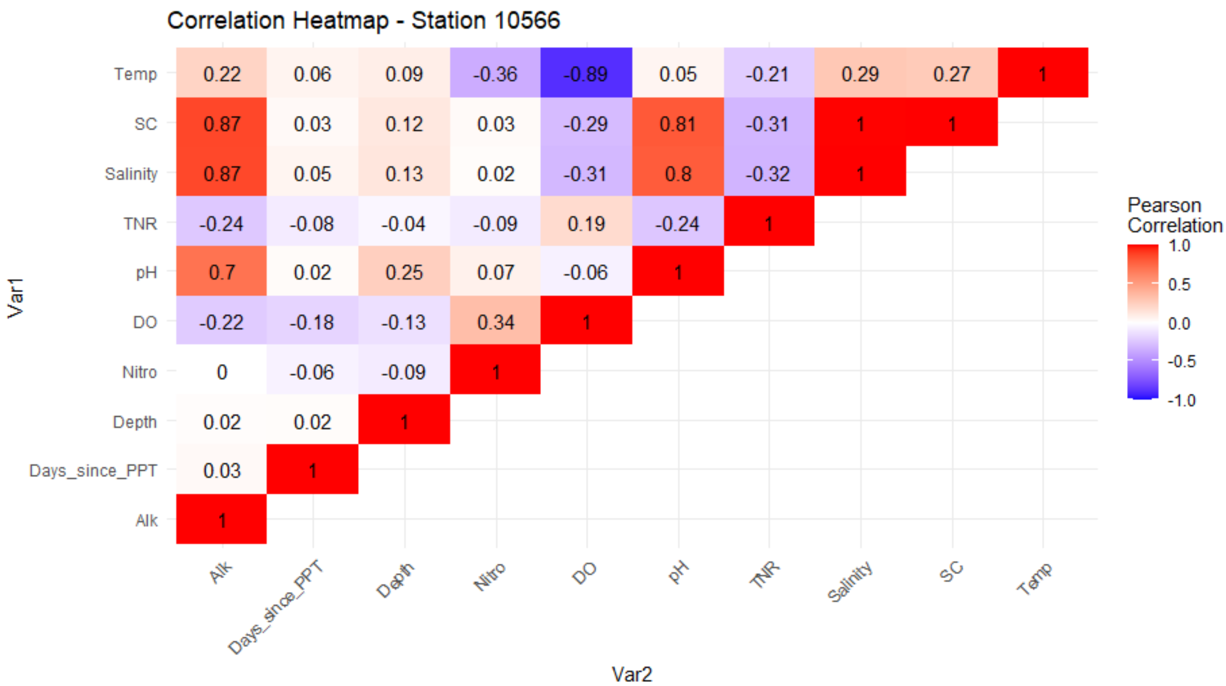


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Carbon	0.183609
PPT	0.089135
Nitro	-0.19191
DO	-0.13298
pH	-0.15715
TNR	0.051886
Salinity	-0.04338
SC	-0.04704
Temp	0.120695

Station 10566 (Segment 601)

Relationship between Water Quality Parameters

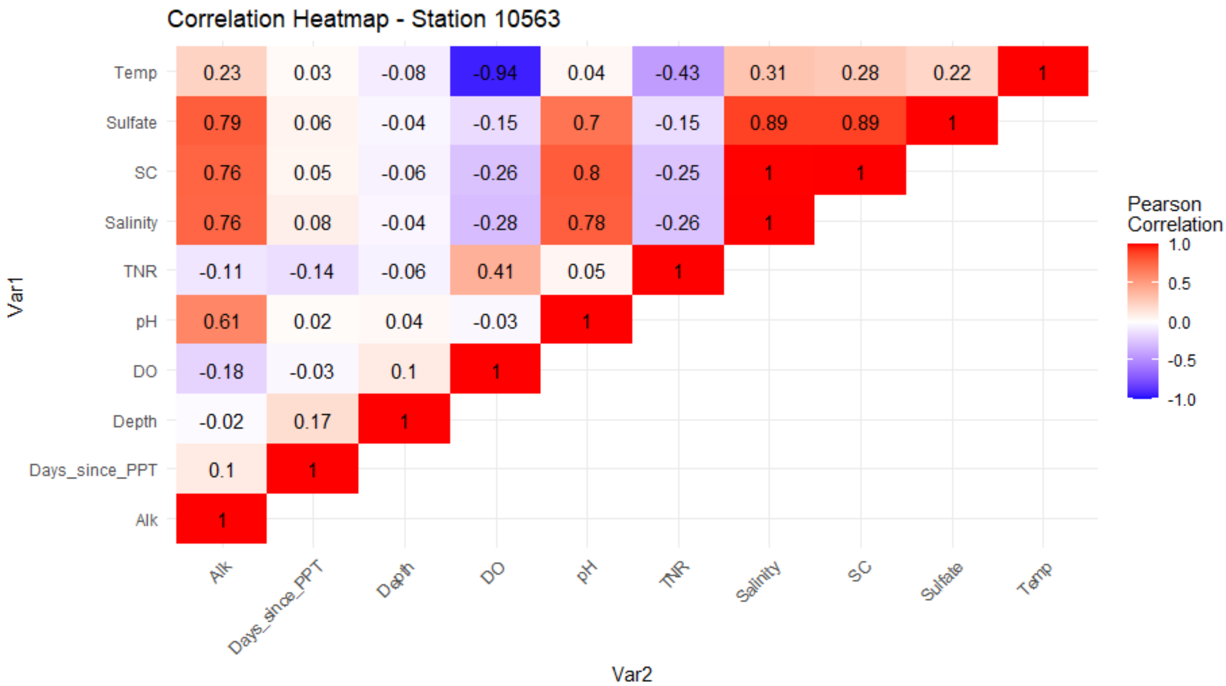


Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	0.016619
Days since PPT	0.017498
Nitro	-0.08567
DO	-0.13223
pH	0.249057
TNR	-0.03677
Salinity	0.132302
SC	0.120113
Temp	0.089887

Station 10563 (Segment 601)

Relationship between Water Quality Parameters



Relationship of Water Quality Parameters with Flow

Parameter	Correlation with Flow
Alkalinity	-0.021
Days Since PPT	0.173
DO	0.104
pH	0.035
TNR	-0.057
Salinity	-0.041
SC	-0.055
Sulfate	-0.037
Temp	-0.076

Summary

- All the codes and aggregated results and correlation parameters are arranged in the GitHub Repository 'WaterQualityModelling' linked below:
    - https://github.com/aaloksk/WaterQualityModeling

- The strongest correlation was observed between the parameters Salinity-Alkalinity and TDS-Specific Conductivity, which are to be expected.
- Strong negative correlation was observed between Total Dissolved Solid and Dissolved Oxygen.
- Chloride shows a strong positive correlation with Sulfate and Specific Conductivity.
- pH and Depth of Sample has negative correlation in most instances.
- Water Quality Parameters are not showing consistent trend of correlation with flow on any of the stations.

Days since last precipitation(DSLP) was also considered as one of the parameters during this analysis. It can have effects on various water quality parameters. For instance, if the period without precipitation is prolonged, the level of dissolved oxygen in a river can decrease, particularly in shallow and slow-moving areas. This is because oxygen enters the water primarily through surface exchange, which can be limited when there is little or no surface disturbance due to low flow. For all the stations in Neches River Segment 601, DSLP has a consistent negative correlation with the dissolved oxygen. However, this correlation is negated if there are not point sources of low-DO water discharged into the water, which can be the case for some station on lower end of Pine Island Bayou stations. Similarly, a positive correlation can be observed between pH and DSLP. This can be an indicator of alkaline water being discharged into the river as a pollutant which is diluted when rain occurs.

Understanding the correlation between water flow and other quality parameters can be essential for managing water quality, habitat quality, floods, and water resources. Such information can be utilized to make informed decisions about how to manage rivers in a way that protects both the environment and human populations.

Water Parameter Standards

2022 Texas Integrated Report of Water Quality Impairments has assigned categories of 5a and 5b to Segment 601 and 607, respectively. Segment 601 has issues of bacteria in water (recreational use) and PCBs in edible tissue. On the other hand, Segment 607 (Pine Island Bayou) has depressed dissolved oxygen in water.

Neches River (Segment 1) has a Total Maximum Daily Loads (TMDL) allocation standards set by stakeholders of the Hillebrandt Bayou and Neches River and adopted by TCEQ on August 11, 2021.

	601-01	601-02	601-03	601-04
TMDL	21974	22231	22842	24761
Margin of Safety	1099	1112	1142	1238
Wasteload Allocation for WWTFs	86	118	125	144
Wasteload Allocation for stormwater	4237	4907	5451	5439
Load Allocation	16531	16064	16093	17903
Future Growth	21.6	29.6	31.3	36.2

Designated Uses and Numeric Criteria

Segment	Aquatic Life Use	DO (mg/L)	pH (SU)	Temp (F)	Ind. Bacteria #/100 mL	Cl^{-1} (mg/L)	SO_4^{-2} (mg/L)	TDS (mg/L)
601	I	3.0	6.0 – 8.5	95	35			
607	H	5.0	6.0 – 8.5	95	126	150	50	300

Appendix A

Code for Part A

Import Necessary Libraries

In [87]:
```
suppressWarnings({
library(dplyr)
library(xts)
library(ggplot2)
library(tidyr)
})
```

Setting Working Directory

In [88]:
```
path <- 'C:\\Users\\Aalok\\OneDrive - lamar.edu\\000Water_Q_Modelling\\WD'
setwd(path)
```

Importing file with water quality parameters

In [89]:
```
a <- read.csv('Combined_PIB_WQ.csv')
colnames(a) <- c('Date', 'Time', 'TempC','Depth','SpCond','WatTurb','TDS','DisOx','
print(head(a))
```

```
        Date     Time   TempC     Depth SpCond  WatTurb    TDS  DisOx      pH
1 2008-07-01 00:00:00 27.6AQI 0.700AQI 173AQI 53.81AQI 112AQI 3.9AQI 6.7AQI
2 2008-07-01 00:15:00 27.5AQI 0.600AQI 175AQI 54.51AQI 114AQI 3.7AQI 6.6AQI
3 2008-07-01 00:30:00 27.5AQI 0.700AQI 175AQI 54.21AQI 114AQI 3.7AQI 6.6AQI
4 2008-07-01 00:45:00 27.4AQI 0.700AQI 175AQI 54.60AQI 114AQI 3.7AQI 6.6AQI
5 2008-07-01 01:00:00 27.4AQI 0.700AQI 174AQI 55.10AQI 113AQI 3.6AQI 6.7AQI
6 2008-07-01 01:15:00 27.4AQI 0.600AQI 175AQI 54.81AQI 114AQI 3.5AQI 6.6AQI
```

Reformatting date and time in a new column

In [90]:
```
a$datetime <- paste(a$Date, a$Time)
a$datetime <- as.POSIXct(a$datetime, format = "%Y-%m-%d %H:%M:%S")
a$datetime <- format(a$datetime, "%d/%m/%Y %H:%M")
head(a)
```

A data.frame: 6 × 10

	Date	Time	TempC	Depth	SpCond	WatTurb	TDS	DisOx	pH	datetime
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	2008-07-01	00:00:00	27.6AQI	0.700AQI	173AQI	53.81AQI	112AQI	3.9AQI	6.7AQI	01/07/2008 00:00
2	2008-07-01	00:15:00	27.5AQI	0.600AQI	175AQI	54.51AQI	114AQI	3.7AQI	6.6AQI	01/07/2008 00:15
3	2008-07-01	00:30:00	27.5AQI	0.700AQI	175AQI	54.21AQI	114AQI	3.7AQI	6.6AQI	01/07/2008 00:30
4	2008-07-01	00:45:00	27.4AQI	0.700AQI	175AQI	54.60AQI	114AQI	3.7AQI	6.6AQI	01/07/2008 00:45
5	2008-07-01	01:00:00	27.4AQI	0.700AQI	174AQI	55.10AQI	113AQI	3.6AQI	6.7AQI	01/07/2008 01:00
6	2008-07-01	01:15:00	27.4AQI	0.600AQI	175AQI	54.81AQI	114AQI	3.5AQI	6.6AQI	01/07/2008 01:15

Remove words from the dataframe and extract values only

In [91]:
```
my_df <- a %>%
  mutate_at(vars('TempC','Depth','SpCond','WatTurb','TDS','DisOx','pH'), ~ as.numer
print(head(my_df))
```

```
        Date     Time TempC Depth SpCond WatTurb TDS DisOx  pH         datetime
1 2008-07-01 00:00:00  27.6   0.7    173   53.81 112   3.9 6.7 01/07/2008 00:00
2 2008-07-01 00:15:00  27.5   0.6    175   54.51 114   3.7 6.6 01/07/2008 00:15
3 2008-07-01 00:30:00  27.5   0.7    175   54.21 114   3.7 6.6 01/07/2008 00:30
4 2008-07-01 00:45:00  27.4   0.7    175   54.60 114   3.7 6.6 01/07/2008 00:45
5 2008-07-01 01:00:00  27.4   0.7    174   55.10 113   3.6 6.7 01/07/2008 01:00
6 2008-07-01 01:15:00  27.4   0.6    175   54.81 114   3.5 6.6 01/07/2008 01:15
```

Import the file with flow data

In [92]:
```
b <- read.csv('FlowData_15Min Interval.csv')
print(head(b))
```

```
  agency_cd site_no       datetime tz_cd flow_cfs
1      USGS 8041749 1/10/2003 0:00   CDT     -370
2      USGS 8041749 1/10/2003 0:15   CDT     -371
3      USGS 8041749 1/10/2003 0:30   CDT     -290
4      USGS 8041749 1/10/2003 0:45   CDT     -291
5      USGS 8041749 1/10/2003 1:00   CDT     -347
6      USGS 8041749 1/10/2003 1:15   CDT     -376
```

Reformat DateTime

In [93]:
```
b$datetime <- as.POSIXct(b$datetime, format = "%d/%m/%Y %H:%M")
b$datetime <- format(b$datetime, "%d/%m/%Y %H:%M")
print(head(b))
```

```
  agency_cd site_no         datetime tz_cd flow_cfs
1      USGS 8041749 01/10/2003 00:00   CDT     -370
2      USGS 8041749 01/10/2003 00:15   CDT     -371
3      USGS 8041749 01/10/2003 00:30   CDT     -290
4      USGS 8041749 01/10/2003 00:45   CDT     -291
5      USGS 8041749 01/10/2003 01:00   CDT     -347
6      USGS 8041749 01/10/2003 01:15   CDT     -376
```

Merging dataframes based on common column

In [94]:
```
merged_df <- merge(b, my_df, by = "datetime", all = TRUE)
merged_df[5001:5005,]
```

A data.frame: 5 × 14

	datetime	agency_cd	site_no	tz_cd	flow_cfs	Date	Time	TempC	Depth	SpC
	<chr>	<chr>	<int>	<chr>	<dbl>	<chr>	<chr>	<dbl>	<dbl>	<c
5001	01/03/2016 02:00	USGS	8041749	CST	-90.80	2016-03-01	02:00:00	17.1	0.887	
5002	01/03/2016 02:15	USGS	8041749	CST	-44.20	2016-03-01	02:15:00	17.1	0.883	
5003	01/03/2016 02:30	USGS	8041749	CST	-21.00	2016-03-01	02:30:00	17.0	0.877	
5004	01/03/2016 02:45	USGS	8041749	CST	-90.60	2016-03-01	02:45:00	17.0	0.872	
5005	01/03/2016 03:00	USGS	8041749	CST	2.26	2016-03-01	03:00:00	16.9	0.866	

Extracting parameters to find correlation into a separate dataframe

In [95]:
```
df <- merged_df[, c("TempC","Depth","SpCond","WatTurb","TDS","DisOx","pH","flow_cfs
```

Removing Outliers

In [96]:
```
df[df == 1000000] <- NA
df$TempC[df$TempC > 10000] <- NA
df$Depth[df$Depth > 30] <- NA
df$SpCond[df$SpCond > 550] <- NA
df$WatTurb[df$WatTurb > 500] <- NA
df$TDS[df$TDS > 2500] <- NA
df$DisOx[df$DisOx > 20] <- NA
df$pH[df$pH > 14] <- NA
```

Calculate Correlation

In [97]:
```
cor_matrix <- cor(df, use = "pairwise.complete.obs")
cor_matrix
```

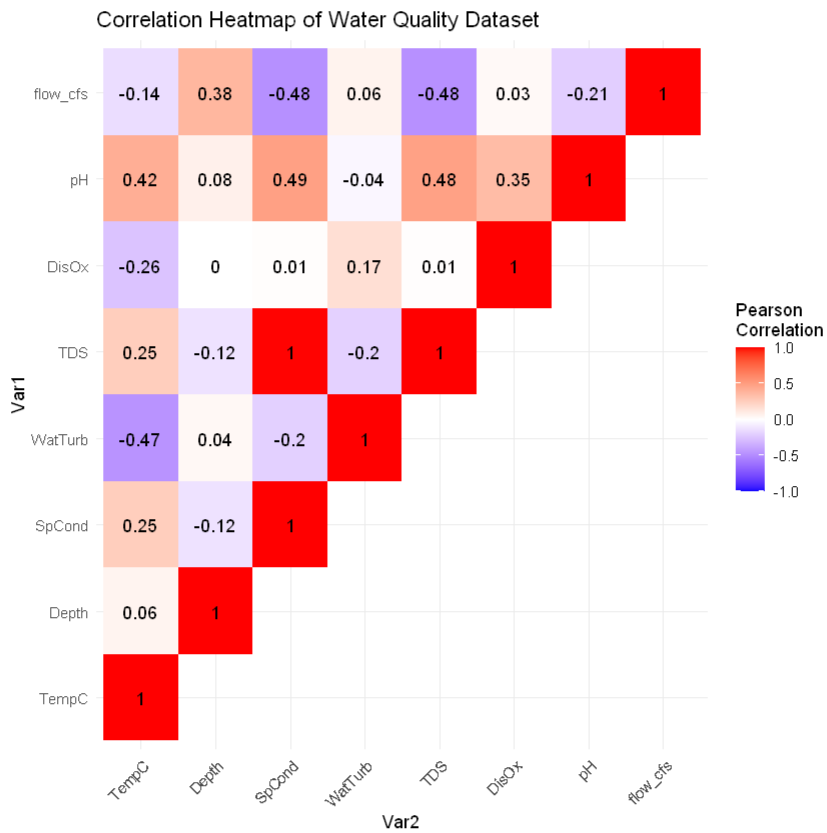
A matrix: 8 × 8 of type dbl

	TempC	Depth	SpCond	WatTurb	TDS	DisOx	p
TempC	1.0000000	0.056179896	0.25485952	-0.46548762	0.254805523	-0.256702102	0.420396
Depth	0.0561799	1.000000000	-0.12469103	0.03643466	-0.124830216	-0.000760466	0.076696
SpCond	0.2548595	-0.124691028	1.00000000	-0.19981323	0.997919078	0.010456146	0.488304
WatTurb	-0.4654876	0.036434656	-0.19981323	1.00000000	-0.197628711	0.167101061	-0.038016
TDS	0.2548055	-0.124830216	0.99791908	-0.19762871	1.000000000	0.008783385	0.483775
DisOx	-0.2567021	-0.000760466	0.01045615	0.16710106	0.008783385	1.000000000	0.349921
pH	0.4203964	0.076696911	0.48830455	-0.03801656	0.483775202	0.349921933	1.000000
flow_cfs	-0.1446149	0.375503874	-0.48099239	0.06368365	-0.477728057	0.028991180	-0.209966

Plot Heatmap of correlation matrix

In [98]:

```
# Create a lower triangular matrix with NA in the upper triangle
lower_tri <- cor_matrix
lower_tri[upper.tri(cor_matrix)] <- NA

# Melt the lower triangular matrix and remove NA values
library(reshape2)
melted_cor <- melt(lower_tri, na.rm = TRUE)

# Create a correlation heatmap using ggplot2
ggplot(data = melted_cor, aes(x=Var2, y=Var1, fill=value, label = round(value, 2)))
  geom_tile() +
  geom_text(color = "black") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Correlation Heatmap of Water Quality Dataset")
```



Preparing the dataframe with datetime to perform data aggregation

In [99]:
```
df <- merged_df[, c("datetime","TempC","Depth","SpCond","WatTurb","TDS","DisOx","pH

#Outlier Removal
df[df == 1000000] <- NA
df$TempC[df$TempC > 10000] <- NA
df$Depth[df$Depth > 30] <- NA
df$SpCond[df$SpCond > 550] <- NA
df$WatTurb[df$WatTurb > 500] <- NA
df$TDS[df$TDS > 2500] <- NA
df$DisOx[df$DisOx > 20] <- NA
df$pH[df$pH > 14] <- NA
```

Remove rows without flow data and reformat datetime

In [100…
```
df <- df[complete.cases(df$datetime), ]
df$datetime <- as.POSIXct(df$datetime, format = "%d/%m/%Y %H:%M")
```

Convert to xts object with 15-minute intervals

In [101…
```
xts_data <- xts(df[,2:9], order.by = df$datetime) #Line 2 to 9 includes all the par
```

Create hourly data aggregation and a respective dataframe

In [102…
```
hourly_data <- aggregate(xts_data, as.POSIXct(cut(index(xts_data), breaks="hour")),
hourly_df <- as.data.frame(hourly_data)
```

Create daily data aggregation and a respective dataframe

In [103…
```
daily_data <- aggregate(xts_data, as.Date(index(xts_data)), mean)
daily_df <- as.data.frame(daily_data)
```

Create daily minimum data aggregation and a respective dataframe

In [104…
```
daily_min <- aggregate(xts_data, as.Date(index(xts_data)), min)
dailymin_df <- as.data.frame(daily_min)
```

Create daily maximum data aggregation and a respective dataframe

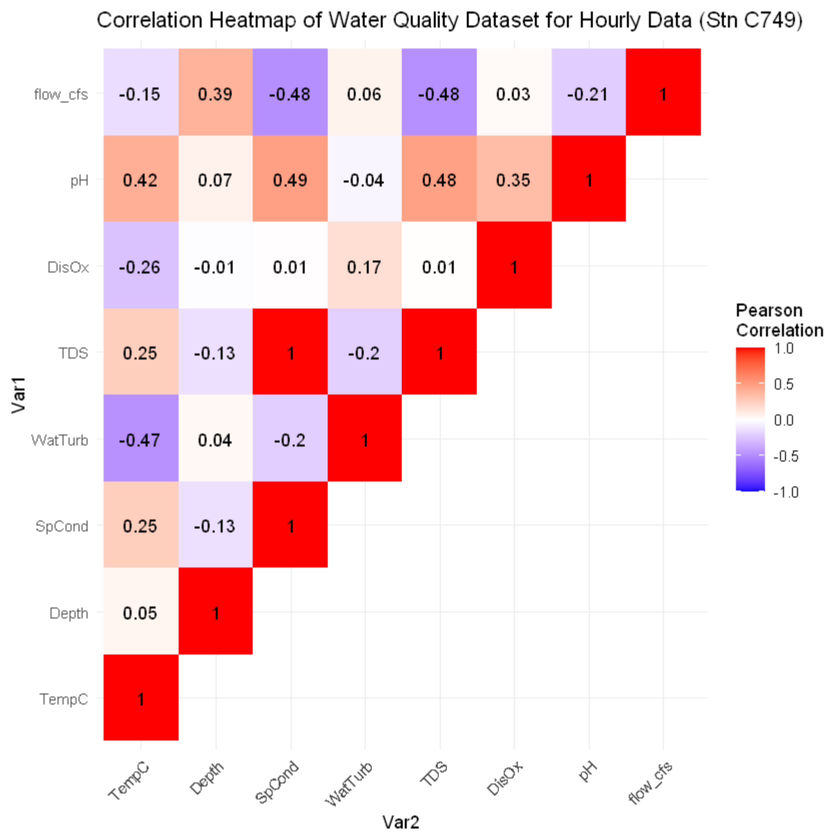
In [105…
```
daily_max <- aggregate(xts_data, as.Date(index(xts_data)), max)
dailymax_df <- as.data.frame(daily_max)
```

Correlation for each aggregation scenario

In [106…
```
cor3 <- cor(daily_df, use = "pairwise.complete.obs")
cor4 <- cor(dailymin_df, use = "pairwise.complete.obs")
cor5 <- cor(dailymax_df, use = "pairwise.complete.obs")
```

Hourly Data

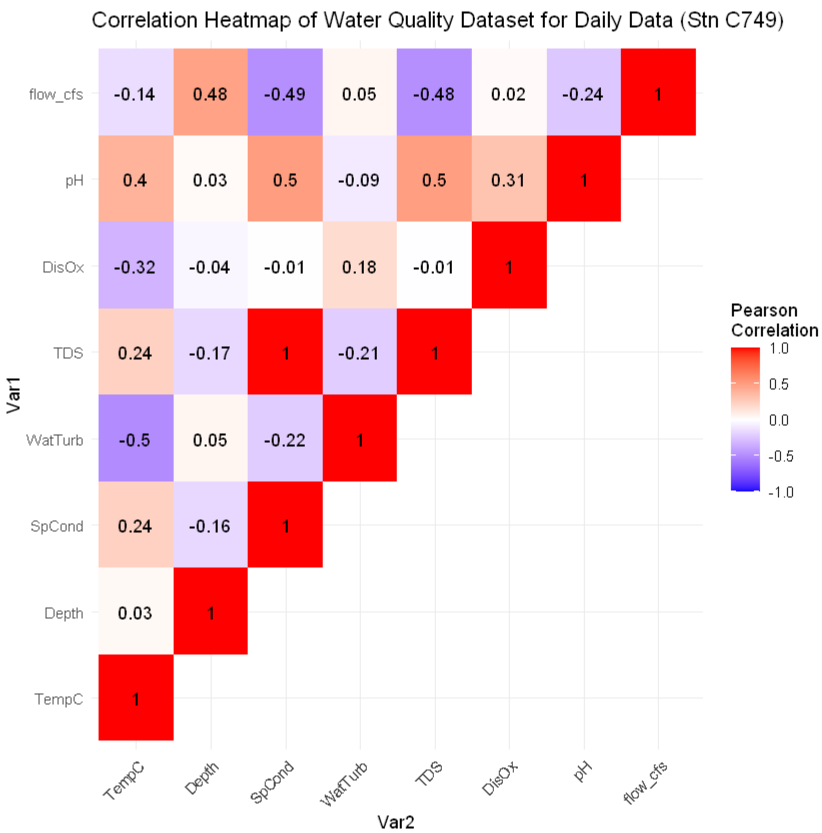
In [107…
```
#Calculate correlation
cor2 <- cor(hourly_df, use = "pairwise.complete.obs")

# Create a lower triangular matrix with NA in the upper triangle
lower_tri <- cor2
lower_tri[upper.tri(cor2)] <- NA

# Melt the lower triangular matrix and remove NA values
melted_cor <- melt(lower_tri, na.rm = TRUE)

# Create a correlation heatmap using ggplot2
ggplot(data = melted_cor, aes(x=Var2, y=Var1, fill=value, label = round(value, 2)))
  geom_tile() +
  geom_text(color = "black") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Correlation Heatmap of Water Quality Dataset for Hourly Data (Stn C749)"
```



Daily Data

In [108…

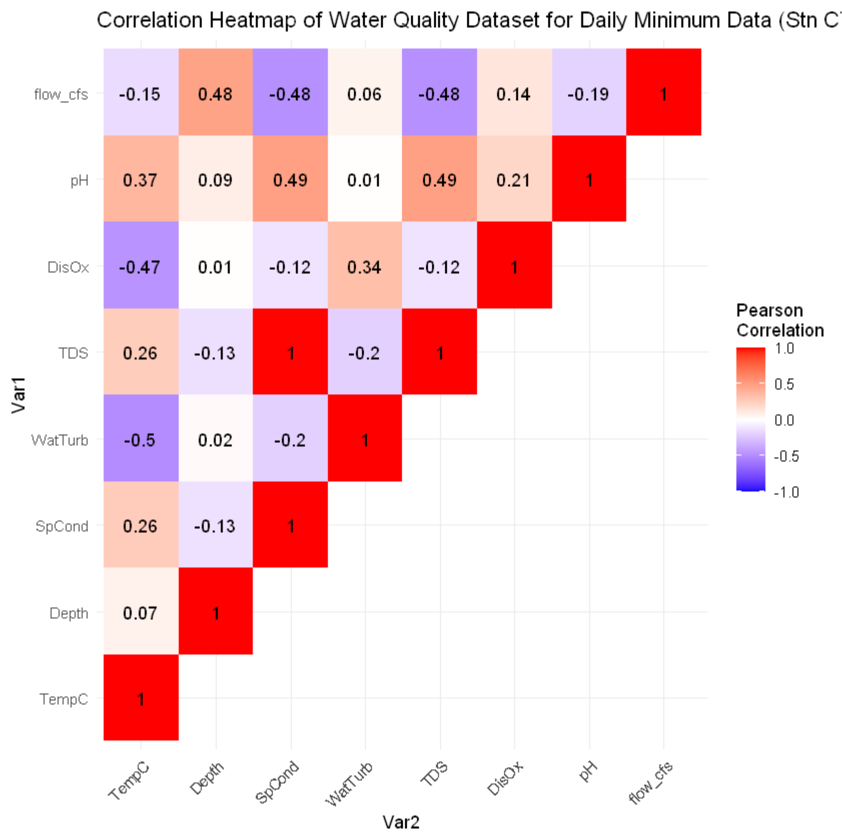
```r
# Create a lower triangular matrix with NA in the upper triangle
lower_tri <- cor3
lower_tri[upper.tri(cor3)] <- NA

# Melt the lower triangular matrix and remove NA values
melted_cor <- melt(lower_tri, na.rm = TRUE)

# Create a correlation heatmap using ggplot2
ggplot(data = melted_cor, aes(x=Var2, y=Var1, fill=value, label = round(value, 2)))
  geom_tile() +
  geom_text(color = "black") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Correlation Heatmap of Water Quality Dataset for Daily Data (Stn C749)")
```



Daily Minimum Data

In [109…

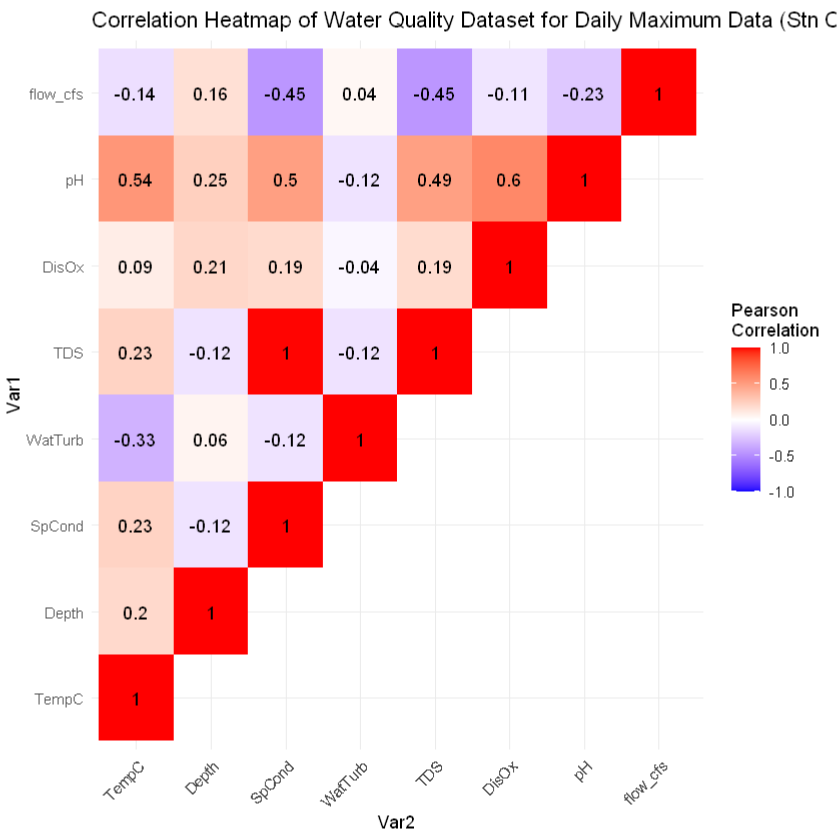
```
# Create a lower triangular matrix with NA in the upper triangle
lower_tri <- cor4
lower_tri[upper.tri(cor4)] <- NA

# Melt the lower triangular matrix and remove NA values
melted_cor <- melt(lower_tri, na.rm = TRUE)

# Create a correlation heatmap using ggplot2
ggplot(data = melted_cor, aes(x=Var2, y=Var1, fill=value, label = round(value, 2)))
  geom_tile() +
  geom_text(color = "black") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Correlation Heatmap of Water Quality Dataset for Daily Minimum Data (Stn
```



Daily Maximum Data

In [110…

```
# Create a lower triangular matrix with NA in the upper triangle
lower_tri <- cor5
lower_tri[upper.tri(cor5)] <- NA

# Melt the lower triangular matrix and remove NA values
melted_cor <- melt(lower_tri, na.rm = TRUE)

# Create a correlation heatmap using ggplot2
ggplot(data = melted_cor, aes(x=Var2, y=Var1, fill=value, label = round(value, 2)))
  geom_tile() +
  geom_text(color = "black") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Correlation Heatmap of Water Quality Dataset for Daily Maximum Data (Stn
```


Correlation Heatmap of Water Quality Dataset for Daily Maximum Data (Stn C
flow_cfs
pH
DisOx
TDS
WatTurb
SpCond
Depth
TempC
Var1
-0.14
0.16
-0.45
0.04
-0.45
-0.11
-0.23
1
0.54
0.25
0.5
-0.12
0.49
0.6
1
0.09
0.21
0.19
-0.04
0.19
1
0.23
-0.12
1
-0.12
1
-0.33
0.06
-0.12
1
0.23
-0.12
1
0.2
1
1
TempC
Depth
SpCond
WatTurb
TDS
DisOx
pH
flow_cfs
Var2
Pearson
Correlation
1.0
0.5
0.0
-0.5
-1.0