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

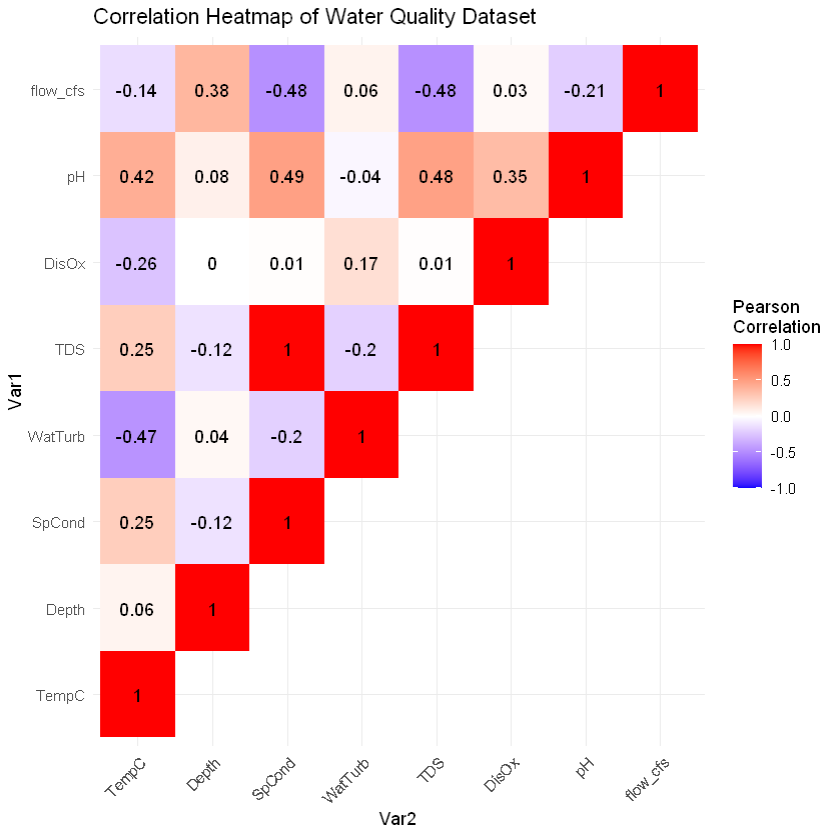
	TempC	Depth	SpCond	WatTurb	TDS	DisOx	p
TempC	1.0000000	0.056179896	0.25485952	-0.46548762	0.254805523	-0.256702102	0.420396
Depth	0.0561799	1.000000000	-0.12469103	0.03643466	-0.124830216	-0.000760466	0.076696
SpCond	0.2548595	-0.124691028	1.00000000	-0.19981323	0.997919078	0.010456146	0.488304
WatTurb	-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")
```

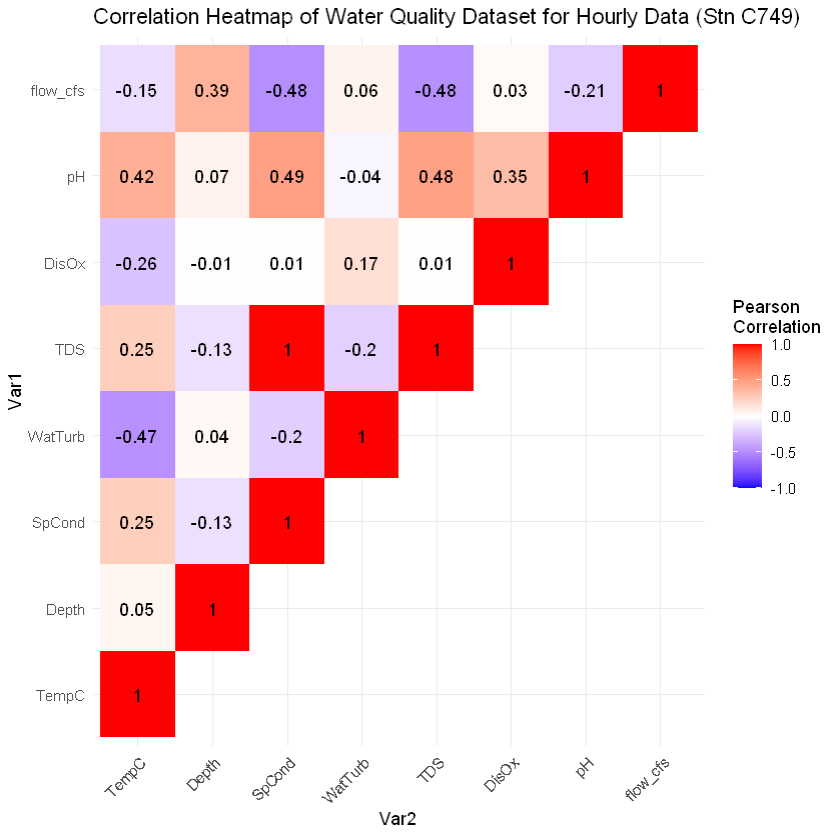
Hourly Data

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)"
```



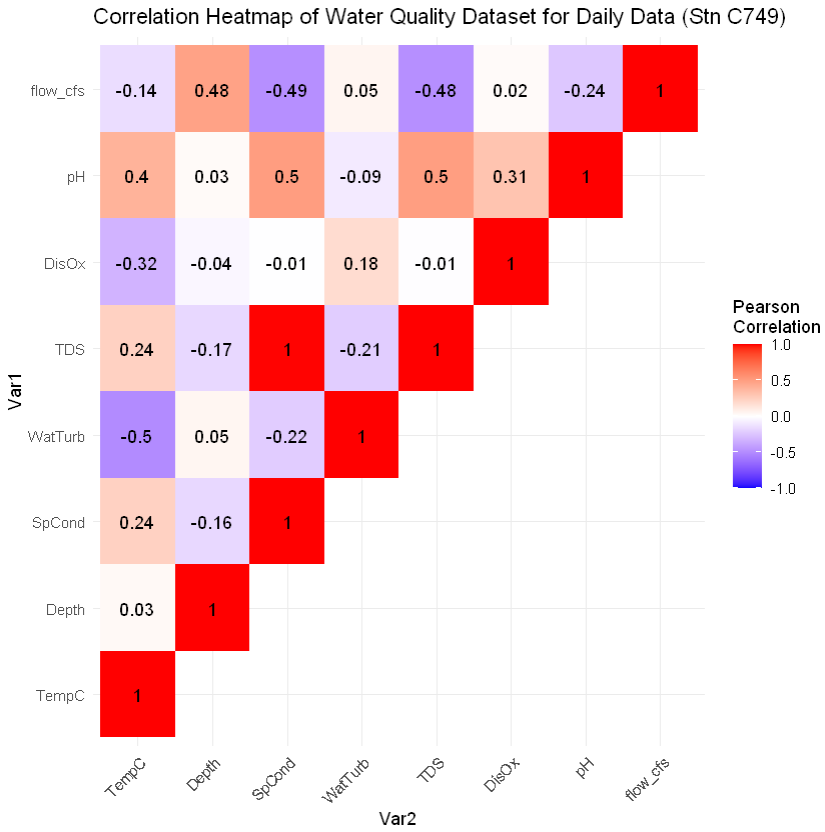
Daily Data

In [108…

```
# 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)")
```



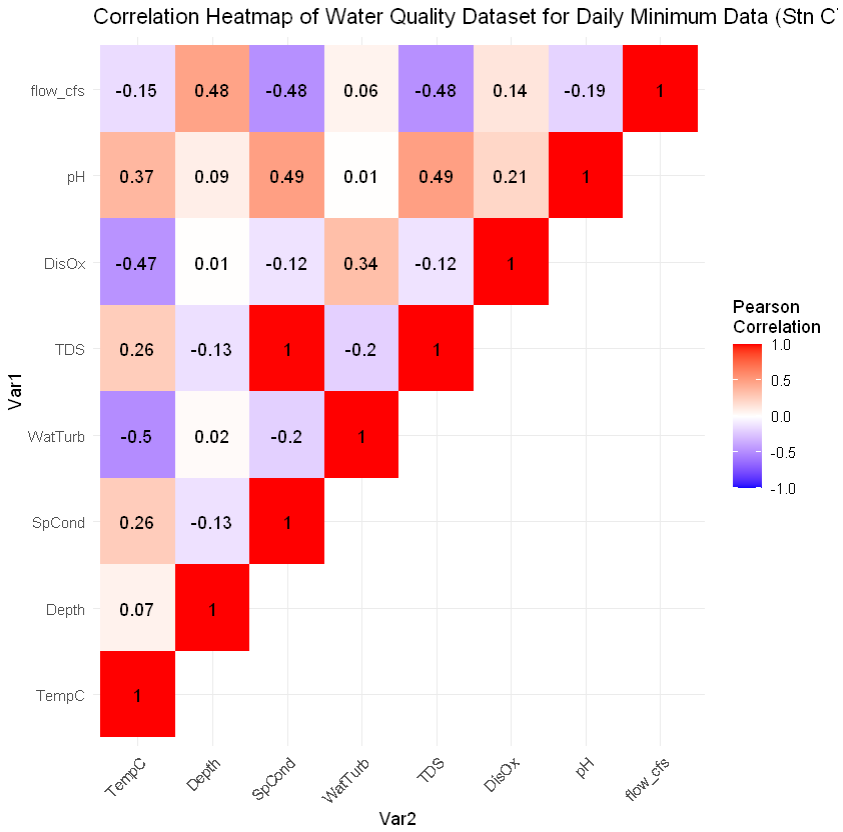
Daily Minimum Data

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
```



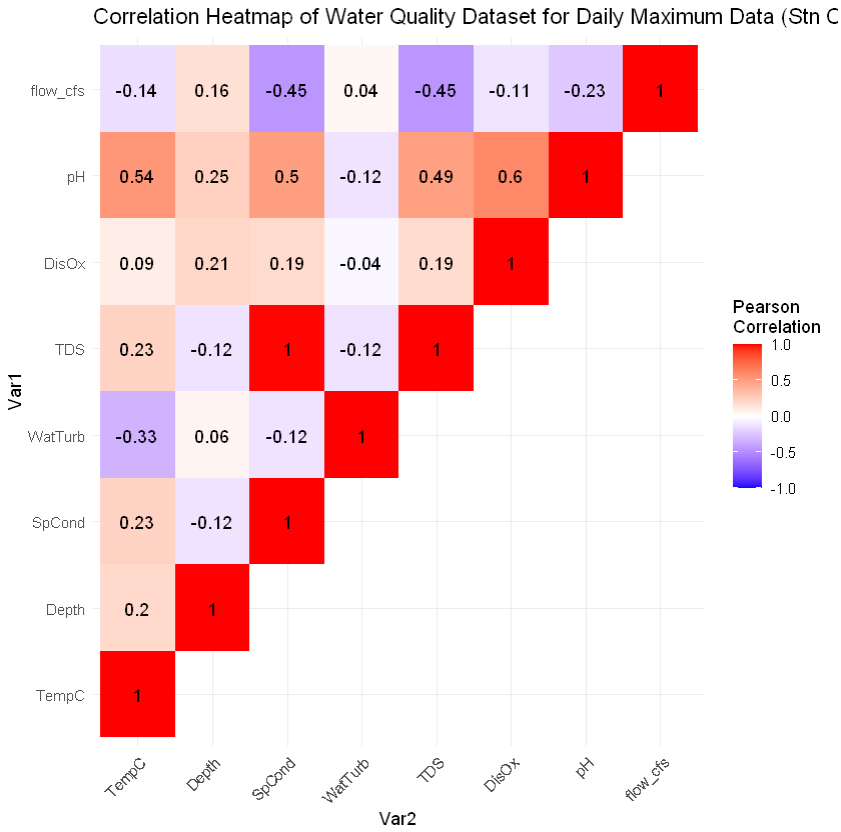
Daily Maximum Data

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
-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
Var1
TempC
Depth
SpCond
WatTurb
TDS
DisOx
pH
flow_cfs
Var2
Pearson Correlation
1.0
0.5
0.0
-0.5
-1.0