W2_CQ_Analysis

Rob Rioux

2024-12-03

Part I: Processing storm concentration - discharge data

Our dataset is 612 observations of 46 variables, of stormwater chemistry from a small, headwater stream found in North Danville, VT. This data is part of a larger set of data our lab has been collecting that includes baseline streamflow, tile drain discharge, and groundwater wells. We will focus on the Northern Weir for this analysis as this has stormwater samples before our basalt application, and we will look at a narrower range of solutes (14). Part I of this script is loading in and preparing the data for analysis.

```r
# Read in the data
W2_Stormwater <- read_excel("/Users/robertrioux/Desktop/W2_Stormwater_2022-2023.xlsx")
names(W2_Stormwater)
```

```
##  [1] "Rain_Event_Number"     "Date"                  "Time"
##  [4] "DateTime"              "ISCO_Location"         "Total_Sample_Count"
##  [7] "Q"                     "Temp_C"                "pH"
## [10] "SpCond_uS_per_cm"      "HDO_ppm"               "HDO_perc"
## [13] "Turb"                  "Chl-a"                 "CDOM"
## [16] "CDOM2"                 "F_ppm"                 "Cl_ppm"
## [19] "N-NO2_ppm"             "Br_ppm"                "N-NO3_ppm"
## [22] "P-PO4_ppm"             "S-SO4_ppm"             "Na_ppm"
## [25] "NH4_ppm"               "K_ppm"                 "Mg_ppm"
## [28] "Ca_ppm"                "npoc conc"             "Total Alkalinity (uM)"
## [31] "Al27ugL"               "AsugL"                 "Ba138ugL"
## [34] "Cd111ugL"              "Cr52ugL"               "Cs133ugL"
## [37] "Cu63ugL"               "Fe57ugL"               "Li7ugL"
## [40] "Mn55ugL"               "Ni60ugL"               "Pb208ugL"
## [43] "Rb85ugL"               "Si28ugL"               "Sr88ugL"
## [46] "Zn66ugL"
```

```r
# Get only complete data for ease of analysis, subset for
# specific data
W2_Stormwater_2 <- W2_Stormwater[, c("Rain_Event_Number", "DateTime",
    "ISCO_Location", "Q", "Cl_ppm", "S-SO4_ppm", "Na_ppm", "Mg_ppm",
    "Ca_ppm", "Total Alkalinity (uM)", "npoc conc", "AsugL",
    "Ba138ugL", "Cr52ugL", "Li7ugL", "Mn55ugL", "Si28ugL", "Sr88ugL")]

W2_Stormwater_2 <- W2_Stormwater_2 |>
    rename(SO4_ppm = `S-SO4_ppm`, DOC = `npoc conc`, Alkalinity_uM = `Total Alkalinity (uM)`,
        As_ugL = AsugL, Ba_ugL = Ba138ugL, Cr_ugL = Cr52ugL,
        Li_ugL = Li7ugL, Mn_ugL = Mn55ugL, Si_ugL = Si28ugL,
```

```r
        Sr_ugL = Sr88ugL)

W2_Stormwater_2 <- W2_Stormwater_2 |>
    mutate(Cl_ppm = replace_na(Cl_ppm, 0.01), SO4_ppm = replace_na(SO4_ppm,
        0.01), Na_ppm = replace_na(Na_ppm, 0.01), Mg_ppm = replace_na(Mg_ppm,
        0.01), Ca_ppm = replace_na(Ca_ppm, 0.01), DOC = replace_na(DOC,
        0.8))

W2_Stormwater_2 <- W2_Stormwater_2 |>
    group_by(Rain_Event_Number) |>
    mutate(Treatment = case_when(Rain_Event_Number %in% c("2022-01",
        "2022-02", "2022-03", "2022-04", "2022-05", "2022-07",
        "2022-08", "2022-09", "2023-01", "2023-02", "2023-03",
        "2023-04") ~ "Pre", TRUE ~ "Post")) |>
    filter(ISCO_Location %in% c("Northern Weir"))  # Southern Weir currently is missing too much data

W2_Stormwater_2 <- W2_Stormwater_2[complete.cases(W2_Stormwater_2),
    ]  # Remove any incomplete cases
W2_Stormwater_2$DateTime <- as.POSIXct(W2_Stormwater_2$DateTime,
    format = "%m/%d/%Y %H:%M", tz = "EST")
W2_Stormwater_2 <- W2_Stormwater_2 |>
    select(Rain_Event_Number, Treatment, everything())


print(W2_Stormwater_2)
```

```
## # A tibble: 354 x 19
## # Groups:   Rain_Event_Number [25]
##    Rain_Event_Number Treatment DateTime            ISCO_Location     Q Cl_ppm
##    <chr>             <chr>     <dttm>              <chr>         <dbl>  <dbl>
##  1 2022-01           Pre       2022-06-27 12:02:00 Northern Weir  8.78  1.28
##  2 2022-01           Pre       2022-06-27 13:02:00 Northern Weir  6.93  1.14
##  3 2022-01           Pre       2022-06-27 14:02:00 Northern Weir  5.64  0.993
##  4 2022-01           Pre       2022-06-27 15:02:00 Northern Weir  4.56  0.896
##  5 2022-01           Pre       2022-06-27 16:02:00 Northern Weir  3.73  0.833
##  6 2022-01           Pre       2022-06-27 17:02:00 Northern Weir  3.22  0.805
##  7 2022-01           Pre       2022-06-27 18:02:00 Northern Weir  2.82  0.788
##  8 2022-01           Pre       2022-06-27 19:02:00 Northern Weir  2.58  0.792
##  9 2022-01           Pre       2022-06-27 20:02:00 Northern Weir  2.41  0.767
## 10 2022-01           Pre       2022-06-27 21:02:00 Northern Weir  2.35  0.764
## # i 344 more rows
## # i 13 more variables: SO4_ppm <dbl>, Na_ppm <dbl>, Mg_ppm <dbl>, Ca_ppm <dbl>,
## #   Alkalinity_uM <dbl>, DOC <dbl>, As_ugL <dbl>, Ba_ugL <dbl>, Cr_ugL <dbl>,
## #   Li_ugL <dbl>, Mn_ugL <dbl>, Si_ugL <dbl>, Sr_ugL <dbl>
```

Now that we have this data, let's do some additional calculations, including total flux of solutes, along with event average discharge. We will save these in their own dataframes for now; they can easily be combined into a single data frame later on

```r
# First, we'll start by calculating the fluxes of all the solutes for each storm
W2_Stormwater_Flux <- W2_Stormwater_2 |>
                     mutate(t = as.numeric(DateTime)) |>
                     group_by(Rain_Event_Number) |>
```

```r
                    mutate(Q_Sum = trapz(t, Q)) |> #Total Q in L
                         # We need to get total discharge per rain event
                    mutate(across(5:18, ~ . * Q))
                         # Multiply all solutes by their Q to get an instantaneous flux

W2_Stormwater_Flux <- W2_Stormwater_Flux |>
                    group_by(Rain_Event_Number) |>
                        # We need to get total solute mass being exported per rain event
                     mutate(Cl_Sum = trapz(t, Cl_ppm),
                            SO4_Sum = trapz(t, SO4_ppm),
                            Na_Sum = trapz(t, Na_ppm),
                            Mg_Sum = trapz(t, Mg_ppm),
                            Ca_Sum = trapz(t, Ca_ppm),
                            Alkalinity_Sum = trapz(t, Alkalinity_uM),
                            DOC_Sum = trapz(t, DOC),
                            As_Sum = trapz(t, As_ugL),
                            Ba_Sum = trapz(t, Ba_ugL),
                            Cr_Sum = trapz(t, Cr_ugL),
                            Li_Sum = trapz(t, Li_ugL),
                            Mn_Sum = trapz(t, Mn_ugL),
                            Si_Sum = trapz(t, Si_ugL),
                            Sr_Sum = trapz(t, Sr_ugL)) |>
                               distinct(Rain_Event_Number, .keep_all = TRUE)
                                   # Only keep storm specific data



# Now, let's calculate Flux Weighted Averages

W2_Stormwater_AvgConc <- W2_Stormwater_Flux |>
                            mutate(across(22:34, ~ . / Q_Sum)) |>
                                 # Divide flux by total Q to back to a concentration, now an average of
                            distinct(Rain_Event_Number, .keep_all = TRUE)
                                 # Only keep storm specific data

# And finally, let's clean up these two dataframes
columns_to_exclude <- c(5:21)
W2_Stormwater_AvgConc <- W2_Stormwater_AvgConc[, -columns_to_exclude] # Remove concentration informatio

# While we're here, change column names from "_Sum" to "_AvgConc"
colnames(W2_Stormwater_AvgConc) <- gsub("_Sum$", "_AvgConc", colnames(W2_Stormwater_AvgConc))
W2_Stormwater_Flux <- W2_Stormwater_Flux[, -columns_to_exclude] # Remove concentration information


print(W2_Stormwater_Flux)
```

```
## # A tibble: 25 x 18
## # Groups:   Rain_Event_Number [25]
##    Rain_Event_Number Treatment DateTime            ISCO_Location  Cl_Sum SO4_Sum
##    <chr>             <chr>     <dttm>              <chr>           <dbl>   <dbl>
##  1 2022-01           Pre       2022-06-27 12:02:00 Northern Weir  1.37e5  8.78e5
##  2 2022-02           Pre       2022-06-29 21:19:00 Northern Weir  1.61e5  1.38e6
##  3 2022-03           Pre       2022-07-05 19:51:00 Northern Weir  3.15e5  2.13e6
```

```
##  4 2022-04            Pre       2022-07-12 03:32:00 Northern Weir  2.94e5  1.29e6
##  5 2022-05            Pre       2022-07-18 14:17:00 Northern Weir  1.03e6  5.06e6
##  6 2022-07            Pre       2022-09-18 22:29:00 Northern Weir  2.43e6  1.88e7
##  7 2022-08            Pre       2022-10-13 15:52:00 Northern Weir  4.31e6  2.12e7
##  8 2022-09            Pre       2022-11-11 19:22:00 Northern Weir  1.65e6  1.11e7
##  9 2023-01            Pre       2023-06-13 10:33:00 Northern Weir  2.73e5  2.03e6
## 10 2023-02            Pre       2023-06-17 06:25:00 Northern Weir  2.45e5  1.42e6
## # i 15 more rows
## # i 12 more variables: Na_Sum <dbl>, Mg_Sum <dbl>, Ca_Sum <dbl>,
## #   Alkalinity_Sum <dbl>, DOC_Sum <dbl>, As_Sum <dbl>, Ba_Sum <dbl>,
## #   Cr_Sum <dbl>, Li_Sum <dbl>, Mn_Sum <dbl>, Si_Sum <dbl>, Sr_Sum <dbl>
```

```
print(W2_Stormwater_AvgConc)
```

```
## # A tibble: 25 x 18
## # Groups:   Rain_Event_Number [25]
##    Rain_Event_Number Treatment DateTime            ISCO_Location Cl_AvgConc
##    <chr>             <chr>     <dttm>              <chr>              <dbl>
##  1 2022-01           Pre       2022-06-27 12:02:00 Northern Weir    136546.
##  2 2022-02           Pre       2022-06-29 21:19:00 Northern Weir    160997.
##  3 2022-03           Pre       2022-07-05 19:51:00 Northern Weir    315341.
##  4 2022-04           Pre       2022-07-12 03:32:00 Northern Weir    293898.
##  5 2022-05           Pre       2022-07-18 14:17:00 Northern Weir   1033847.
##  6 2022-07           Pre       2022-09-18 22:29:00 Northern Weir   2430060.
##  7 2022-08           Pre       2022-10-13 15:52:00 Northern Weir   4309821.
##  8 2022-09           Pre       2022-11-11 19:22:00 Northern Weir   1651210.
##  9 2023-01           Pre       2023-06-13 10:33:00 Northern Weir    273448.
## 10 2023-02           Pre       2023-06-17 06:25:00 Northern Weir    244899.
## # i 15 more rows
## # i 13 more variables: SO4_AvgConc <dbl>, Na_AvgConc <dbl>, Mg_AvgConc <dbl>,
## #   Ca_AvgConc <dbl>, Alkalinity_AvgConc <dbl>, DOC_AvgConc <dbl>,
## #   As_AvgConc <dbl>, Ba_AvgConc <dbl>, Cr_AvgConc <dbl>, Li_AvgConc <dbl>,
## #   Mn_AvgConc <dbl>, Si_AvgConc <dbl>, Sr_AvgConc <dbl>
```

Part II: Environmental Controls for each storm

For this part, we want to be able to consider how changes in the hydrology between storms impacts our C-Q data **expand a bit here**

We will save everything in a new data frame

```
# Step 1: Precompute Storm Time/Duration Metrics
storm_times <- read_excel("/Users/robertrioux/Desktop/CQ_Data/StormTimes.xlsx") %>%
    mutate(DateTime = force_tz(as.POSIXct(DateTime, format = "%m/%d/%Y %H:%M"),
        "EST") %>%
        with_tz("GMT"), doy = yday(DateTime))

# Create start and end times
time_lookup <- storm_times %>%
    filter(Range == "Start") %>%
    rename(start_time = DateTime) %>%
    inner_join(storm_times %>%
```

```r
        filter(Range == "End") %>%
        rename(end_time = DateTime), by = "Rain_Event_Number")

# Step 2: Add Time and Q Metrics
W2_Stormwater_EnvCont <- W2_Stormwater_2 %>%
    select(1:5) %>%
    mutate(t = as.numeric(DateTime)) %>%
    group_by(Rain_Event_Number) %>%
    summarise(Q_Tot = trapz(t, Q), Range_Q = max(Q) - min(Q),
        Max_Q = max(Q), Mean_Q = mean(Q)) %>%
    inner_join(time_lookup, by = "Rain_Event_Number") %>%
    mutate(Range_t = as.numeric(difftime(end_time, start_time,
        units = "secs"))/60, doy = yday(start_time), SW = ifelse(doy <
        183, doy/182.5, (365 - doy)/182.5))

# Step 3: Add Temperature and Precipitation Stats
temp <- read_excel("/Users/robertrioux/Documents/Grad School/YSE/Research/Saiers Lab/Basalt Project/Wya
    mutate(DateTime = as.POSIXct(DateTime, format = "%m/%d/%Y %H:%M",
        tz = "GMT"))

rain <- read_excel("/Users/robertrioux/Documents/Grad School/YSE/Research/Saiers Lab/Basalt Project/Wya
    mutate(DateTime = as.POSIXct(DateTime, format = "%m/%d/%Y %H:%M",
        tz = "GMT"))

W2_Stormwater_EnvCont <- W2_Stormwater_EnvCont %>%
    rowwise() %>%
    mutate(Avg_Temp_7day = mean(temp$temp[temp$DateTime >= (start_time -
        days(7)) & temp$DateTime < start_time], na.rm = TRUE),
        Tot_Precip_7day = sum(rain$Rain_in[rain$DateTime >= (start_time -
            days(7)) & rain$DateTime < start_time], na.rm = TRUE),
        Avg_Temp_24hr = mean(temp$temp[temp$DateTime >= (start_time -
            days(1)) & temp$DateTime < start_time], na.rm = TRUE),
        Tot_Precip_24hr = sum(rain$Rain_in[rain$DateTime >= (start_time -
            days(1)) & rain$DateTime < start_time], na.rm = TRUE),
        Avg_Temp_Storm = mean(temp$temp[temp$DateTime >= start_time &
            temp$DateTime < end_time], na.rm = TRUE), Tot_Precip_Storm = sum(rain$Rain_in[rain$DateTime
            start_time & rain$DateTime < end_time], na.rm = TRUE),
        Max_Precip_Rate = max(slide_dbl(rain$Rain_in[rain$DateTime >=
            start_time & rain$DateTime <= end_time], sum, .before = 3,
            .complete = TRUE), na.rm = TRUE)/4, QP_ratio = Q_Tot/Tot_Precip_Storm)

# Final Output
W2_Stormwater_EnvCont <- ungroup(W2_Stormwater_EnvCont)
# Assuming df is your data frame and you want to keep
# columns 'column3' and 'column5'
W2_Stormwater_EnvCont <- subset(W2_Stormwater_EnvCont, select = -c(start_time,
    Range.x, doy.x, end_time, Range.y, doy.y, doy))

print(W2_Stormwater_EnvCont)
```

```
## # A tibble: 25 x 15
##    Rain_Event_Number    Q_Tot Range_Q Max_Q Mean_Q Range_t    SW Avg_Temp_7day
##    <chr>                <dbl>   <dbl> <dbl>  <dbl>   <dbl> <dbl>         <dbl>
```

```
##  1 2022-01            144463.    6.44  8.78   3.74     608 0.975         19.8
##  2 2022-02            223861.    3.33  4.79   3.09    1560 0.992         20.3
##  3 2022-03            372766.    7.24  8.61   4.10    1903 0.975         18.9
##  4 2022-04            252858.    6.86  7.36   2.64    1718 0.942         17.2
##  5 2022-05            972445.   25.4  26.3   11.5     1327 0.910        NaN
##  6 2022-07           2541009.   47.8  50.6   17.4     2766 0.564         14.6
##  7 2022-08           2255787.   52.3  53.3   12.8     2676 0.433          7.53
##  8 2022-09           1191444.   29.7  32.2   10.6     2139 0.268         10.2
##  9 2023-01            300724.   10.6  17.2   11.7      450 0.899         14.5
## 10 2023-02            224258.    5.76 12.5    9.91     391 0.921         17.7
## # i 15 more rows
## # i 7 more variables: Tot_Precip_7day <dbl>, Avg_Temp_24hr <dbl>,
## #   Tot_Precip_24hr <dbl>, Avg_Temp_Storm <dbl>, Tot_Precip_Storm <dbl>,
## #   Max_Precip_Rate <dbl>, QP_ratio <dbl>
```

Part III: C-Q Analysis

For this part, we need to log (base 10) transform our data to be able to estimate the linear form of the C-Q relationship: log(C) = log(a) + b * log(Q)

```r
#### Extracting lm model coefficients per solute and event
#### #### Replace negative values with 0.01
W2_Stormwater_2[W2_Stormwater_2 < 0] <- 0.01

# Log10 transform the variables of interest and Q
w2sw_log <- W2_Stormwater_2 |>
    mutate_at(vars(Cl_ppm:Sr_ugL, Q), ~log10(.)) |>
    group_by(Rain_Event_Number)  # Group by Rain_Event_Number


# These next two outputs are a simple example of what we
# will be doing next for all solutes; here they are useful
# to make sure functions are running properly and we have
# the outputs we desire Performing linear regressions using
# lmList
lm_results <- lmList(Cl_ppm ~ Q | Rain_Event_Number, w2sw_log)
print(lm_results)
```

```
## Call: lmList(formula = Cl_ppm ~ Q | Rain_Event_Number, data = w2sw_log)
## Coefficients:
##         (Intercept)            Q
## 2022-01 -0.26504083  0.368308694
## 2022-02 -0.17542438  0.071330994
## 2022-03 -0.18962301  0.218025181
## 2022-04 -0.09763309  0.134099709
## 2022-05  0.14971608 -0.128357652
## 2022-07 -1.66428855  0.972130855
## 2022-08  0.16912534  0.039915573
## 2022-09 -0.27551468  0.320415539
## 2023-01 -0.22719199  0.189949906
## 2023-02  0.48156841 -0.512868373
## 2023-03  0.28307173 -0.240493970
```

```
## 2023-04 -0.10364230 -0.027881996
## 2023-05 -1.18738791  0.782992003
## 2023-06 -0.27787438  0.139766756
## 2023-07 -0.21992203  0.073895989
## 2023-08 -0.19500445 -0.002317873
## 2023-09  0.05791015 -0.126372932
## 2023-12  0.05110964 -0.074378534
## 2023-13 -0.56738097  0.404569274
## 2023-14  0.11149393 -0.069304795
## 2023-15 -0.41425044  0.369958576
## 2023-16 -0.42067055  0.405878656
## 2023-17 -0.38716981  0.414908993
## 2023-18  0.04776427  0.224599138
## 2023-19 -0.08146087  0.146661825
##
## Degrees of freedom: 354 total; 304 residual
## Residual standard error: 0.3500121
```

```
# Extracting coefficients
output <- tibble(Rain_Event_Number = names(lm_results), intercept = summary(lm_results)$coef[,
    1, 1], slope = summary(lm_results)$coef[, 1, 2], r_sq = summary(lm_results)$r.squared,
    std_err_slope = summary(lm_results)$coef[, 2, 2], std_err_int = summary(lm_results)$coef[,
        2, 1], p_value_intercept = summary(lm_results)$coef[,
        4, 1], p_value_slope = summary(lm_results)$coef[, 4,
        2], )
print(output)
```

```
## # A tibble: 25 x 8
##    Rain_Event_Number intercept   slope   r_sq std_err_slope std_err_int
##    <chr>                 <dbl>   <dbl>  <dbl>         <dbl>       <dbl>
##  1 2022-01             -0.265   0.368  0.958          0.493      0.276
##  2 2022-02             -0.175   0.0713 0.174          0.800      0.392
##  3 2022-03             -0.190   0.218  0.303          0.262      0.156
##  4 2022-04             -0.0976  0.134  0.0160         0.170      0.0802
##  5 2022-05              0.150  -0.128  0.0992         0.188      0.192
##  6 2022-07             -1.66    0.972  0.120          0.216      0.251
##  7 2022-08              0.169   0.0399 0.0468         0.156      0.156
##  8 2022-09             -0.276   0.320  0.0267         0.229      0.221
##  9 2023-01             -0.227   0.190  0.0564         0.738      0.778
## 10 2023-02              0.482  -0.513  0.0820         1.17       1.16
## # i 15 more rows
## # i 2 more variables: p_value_intercept <dbl>, p_value_slope <dbl>
```

```
# List of variable names
variables <- c("Cl_ppm", "SO4_ppm", "Na_ppm", "Mg_ppm", "Ca_ppm",
    "Alkalinity_uM", "DOC", "As_ugL", "Ba_ugL", "Cr_ugL", "Li_ugL",
    "Mn_ugL", "Si_ugL", "Sr_ugL")

# Perform linear regression for each variable
for (var in variables) {
    lm_results <- lmList(as.formula(paste(var, "~ Q | Rain_Event_Number")),
        w2sw_log)
```

```r
    # Extract coefficients and create tibble
    output <- tibble(Rain_Event_Number = names(lm_results), intercept = summary(lm_results)$coef[,
        1, 1], slope = summary(lm_results)$coef[, 1, 2], r_sq = summary(lm_results)$r.squared,
        std_err_slope = summary(lm_results)$coef[, 2, 2], std_err_int = summary(lm_results)$coef[,
            2, 1], p_value_intercept = summary(lm_results)$coef[,
            4, 1], p_value_slope = summary(lm_results)$coef[,
            4, 2], )

    # Dynamically name the objects
    lm_name <- paste("lm_results_", var, sep = "")
    output_name <- paste("output_", var, sep = "")

    # Assign results to dynamically named objects
    assign(lm_name, lm_results)
    assign(output_name, output)
}

# Viewing the output for one of the variables (e.g.,
# Cl_ppm)
print(output_Cl_ppm)
```

```
## # A tibble: 25 x 8
##    Rain_Event_Number intercept   slope   r_sq std_err_slope std_err_int
##    <chr>                 <dbl>   <dbl>  <dbl>         <dbl>       <dbl>
##  1 2022-01              -0.265   0.368  0.958         0.493      0.276
##  2 2022-02              -0.175   0.0713 0.174         0.800      0.392
##  3 2022-03              -0.190   0.218  0.303         0.262      0.156
##  4 2022-04              -0.0976  0.134  0.0160        0.170      0.0802
##  5 2022-05               0.150  -0.128  0.0992        0.188      0.192
##  6 2022-07              -1.66    0.972  0.120         0.216      0.251
##  7 2022-08               0.169   0.0399 0.0468        0.156      0.156
##  8 2022-09              -0.276   0.320  0.0267        0.229      0.221
##  9 2023-01              -0.227   0.190  0.0564        0.738      0.778
## 10 2023-02               0.482  -0.513  0.0820        1.17       1.16
## # i 15 more rows
## # i 2 more variables: p_value_intercept <dbl>, p_value_slope <dbl>
```

```r
# Initialize an empty data frame to store combined results
W2_Stormwater_lmcoef <- data.frame(Rain_Event_Number = unique(w2sw_log$Rain_Event_Number))

# Iterate through each variable
for (var in variables) {
    # Define column names for the current variable
    col_names <- paste(c("intercept", "slope", "r_sq", "std_err_slope",
        "std_err_int", "p_value_intercept", "p_value_slope"),
        "_", var, sep = "")

    # Extract output for the current variable
    output_name <- paste("output_", var, sep = "")
    output <- get(output_name)

    # Assign output columns to the combined results
    W2_Stormwater_lmcoef[col_names] <- output[, c("intercept",
```

```r
        "slope", "r_sq", "std_err_slope", "std_err_int", "p_value_intercept",
        "p_value_slope")]
}

# Look at the first couple rows of 'W2_Stormwater_lmcoef'
# head(W2_Stormwater_lmcoef) - this is really long - no
# need unless you really want to. Its fine, I promise.


# Now you can write out and save these results if you so
# desrire Define the file path and filename file_path <-
# '/Users/robertrioux/Desktop/CQ_Data' file_name <-
# 'W2_Storm_lm_results.xlsx' file_full_path <-
# file.path(file_path, file_name)

# Write the combined results to an Excel file
# write.xlsx(W2_Stormwater_lmcoef, file_full_path, rowNames
# = FALSE)


# Print confirmation message cat('Data exported to:',
# file_full_path, '\n')
```

Now we can visualize these regressions with plots!

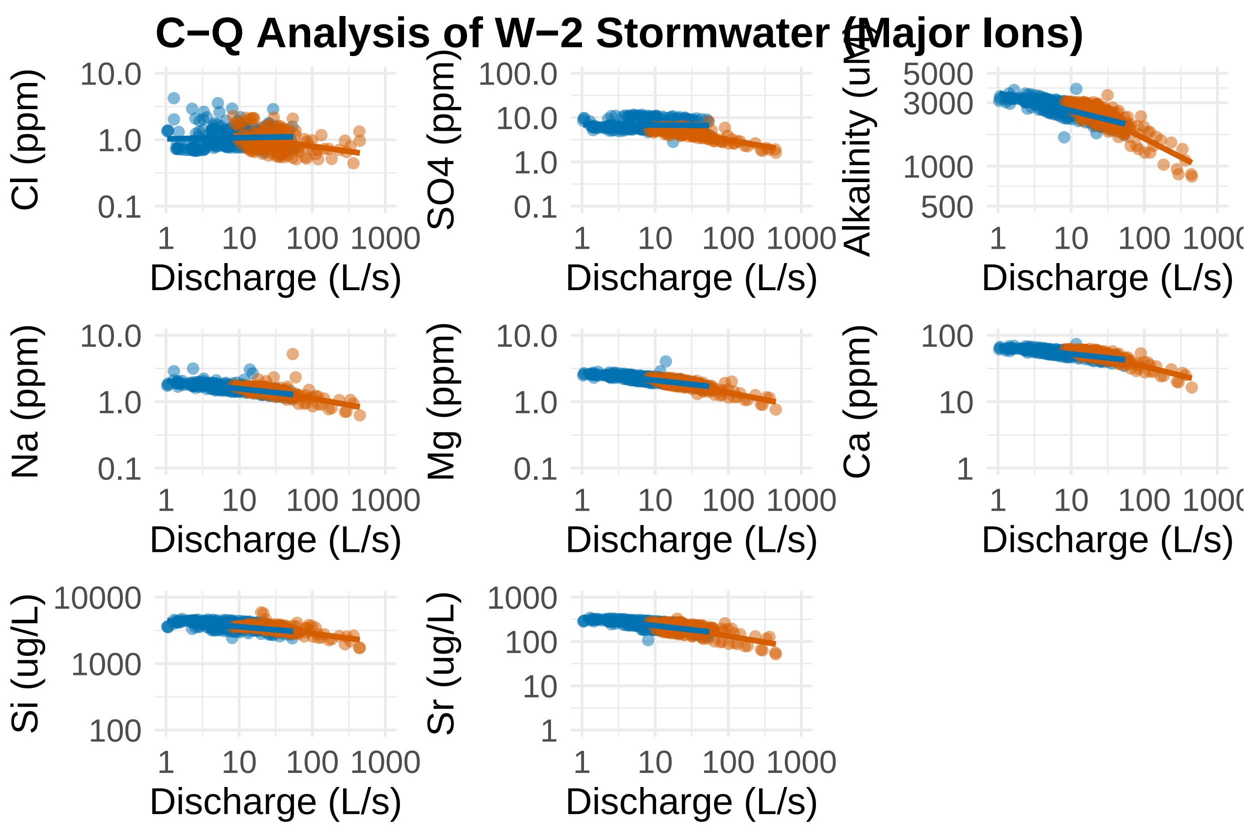
```r
p1 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Cl_ppm, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.1, 10)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Cl (ppm)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

p2 <- ggplot(W2_Stormwater_2, aes(x = Q, y = SO4_ppm, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.1, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("SO4 (ppm)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

p3 <- ggplot(W2_Stormwater_2, aes(x = Q, y = DOC, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(1, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("DOC") +
  xlab("Discharge (L/s)") +
```

```
  theme_minimal()

p4 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Na_ppm, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.1, 10)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Na (ppm)") +
  xlab("Discharge (L/s)") +
  theme_minimal()


p5 <- ggplot(W2_Stormwater_2, aes(x = Q, y = As_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.01, 10)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("As (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()


p6 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Ca_ppm, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(1, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Ca (ppm)") +
  xlab("Discharge (L/s)") +
  theme_minimal()


p7 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Mg_ppm, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.1, 10)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Mg (ppm)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

p8 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Alkalinity_uM, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(500, 5000)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Alkalinity (uM)") +
  xlab("Discharge (L/s)") +
```

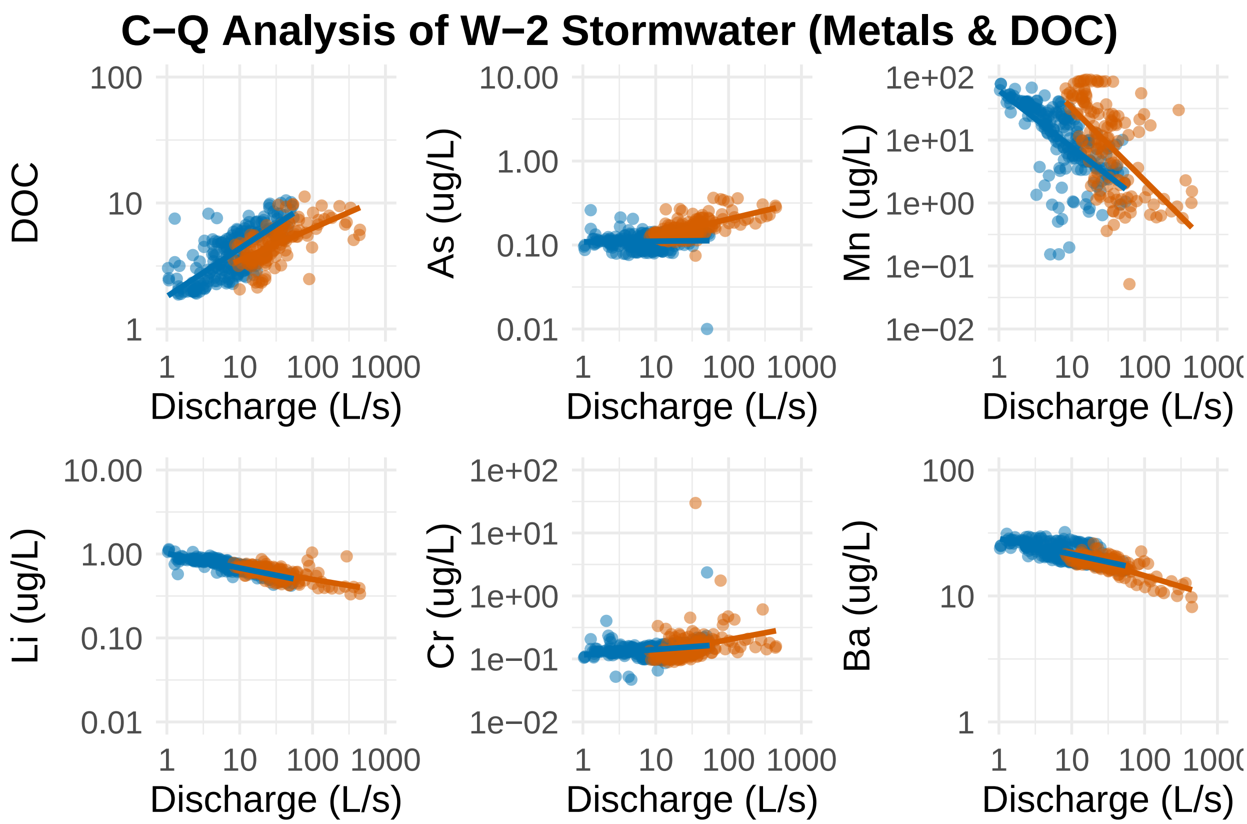
```
  theme_minimal()


p9 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Si_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(100, 10000)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Si (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

p10 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Sr_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(1, 1000)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Sr (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()


p11 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Li_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.01, 10)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Li (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

p12 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Mn_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.01, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Mn (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

p13 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Cr_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(0.01, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Cr (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()
```

```r
p14 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Ba_ugL, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(1, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Ba (ug/L)") +
  xlab("Discharge (L/s)") +
  theme_minimal()



# Define a common theme for all plots with increased text sizes
common_theme <- theme_minimal() +
  theme(
    text = element_text(size = 14),           # General text size
    axis.title = element_text(size = 14),   # Axis titles size
    axis.text = element_text(size = 12),    # Axis text size
    legend.title = element_text(size = 12), # Legend title size
    legend.text = element_text(size = 10),  # Legend text size
    plot.margin = unit(c(0.2, 0.2, 0.2, 0.2), "cm") # Small plot margins
  )

# Update each plot with the common theme and suppress legends
p1 <- p1 + common_theme + guides(color = "none")
p2 <- p2 + common_theme + guides(color = "none")
p3 <- p3 + common_theme + guides(color = "none")
p4 <- p4 + common_theme + guides(color = "none")
p5 <- p5 + common_theme + guides(color = "none")
p6 <- p6 + common_theme + guides(color = "none")
p7 <- p7 + common_theme + guides(color = "none")
p8 <- p8 + common_theme + guides(color = "none")
p9 <- p9 + common_theme + guides(color = "none")
p10 <- p10 + common_theme + guides(color = "none")
p11 <- p11 + common_theme + guides(color = "none")
p12 <- p12 + common_theme + guides(color = "none")
p13 <- p13 + common_theme + guides(color = "none")
p14 <- p14 + common_theme + guides(color = "none")

# Create the legend from one of the plots
legend <- get_legend(p1 + theme(legend.position = "bottom"))

# Create plot grids without legends
CQ_grid1 <- plot_grid(p1, p2, p8, p4, p7, p6, p9, p10,
                      nrow = 3, align = 'hv', vjust = 0.1, hjust = 0.1)
CQ_grid2 <- plot_grid(p3, p5, p12, p11, p13, p14,
                      nrow = 2, align = 'hv', vjust = 0.1, hjust = 0.1)

# Add titles and combine grids with the common legend
title_plot1 <- ggdraw() +
  draw_label("C-Q Analysis of W-2 Stormwater (Major Ions)", fontface = 'bold', size = 16, hjust = 0.5)
  theme(plot.margin = margin(20, 10, 20, 10))  # Add padding: top, right, bottom, left
```

```
final_plot1 <- plot_grid(title_plot1, CQ_grid1, legend, ncol = 1, rel_heights = c(0.1, 2, 0.1))

title_plot2 <- ggdraw() +
  draw_label("C-Q Analysis of W-2 Stormwater (Metals & DOC)", fontface = 'bold', size = 16, hjust = 0.5)
  theme(plot.margin = margin(20, 10, 20, 10))  # Add padding: top, right, bottom, left

final_plot2 <- plot_grid(title_plot2, CQ_grid2, legend, ncol = 1, rel_heights = c(0.1, 2, 0.1))

# Display the final combined plots
print(final_plot1)
```




```
print(final_plot2)
```




```r
# Save the final plots
ggsave("final_plot1.png", plot = final_plot1, width = 8, height = 6, dpi = 600, units = "in")
ggsave("final_plot2.png", plot = final_plot2, width = 8, height = 6, dpi = 600, units = "in")




# Ensure p1 contains the legend
p6 <- ggplot(W2_Stormwater_2, aes(x = Q, y = Ca_ppm, color = Treatment)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10(limits = c(1, 1000)) +
  scale_y_log10(limits = c(1, 100)) +
  scale_color_manual(values = c("Pre" = "#0072B2", "Post" = "#D55E00")) +
  ylab("Ca (ppm)") +
  xlab("Discharge (L/s)") +
  theme_minimal()

# Extract the legend
legend <- get_legend(p6)

# Save the legend as a standalone plot
legend_plot <- ggdraw(legend)

# Save the legend to a PNG file
ggsave("legend_only.png", plot = legend_plot, width = 4, height = 1, dpi = 600, units = "in")
```

Another important part of C-Q Analysis is determining if there the relationship between C and Q explain the variation of those those two variables, to do this, we will calculate the covariance ratio for each of the solutes.

```
# Function to calculate the ratio of CVs
calculate_ratio_CV <- function(df, solute_col, discharge_col) {
    mean_solute <- mean(df[[solute_col]], na.rm = TRUE)
    mean_discharge <- mean(df[[discharge_col]], na.rm = TRUE)
    sd_solute <- sd(df[[solute_col]], na.rm = TRUE)
    sd_discharge <- sd(df[[discharge_col]], na.rm = TRUE)

    ratio_CV <- (mean_discharge/mean_solute) * (sd_solute/sd_discharge)
    return(ratio_CV)
}


# List of variable names - same as above, no need to run
# again variables <- c('Cl_ppm', 'SO4_ppm', 'Na_ppm',
# 'Mg_ppm', 'Ca_ppm',
#'Alkalinity_uM', 'DOC', 'As_ugL', 'Ba_ugL', 'Cr_ugL', 'Li_ugL',
#'Mn_ugL', 'Si_ugL', 'Sr_ugL')

# Initialize an empty data frame to store results
W2_Stormwater_ratioCV <- data.frame()

## Perform function for each variable
for (var in variables) {
    results <- W2_Stormwater_2 %>%
        group_by(Rain_Event_Number) %>%
        summarise(ratio_CV = calculate_ratio_CV(pick(everything()),
            var, "Q")) %>%
        mutate(variable = var)  # Add a column for the variable name

    # Bind the results to the W2_Stormwater_ratioCV data
    # frame
    W2_Stormwater_ratioCV <- bind_rows(W2_Stormwater_ratioCV,
        results)
}

# Pivot wider for viewing ease
W2_Stormwater_ratioCV <- W2_Stormwater_ratioCV %>%
    pivot_wider(names_from = variable, values_from = ratio_CV)

# Rename columns to add 'CV_' prefix
names(W2_Stormwater_ratioCV)[-1] <- paste0("CV_", gsub("_ppm|_uM|_ugL",
    "", names(W2_Stormwater_ratioCV)[-1]))

# Print the combined results
print(W2_Stormwater_ratioCV)
```

```
## # A tibble: 25 x 15
##    Rain_Event_Number CV_Cl CV_SO4  CV_Na  CV_Mg  CV_Ca CV_Alkalinity CV_DOC
##    <chr>             <dbl>  <dbl>  <dbl>  <dbl>  <dbl>         <dbl>  <dbl>
##  1 2022-01           0.353 0.0798 0.398  0.0471 0.0436        0.0487  0.364
```

```
##  2 2022-02          0.183 0.0617 0.0908 0.0889 0.0826        0.0840  0.703
##  3 2022-03          0.531 0.0990 0.0938 0.0552 0.0631        0.0625  0.395
##  4 2022-04          0.942 0.294  0.136  0.0726 0.0911        0.0889  0.731
##  5 2022-05          0.679 0.208  0.312  0.269  0.284         0.288   0.354
##  6 2022-07          0.932 0.926  0.126  0.158  0.174         0.193   0.374
##  7 2022-08          0.248 0.0851 0.131  0.150  0.168         0.175   0.383
##  8 2022-09          0.491 0.396  0.112  0.109  0.131         0.138   0.386
##  9 2023-01          1.05  0.125  0.0804 0.0699 0.0866        0.0817  0.345
## 10 2023-02          2.95  0.451  0.0724 0.0719 0.0559        0.626   0.453
## # i 15 more rows
## # i 7 more variables: CV_As <dbl>, CV_Ba <dbl>, CV_Cr <dbl>, CV_Li <dbl>,
## #   CV_Mn <dbl>, CV_Si <dbl>, CV_Sr <dbl>
```

```r
# Before we begin, we need to clean up the
# W2_Stormwater_lmcoef so its easier to use later on Create
# a copy of W2_Stormwater_lmcoef for preprocessing
W2_Stormwater_lmcoef_processed <- W2_Stormwater_lmcoef

# Extract unique solute names by finding all unique names
# after removing intercept_ or slope_
solute_names <- names(W2_Stormwater_lmcoef) %>%
    str_remove("^(intercept_|slope_|r_sq_|std_err_slope_|std_err_int_|p_value_intercept_|p_value_slope_
    unique() %>%
    na.omit()

# Preprocess the dataframe
for (solute in solute_names) {

    # Update intercept values to zero if p-value is > 0.05
    p_value_intercept_col <- paste0("p_value_intercept_", solute)
    intercept_col <- paste0("intercept_", solute)

    if (p_value_intercept_col %in% names(W2_Stormwater_lmcoef_processed)) {
        W2_Stormwater_lmcoef_processed[[intercept_col]][W2_Stormwater_lmcoef_processed[[p_value_interce
            0.05] <- 0
    }

    # Update slope values to zero if p-value is > 0.05
    p_value_slope_col <- paste0("p_value_slope_", solute)
    slope_col <- paste0("slope_", solute)

    if (p_value_slope_col %in% names(W2_Stormwater_lmcoef_processed)) {
        W2_Stormwater_lmcoef_processed[[slope_col]][W2_Stormwater_lmcoef_processed[[p_value_slope_col]]
            0.05] <- 0
    }
}


# Now let's get a dataframe with just slopes and intercepts
# for t-tests

W2_Stormwater_coefs <- W2_Stormwater_lmcoef_processed %>%
    select(Rain_Event_Number, starts_with("intercept_"), starts_with("slope_")) %>%
    mutate(Treatment = case_when(Rain_Event_Number %in% c("2022-01",
```

```r
        "2022-02", "2022-03", "2022-04", "2022-05", "2022-07",
        "2022-08", "2022-09", "2023-01", "2023-02", "2023-03",
        "2023-04") ~ "Pre", TRUE ~ "Post"))

# Make rownames
W2_Stormwater_coefs <- column_to_rownames(W2_Stormwater_coefs,
    var = "Rain_Event_Number")

# View the dataframe head(W2_Stormwater_coefs) we'll skip
# showing this one as well

# Now let's do a t-test to see which variables have changed

# Exclude 'Treatment' from the list of variables to test
variables_to_test <- colnames(W2_Stormwater_coefs)[colnames(W2_Stormwater_coefs) !=
    "Treatment"]

# Ensure the dataset is ordered by Treatment
W2_Stormwater_coefs <- W2_Stormwater_coefs |>
    arrange(Treatment)

## So now, we run into an issue where we have an unequal
## amount of Pre (12) and Post (13) storm events.  We still
## want to be able to conduct a paired t test to understand
## how our treatment affected these variables.  We will
## test the effects of removing a single storm, for each of
## the storms and determine how we can best minimize the
## affect of removing a storm - we'll do this as
## systematically and robustly as possible.

# List of Post storm indices
post_indices <- which(W2_Stormwater_coefs$Treatment == "Post")

# Initialize a list to store results
robust_t_test_results <- list()

# Loop through each Post storm to exclude one at a time
for (excluded_post in post_indices) {
    # Subset data excluding the current Post storm
    subset_data <- W2_Stormwater_coefs[-excluded_post, ]

    # Ensure Pre and Post groups are balanced
    pre_values <- subset_data[subset_data$Treatment == "Pre",
        ]
    post_values <- subset_data[subset_data$Treatment == "Post",
        ]

    # Perform paired t-tests
    t_test_results <- lapply(variables_to_test, function(var) {
        pre <- pre_values[[var]]
        post <- post_values[[var]]

        if (length(pre) != length(post)) {
```

```r
            warning(paste("Length mismatch for variable:", var))
            return(data.frame(Variable = var, Excluded_Post = excluded_post,
                P_Value = NA, Mean_Pre = NA, Mean_Post = NA,
                Mean_Difference = NA))
        }

        # Perform the paired t-test
        t_test <- t.test(pre, post, paired = TRUE)
        data.frame(Variable = var, Excluded_Post = excluded_post,
            P_Value = t_test$p.value, Mean_Pre = mean(pre, na.rm = TRUE),
            Mean_Post = mean(post, na.rm = TRUE), Mean_Difference = mean(pre,
                na.rm = TRUE) - mean(post, na.rm = TRUE))
    })

    # Combine results for this iteration
    robust_t_test_results[[as.character(excluded_post)]] <- do.call(rbind,
        t_test_results)
}

# Combine results across all iterations
robust_t_test_summary <- do.call(rbind, robust_t_test_results)

# Analyze the impact of exclusions for each variable
impact_analysis <- robust_t_test_summary %>%
    group_by(Variable) %>%
    summarise(Max_Difference = max(Mean_Difference, na.rm = TRUE),
        Min_Difference = min(Mean_Difference, na.rm = TRUE),
        Range_Difference = Max_Difference - Min_Difference, Max_P_Value = max(P_Value,
            na.rm = TRUE), Min_P_Value = min(P_Value, na.rm = TRUE),
        Range_P_Value = Max_P_Value - Min_P_Value, Excluded_Post_Max_Difference = Excluded_Post[which.ma
        Excluded_Post_Min_Difference = Excluded_Post[which.min(Mean_Difference)])

# View the impact analysis summary
print(impact_analysis)
```

```
## # A tibble: 28 x 9
##    Variable           Max_Difference Min_Difference Range_Difference Max_P_Value
##    <chr>                       <dbl>          <dbl>            <dbl>       <dbl>
##  1 intercept_Alkalin~       -0.121         -0.179             0.0585      0.0171
##  2 intercept_As_ugL         -0.0464        -0.165             0.119       0.675
##  3 intercept_Ba_ugL         -0.00175       -0.0240            0.0223      0.965
##  4 intercept_Ca_ppm          0.0569        -0.219             0.275       0.838
##  5 intercept_Cl_ppm         -0.139         -0.139             0           0.339
##  6 intercept_Cr_ugL         -0.312         -0.404             0.0922      0.202
##  7 intercept_DOC             0.0527        -0.0746            0.127       0.874
##  8 intercept_Li_ugL          0.00682       -0.00478           0.0116      0.700
##  9 intercept_Mg_ppm          0.0280        -0.0786            0.107       0.762
## 10 intercept_Mn_ugL         -0.605         -1.03              0.422       0.272
## # i 18 more rows
## # i 4 more variables: Min_P_Value <dbl>, Range_P_Value <dbl>,
## #   Excluded_Post_Max_Difference <int>, Excluded_Post_Min_Difference <int>
```

```r
# Identify the Post event that minimizes overall
# variability
best_event_to_exclude <- robust_t_test_summary %>%
    group_by(Excluded_Post) %>%
    summarise(Total_Difference_Variability = sum(abs(Mean_Difference -
        mean(Mean_Difference, na.rm = TRUE)), na.rm = TRUE),
        Total_P_Value_Variability = sum(abs(P_Value - mean(P_Value,
            na.rm = TRUE)), na.rm = TRUE)) %>%
    arrange(Total_Difference_Variability, Total_P_Value_Variability)

# View the best Post event to exclude
print(best_event_to_exclude)
```

```
## # A tibble: 13 x 3
##    Excluded_Post Total_Difference_Variability Total_P_Value_Variability
##            <int>                        <dbl>                     <dbl>
##  1             9                         2.94                      7.27
##  2            11                         2.99                      6.46
##  3             8                         3.04                      7.17
##  4            13                         3.15                      6.77
##  5            10                         3.15                      6.59
##  6             2                         3.23                      7.65
##  7             5                         3.27                      7.93
##  8             4                         3.35                      7.37
##  9             3                         3.41                      6.58
## 10             1                         3.60                      6.59
## 11             6                         3.62                      7.26
## 12            12                         3.71                      6.14
## 13             7                         3.71                      6.55
```

```r
# Based on this, we should exclude Post Event 9 - which is
# 2023-15


# Exclude Event 2023-09
final_data <- W2_Stormwater_coefs[rownames(W2_Stormwater_coefs) !=
    "2023-15", ]

# Perform paired t-tests on the final dataset
final_t_test_results <- lapply(variables_to_test, function(var) {
    pre_values <- final_data[[var]][final_data$Treatment == "Pre"]
    post_values <- final_data[[var]][final_data$Treatment ==
        "Post"]

    t_test <- t.test(pre_values, post_values, paired = TRUE)
    data.frame(Variable = var, P_Value = t_test$p.value, Mean_Pre = mean(pre_values,
        na.rm = TRUE), Mean_Post = mean(post_values, na.rm = TRUE),
        Mean_Difference = mean(pre_values, na.rm = TRUE) - mean(post_values,
            na.rm = TRUE))
})

# Combine results into a single dataframe
final_t_test_summary <- do.call(rbind, final_t_test_results)
```

```r
# View the summary of results
print(final_t_test_summary)
```

```
##                   Variable    P_Value     Mean_Pre   Mean_Post Mean_Difference
## 1          intercept_Cl_ppm 0.33880070 -0.138690713  0.00000000    -0.138690713
## 2         intercept_SO4_ppm 0.94835971  0.197505901  0.18349140     0.014014505
## 3          intercept_Na_ppm 0.65915426  0.195461856  0.23530735    -0.039845496
## 4          intercept_Mg_ppm 0.44217505  0.277220013  0.35583833    -0.078618315
## 5          intercept_Ca_ppm 0.78923093  1.833682395  1.89115008    -0.057467690
## 6  intercept_Alkalinity_uM 0.01314822  3.478572809  3.62875548    -0.150182673
## 7             intercept_DOC 0.38687802  0.197106311  0.14443929     0.052667019
## 8          intercept_As_ugL 0.30242885 -0.998457962 -0.85286078    -0.145597184
## 9          intercept_Ba_ugL 0.76290353  1.422623373  1.43493274    -0.012309365
## 10         intercept_Cr_ugL 0.15556640 -0.870927212 -0.55894228    -0.311984936
## 11         intercept_Li_ugL 0.63766427 -0.004782131 -0.01160710     0.006824973
## 12         intercept_Mn_ugL 0.26307602  1.879667254  2.49553979    -0.615872531
## 13         intercept_Si_ugL 0.03240005  3.629607552  3.73187199    -0.102264438
## 14         intercept_Sr_ugL 0.03779729  2.437296413  2.53300733    -0.095710918
## 15             slope_Cl_ppm 0.33880070  0.081010905  0.00000000     0.081010905
## 16            slope_SO4_ppm 0.33880070  0.135212034  0.00000000     0.135212034
## 17             slope_Na_ppm 0.09787224 -0.009783227 -0.12843967     0.118656440
## 18             slope_Mg_ppm 0.17990008 -0.020954093 -0.11862677     0.097672679
## 19             slope_Ca_ppm 0.29715290 -0.012792248 -0.12732567     0.114533422
## 20      slope_Alkalinity_uM 0.04030749 -0.049192991 -0.13954897     0.090355984
## 21                slope_DOC 0.92397959  0.183343297  0.17621156     0.007131735
## 22             slope_As_ugL 0.08530532  0.003990296  0.08575560    -0.081765299
## 23             slope_Ba_ugL 0.53063019 -0.073982209 -0.10132450     0.027342290
## 24             slope_Cr_ugL 0.23993835  0.066816447  0.00000000     0.066816447
## 25             slope_Li_ugL 0.85686020 -0.081135603 -0.08676751     0.005631911
## 26             slope_Mn_ugL 0.99828834 -1.000895757 -0.99984013    -0.001055624
## 27             slope_Si_ugL 0.09184247 -0.043360116 -0.09784684     0.054486720
## 28             slope_Sr_ugL 0.16398017 -0.075179830 -0.14500517     0.069825344
```

```r
# Filter significant results (p < 0.05)
final_significant_results <- final_t_test_summary %>%
    filter(P_Value < 0.05)
print(final_significant_results)
```

```
##                  Variable    P_Value    Mean_Pre Mean_Post Mean_Difference
## 1 intercept_Alkalinity_uM 0.01314822  3.47857281  3.628755     -0.15018267
## 2        intercept_Si_ugL 0.03240005  3.62960755  3.731872     -0.10226444
## 3        intercept_Sr_ugL 0.03779729  2.43729641  2.533007     -0.09571092
## 4     slope_Alkalinity_uM 0.04030749 -0.04919299 -0.139549      0.09035598
```

Part IV: PCA

In this part, we will consider how the slope and intercept vary from storm to storm and why

```r
# Before moving on, we'll need to separate intercepts from
# solutes so we can perform PCAs on them independently
```

```r
# Create a dataframe for intercepts
W2_Stormwater_intercept <- W2_Stormwater_lmcoef_processed %>%
    select(Rain_Event_Number, starts_with("intercept_")) %>%
    mutate(Treatment = case_when(Rain_Event_Number %in% c("2022-01",
        "2022-02", "2022-03", "2022-04", "2022-05", "2022-07",
        "2022-08", "2022-09", "2023-01", "2023-02", "2023-03",
        "2023-04") ~ "Pre", TRUE ~ "Post"))

W2_Stormwater_intercept <- W2_Stormwater_intercept |>
    select(Rain_Event_Number, Treatment, everything())

# Create a dataframe for slopes
W2_Stormwater_slope <- W2_Stormwater_lmcoef_processed %>%
    select(Rain_Event_Number, starts_with("slope_")) %>%
    mutate(Treatment = case_when(Rain_Event_Number %in% c("2022-01",
        "2022-02", "2022-03", "2022-04", "2022-05", "2022-07",
        "2022-08", "2022-09", "2023-01", "2023-02", "2023-03",
        "2023-04") ~ "Pre", TRUE ~ "Post"))

W2_Stormwater_slope <- W2_Stormwater_slope |>
    select(Rain_Event_Number, Treatment, everything())
```

Okay! Now we can do some fun statistics! We will start by performing separate PCAs for slope and intercept of pre-basalt application storms

```r
# We will first start by thinking just about pre-basalt application storm slopes
# First, standardize our data (excluding the first three columns that aren't numeric)
w2_std_pre_slope <- W2_Stormwater_slope %>%
  filter(Treatment == "Pre")

w2_std_pre_slope <- scale(w2_std_pre_slope[, -c(1:2)])

# Convert it back to a dataframe (since scale returns a matrix)
w2_std_pre_slope <- as.data.frame(w2_std_pre_slope)

# let's also clean up the names while we're at it:
# Rename columns of w2_std_pre_intercept
colnames(w2_std_pre_slope) <- gsub("slope_|_ppm|_uM|_ugL", "", colnames(w2_std_pre_slope))

# Check the new column names
colnames(w2_std_pre_slope)
```

```
##  [1] "Cl"         "SO4"        "Na"         "Mg"         "Ca"
##  [6] "Alkalinity" "DOC"        "As"         "Ba"         "Cr"
## [11] "Li"         "Mn"         "Si"         "Sr"
```

```r
# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_std_pre_slope) <- c("2022-01", "2022-02", "2022-03", "2022-04", "2022-05", "2022-07", "2022-
                            "2022-09", "2023-01", "2023-02", "2023-03", "2023-04")

# Viewing the first few rows of the standardized data
head(w2_std_pre_slope) # looks good to me!
```

```
##                 Cl        SO4        Na         Mg        Ca   Alkalinity
## 2022-01 -0.2886751 -0.2886751 0.2886751  0.4271718 0.2886751  0.693803798
## 2022-02 -0.2886751 -0.2886751 0.2886751  0.4271718 0.2886751  0.693803798
## 2022-03 -0.2886751 -0.2886751 0.2886751  0.4271718 0.2886751  0.693803798
## 2022-04 -0.2886751 -0.2886751 0.2886751  0.4271718 0.2886751 -0.003514415
## 2022-05 -0.2886751 -0.2886751 0.2886751  0.4271718 0.2886751 -0.107347696
## 2022-07  3.1754265  3.1754265 0.2886751 -1.9754599 0.2886751 -1.840461011
##                DOC         As         Ba         Cr          Li          Mn
## 2022-01  1.2198535 -0.0588311 -0.3697237 -0.3586376  0.80837733  0.82783831
## 2022-02 -1.0590024 -0.0588311  0.8405876 -0.3586376  0.80837733  0.82783831
## 2022-03  0.2856525 -0.0588311  0.1534188 -0.3586376  0.08332534  0.19394546
## 2022-04  0.8732216 -0.0588311  0.3304713 -0.3586376  0.08079618 -0.19706253
## 2022-05 -0.1063174 -0.0588311  0.1734018 -0.3586376 -0.53649487  0.36259014
## 2022-07  0.9972309 -2.5070167 -0.2680714  3.0743486 -1.63313396 -0.08994108
##                  Si         Sr
## 2022-01  0.90045216  0.9172717
## 2022-02  0.90045216  0.9172717
## 2022-03 -0.05089628  0.2959040
## 2022-04  0.00189880  0.1748210
## 2022-05  0.08052910 -0.1791263
## 2022-07 -1.11020422 -1.0866443
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
pca_result_pre_slope <- prcomp(w2_std_pre_slope, center = FALSE, scale. = FALSE)
summary(pca_result_pre_slope)  # View variance explained by each component
```

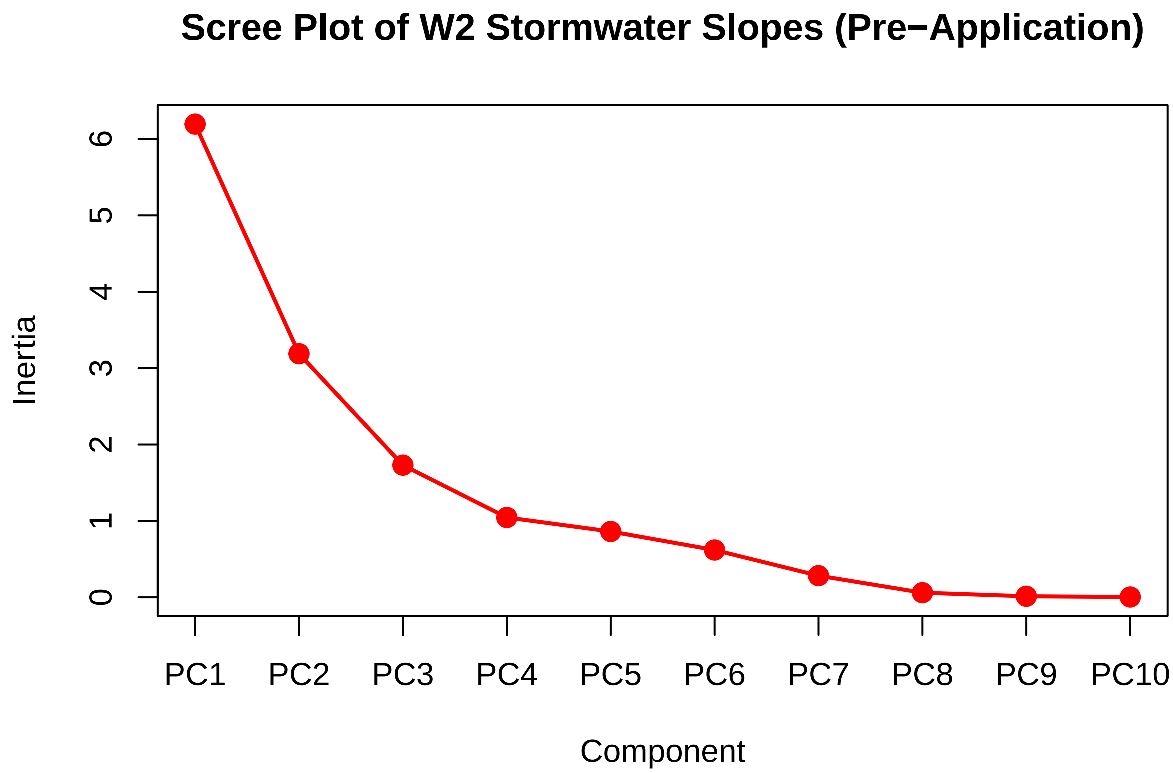
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4889 1.7857 1.3153 1.02222 0.92826 0.78681 0.53211
## Proportion of Variance 0.4425 0.2278 0.1236 0.07464 0.06155 0.04422 0.02022
## Cumulative Proportion  0.4425 0.6703 0.7938 0.86847 0.93002 0.97424 0.99446
##                            PC8     PC9    PC10      PC11      PC12
## Standard deviation     0.24471 0.11720 0.06252 2.488e-16 4.466e-17
## Proportion of Variance 0.00428 0.00098 0.00028 0.000e+00 0.000e+00
## Cumulative Proportion  0.99874 0.99972 1.00000 1.000e+00 1.000e+00
```

```r
# View the loadings
loadings_pre_slope <- pca_result_pre_slope$rotation
print(loadings_pre_slope)
```

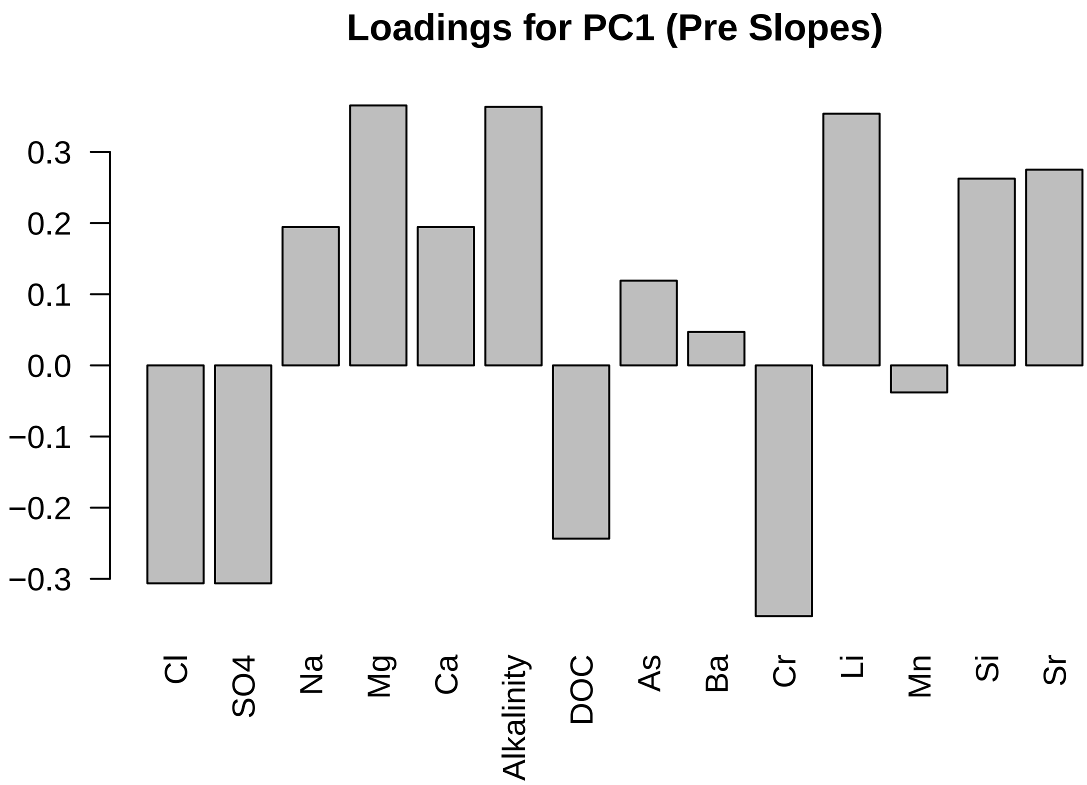
```
##                    PC1         PC2         PC3         PC4         PC5
## Cl         -0.30634477  0.35753522  0.04541880 -0.02182304 -0.05180063
## SO4        -0.30634477  0.35753522  0.04541880 -0.02182304 -0.05180063
## Na          0.19449218  0.42749768 -0.17335558  0.28054119  0.21129621
## Mg          0.36538174  0.08811359 -0.16779139  0.23569371  0.20204610
## Ca          0.19449218  0.42749768 -0.17335558  0.28054119  0.21129621
## Alkalinity  0.36337304  0.09885505 -0.05114997 -0.22597522 -0.07434474
## DOC        -0.24357680 -0.11346781  0.02608948  0.64157435 -0.15412475
## As          0.11907901 -0.49511919 -0.01727330  0.19971128  0.15624310
## Ba          0.04706379  0.01782253 -0.50613665 -0.25515362 -0.48690725
## Cr         -0.35247662  0.24687680  0.08858190 -0.09213804 -0.10444237
## Li          0.35374448  0.13053951 -0.03185549 -0.25247223 -0.06061193
```

```
## Mn         -0.03800985 -0.07084118 -0.52178252  0.29563227 -0.45157223
## Si          0.26247937  0.08396565  0.43616508  0.21911863 -0.39764534
## Sr          0.27507817  0.11357489  0.42010456  0.10380652 -0.44398775
##                    PC6         PC7         PC8          PC9         PC10
## Cl         -0.005717817  0.07205599  0.21249998  0.176841394 -0.035371417
## SO4        -0.005717817  0.07205599  0.21249998  0.176841394 -0.035371417
## Na          0.112643031  0.08798410  0.04174337  0.056123987  0.029093489
## Mg          0.092524189  0.01919532 -0.11456767 -0.078529743  0.047405816
## Ca          0.112643031  0.08798410  0.04174337  0.056123987  0.029093489
## Alkalinity -0.254257291 -0.37735323 -0.15755092  0.652867026 -0.375339281
## DOC         0.103020289 -0.68414582  0.09468984 -0.009582802 -0.063371773
## As          0.179120132  0.29204850  0.58012127  0.471337123 -0.002566408
## Ba          0.658614285 -0.06387078  0.03211182  0.007489754 -0.013095477
## Cr         -0.033978074  0.04929490  0.20009950  0.161146797 -0.042366038
## Li         -0.254165529 -0.31824964  0.68724936 -0.377929362  0.091478115
## Mn         -0.586251234  0.29242093 -0.01008204 -0.005359083  0.025159066
## Si          0.120812284  0.29009251  0.05471096 -0.219497324 -0.614620163
## Sr          0.028973995  0.02378566 -0.07918336  0.243800877  0.678181294
##                     PC11          PC12
## Cl         -4.426833e-01 -3.385739e-01
## SO4         2.017348e-01  3.433351e-01
## Na          6.886876e-01 -2.077692e-01
## Mg         -2.826623e-01  6.784754e-01
## Ca         -4.550751e-01 -2.074999e-01
## Alkalinity -3.399681e-15 -1.023218e-15
## DOC         9.014004e-17  5.673655e-17
## As         -1.884147e-15 -5.390075e-16
## Ba          4.419049e-17  5.896812e-17
## Cr          4.530656e-02  4.700379e-01
## Li          1.699167e-15  4.889600e-16
## Mn         -1.250092e-16  1.612968e-17
## Si         -7.031277e-16 -4.208980e-16
## Sr          6.766896e-16  5.237322e-16
```

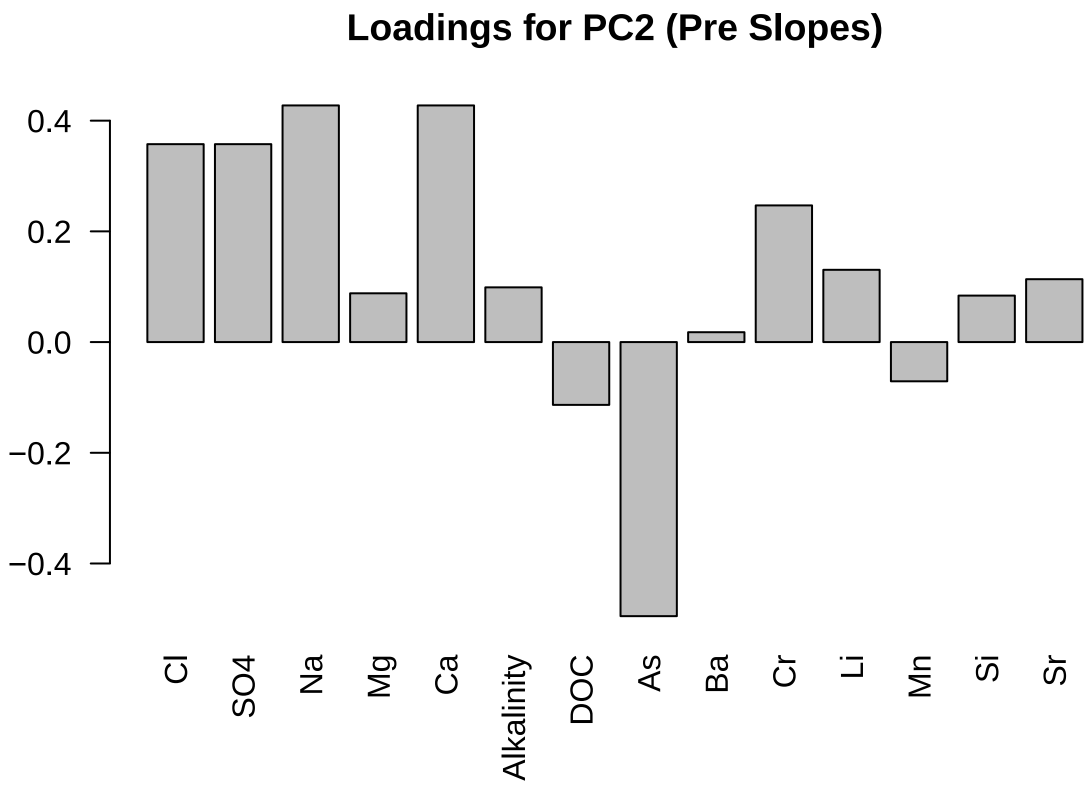
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_pre_slope, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Slopes (Pre-Application)")
```



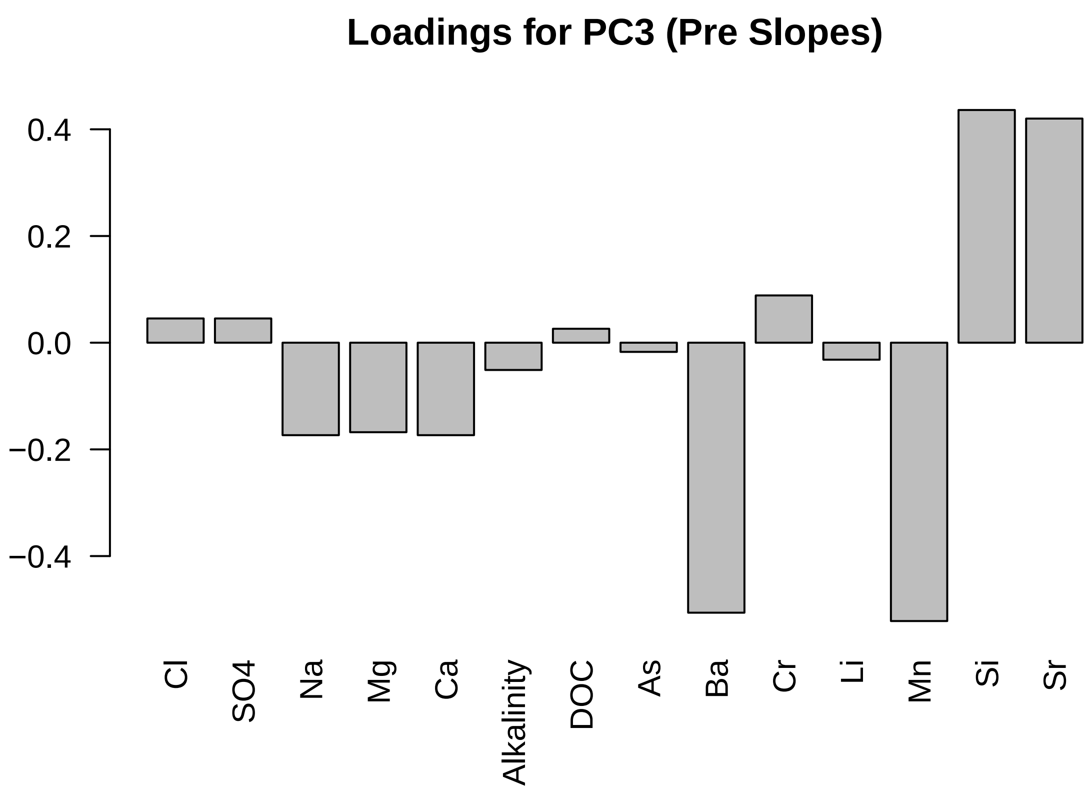

```
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_pre_slope[, 1], main = "Loadings for PC1 (Pre Slopes)", las = 2)
```



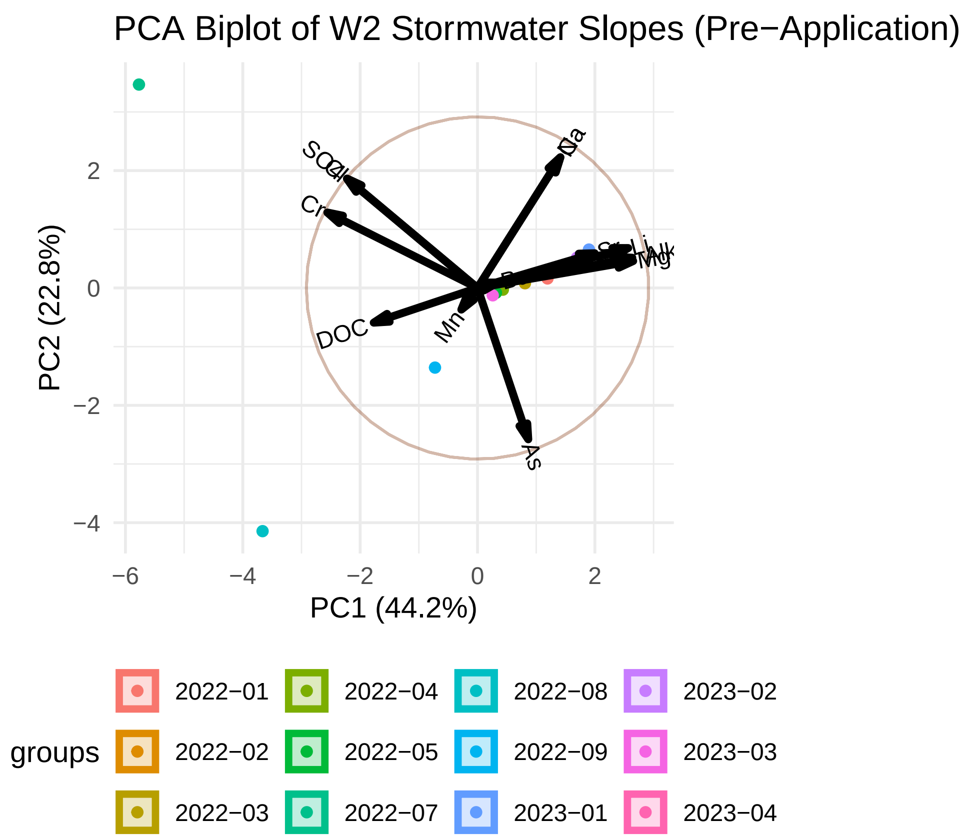

```
barplot(loadings_pre_slope[, 2], main = "Loadings for PC2 (Pre Slopes)", las = 2)
```



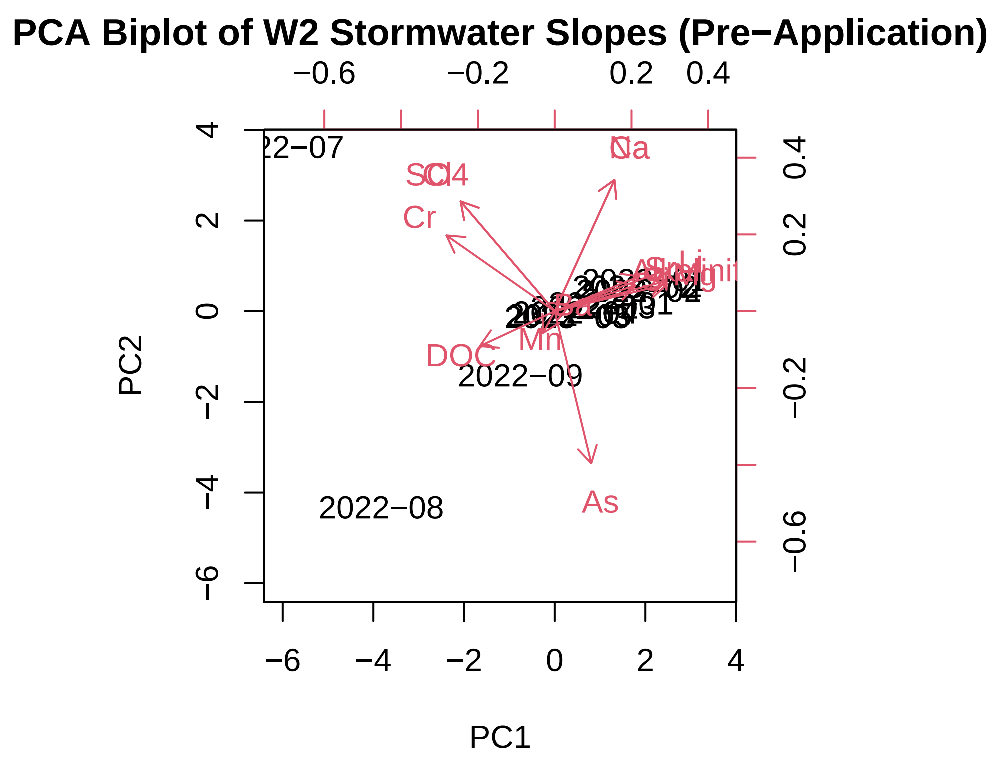

```r
barplot(loadings_pre_slope[, 3], main = "Loadings for PC3 (Pre Slopes)", las = 2)
```




```
# Generate the biplot
biplot_pre_slope <- ggbiplot(pca_result_pre_slope,
                             obs.scale = 1,
                             var.scale = 1,
                             groups = rownames(w2_std_pre_slope),  # Use Rain_Event_Number as group lab
                             ellipse = TRUE,  # Add confidence ellipses
                             circle = TRUE)   # Add a unit circle

# Customize the biplot
biplot_pre_slope <- biplot_pre_slope +
  ggtitle("PCA Biplot of W2 Stormwater Slopes (Pre-Application)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Print the biplot
print(biplot_pre_slope)
```




```r
# Simple base R biplot
biplot(pca_result_pre_slope, scale = 0,
       main = "PCA Biplot of W2 Stormwater Slopes (Pre-Application)")
```




```r
# We will first start by thinking just about pre-basalt application storm intercepts
# First, standardize our data (excluding the first three columns that aren't numeric)
w2_std_pre_intercept <- W2_Stormwater_intercept %>%
  filter(Treatment == "Pre")

w2_std_pre_intercept <- scale(w2_std_pre_intercept[, -c(1:2)])

# Convert it back to a dataframe (since scale returns a matrix)
w2_std_pre_intercept <- as.data.frame(w2_std_pre_intercept)




# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_std_pre_intercept) <- c("2022-01", "2022-02", "2022-03", "2022-04", "2022-05", "2022-07", "2
                          "2022-09", "2023-01", "2023-02", "2023-03", "2023-04")

# let's also clean up the names while we're at it:
# Rename columns of w2_std_pre_intercept
colnames(w2_std_pre_intercept) <- gsub("intercept_|_ppm|_uM|_ugL", "", colnames(w2_std_pre_intercept))

# Check the new column names
colnames(w2_std_pre_intercept)
```

```
##  [1] "Cl"         "SO4"        "Na"         "Mg"         "Ca"
##  [6] "Alkalinity" "DOC"        "As"         "Ba"         "Cr"
```

```
## [11] "Li"         "Mn"         "Si"         "Sr"
```

```r
# Viewing the first few rows of the standardized data
head(w2_std_pre_intercept) # looks good to me!
```

```
##                 Cl        SO4        Na        Mg         Ca Alkalinity
## 2022-01  0.2886751  0.8377987 0.9351360 0.5852743 -0.6169582  0.2803558
## 2022-02  0.2886751 -0.2614734 0.3014065 0.5143555 -0.7784013  0.1540760
## 2022-03  0.2886751  0.8034192 0.6100230 0.6536671 -0.4159970  0.4020312
## 2022-04  0.2886751  0.6412757 0.7313665 0.7173084 -0.2168643  0.5496577
## 2022-05  0.2886751  0.8431930 1.0400895 0.7708044 -0.2349155  0.5296932
## 2022-07 -3.1754265 -2.6369583 0.6859073 0.7942569  0.6214634  1.3448539
##                DOC         As          Ba          Cr         Li          Mn
## 2022-01 -0.1929098 0.12642896  0.08299012 -0.04708965  0.1140722 -0.06089418
## 2022-02 -0.9775343 0.06715948 -0.12274372 -0.13898788 -2.0245711 -0.15577320
## 2022-03  0.2875371 0.37738696  0.35546365  0.21636037 -0.9640131 -0.01532663
## 2022-04  0.5500840 0.61092541  0.41258252  0.01587638 -0.9335329 -0.12750849
## 2022-05  1.8568488 1.34876636  0.64151953  0.06296612  0.1140722 -0.50418433
## 2022-07  1.0985922 1.63509912  0.25818260 -1.53744197  0.1140722  0.20441754
##                 Si        Sr
## 2022-01  0.1529329 0.3182347
## 2022-02  0.2547796 0.2968901
## 2022-03  0.4109237 0.4813263
## 2022-04  0.2815999 0.5265089
## 2022-05  0.6398090 0.6577172
## 2022-07 -0.3768575 0.9696541
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
pca_result_pre_intercept <- prcomp(w2_std_pre_intercept, center = FALSE, scale. = FALSE)
summary(pca_result_pre_intercept)  # View variance explained by each component
```

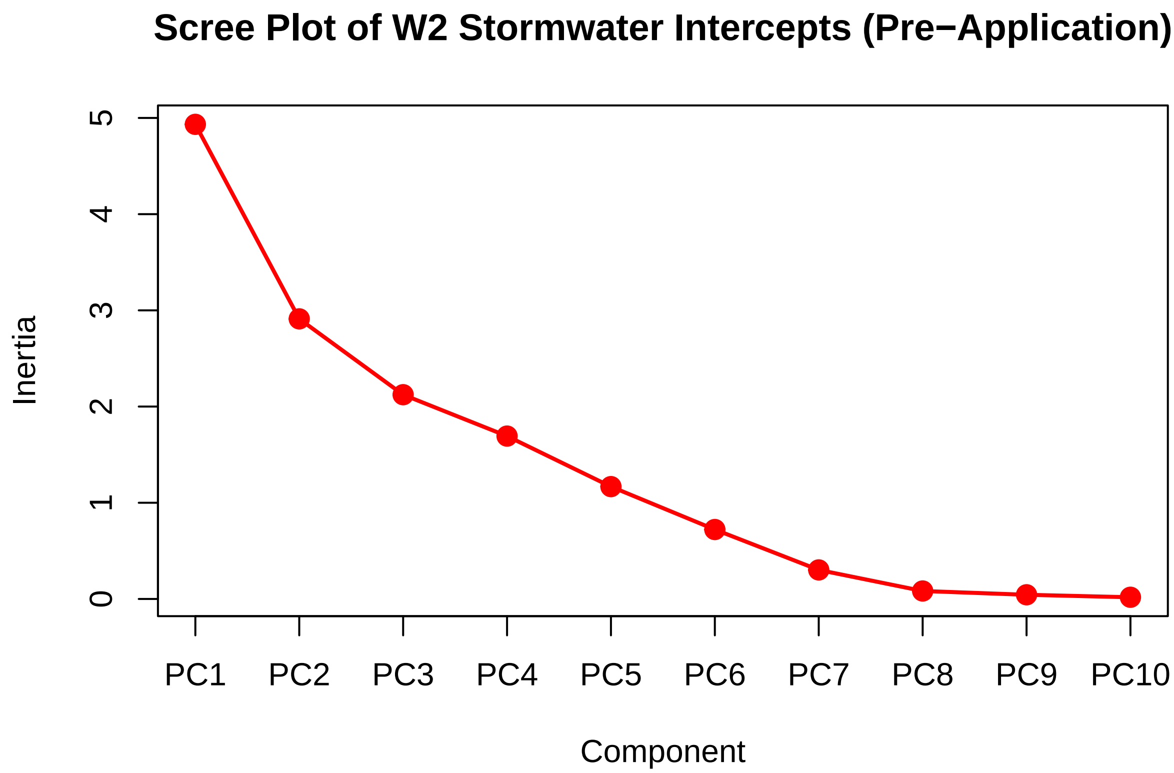
```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.2211 1.7064 1.4572 1.3012 1.08072 0.84963 0.54955
## Proportion of Variance 0.3524 0.2080 0.1517 0.1209 0.08343 0.05156 0.02157
## Cumulative Proportion  0.3524 0.5604 0.7120 0.8330 0.91639 0.96795 0.98952
##                            PC8     PC9    PC10    PC11      PC12
## Standard deviation     0.28781 0.20863 0.13388 0.04928 6.244e-15
## Proportion of Variance 0.00592 0.00311 0.00128 0.00017 0.000e+00
## Cumulative Proportion  0.99544 0.99855 0.99983 1.00000 1.000e+00
```

```r
# View the loadings
loadings_pre_intercept <- pca_result_pre_intercept$rotation
print(loadings_pre_intercept)
```

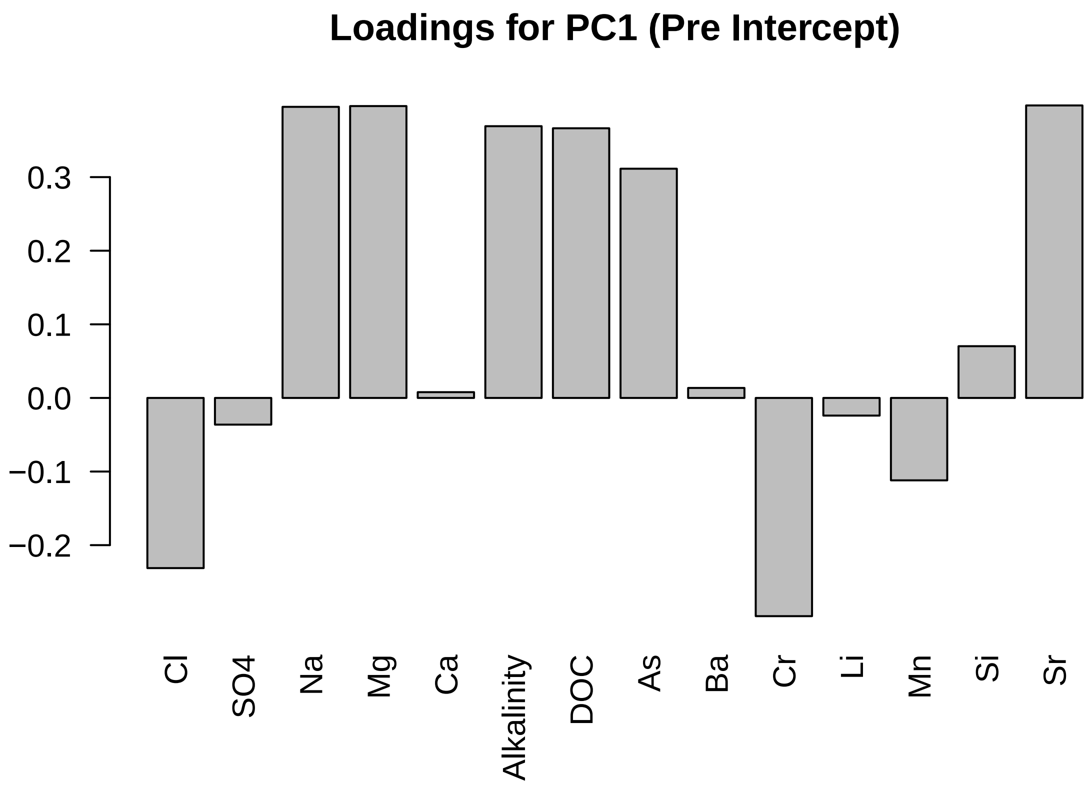
```
##                     PC1         PC2         PC3         PC4          PC5
## Cl         -0.231177222  0.09606190  0.53581854 -0.12190024  0.011983980
## SO4        -0.036257297  0.20348371  0.57999333 -0.15344665  0.110333986
## Na          0.395465835  0.22168397  0.15524168 -0.02271894  0.005368087
## Mg          0.396524964  0.21838318  0.13109653  0.01639066 -0.016628540
## Ca          0.007848634 -0.54161391 -0.04881173 -0.03887592  0.321262974
```

```
## Alkalinity  0.369209622  0.06890347  0.05136930  0.29956949 -0.338382613
## DOC         0.366377867  0.17125526  0.03209044  0.07064187  0.328839310
## As          0.311409178  0.04628683 -0.25350052 -0.43928878  0.086991771
## Ba          0.013593529  0.27410037 -0.23295615 -0.53065228  0.383690498
## Cr         -0.296464853  0.31659786  0.14111914 -0.03954061  0.042991007
## Li         -0.023985716  0.02421185 -0.01796762  0.53955576  0.609379208
## Mn         -0.111952383  0.29172978 -0.33302332  0.06186840 -0.330738323
## Si          0.070308503 -0.46240901  0.21545948 -0.30536146 -0.148423280
## Sr          0.397370524 -0.21928875  0.18185959  0.01557175 -0.046205697
##                    PC6         PC7         PC8          PC9         PC10
## Cl         -0.263444519 -0.14724805  0.11194013  0.617400408  0.014266148
## SO4        -0.326973559  0.10836905 -0.35224248 -0.556947248 -0.060456300
## Na          0.085576495 -0.22535486  0.35289146 -0.212472720  0.025630476
## Mg          0.120549041 -0.34399941 -0.04269316  0.136219669  0.213178044
## Ca         -0.072199578  0.03984048 -0.32506713 -0.060016346  0.466022403
## Alkalinity  0.005349521  0.25887631  0.02396533 -0.063220218  0.287039761
## DOC        -0.070161791  0.57746148 -0.18774608  0.433894402  0.002403474
## As         -0.182586167  0.22028999  0.06243938 -0.065280136 -0.469723148
## Ba          0.011647102 -0.20372840  0.08592953 -0.001747892  0.419753121
## Cr          0.471725131  0.45187432  0.18137897 -0.139421629  0.297519926
## Li         -0.267461405 -0.05648531  0.41916527 -0.148616516 -0.053097898
## Mn         -0.662629339  0.06734947  0.02433138 -0.022321457  0.350583792
## Si         -0.158102018  0.27028626  0.60258902 -0.050381193  0.148724457
## Sr         -0.007638210 -0.14276992 -0.10937108 -0.015382221  0.131404946
##                   PC11        PC12
## Cl         -0.31678009  0.16705504
## SO4         0.06984223 -0.05115312
## Na         -0.39161023 -0.26073934
## Mg          0.27844747 -0.41366195
## Ca         -0.38499306 -0.20984063
## Alkalinity -0.43145074  0.31533016
## DOC         0.18269648 -0.18844948
## As         -0.25354296  0.13619708
## Ba          0.01749660  0.27748045
## Cr          0.02911648  0.09597598
## Li          0.04794181  0.13796949
## Mn          0.11538943 -0.08795911
## Si          0.26163933 -0.18423782
## Sr          0.38291939  0.62122698
```

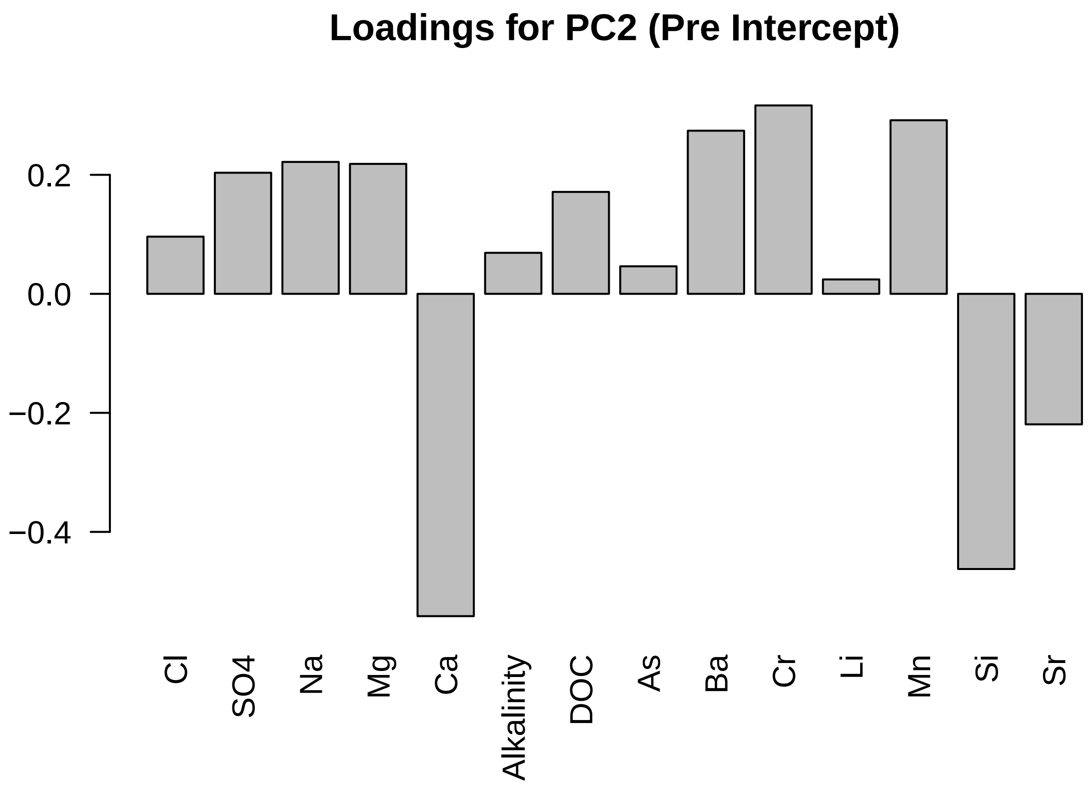
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_pre_intercept, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Intercepts (Pre-Application)")
```



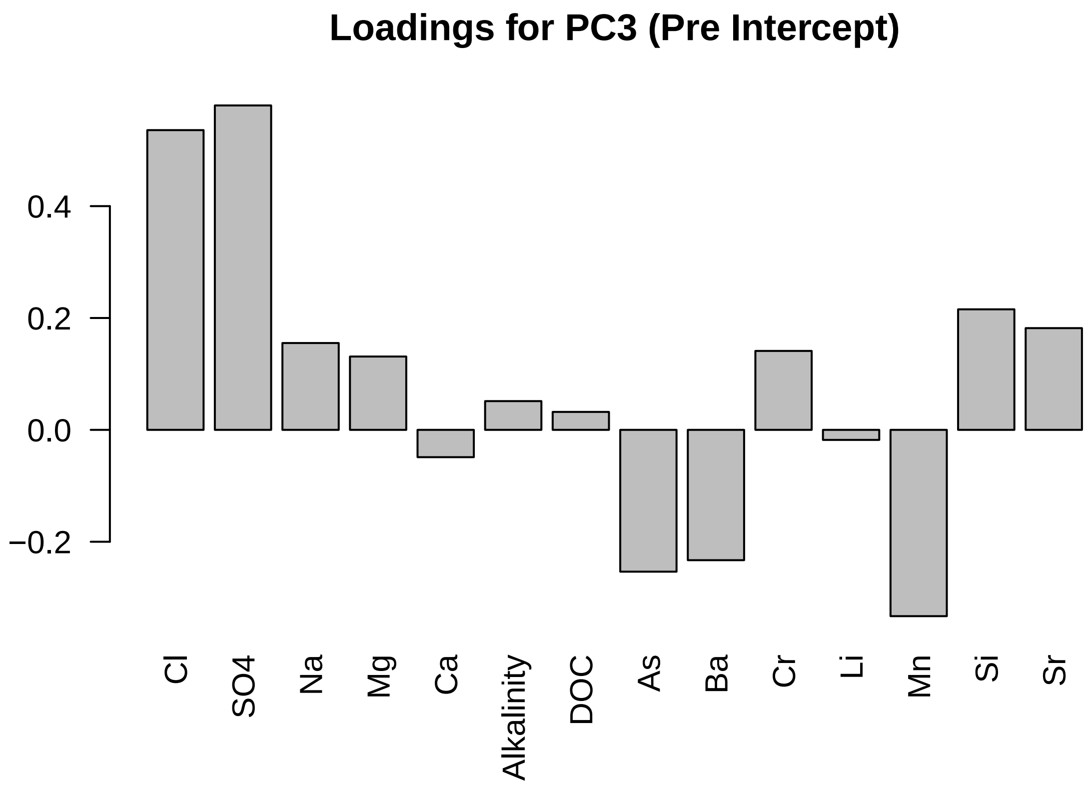

```r
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_pre_intercept[, 1], main = "Loadings for PC1 (Pre Intercept)", las = 2)
```



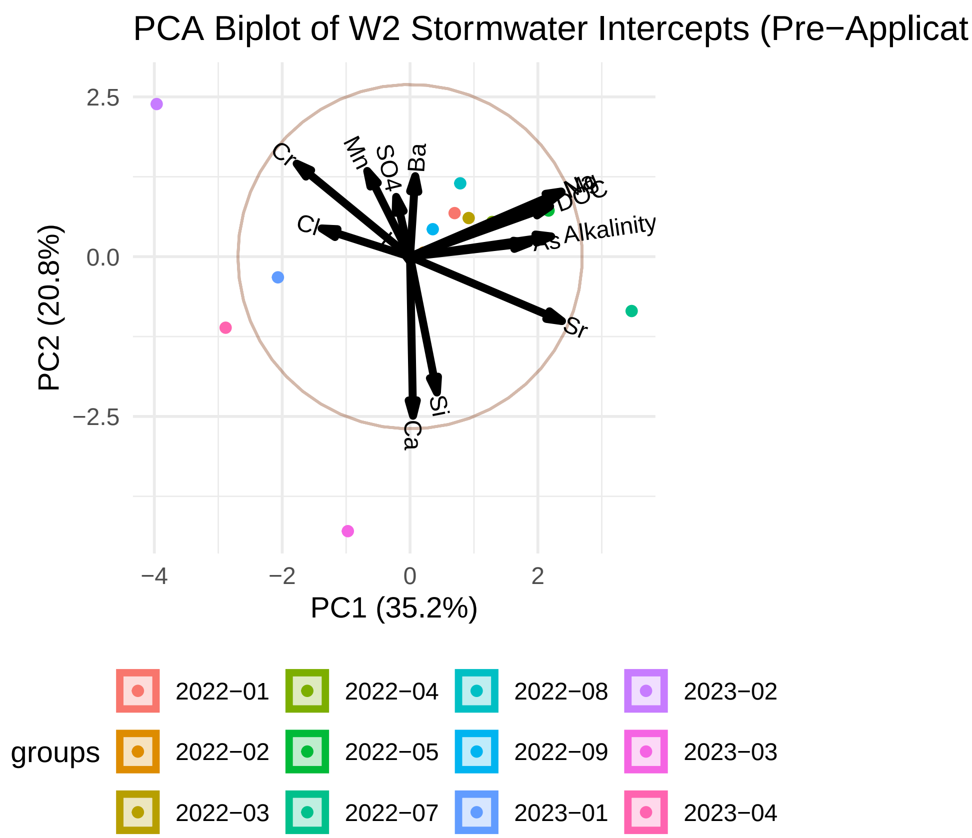

```r
barplot(loadings_pre_intercept[, 2], main = "Loadings for PC2 (Pre Intercept)", las = 2)
```



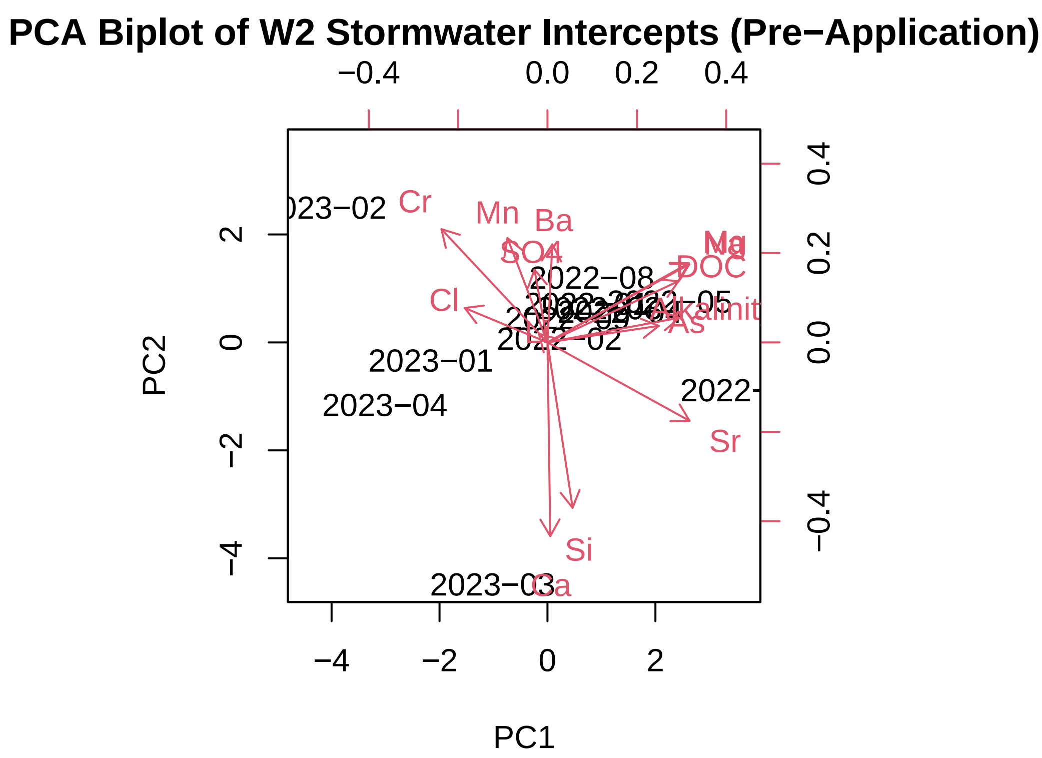

```r
barplot(loadings_pre_intercept[, 3], main = "Loadings for PC3 (Pre Intercept)", las = 2)
```




```
# Generate the biplot
biplot_pre_intercept <- ggbiplot(pca_result_pre_intercept,
                                 obs.scale = 1,
                                 var.scale = 1,
                                 groups = rownames(w2_std_pre_intercept),  # Use Rain_Event_Number as group
                                 ellipse = TRUE,  # Add confidence ellipses
                                 circle = TRUE)   # Add a unit circle

# Customize the biplot
biplot_pre_intercept <- biplot_pre_intercept +
  ggtitle("PCA Biplot of W2 Stormwater Intercepts (Pre-Application)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Print the biplot
print(biplot_pre_intercept)
```




```r
# Simple base R biplot
biplot(pca_result_pre_intercept, scale = 0,
       main = "PCA Biplot of W2 Stormwater Intercepts (Pre-Application)")
```



Great, we're off to a great start! Now let's do this again for post-storms

```
# We will first start by thinking just about post-basalt application storm slopes
# First, standardize our data (excluding the first three columns that aren't numeric)
w2_std_post_slope <- W2_Stormwater_slope %>%
  filter(Treatment == "Post")

w2_std_post_slope <- scale(w2_std_post_slope[, -c(1:2)])

# Convert it back to a dataframe (since scale returns a matrix)
w2_std_post_slope <- as.data.frame(w2_std_post_slope)

# Rename columns of w2_std_pre_intercept
colnames(w2_std_post_slope) <- gsub("slope_|_ppm|_uM|_ugL", "", colnames(w2_std_post_slope))

# Check the new column names
colnames(w2_std_post_slope)
```

```
##  [1] "Cl"         "SO4"        "Na"         "Mg"         "Ca"
##  [6] "Alkalinity" "DOC"        "As"         "Ba"         "Cr"
## [11] "Li"         "Mn"         "Si"         "Sr"
```

```
# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_std_post_slope) <- c("2023-05", "2023-06", "2023-07", "2023-08", "2023-09", "2023-12", "202
                            "2023-14", "2023-15", "2023-16", "2023-17", "2023-18", "2023-19")
```

```r
# Viewing the first few rows of the standardized data
head(w2_std_post_slope) # looks good to me!
```

```
##          Cl SO4         Na         Mg         Ca Alkalinity        DOC
## 2023-05 NaN NaN  0.5586417  0.4956966  0.3441058  0.9802014 -0.2776450
## 2023-06 NaN NaN -0.5331273 -0.5127095  0.3441058 -1.0014101  0.9114820
## 2023-07 NaN NaN -2.9574229 -3.0105925 -3.2218242 -1.2302407  0.3628312
## 2023-08 NaN NaN -0.6426660 -0.5741833 -0.5633400 -1.4325055  0.3807299
## 2023-09 NaN NaN -0.1270901  0.4956966  0.3441058 -0.2760112 -0.2776450
## 2023-12 NaN NaN -0.2088275 -0.3637842  0.3441058 -1.0280207  0.6481013
##                 As         Ba  Cr           Li         Mn         Si         Sr
## 2023-05 -0.5498953  0.9737287 NaN  0.925234738  0.8957228  1.0059068  0.9699187
## 2023-06  2.7378231 -1.3091253 NaN -0.916663359 -0.2730181 -0.9645235 -1.2187599
## 2023-07  0.4544612 -1.0612752 NaN -0.921908760  0.2943797 -1.2969968 -1.2991275
## 2023-08  0.7495470 -1.1639152 NaN  0.160996726  0.4148695 -0.9519496 -1.3256803
## 2023-09 -0.5498953 -0.6606360 NaN -0.004701235  0.3729829 -0.4163017 -0.4296311
## 2023-12 -0.5498953 -0.9234009 NaN -1.420101329  0.8957228 -1.0058492 -0.9294811
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
# prcomp() won't work immediately because Cl slope and intercept, SO4 slope, and Cr slope being 0;
# Having three rows of 0s will not be helpful for this analysis anyways, let's remove it
w2_std_post_slope <- w2_std_post_slope[, !colnames(w2_std_post_slope) %in% c("Cl", "SO4", "Cr")]

pca_result_post_slope <- prcomp(w2_std_post_slope, center = FALSE, scale. = FALSE)
summary(pca_result_post_slope)  # View variance explained by each component
```

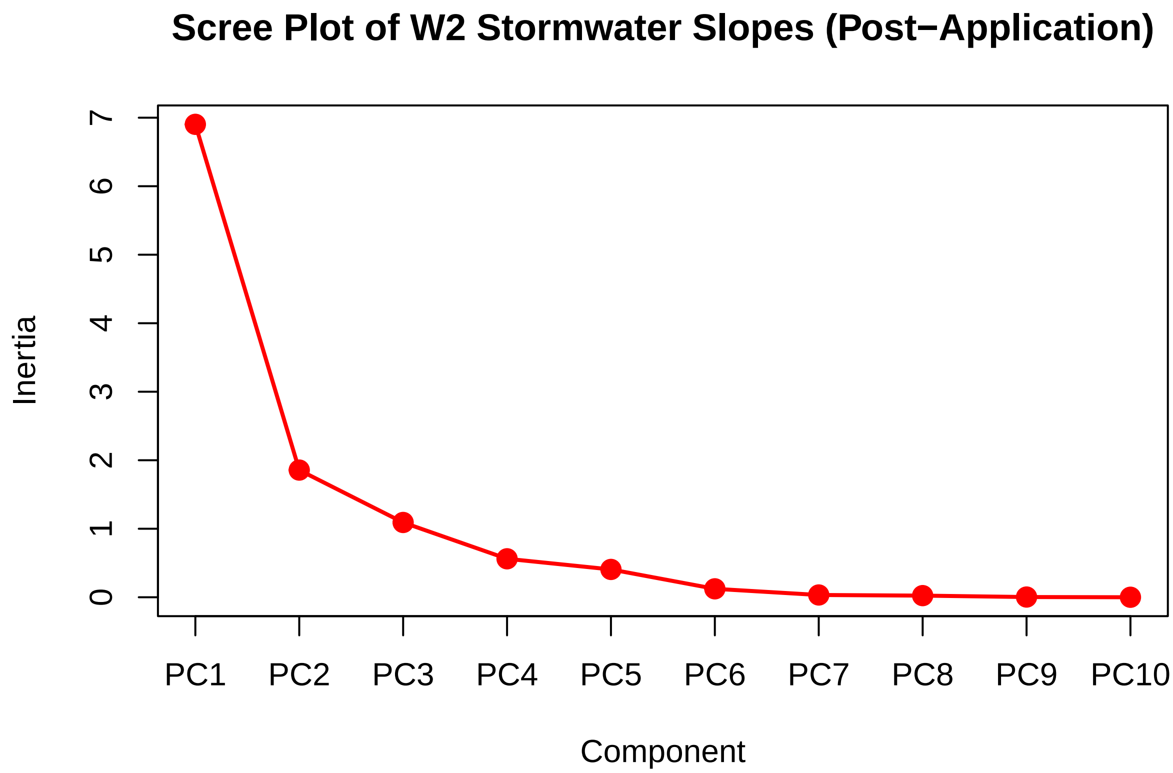
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.6274 1.3624 1.04477 0.74907 0.63715 0.35028 0.18132
## Proportion of Variance 0.6275 0.1688 0.09923 0.05101 0.03691 0.01115 0.00299
## Cumulative Proportion  0.6275 0.7963 0.89552 0.94653 0.98344 0.99459 0.99758
##                            PC8     PC9     PC10     PC11
## Standard deviation     0.15368 0.05483 1.78e-16 7.14e-17
## Proportion of Variance 0.00215 0.00027 0.00e+00 0.00e+00
## Cumulative Proportion  0.99973 1.00000 1.00e+00 1.00e+00
```

```r
# View the loadings
loadings_post_slope <- pca_result_post_slope$rotation
print(loadings_post_slope)
```

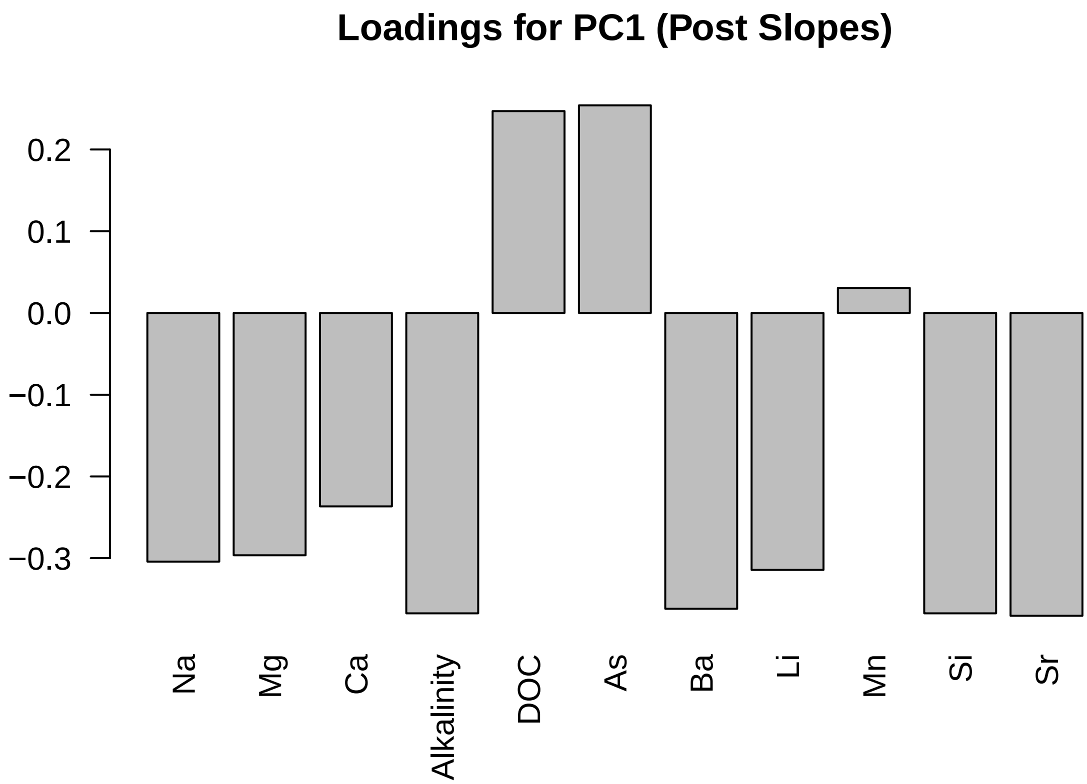
```
##                    PC1         PC2          PC3         PC4         PC5
## Na         -0.30416670 -0.43083032 -0.061375618 -0.03524140  0.01333912
## Mg         -0.29649076 -0.44029128 -0.073050515 -0.02433793  0.16061149
## Ca         -0.23665489 -0.54122335 -0.116122272  0.12139852  0.26599066
## Alkalinity -0.36748227  0.10600139  0.009442126  0.06374708 -0.25760945
## DOC         0.24698896 -0.30333907 -0.399334479 -0.12721220 -0.71620904
## As          0.25399775 -0.21423670  0.192793998  0.84463305 -0.23173321
## Ba         -0.36188891  0.15007500  0.001992801 -0.06447220 -0.32609375
## Li         -0.31434481  0.28246577 -0.025471982  0.41359365  0.10466438
## Mn          0.03065683  0.21893173 -0.883068686  0.21623862  0.22008740
```

```
## Si          -0.36748829  0.14389136 -0.014883785  0.16388207 -0.16650936
## Sr          -0.37046587  0.08667532  0.000970918 -0.03071850 -0.27713008
##                      PC6          PC7          PC8         PC9          PC10
## Na          -0.009571395 -0.374480488 -0.528751542 -0.48545212  2.242454e-01
## Mg          -0.209761098 -0.460841799  0.396111488  0.45658417 -1.983842e-02
## Ca           0.132621337  0.692677168  0.028412332  0.04744290 -1.402287e-01
## Alkalinity  0.283178472 -0.009624287  0.569338128 -0.20491694  4.700924e-01
## DOC         -0.359439536  0.145899719  0.080166014 -0.03828055  7.379529e-17
## As           0.218809289 -0.159691978 -0.046955684  0.09251515 -3.404962e-02
## Ba           0.136594562  0.035698443 -0.460620381  0.65923245  1.008998e-01
## Li          -0.747143560  0.065986919 -0.007484622 -0.08218544 -6.728266e-02
## Mn           0.235663615 -0.134512464 -0.039443501  0.03056814 -8.591601e-17
## Si           0.021180706  0.275863776 -0.085869542 -0.11260703  1.998272e-01
## Sr           0.219518525 -0.144511523  0.093682348 -0.21597284 -8.080046e-01
##                     PC11
## Na          -1.040470e-01
## Mg           2.537128e-01
## Ca          -1.788476e-01
## Alkalinity -3.402920e-01
## DOC         -3.871388e-17
## As           3.267314e-03
## Ba          -2.431961e-01
## Li          -2.546018e-01
## Mn           4.521627e-17
## Si           8.080279e-01
## Sr          -1.151954e-02
```

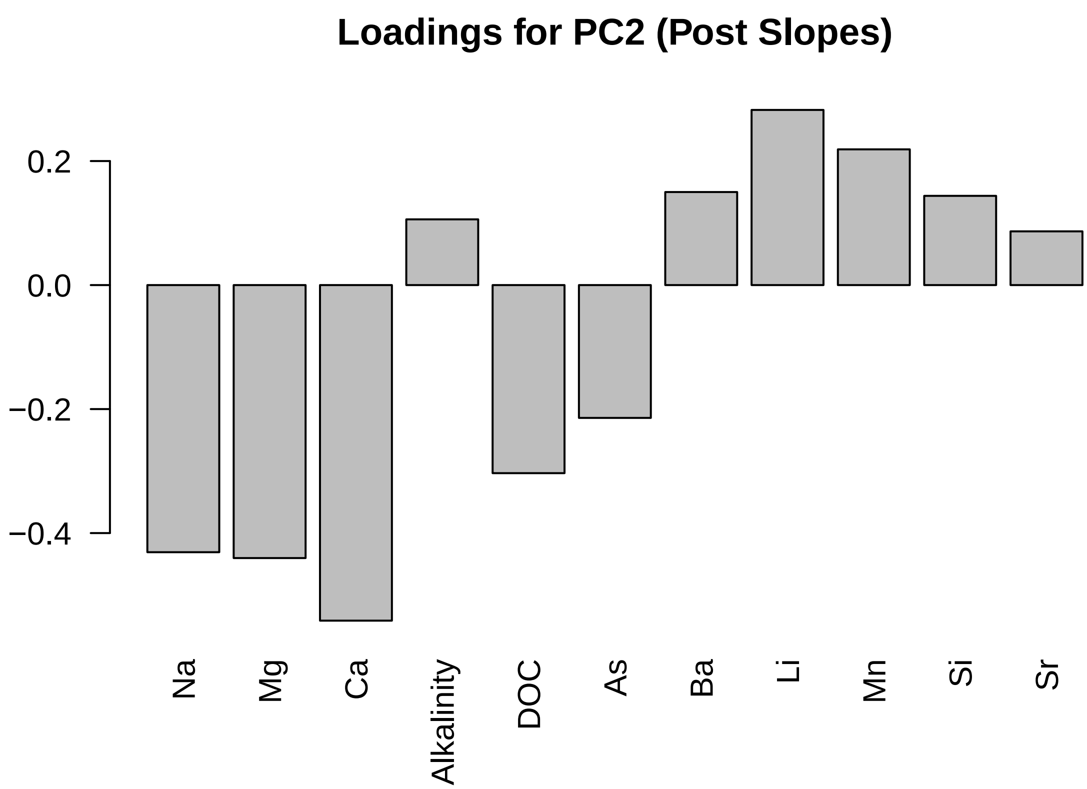
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_post_slope, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Slopes (Post-Application)")
```



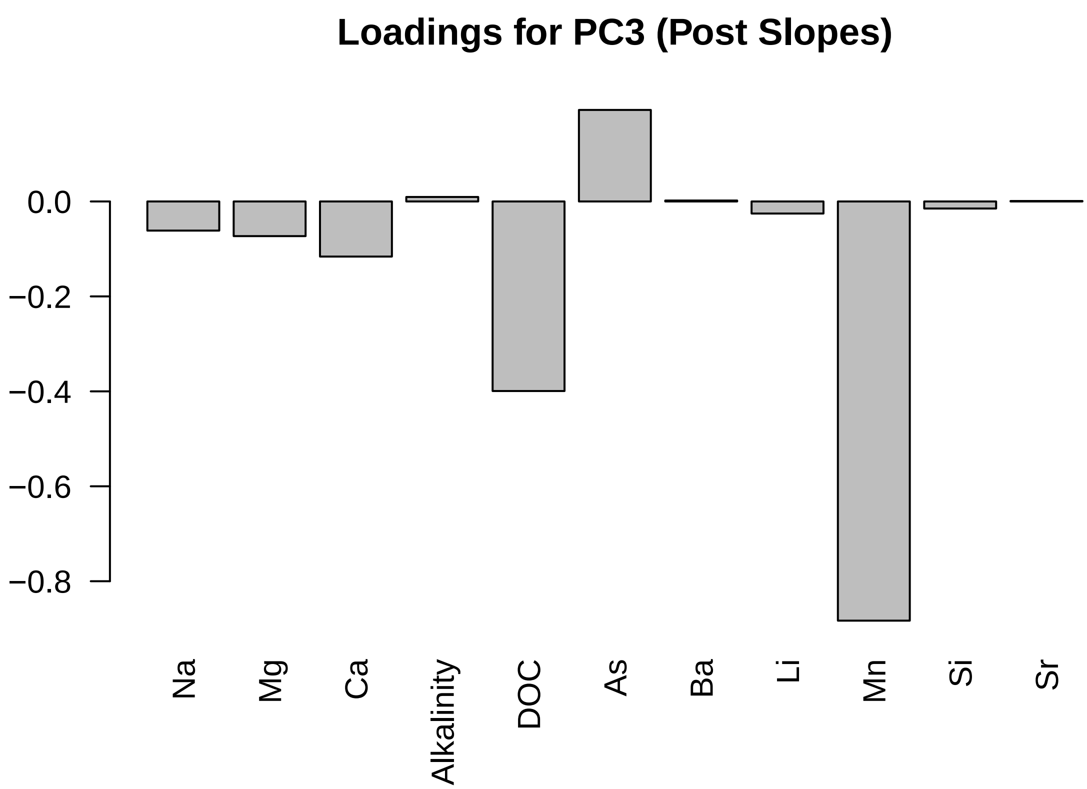

```r
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_post_slope[, 1], main = "Loadings for PC1 (Post Slopes)", las = 2)
```



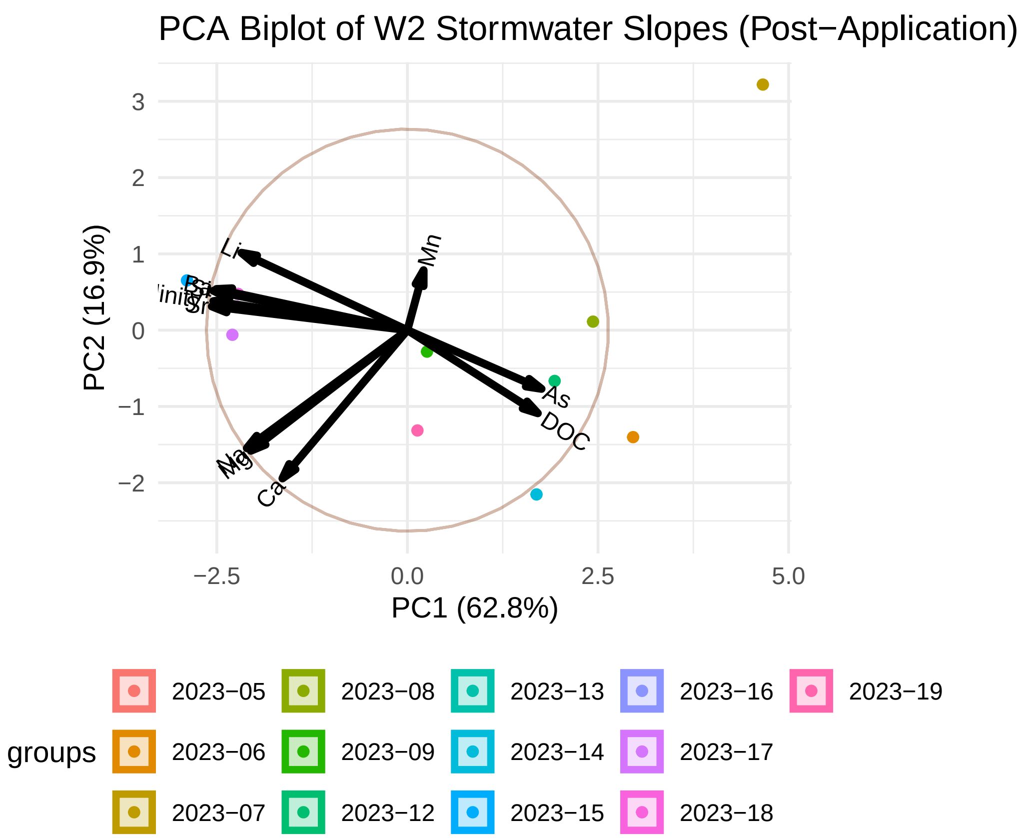

```r
barplot(loadings_post_slope[, 2], main = "Loadings for PC2 (Post Slopes)", las = 2)
```



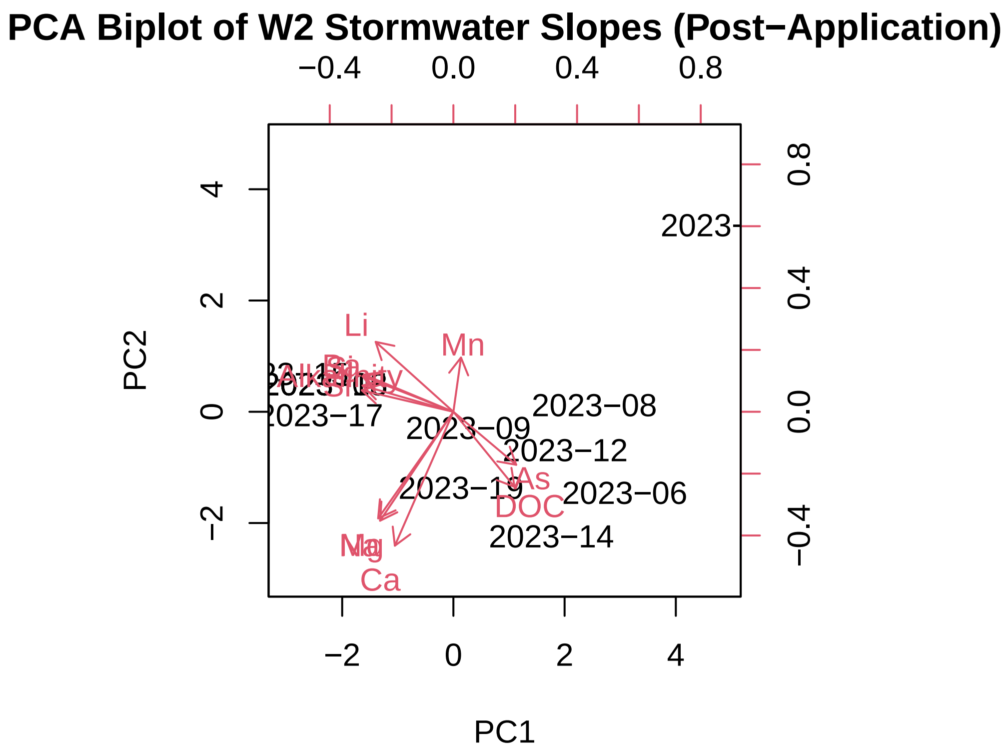

```
barplot(loadings_post_slope[, 3], main = "Loadings for PC3 (Post Slopes)", las = 2)
```




```r
# Generate the biplot
biplot_post_slope <- ggbiplot(pca_result_post_slope,
                              obs.scale = 1,
                              var.scale = 1,
                              groups = rownames(w2_std_post_slope),  # Use Rain_Event_Number as group la
                              ellipse = TRUE,  # Add confidence ellipses
                              circle = TRUE)   # Add a unit circle

# Customize the biplot
biplot_post_slope <- biplot_post_slope +
  ggtitle("PCA Biplot of W2 Stormwater Slopes (Post-Application)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Print the biplot
print(biplot_post_slope)
```




```r
# Simple base R biplot
biplot(pca_result_post_slope, scale = 0,
       main = "PCA Biplot of W2 Stormwater Slopes (Post-Application)")
```




```r
# We will first start by thinking just about pre-basalt application storms
# First, standardize our data (excluding the first three columns that aren't numeric)
w2_std_post_intercept <- W2_Stormwater_intercept %>%
  filter(Treatment == "Post")

w2_std_post_intercept <- scale(w2_std_post_intercept[, -c(1:2)])

# Convert it back to a dataframe (since scale returns a matrix)
w2_std_post_intercept <- as.data.frame(w2_std_post_intercept)

# Rename columns of w2_std_pre_intercept
colnames(w2_std_post_intercept) <- gsub("intercept_|_ppm|_uM|_ugL", "", colnames(w2_std_post_intercept))

# Check the new column names
colnames(w2_std_post_intercept)
```

```
##  [1] "Cl"         "SO4"        "Na"         "Mg"         "Ca"
##  [6] "Alkalinity" "DOC"        "As"         "Ba"         "Cr"
## [11] "Li"         "Mn"         "Si"         "Sr"
```

```r
# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_std_post_intercept) <- c("2023-05", "2023-06", "2023-07", "2023-08", "2023-09", "2023-12",
                          "2023-14", "2023-15", "2023-16", "2023-17", "2023-18", "2023-19")

# Viewing the first few rows of the standardized data
head(w2_std_post_intercept) # looks good to me!
```

```
##          Cl        SO4         Na         Mg          Ca Alkalinity        DOC
## 2023-05 NaN -0.4094091 -0.6419600 -0.8611717 -0.18202728 -1.2774675  0.3870766
## 2023-06 NaN -0.4094091  0.6332090  0.5465100  0.24706059  0.3214376 -0.5699354
## 2023-07 NaN  2.1638039  2.6667311  2.4938471  2.04227484  0.4895760  0.2258070
## 2023-08 NaN  2.3396962  0.8052088  0.5577456  0.25351426  0.6439460  0.1757677
## 2023-09 NaN -0.4094091  0.4399899  0.1553489  0.04074547 -0.3218917  0.8705042
## 2023-12 NaN -0.4094091  0.5905409  0.5658263  0.27806198  0.9143621 -0.5699354
##                 As         Ba         Cr         Li         Mn         Si
## 2023-05 -0.1582196 -1.2422843 -0.7965264  0.2773501 -0.3685909 -1.3699961
## 2023-06 -1.2905226  0.8382346 -0.8196047  0.2773501  0.1773054  0.3259206
## 2023-07 -0.2625318  0.5064675 -0.9312965  0.2773501 -0.4959238  0.7468591
## 2023-08 -0.4018591  0.7508031 -0.6427507 -3.3282012 -0.2670484  0.4717829
## 2023-09  0.1588602  1.1842500  1.0305922  0.2773501 -0.2591821  0.4868160
## 2023-12 -0.1467940  1.1464408 -0.8577811  0.2773501 -1.3725648  0.8217398
##                 Sr
## 2023-05 -2.0563786
## 2023-06  0.4113331
## 2023-07  0.3818467
## 2023-08  0.4525672
## 2023-09  0.9654409
## 2023-12  1.6086974
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
# prcomp() won't work immediately because Cl slope and intercept, SO4 slope, and Cr slope being 0;
# Having three rows of 0s will not be helpful for this analysis anyways, let's remove it
w2_std_post_intercept <- w2_std_post_intercept[, !colnames(w2_std_post_intercept) %in% c("Cl")]

pca_result_post_intercept <- prcomp(w2_std_post_intercept, center = FALSE, scale. = FALSE)
summary(pca_result_post_intercept)  # View variance explained by each component
```

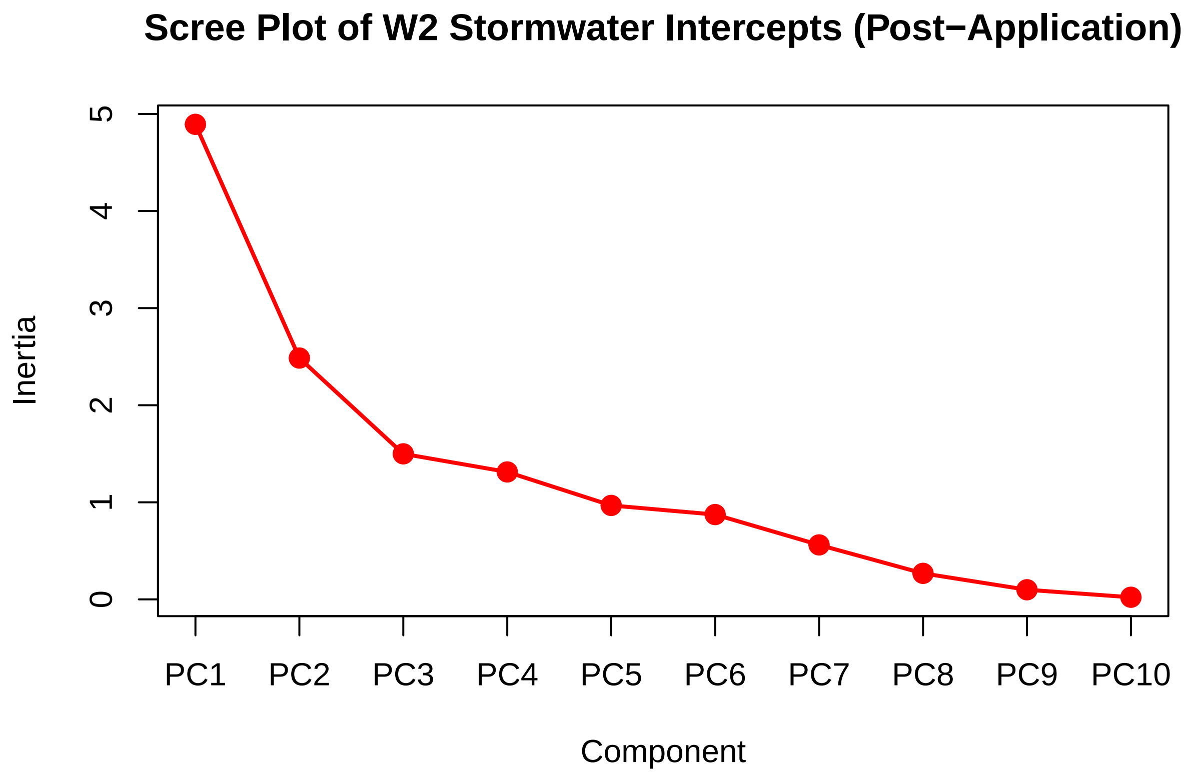
```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.2121 1.5766 1.2244 1.1457 0.98374 0.93475 0.74911
## Proportion of Variance 0.3764 0.1912 0.1153 0.1010 0.07444 0.06721 0.04317
## Cumulative Proportion  0.3764 0.5676 0.6829 0.7839 0.85834 0.92555 0.96872
##                            PC8     PC9    PC10    PC11    PC12      PC13
## Standard deviation     0.51756 0.31475 0.14808 0.11519 0.06724 2.004e-15
## Proportion of Variance 0.02061 0.00762 0.00169 0.00102 0.00035 0.000e+00
## Cumulative Proportion  0.98932 0.99694 0.99863 0.99965 1.00000 1.000e+00
```

```r
# View the loadings
loadings_post_intercept <- pca_result_post_intercept$rotation
print(loadings_post_intercept)
```

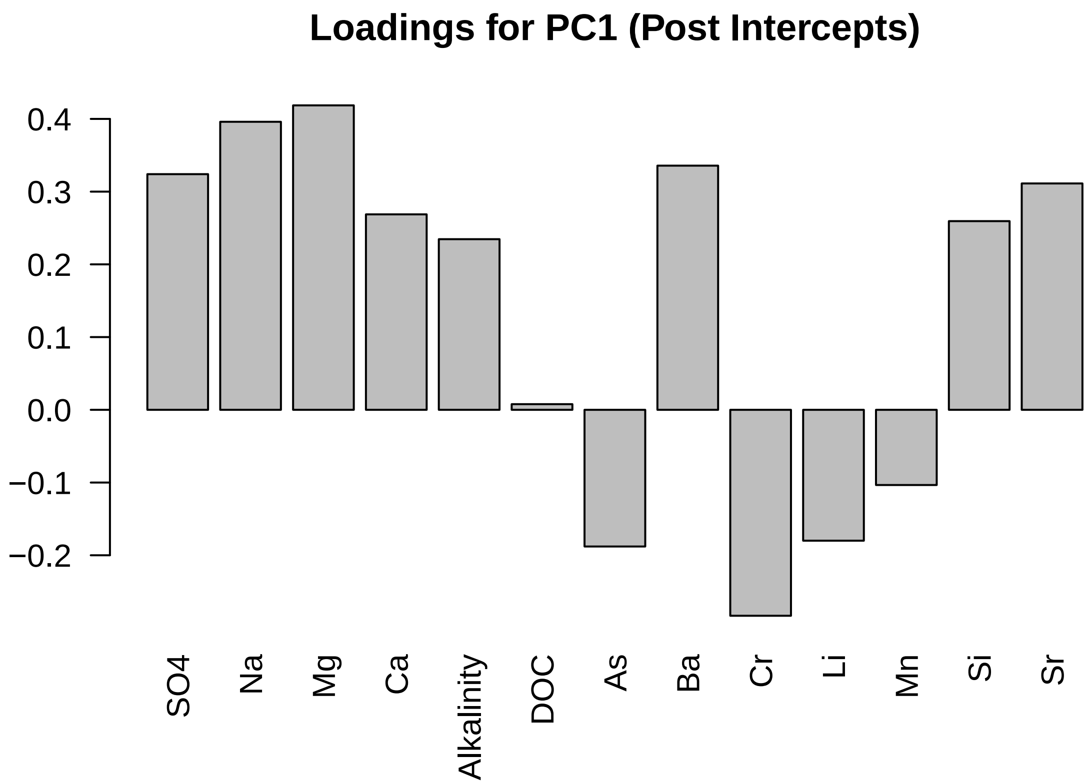
```
##                    PC1         PC2         PC3          PC4         PC5
## SO4        0.324069787  0.071974403 -0.40287069 -0.371673296  0.13563238
## Na         0.396056172  0.046644218 -0.11932701 -0.036168140  0.24624892
## Mg         0.418633568  0.085478146 -0.05649867  0.189197943  0.10993822
## Ca         0.268761487  0.475407830  0.02498970  0.063339863  0.19081973
```

```
## Alkalinity  0.234622133 -0.446793748 -0.01037884  0.045421554 -0.42968473
## DOC          0.007777982  0.174318010  0.31037604 -0.562172890 -0.39000607
## As          -0.187947614 -0.359862849 -0.04466718 -0.162975837  0.62748136
## Ba           0.335736806 -0.053417592  0.43436535 -0.064279861  0.19074108
## Cr          -0.283198831 -0.154253438  0.18985386 -0.425501364  0.26781067
## Li          -0.179952979  0.009819622  0.35431267  0.526503783  0.11565559
## Mn          -0.103414097  0.366287064  0.31757467 -0.110474425  0.01718579
## Si           0.259443394 -0.478136191  0.13315176 -0.008816069 -0.07668466
## Sr           0.311271467 -0.101428665  0.50235968 -0.037530800  0.11976382
##                      PC6         PC7         PC8         PC9        PC10
## SO4         -0.008891585 -0.18550930 -0.07308824  0.21924260  0.02978833
## Na          -0.358077505 -0.07035349 -0.03967681 -0.48661336 -0.50184522
## Mg          -0.113425351 -0.25022616 -0.05799710  0.05523725  0.73178788
## Ca          -0.167754331  0.04781927 -0.14484442  0.32385809 -0.17449889
## Alkalinity -0.035595563 -0.19802765 -0.27837988  0.26883407 -0.21785845
## DOC         -0.432661429 -0.11888237  0.40659704  0.13038261  0.05250361
## As           0.028506078 -0.25502377  0.36566182  0.35595793 -0.10005536
## Ba           0.227635910  0.23930464  0.32144692 -0.31575455  0.13122220
## Cr          -0.289123662  0.08885207 -0.62553931 -0.19363577  0.24208160
## Li          -0.517615796 -0.26266716  0.03103497  0.09543687 -0.03953195
## Mn           0.462511950 -0.66401762 -0.19773369 -0.11617101 -0.14186537
## Si           0.009337881 -0.33214020  0.02944752 -0.23276668  0.04005477
## Sr           0.155170390  0.30324879 -0.23803091  0.42100772 -0.14817204
##                    PC11         PC12        PC13
## SO4         -0.62582055  0.048196860 -0.29697284
## Na           0.24674592 -0.245714332 -0.14189642
## Mg           0.30672465 -0.232541172 -0.02833521
## Ca           0.05467205  0.392032898  0.56328075
## Alkalinity -0.08761533 -0.391632897  0.39970466
## DOC          0.12886713 -0.031375740 -0.03284572
## As           0.19103895 -0.123913013  0.15735170
## Ba          -0.42174794 -0.190657684  0.33419339
## Cr          -0.05704581 -0.009488891  0.15797675
## Li          -0.41694683 -0.027219117 -0.16810935
## Mn           0.02154207 -0.115000626 -0.02323106
## Si           0.06053405  0.713707111 -0.01699613
## Sr           0.17469499  0.006073397 -0.47143510
```

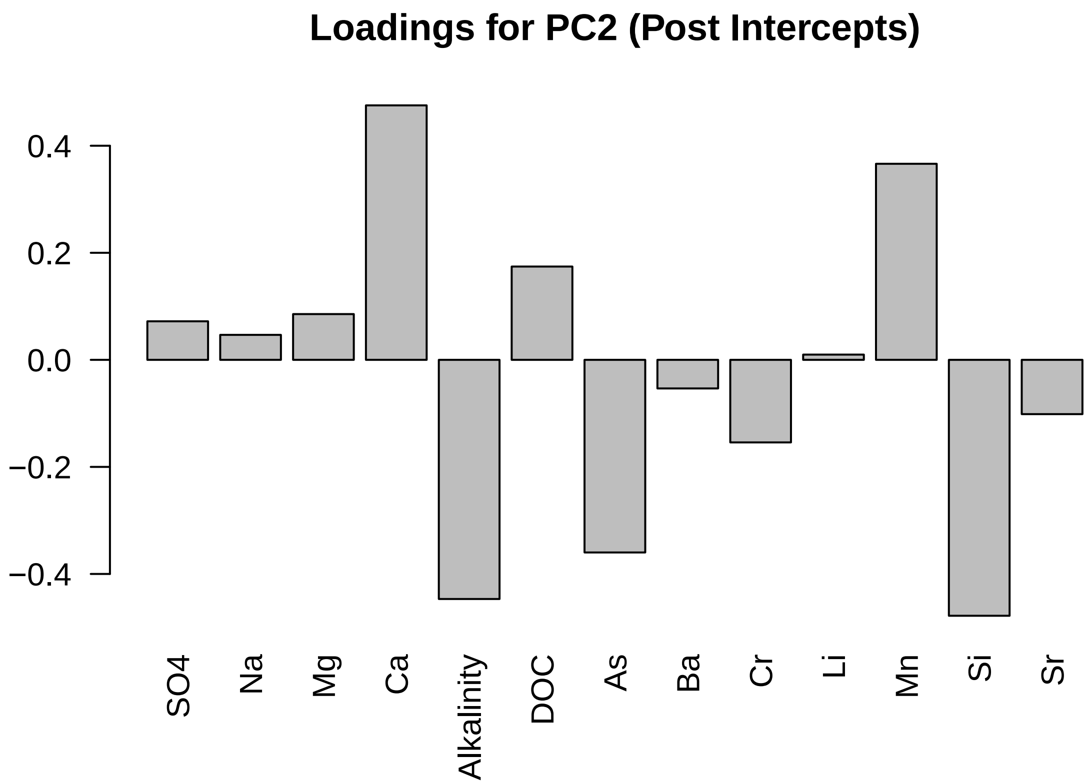
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_post_intercept, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Intercepts (Post-Application)")
```



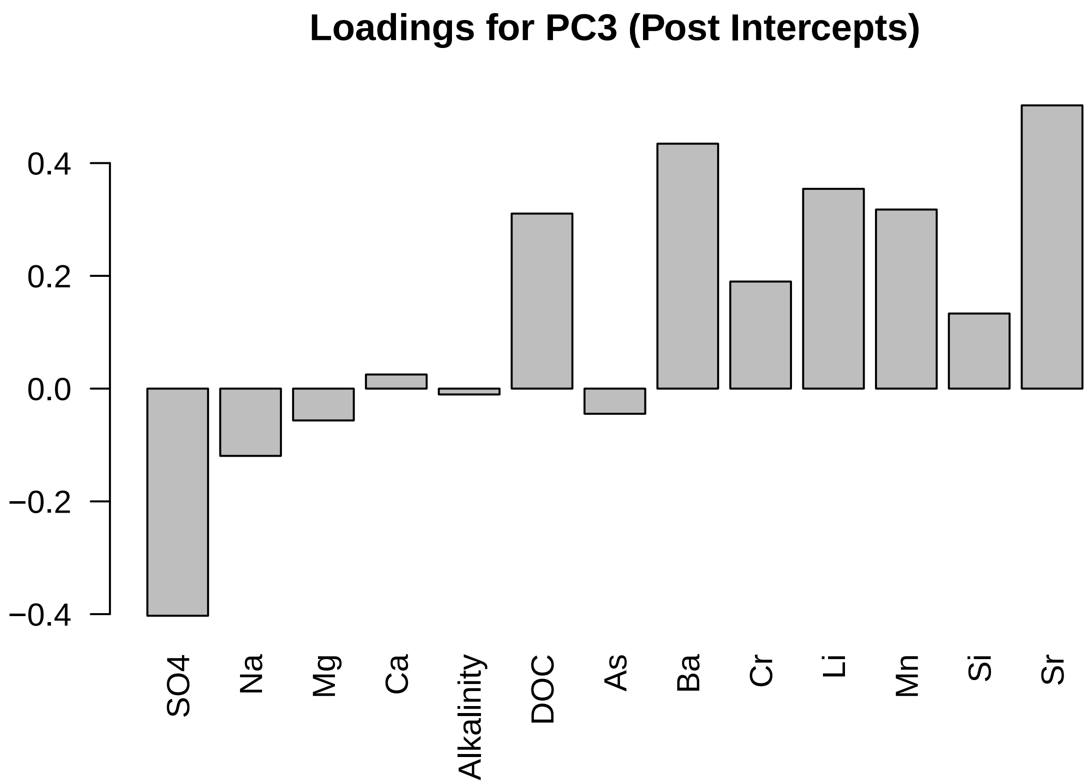

```r
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_post_intercept[, 1], main = "Loadings for PC1 (Post Intercepts)", las = 2)
```



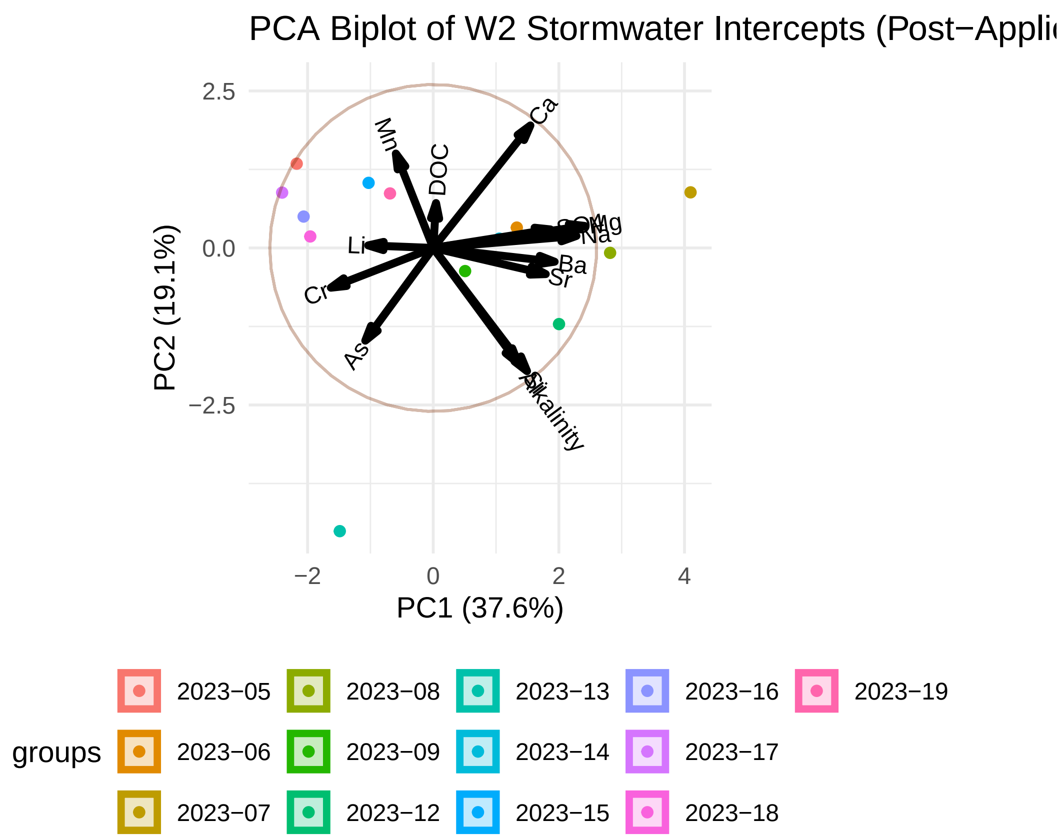

```
barplot(loadings_post_intercept[, 2], main = "Loadings for PC2 (Post Intercepts)", las = 2)
```



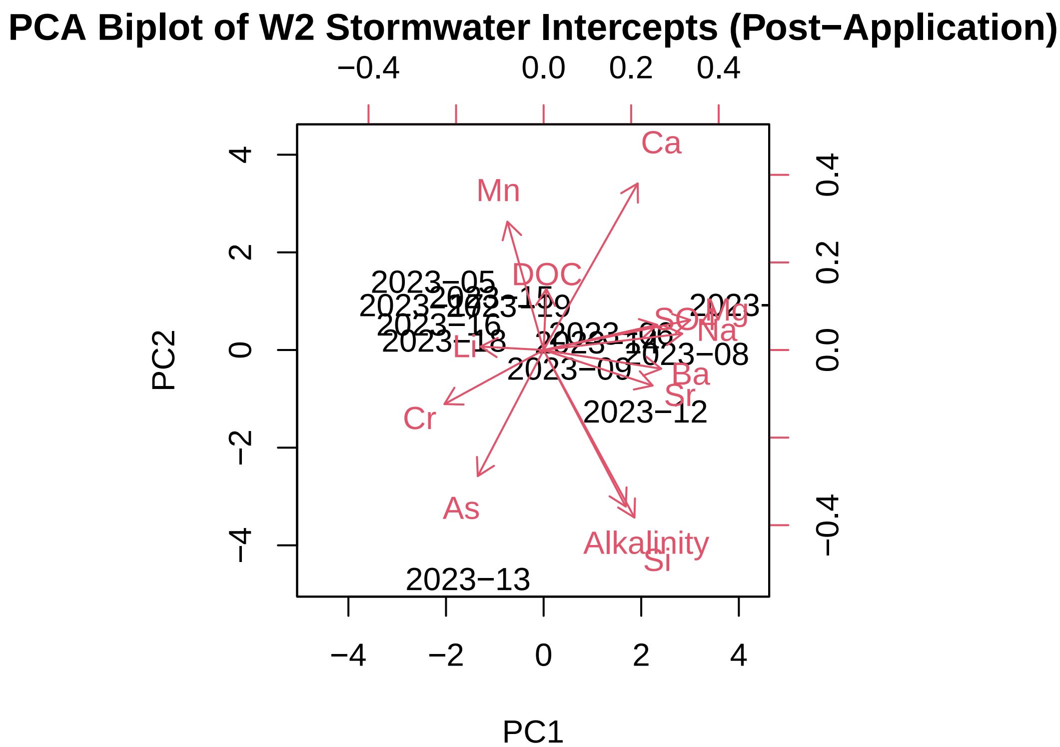

```
barplot(loadings_post_intercept[, 3], main = "Loadings for PC3 (Post Intercepts)", las = 2)
```




```r
# Generate the biplot
biplot_post_intercept <- ggbiplot(pca_result_post_intercept,
                                  obs.scale = 1,
                                  var.scale = 1,
                                  groups = rownames(w2_std_post_intercept),  # Use Rain_Event_Number as grou
                                  ellipse = TRUE,  # Add confidence ellipses
                                  circle = TRUE)   # Add a unit circle

# Customize the biplot
biplot_post_intercept <- biplot_post_intercept +
  ggtitle("PCA Biplot of W2 Stormwater Intercepts (Post-Application)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Print the biplot
print(biplot_post_intercept)
```




```r
# Simple base R biplot
biplot(pca_result_post_intercept, scale = 0,
       main = "PCA Biplot of W2 Stormwater Intercepts (Post-Application)")
```



Great! Now that we've considered pre and post basalt appplication storms seperately, let's do PCA with them together!

```r
# We will first start by thinking just about storm slopes
# First, standardize our data (excluding the first three columns that aren't numeric)
w2_std_all_slope <- W2_Stormwater_slope

w2_std_all_slope <- scale(w2_std_all_slope[, -c(1:2)])

# Convert it back to a dataframe (since scale returns a matrix)
w2_std_all_slope <- as.data.frame(w2_std_all_slope)

# Rename columns of w2_std_pre_intercept
colnames(w2_std_all_slope) <- gsub("slope_|_ppm|_uM|_ugL", "", colnames(w2_std_all_slope))

# Check the new column names
colnames(w2_std_all_slope)
```

```
##  [1] "Cl"          "SO4"         "Na"          "Mg"          "Ca"
##  [6] "Alkalinity" "DOC"         "As"          "Ba"          "Cr"
## [11] "Li"          "Mn"          "Si"          "Sr"
```

```r
# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_std_all_slope) <- c("2022-01", "2022-02", "2022-03", "2022-04", "2022-05", "2022-07", "2022-
                                "2022-09", "2023-01", "2023-02", "2023-03", "2023-04", "2023-05", "2023-06",
                                "2023-07", "2023-08", "2023-09", "2023-12", "2023-13", "2023-14", "2023-15",
```

```r
                                        "2023-16", "2023-17", "2023-18", "2023-19")

# Viewing the first few rows of the standardized data
head(w2_std_all_slope) # looks good to me!
```

```
##           Cl  SO4        Na         Mg        Ca Alkalinity        DOC
## 2022-01 -0.2 -0.2 0.4104945  0.4037185 0.2699294  0.8074926  0.9330072
## 2022-02 -0.2 -0.2 0.4104945  0.4037185 0.2699294  0.8074926 -0.5017627
## 2022-03 -0.2 -0.2 0.4104945  0.4037185 0.2699294  0.8074926  0.3448333
## 2022-04 -0.2 -0.2 0.4104945  0.4037185 0.2699294  0.3668095  0.7147674
## 2022-05 -0.2 -0.2 0.4104945  0.4037185 0.2699294  0.3011901  0.0980487
## 2022-07  4.8  4.8 0.4104945 -0.3064559 0.2699294 -0.7940828  0.7928437
##                 As         Ba         Cr          Li           Mn         Si
## 2022-01 -0.3648739 -0.2461521 -0.2454787  0.88122602  0.879568751  0.9003503
## 2022-02 -0.3648739  0.9257094 -0.2454787  0.88122602  0.879568751  0.9003503
## 2022-03 -0.3648739  0.2603709 -0.2454787  0.08551787  0.266460175  0.2918180
## 2022-04 -0.3648739  0.4317987 -0.2454787  0.08274224 -0.111727320  0.3255885
## 2022-05 -0.3648739  0.2797191 -0.2454787 -0.59470369  0.429575280  0.3758846
## 2022-07 -1.7713406 -0.1477291  4.6499296 -1.79821003 -0.008118198 -0.3857707
##                 Sr
## 2022-01  0.9097541
## 2022-02  0.9097541
## 2022-03  0.4713778
## 2022-04  0.3859534
## 2022-05  0.1362429
## 2022-07 -0.5040130
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
pca_result_all_slope <- prcomp(w2_std_all_slope, center = FALSE, scale. = FALSE)
summary(pca_result_all_slope)  # View variance explained by each component
```

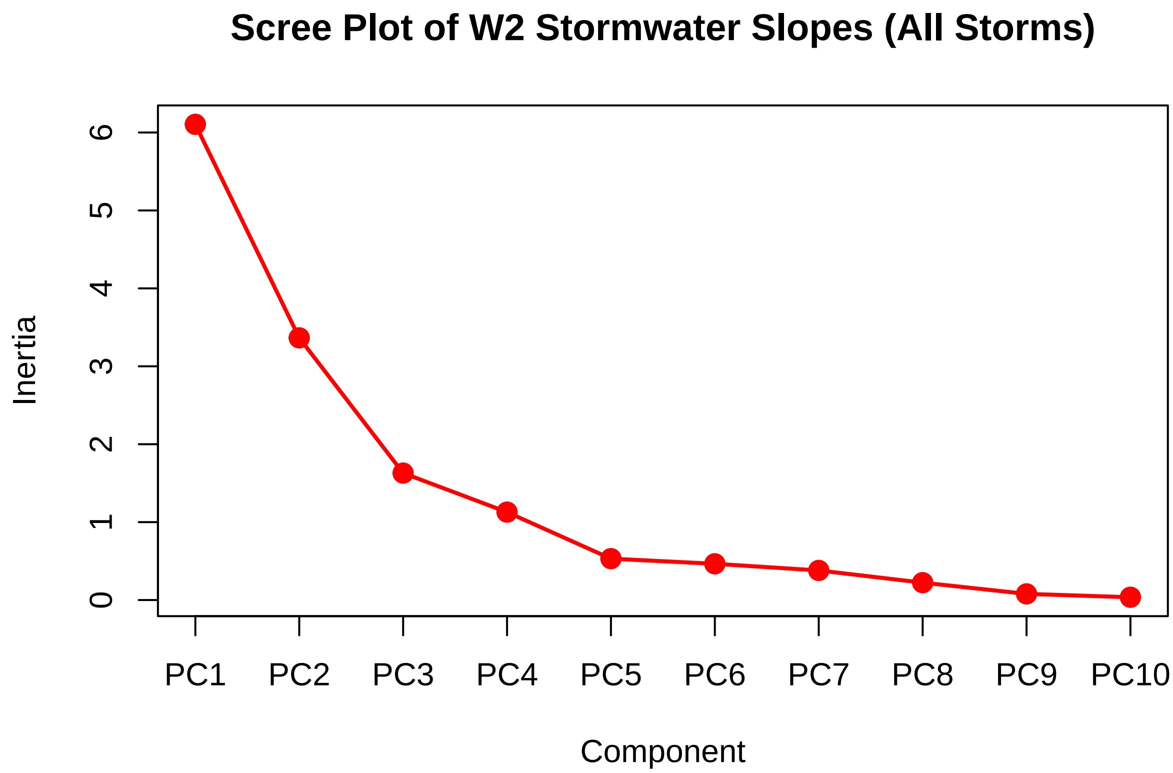
```
## Importance of components:
##                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.471 1.8343 1.2765 1.0623 0.72890 0.68187 0.61641
## Proportion of Variance 0.436 0.2403 0.1164 0.0806 0.03795 0.03321 0.02714
## Cumulative Proportion  0.436 0.6764 0.7928 0.8734 0.91132 0.94453 0.97167
##                            PC8     PC9    PC10    PC11    PC12    PC13
## Standard deviation     0.47250 0.28107 0.18991 0.17195 0.12727 0.11170
## Proportion of Variance 0.01595 0.00564 0.00258 0.00211 0.00116 0.00089
## Cumulative Proportion  0.98762 0.99326 0.99584 0.99795 0.99911 1.00000
##                             PC14
## Standard deviation     8.629e-17
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```r
# View the loadings
loadings_all_slope <- pca_result_all_slope$rotation
print(loadings_all_slope)
```

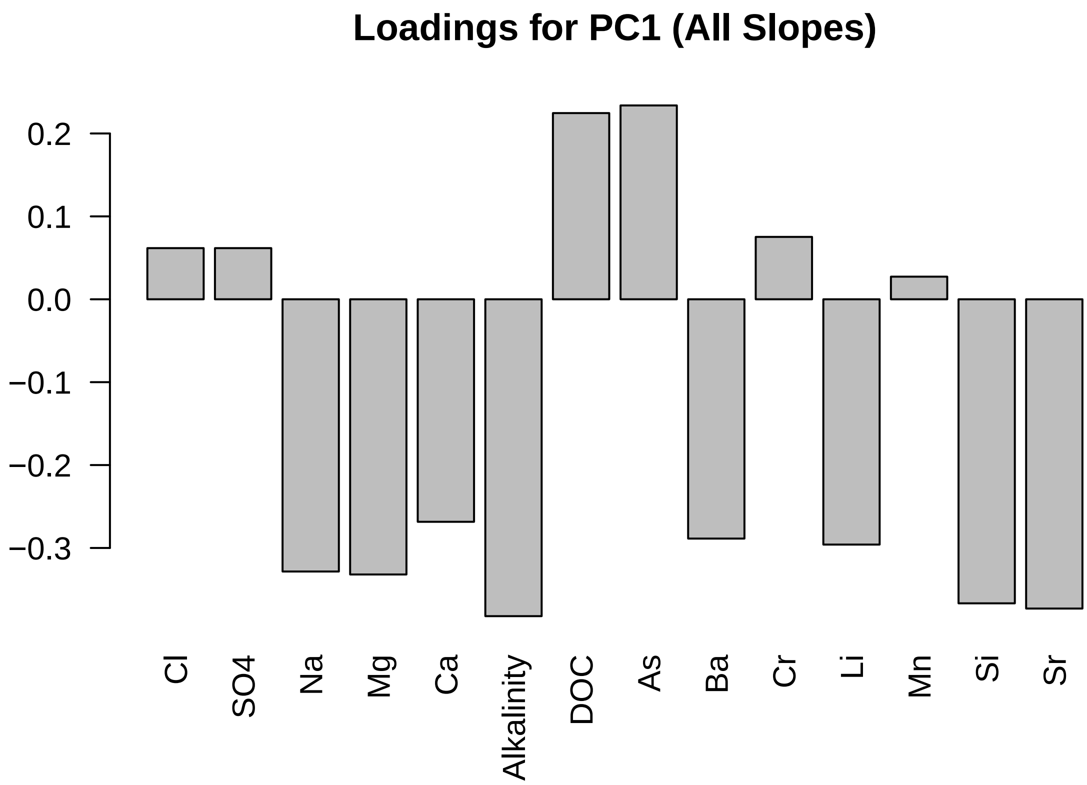
```
##                   PC1          PC2         PC3           PC4          PC5
## Cl         0.06171217 -0.524066771 -0.12090223 -0.0586237917  0.046570254
```

```
## SO4         0.06171217 -0.524066771 -0.12090223 -0.0586237917  0.046570254
## Na         -0.32840007 -0.161897948  0.38107776  0.0052898915  0.060142389
## Mg         -0.33196532 -0.081024610  0.41100822  0.0080454468  0.121360137
## Ca         -0.26845782 -0.137955735  0.51244423  0.0039119279  0.240850728
## Alkalinity -0.38226553  0.026004678 -0.12446988  0.0445398655 -0.159958570
## DOC         0.22460271 -0.108000927  0.38214300  0.3450569583 -0.611586878
## As          0.23381814  0.264095178  0.27934999 -0.1303723214  0.125404978
## Ba         -0.28868647 -0.038229416 -0.18841925  0.2039530148  0.392581917
## Cr          0.07525409 -0.520996273 -0.10060448 -0.0462134623  0.007120207
## Li         -0.29591070  0.208514695 -0.25923027 -0.0009867243  0.074104401
## Mn          0.02726572 -0.006195399 -0.09396743  0.8996400649  0.150869693
## Si         -0.36678194 -0.017245136 -0.13794090 -0.0130108903 -0.382260502
## Sr         -0.37307264 -0.010042610 -0.11736598 -0.0414538408 -0.418818364
##                      PC6         PC7         PC8          PC9         PC10
## Cl         -0.1096636096 -0.16471286  0.12658458  0.059995569  0.035014364
## SO4        -0.1096636096 -0.16471286  0.12658458  0.059995569  0.035014364
## Na          0.0087855677  0.04240623  0.01897937 -0.133415973  0.398736764
## Mg         -0.0664714983  0.11954238  0.02984424 -0.009132966  0.488454345
## Ca         -0.1780411380 -0.02990091 -0.04793822  0.202037258 -0.715361824
## Alkalinity  0.0001066788 -0.08926135  0.14233751 -0.827493409 -0.249065965
## DOC         0.2671845458 -0.07180438  0.45317744  0.076558953 -0.064070454
## As         -0.0253873247 -0.84748581 -0.08922479 -0.126279100  0.114042502
## Ba          0.7743577880 -0.19384891  0.14416259  0.153747958 -0.036103354
## Cr         -0.0148959256 -0.16156900 -0.10352691 -0.179553338  0.000660414
## Li         -0.4227314560 -0.19143794  0.68098868  0.241416460  0.054329665
## Mn         -0.2948972556 -0.08812323 -0.23067551 -0.029416701  0.062640306
## Si         -0.0336605671 -0.26694214 -0.35707302  0.265491419  0.028470056
## Sr          0.0327139374 -0.13433812 -0.23420997  0.205024320 -0.009605880
##                   PC11         PC12         PC13          PC14
## Cl          0.35800421  0.084508753  0.077514080 -7.071068e-01
## SO4         0.35800421  0.084508753  0.077514080  7.071068e-01
## Na          0.10847857 -0.048979458 -0.723798312  1.456049e-16
## Mg         -0.11182272  0.047517150  0.653054742  2.644237e-16
## Ca         -0.02676418 -0.009966775  0.002666770 -1.991482e-16
## Alkalinity  0.11460416  0.095232207  0.113681486 -1.405688e-16
## DOC        -0.06546111  0.044963729  0.023809636  1.092044e-16
## As          0.05463494 -0.067872462  0.037967373 -7.429568e-17
## Ba         -0.01974124  0.007299054  0.024582744 -1.447396e-17
## Cr         -0.76573739 -0.229358119 -0.048467246 -1.753513e-17
## Li         -0.21313354 -0.059181525 -0.083712525 -1.400244e-16
## Mn          0.04649993 -0.057908727 -0.005320449  1.004950e-16
## Si         -0.15215366  0.634099054 -0.058589064 -4.709214e-17
## Sr          0.20196813 -0.709694899  0.096215753 -3.007272e-17
```

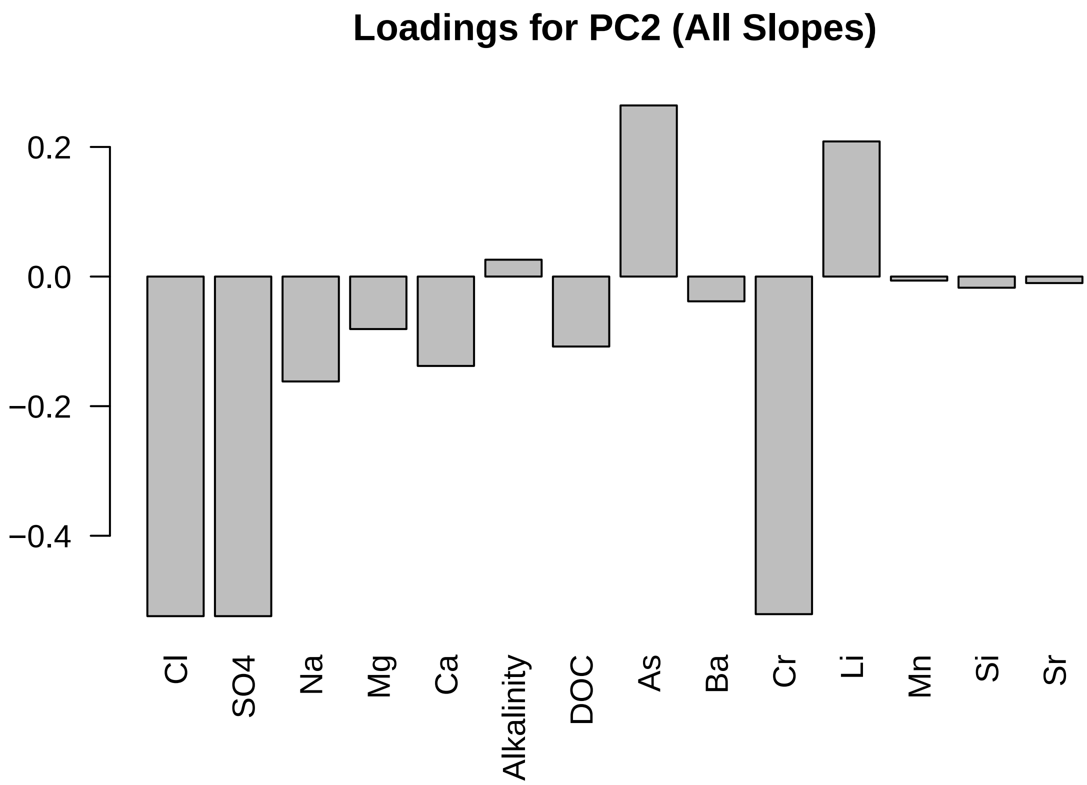
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_all_slope, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Slopes (All Storms)")
```



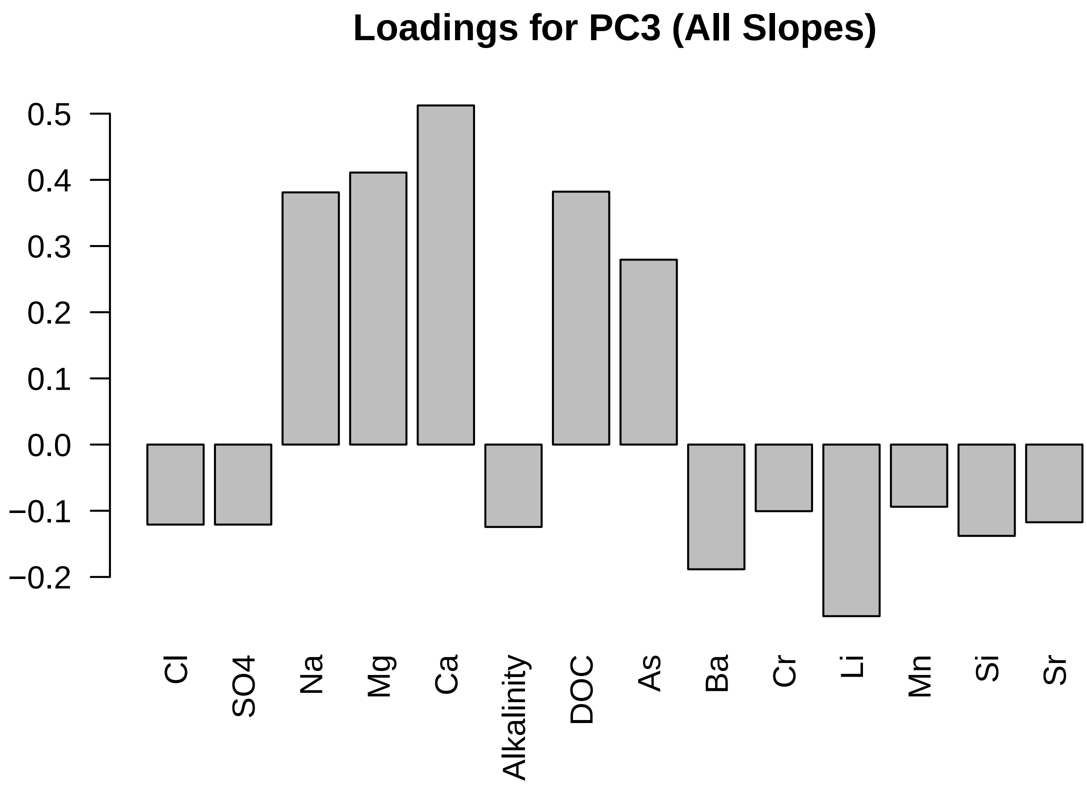

```r
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_all_slope[, 1], main = "Loadings for PC1 (All Slopes)", las = 2)
```



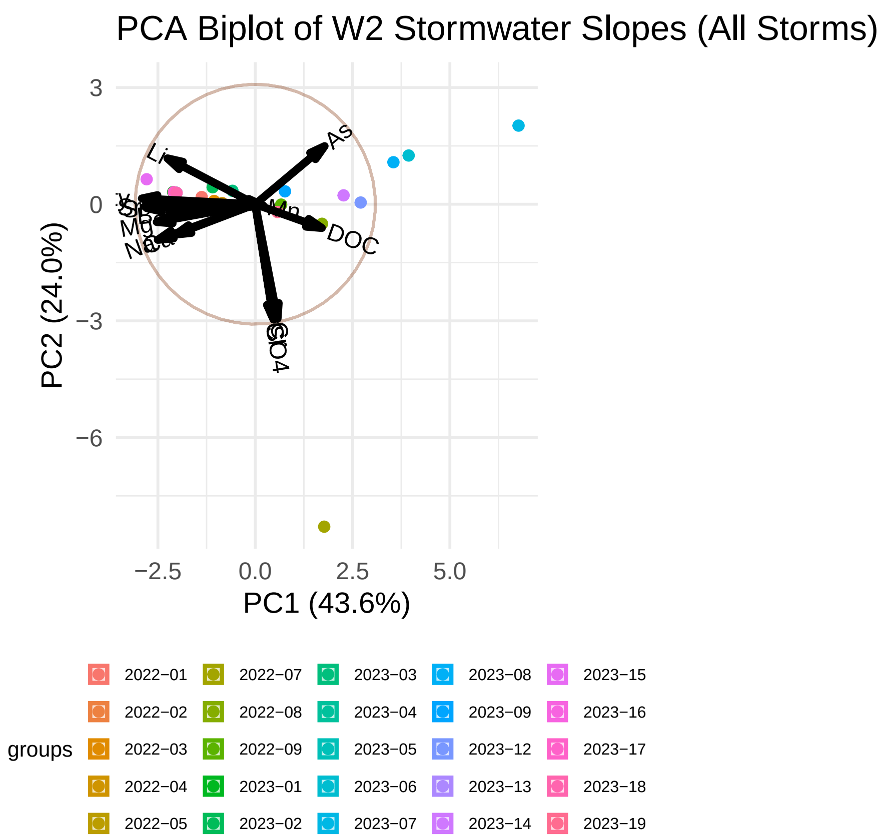

```r
barplot(loadings_all_slope[, 2], main = "Loadings for PC2 (All Slopes)", las = 2)
```



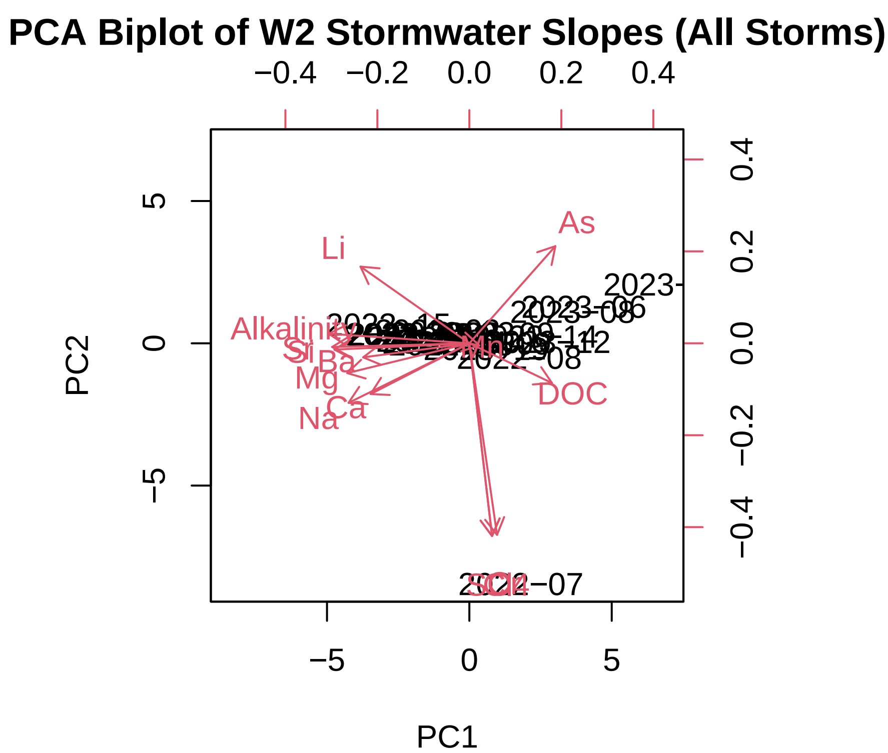

```
barplot(loadings_all_slope[, 3], main = "Loadings for PC3 (All Slopes)", las = 2)
```




```
# Generate the biplot
biplot_all_slope <- ggbiplot(pca_result_all_slope,
                              obs.scale = 1,
                              var.scale = 1,
                              groups = rownames(w2_std_all_slope),  # Use Rain_Event_Number as group lab
                              ellipse = TRUE,  # Add confidence ellipses
                              circle = TRUE)   # Add a unit circle

# Customize the biplot
 biplot_all_slope <- biplot_all_slope +
  ggtitle("PCA Biplot of W2 Stormwater Slopes (All Storms)") +
  theme_minimal() +
  theme(
    legend.position = "bottom",           # Position legend at the bottom
    legend.title = element_text(size = 8),  # Reduce legend title size
    legend.text = element_text(size = 6),   # Reduce legend label size
    legend.key.size = unit(0.3, "cm"),      # Reduce legend key size (symbols)
    legend.spacing.x = unit(0.1, "cm"),     # Reduce horizontal spacing between legend items
    legend.spacing.y = unit(0.08, "cm")      # Reduce vertical spacing between legend items (if vertica
  )


# Print the biplot
print(biplot_all_slope)
```




```r
# Simple base R biplot
biplot(pca_result_all_slope, scale = 0,
       main = "PCA Biplot of W2 Stormwater Slopes (All Storms)")
```




```r
# We will first start by thinking just about pre-basalt application storms
# First, standardize our data (excluding the first three columns that aren't numeric)
w2_std_all_intercept <- W2_Stormwater_intercept

w2_std_all_intercept <- scale(w2_std_all_intercept[, -c(1:2)])

# Convert it back to a dataframe (since scale returns a matrix)
w2_std_all_intercept <- as.data.frame(w2_std_all_intercept)

# Rename columns of w2_std_pre_intercept
colnames(w2_std_all_intercept) <- gsub("intercept_|_ppm|_uM|_ugL", "", colnames(w2_std_all_intercept))

# Check the new column names
colnames(w2_std_all_intercept)
```

```
##  [1] "Cl"          "SO4"         "Na"          "Mg"          "Ca"
##  [6] "Alkalinity" "DOC"         "As"          "Ba"          "Cr"
## [11] "Li"          "Mn"          "Si"          "Sr"
```

```r
# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_std_all_intercept) <-  c("2022-01", "2022-02", "2022-03", "2022-04", "2022-05", "2022-07",
                                     "2022-08", "2022-09", "2023-01", "2023-02", "2023-03", "2023-04",
                                     "2023-05", "2023-06", "2023-07", "2023-08", "2023-09", "2023-12",
                                     "2023-13", "2023-14", "2023-15", "2023-16", "2023-17", "2023-18",
```

```r
# Viewing the first few rows of the standardized data
head(w2_std_all_intercept) # looks good to me!
```

```
##           Cl        SO4        Na        Mg          Ca Alkalinity         DOC
## 2022-01  0.2  1.0986726 0.4867690 0.3070590 -0.14266371 -0.3128753 -0.19600645
## 2022-02  0.2 -0.3103240 0.1273499 0.2592490 -0.16319958 -0.3874910 -0.66012033
## 2022-03  0.2  1.0546065 0.3023815 0.3531660 -0.11710106 -0.2409802  0.08818313
## 2022-04  0.2  0.8467784 0.3712013 0.3960698 -0.09177101 -0.1537512  0.24348246
## 2022-05  0.2  1.1055868 0.5462932 0.4321343 -0.09406716 -0.1655477  1.01644798
## 2022-07 -4.8 -3.3551116 0.3454192 0.4479448  0.01486585  0.3161113  0.56793096
##                  As           Ba         Cr         Li         Mn         Si
## 2022-01 -0.13978173  0.004883986 -0.4010283  0.1991241 -0.2913304 -0.4321157
## 2022-02 -0.16586915 -0.176737078 -0.4879982 -2.0702399 -0.3687795 -0.3781283
## 2022-03 -0.02932271  0.245422612 -0.1517064 -0.9448572 -0.2541339 -0.2953588
## 2022-04  0.07346909  0.295846934 -0.3414390 -0.9125139 -0.3457072 -0.3639113
## 2022-05  0.39822935  0.497951676 -0.2968745  0.1991241 -0.6531851 -0.1740302
## 2022-07  0.52425854  0.159543256 -1.8114567  0.1991241 -0.0747582 -0.7129494
##                 Sr
## 2022-01 -0.0459834
## 2022-02 -0.0678548
## 2022-03  0.1211331
## 2022-04  0.1674307
## 2022-05  0.3018771
## 2022-07  0.6215123
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability

pca_result_all_intercept <- prcomp(w2_std_all_intercept, center = FALSE, scale. = FALSE)
summary(pca_result_all_intercept)  # View variance explained by each component
```

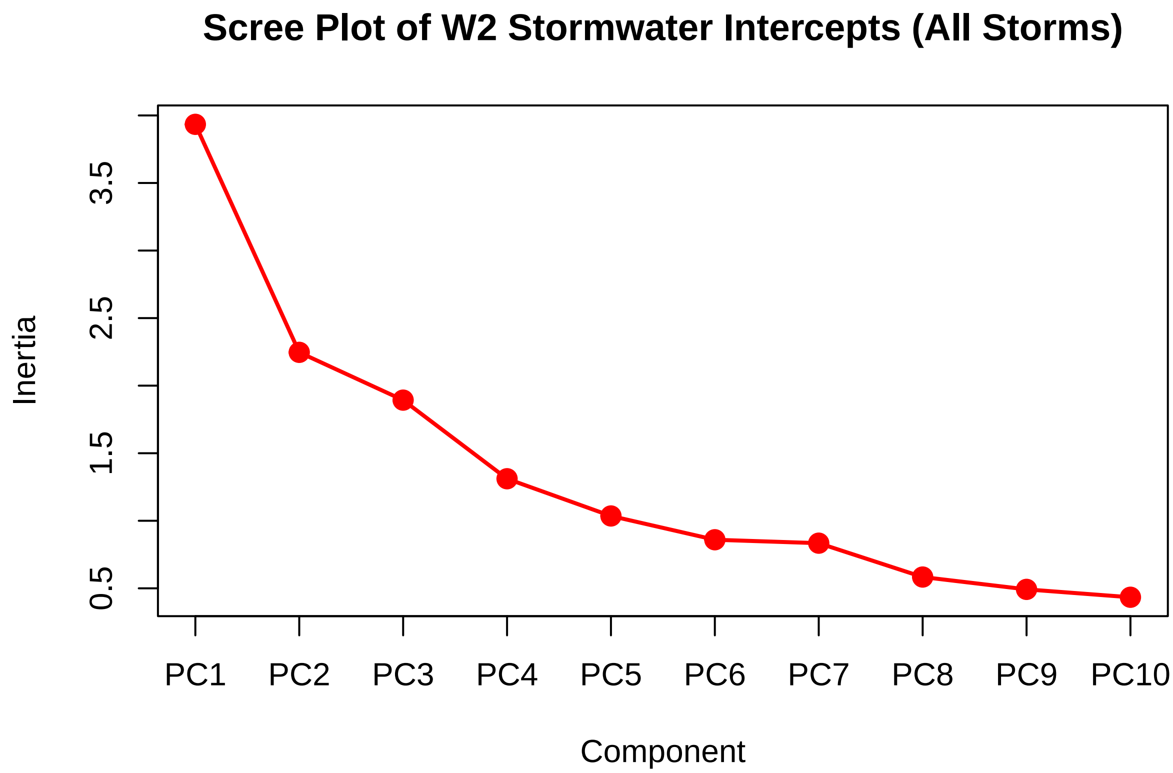
```
## Importance of components:
##                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.983 1.4988 1.3758 1.14501 1.01759 0.92705 0.91331
## Proportion of Variance 0.281 0.1605 0.1352 0.09365 0.07396 0.06139 0.05958
## Cumulative Proportion  0.281 0.4415 0.5767 0.67032 0.74429 0.80567 0.86526
##                            PC8     PC9   PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.76384 0.70148 0.6588 0.46798 0.26165 0.23577 0.18397
## Proportion of Variance 0.04168 0.03515 0.0310 0.01564 0.00489 0.00397 0.00242
## Cumulative Proportion  0.90693 0.94208 0.9731 0.98872 0.99361 0.99758 1.00000
```

```r
# View the loadings
loadings_all_intercept <- pca_result_all_intercept$rotation
print(loadings_all_intercept)
```

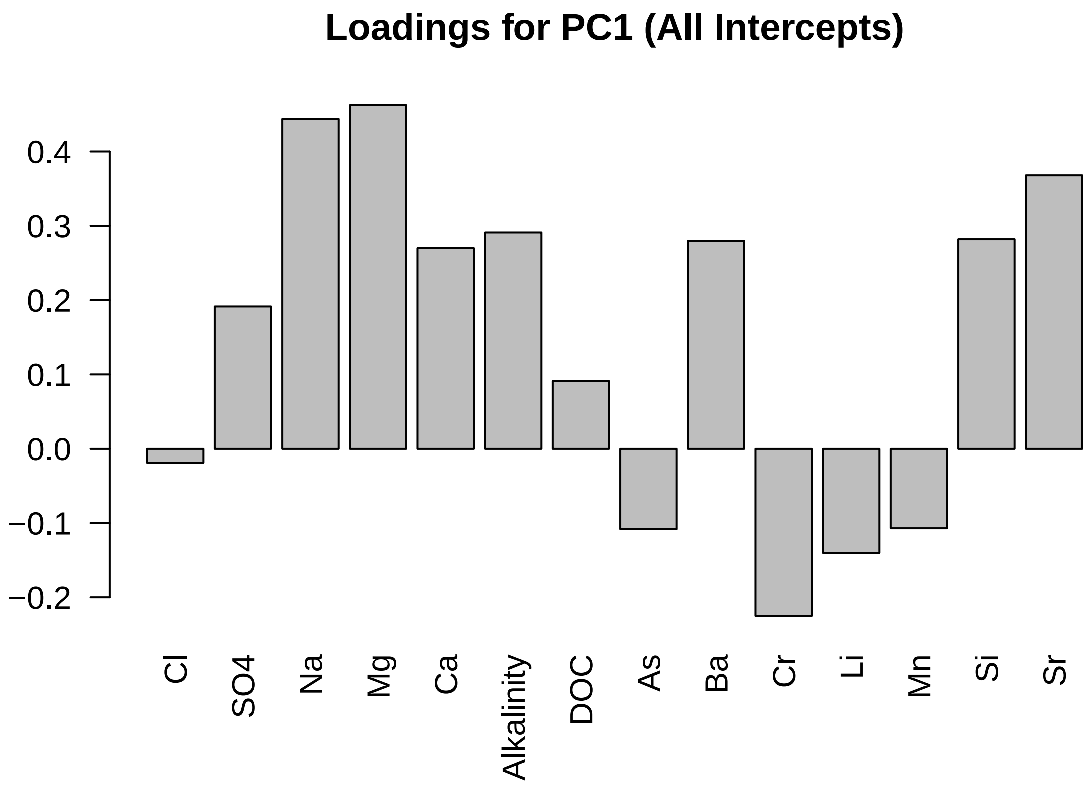
```
##                    PC1         PC2          PC3          PC4          PC5
## Cl         -0.01909922  0.02335628  0.670921828 -0.049085343 -0.131148980
## SO4         0.19144162 -0.07381274  0.602364313 -0.092381409 -0.077713263
## Na          0.44376664 -0.10641684  0.037990666 -0.063906938 -0.023594883
## Mg          0.46234367 -0.12605651 -0.031803784 -0.078734034 -0.006488776
## Ca          0.26990520 -0.48657293  0.020410072  0.078347565  0.040681696
## Alkalinity  0.29098686  0.41903699 -0.075205967  0.009807868 -0.238977538
```

```
## DOC         0.09103226 -0.08165632 -0.033389079  0.594728506 -0.458231516
## As         -0.10831252  0.44639785 -0.037847644  0.114040820  0.241302057
## Ba          0.27953444 -0.01146905  0.022989484  0.325771036  0.415417811
## Cr         -0.22500907  0.21098572  0.346164236  0.373276966 -0.067928238
## Li         -0.14024700 -0.08947359 -0.169422224 -0.010946249 -0.638214349
## Mn         -0.10710110 -0.19077040 -0.003157421  0.575478595  0.227503972
## Si          0.28188875  0.47288495  0.053395257  0.023430862 -0.009195612
## Sr          0.36794629  0.18931936 -0.154995071  0.168891739 -0.113346051
##                    PC6         PC7         PC8         PC9         PC10
## Cl         -0.05807425  0.27168429 -0.11603816  0.22106650  0.064676450
## SO4         0.04346779 -0.14439590  0.25254174  0.20609877 -0.274157298
## Na          0.27760238 -0.15831429  0.20275117 -0.31627211 -0.003814488
## Mg          0.12871895  0.09941912  0.22253515 -0.20277625 -0.125834089
## Ca          0.06311687  0.11634310  0.05322778 -0.06402423  0.515638541
## Alkalinity -0.31252354  0.14608590  0.06952668 -0.34064434 -0.187402788
## DOC        -0.11011426 -0.51519276 -0.01888368  0.13600738 -0.170344504
## As          0.51641149 -0.11555836  0.50888187  0.21887362  0.041178766
## Ba          0.35162505  0.07981456 -0.59495208  0.07502879 -0.336485358
## Cr          0.21556801 -0.06408129 -0.12056967 -0.54391824  0.410873873
## Li          0.51930155  0.46735192 -0.05343231  0.05757732 -0.150445877
## Mn         -0.24891942  0.50556605  0.41858285 -0.02232874 -0.178274321
## Si         -0.12576213  0.25790092 -0.13509761 -0.01486125  0.062974908
## Sr         -0.05554077  0.07570516 -0.01789902  0.52483746  0.482357589
##                    PC11         PC12        PC13         PC14
## Cl          0.101116758  0.559877556  0.11745997 -0.219446094
## SO4        -0.167010261 -0.578084055 -0.05842021  0.104129429
## Na          0.110334125  0.228205748 -0.62215520 -0.301756223
## Mg         -0.186062232  0.339709227  0.37534178  0.582464976
## Ca          0.310555245 -0.287925935  0.38507006 -0.253433119
## Alkalinity -0.224252236 -0.121995669  0.31235619 -0.489008671
## DOC         0.246806678  0.137281047  0.11709789  0.036882305
## As          0.153372414  0.064021013  0.26891105 -0.166737289
## Ba         -0.099367454 -0.061550664  0.08609537 -0.136369688
## Cr         -0.252219750 -0.088608538 -0.03643312  0.196227582
## Li          0.004006884 -0.130315841 -0.04764135 -0.004470572
## Mn         -0.029676866  0.008153675 -0.20774767  0.006021942
## Si          0.638845319 -0.188055001 -0.16050690  0.345566772
## Sr         -0.448788519  0.009660204 -0.20743735  0.039801432
```

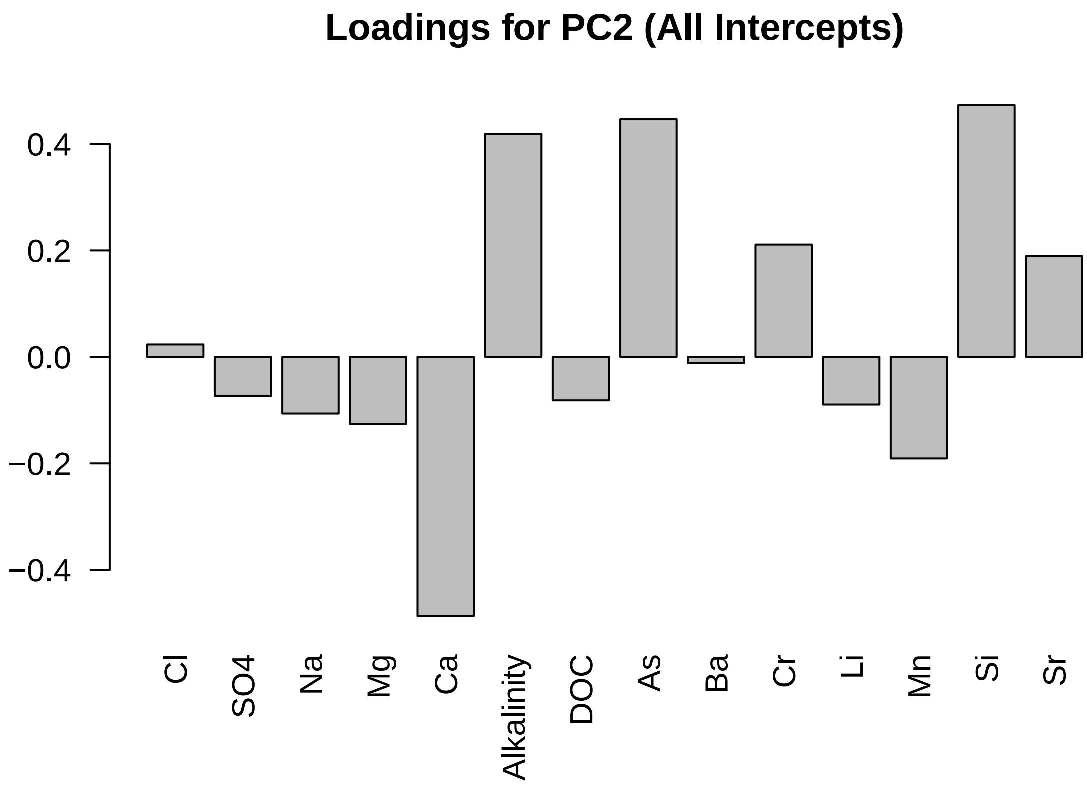
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_all_intercept, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Intercepts (All Storms)")
```



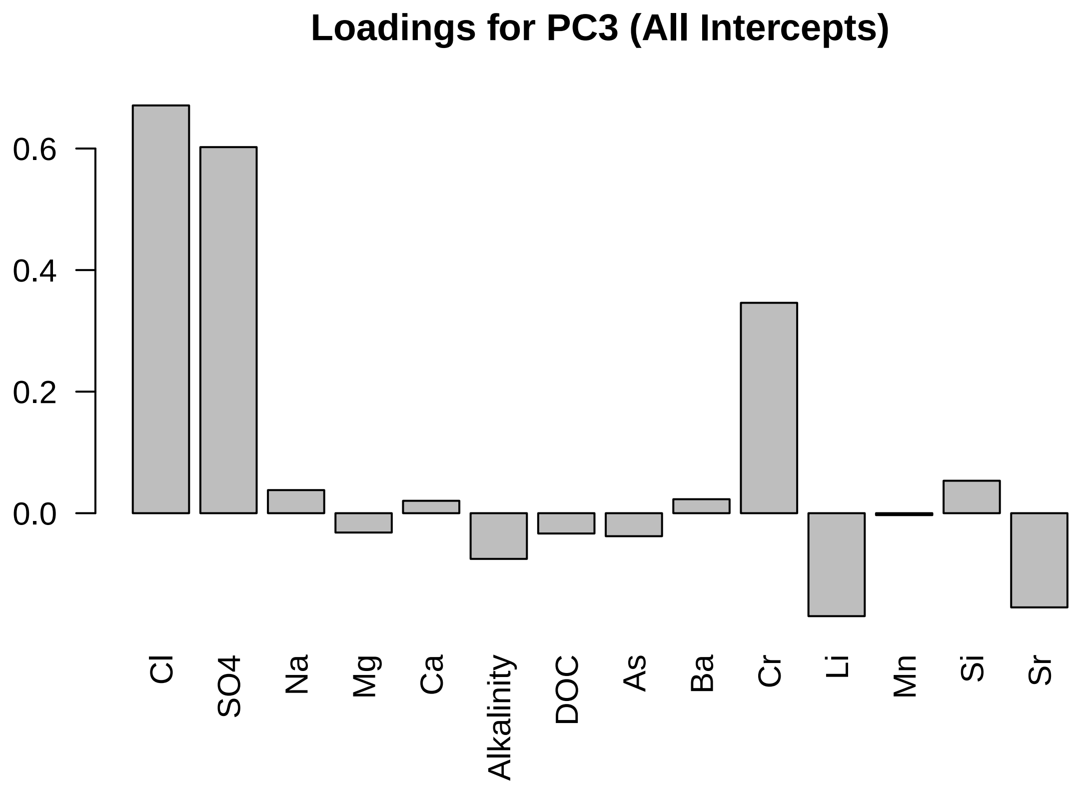

```r
# looks like we should use the first 2!

# Visualize the loadings for the first 2 principal components
barplot(loadings_all_intercept[, 1], main = "Loadings for PC1 (All Intercepts)", las = 2)
```



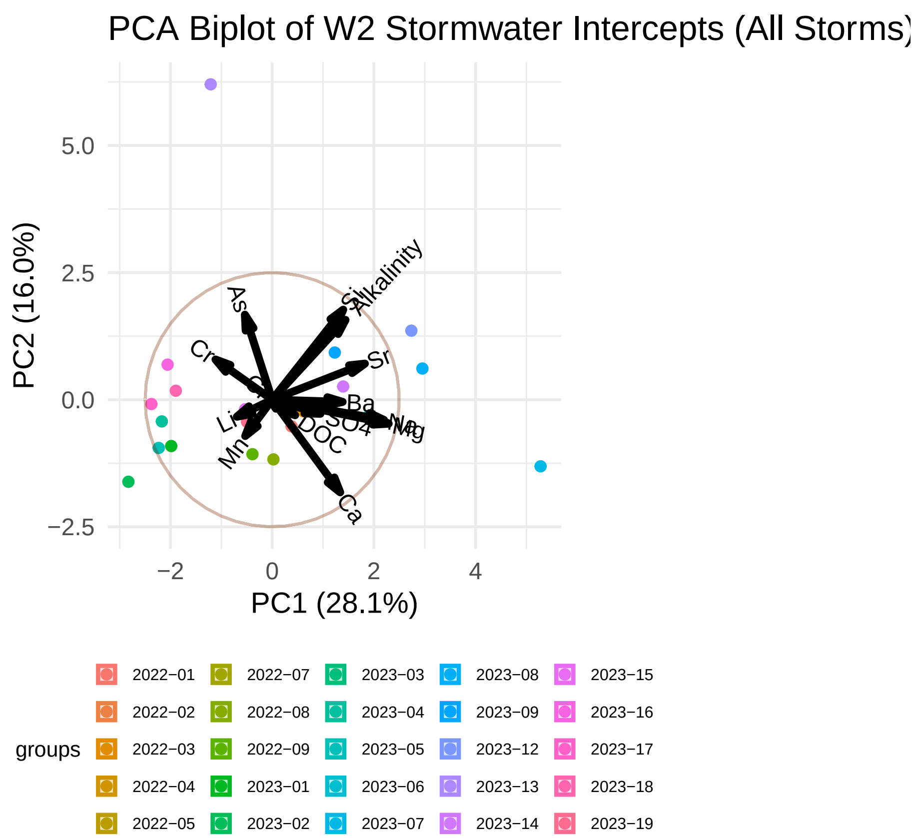

```
barplot(loadings_all_intercept[, 2], main = "Loadings for PC2 (All Intercepts)", las = 2)
```



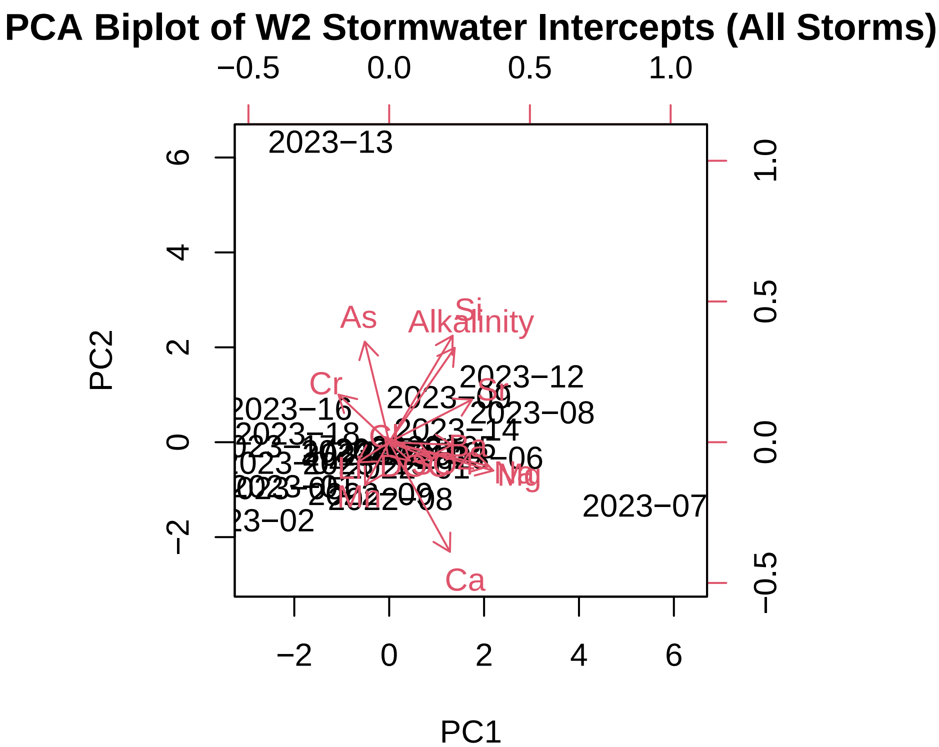

```
barplot(loadings_all_intercept[, 3], main = "Loadings for PC3 (All Intercepts)", las = 2)
```




```r
# Generate the biplot
biplot_all_intercept <- ggbiplot(pca_result_all_intercept,
                                 obs.scale = 1,
                                 var.scale = 1,
                                 groups = rownames(w2_std_all_intercept),  # Use Rain_Event_Number as group
                                 ellipse = TRUE,   # Add confidence ellipses
                                 circle = TRUE)    # Add a unit circle

# Customize the biplot
biplot_all_intercept <- biplot_all_intercept +
  ggtitle("PCA Biplot of W2 Stormwater Intercepts (All Storms)") +
  theme_minimal() +
  theme(
    legend.position = "bottom",           # Position legend at the bottom
    legend.title = element_text(size = 8),   # Reduce legend title size
    legend.text = element_text(size = 6),    # Reduce legend label size
    legend.key.size = unit(0.3, "cm"),       # Reduce legend key size (symbols)
    legend.spacing.x = unit(0.1, "cm"),      # Reduce horizontal spacing between legend items
    legend.spacing.y = unit(0.08, "cm")       # Reduce vertical spacing between legend items (if vertica
  )

# Print the biplot
print(biplot_all_intercept)
```




```r
# Simple base R biplot
biplot(pca_result_all_intercept, scale = 0,
       main = "PCA Biplot of W2 Stormwater Intercepts (All Storms)")
```




```r
# Function to get top 5 variables for each PC
get_top_variables <- function(loadings, n = 5) {
    top_vars <- lapply(1:3, function(pc) {
        # Extract loadings for the PC and get top variables
        # by absolute value
        abs_loadings <- abs(loadings[, pc])
        top_indices <- order(abs_loadings, decreasing = TRUE)[1:n]
        data.frame(PC = pc, Variable = rownames(loadings)[top_indices],
            Loading = loadings[top_indices, pc])
    })
    do.call(rbind, top_vars)
}

# Apply function to each matrix
top_vars_pre_slope <- get_top_variables(loadings_pre_slope)
top_vars_pre_intercept <- get_top_variables(loadings_pre_intercept)
top_vars_post_slope <- get_top_variables(loadings_post_slope)
top_vars_post_intercept <- get_top_variables(loadings_post_intercept)
top_vars_all_slope <- get_top_variables(loadings_all_slope)
top_vars_all_intercept <- get_top_variables(loadings_all_intercept)

# Results for each subset
top_vars_pre_slope
```

```
##              PC    Variable    Loading
## Mg            1          Mg  0.3653817
```

```
## Alkalinity  1 Alkalinity  0.3633730
## Li          1         Li  0.3537445
## Cr          1         Cr -0.3524766
## SO4         1        SO4 -0.3063448
## As          2         As -0.4951192
## Na          2         Na  0.4274977
## Ca          2         Ca  0.4274977
## Cl          2         Cl  0.3575352
## SO41        2        SO4  0.3575352
## Mn          3         Mn -0.5217825
## Ba          3         Ba -0.5061367
## Si          3         Si  0.4361651
## Sr          3         Sr  0.4201046
## Na1         3         Na -0.1733556
```

```
top_vars_pre_intercept
```

```
##            PC   Variable   Loading
## Sr          1         Sr  0.3973705
## Mg          1         Mg  0.3965250
## Na          1         Na  0.3954658
## Alkalinity  1 Alkalinity  0.3692096
## DOC         1        DOC  0.3663779
## Ca          2         Ca -0.5416139
## Si          2         Si -0.4624090
## Cr          2         Cr  0.3165979
## Mn          2         Mn  0.2917298
## Ba          2         Ba  0.2741004
## SO4         3        SO4  0.5799933
## Cl          3         Cl  0.5358185
## Mn1         3         Mn -0.3330233
## As          3         As -0.2535005
## Ba1         3         Ba -0.2329561
```

```
top_vars_post_slope
```

```
##            PC   Variable     Loading
## Sr          1         Sr -0.37046587
## Si          1         Si -0.36748829
## Alkalinity  1 Alkalinity -0.36748227
## Ba          1         Ba -0.36188891
## Li          1         Li -0.31434481
## Ca          2         Ca -0.54122335
## Mg          2         Mg -0.44029128
## Na          2         Na -0.43083032
## DOC         2        DOC -0.30333907
## Li1         2         Li  0.28246577
## Mn          3         Mn -0.88306869
## DOC1        3        DOC -0.39933448
## As          3         As  0.19279400
## Ca1         3         Ca -0.11612227
## Mg1         3         Mg -0.07305052
```

```
top_vars_post_intercept
```

```
##            PC   Variable    Loading
## Mg          1         Mg  0.4186336
## Na          1         Na  0.3960562
## Ba          1         Ba  0.3357368
## SO4         1        SO4  0.3240698
## Sr          1         Sr  0.3112715
## Si          2         Si -0.4781362
## Ca          2         Ca  0.4754078
## Alkalinity  2 Alkalinity -0.4467937
## Mn          2         Mn  0.3662871
## As          2         As -0.3598628
## Sr1         3         Sr  0.5023597
## Ba1         3         Ba  0.4343653
## SO41        3        SO4 -0.4028707
## Li          3         Li  0.3543127
## Mn1         3         Mn  0.3175747
```

```
top_vars_all_slope
```

```
##            PC   Variable    Loading
## Alkalinity  1 Alkalinity -0.3822655
## Sr          1         Sr -0.3730726
## Si          1         Si -0.3667819
## Mg          1         Mg -0.3319653
## Na          1         Na -0.3284001
## Cl          2         Cl -0.5240668
## SO4         2        SO4 -0.5240668
## Cr          2         Cr -0.5209963
## As          2         As  0.2640952
## Li          2         Li  0.2085147
## Ca          3         Ca  0.5124442
## Mg1         3         Mg  0.4110082
## DOC         3        DOC  0.3821430
## Na1         3         Na  0.3810778
## As1         3         As  0.2793500
```

```
top_vars_all_intercept
```

```
##             PC   Variable    Loading
## Mg           1         Mg  0.4623437
## Na           1         Na  0.4437666
## Sr           1         Sr  0.3679463
## Alkalinity   1 Alkalinity  0.2909869
## Si           1         Si  0.2818888
## Ca           2         Ca -0.4865729
## Si1          2         Si  0.4728849
## As           2         As  0.4463978
## Alkalinity1  2 Alkalinity  0.4190370
## Cr           2         Cr  0.2109857
## Cl           3         Cl  0.6709218
```

```
## SO4          3       SO4  0.6023643
## Cr1          3        Cr  0.3461642
## Li           3        Li -0.1694222
## Sr1          3        Sr -0.1549951
```

```r
# let's do some PCAs without standardizing the data for funsies

# We will first start by thinking just about pre-basalt application storm slopes
w2_pre_slope <- W2_Stormwater_slope %>%
  filter(Treatment == "Pre")

w2_pre_slope <- w2_pre_slope[, -c(1:2)]

# let's also clean up the names while we're at it:
# Rename columns of w2_std_pre_intercept
colnames(w2_pre_slope) <- gsub("slope_|_ppm|_uM|_ugL", "", colnames(w2_pre_slope))

# Check the new column names
colnames(w2_pre_slope)
```

```
##  [1] "Cl"         "SO4"        "Na"         "Mg"         "Ca"
##  [6] "Alkalinity" "DOC"        "As"         "Ba"         "Cr"
## [11] "Li"         "Mn"         "Si"         "Sr"
```

```r
# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_pre_slope) <- c("2022-01", "2022-02", "2022-03", "2022-04", "2022-05", "2022-07", "2022-08"
                          "2022-09", "2023-01", "2023-02", "2023-03", "2023-04")

# Viewing the first few rows of the standardized data
head(w2_pre_slope) # looks good to me!
```

```
##                Cl      SO4 Na         Mg Ca  Alkalinity       DOC         As
## 2022-01 0.0000000 0.000000  0  0.0000000  0  0.00000000 0.3945345  0.0000000
## 2022-02 0.0000000 0.000000  0  0.0000000  0  0.00000000 0.0000000  0.0000000
## 2022-03 0.0000000 0.000000  0  0.0000000  0  0.00000000 0.2327978  0.0000000
## 2022-04 0.0000000 0.000000  0  0.0000000  0 -0.04944217 0.3345227  0.0000000
## 2022-05 0.0000000 0.000000  0  0.0000000  0 -0.05680430 0.1649367  0.0000000
## 2022-07 0.9721309 1.622544  0 -0.1178565  0 -0.17968778 0.3559922 -0.1660514
##                  Ba       Cr          Li         Mn          Si          Sr
## 2022-01 -0.10652251 0.000000  0.00000000  0.0000000  0.00000000  0.00000000
## 2022-02  0.00000000 0.000000  0.00000000  0.0000000  0.00000000  0.00000000
## 2022-03 -0.06047943 0.000000 -0.07277237 -0.7664065 -0.04581096 -0.05092746
## 2022-04 -0.04489661 0.000000 -0.07302621 -1.2391537 -0.04326868 -0.06085146
## 2022-05 -0.05872068 0.000000 -0.13498278 -0.5625071 -0.03948234 -0.08986106
## 2022-07 -0.09757584 0.639587 -0.24505078 -1.1096388 -0.09682057 -0.16424148
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
pca_result_pre_slope_nonstd <- prcomp(w2_pre_slope, center = FALSE, scale. = FALSE)
summary(pca_result_pre_slope_nonstd)  # View variance explained by each component
```

```
## Importance of components:
```

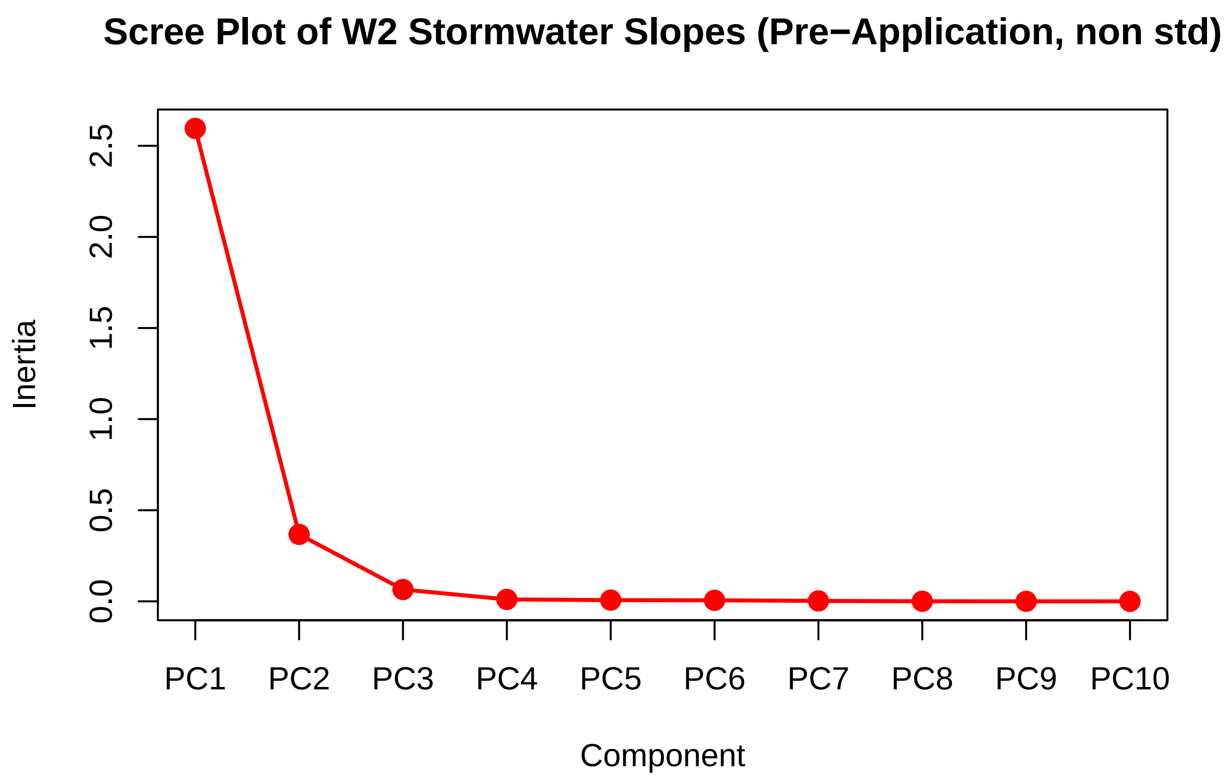
```
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.6110 0.6060 0.25444 0.10268 0.08101 0.07446 0.04846
## Proportion of Variance 0.8501 0.1203 0.02121 0.00345 0.00215 0.00182 0.00077
## Cumulative Proportion  0.8501 0.9704 0.99163 0.99508 0.99723 0.99904 0.99981
##                            PC8      PC9     PC10      PC11 PC12
## Standard deviation     0.02175 0.008894 0.003925 2.384e-17    0
## Proportion of Variance 0.00016 0.000030 0.000010 0.000e+00    0
## Cumulative Proportion  0.99997 0.999990 1.000000 1.000e+00    1
```

```r
# View the loadings
loadings_pre_slope_nonstd <- pca_result_pre_slope_nonstd$rotation
print(loadings_pre_slope_nonstd)
```

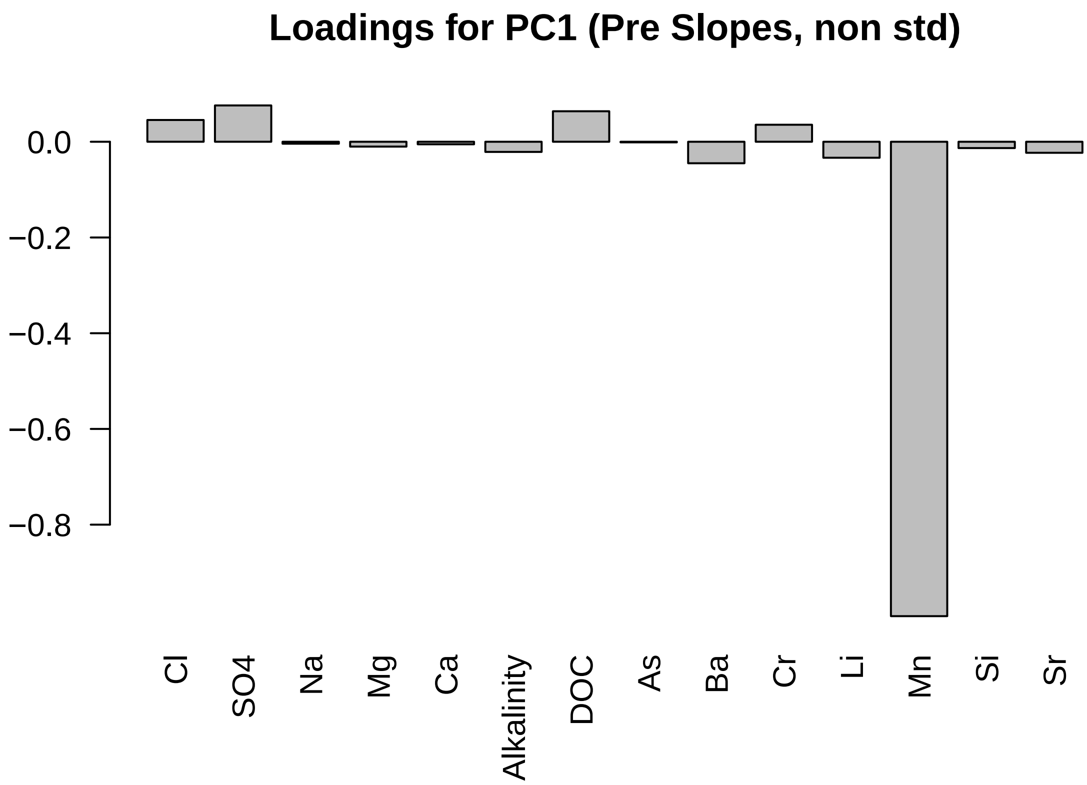
```
##                     PC1          PC2         PC3          PC4         PC5
## Cl          0.045474877 -0.464805943 -0.13341683  0.048228601 -0.01376097
## SO4         0.075900284 -0.775788856 -0.22268065  0.080496412 -0.02296788
## Na         -0.004056557  0.002044161 -0.07421274  0.165222290 -0.08683689
## Mg         -0.010129272  0.058676978 -0.06827478  0.182165939 -0.09714680
## Ca         -0.005304229  0.002672883 -0.09703831  0.216039622 -0.11354527
## Alkalinity -0.021228113  0.088848650 -0.19436342  0.220927469 -0.03808088
## DOC         0.063747849 -0.179216364  0.79967119  0.477163958 -0.25621859
## As         -0.001211809  0.076981460  0.15298656 -0.164197829  0.05446345
## Ba         -0.044913929  0.027148230 -0.16105220 -0.225302198 -0.93987801
## Cr          0.035523920 -0.308630840  0.01476211 -0.196557695  0.11092928
## Li         -0.033445629  0.121046760 -0.32243348  0.316140091 -0.03869552
## Mn         -0.991162107 -0.113818424  0.05998728  0.006547976  0.03081507
## Si         -0.013225517  0.048213314 -0.13185655  0.310449636  0.02605386
## Sr         -0.023087746  0.082416900 -0.24752400  0.542973889  0.02337427
##                     PC6          PC7          PC8           PC9          PC10
## Cl          0.043385362 -0.055205910  0.046488235  0.0389510067  0.0015417445
## SO4         0.072412758 -0.092141959  0.077591639  0.0650115546  0.0025732635
## Na          0.248344357 -0.271379020 -0.003031920  0.0036037103  0.0336172631
## Mg          0.277340980 -0.302119963 -0.009086152 -0.0006214298  0.0380674926
## Ca          0.324727499 -0.354846921 -0.003964446  0.0047121015  0.0439569069
## Alkalinity  0.205474800  0.365552144 -0.269245661  0.7544704391 -0.2933460650
## DOC         0.019778900  0.166707352  0.049665094  0.0004271254 -0.0189984139
## As         -0.152963471 -0.298542827  0.736743129  0.5341619916  0.0097565241
## Ba         -0.188189504  0.034775192  0.018377351  0.0090935977 -0.0114510101
## Cr         -0.314594310  0.338644401  0.034774886  0.0206474913 -0.0454347766
## Li          0.254725888  0.522380643  0.598279210 -0.2851063733  0.0428176649
## Mn          0.006307094 -0.002427008 -0.002362782 -0.0004615576  0.0008127121
## Si         -0.374516657 -0.228384871  0.076220760 -0.2018003387 -0.8025592036
## Sr         -0.585876814 -0.078350711 -0.093061315  0.1309489772  0.5107257200
##                     PC11          PC12
## Cl          5.915806e-01 -4.080890e-01
## SO4        -4.993479e-01  2.382247e-01
## Na          4.504152e-01  7.178788e-01
## Mg         -2.289946e-01 -4.537035e-01
## Ca          1.986859e-01 -2.256882e-01
## Alkalinity  3.813311e-15 -8.432023e-16
## DOC         2.662464e-16 -2.180495e-16
## As          4.546770e-16  2.313655e-16
## Ba          3.367164e-16 -1.694819e-16
## Cr          3.254161e-01 -6.767817e-02
```

```
## Li         -4.953685e-16 -2.055358e-16
## Mn         -6.048905e-17 -1.192809e-17
## Si          5.413108e-15 -3.959827e-15
## Sr         -3.506797e-15  2.481186e-15
```

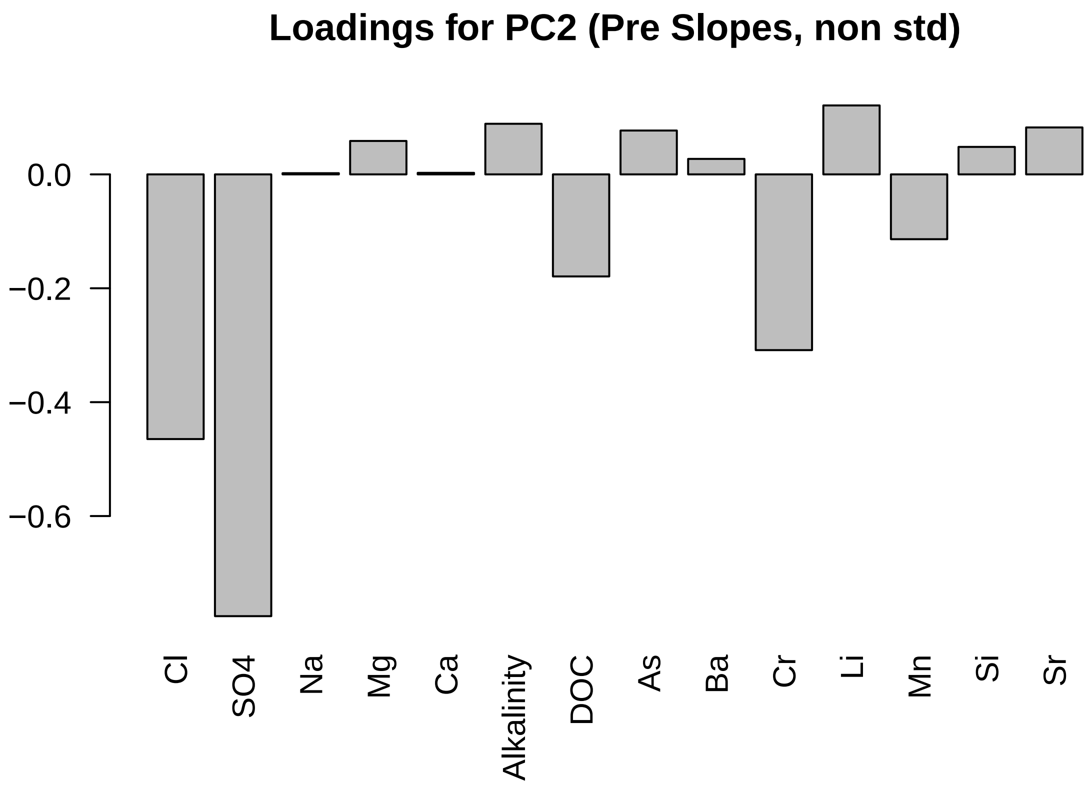
```r
# Screeplot to see how many principal components we should use
screeplot(pca_result_pre_slope_nonstd, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Slopes (Pre-Application, non std)")
```



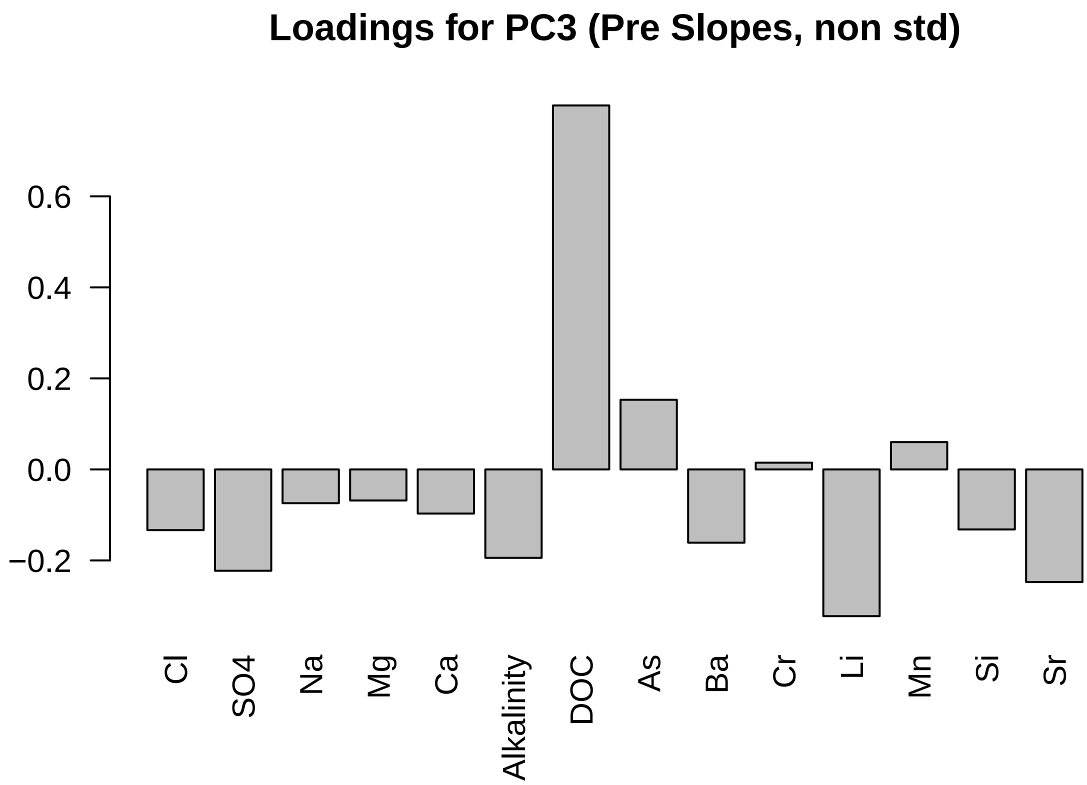

```r
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_pre_slope_nonstd[, 1], main = "Loadings for PC1 (Pre Slopes, non std)", las = 2)
```



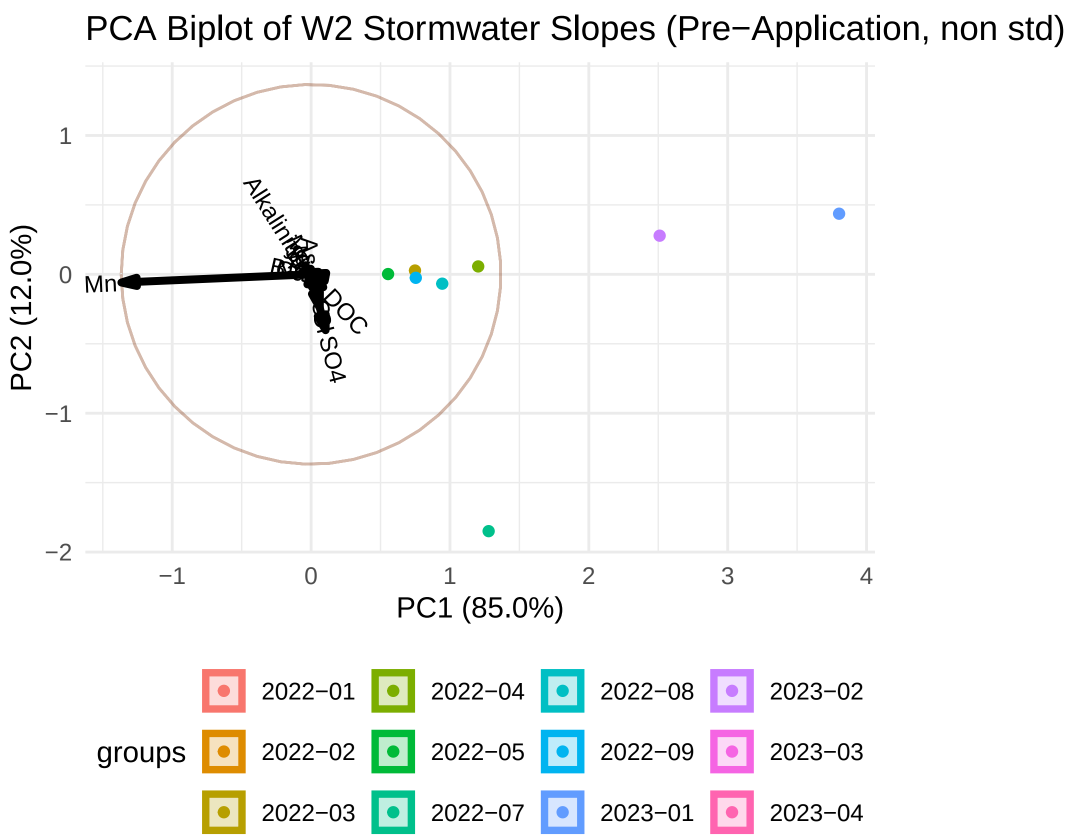

```
barplot(loadings_pre_slope_nonstd[, 2], main = "Loadings for PC2 (Pre Slopes, non std)", las = 2)
```




```r
barplot(loadings_pre_slope_nonstd[, 3], main = "Loadings for PC3 (Pre Slopes, non std)", las = 2)
```

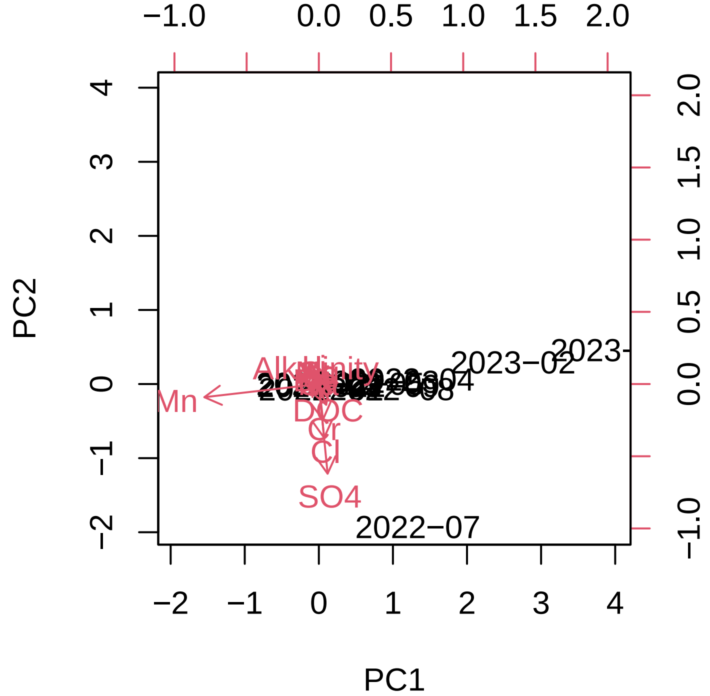



```r
# Generate the biplot
biplot_pre_slope_nonstd <- ggbiplot(pca_result_pre_slope_nonstd,
                             obs.scale = 1,
                             var.scale = 1,
                             groups = rownames(w2_pre_slope),  # Use Rain_Event_Number as group labels
                             ellipse = TRUE,  # Add confidence ellipses
                             circle = TRUE)   # Add a unit circle

# Customize the biplot
biplot_pre_slope_nonstd <- biplot_pre_slope_nonstd +
  ggtitle("PCA Biplot of W2 Stormwater Slopes (Pre-Application, non std)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Print the biplot
print(biplot_pre_slope_nonstd)
```




```r
# Simple base R biplot
biplot(pca_result_pre_slope_nonstd, scale = 0,
       main = "PCA Biplot of W2 Stormwater Slopes (Pre-Application, non std)")
```

PCA Biplot of W2 Stormwater Slopes (Pre−Application, non std)




```r
# Now Intercepts
w2_pre_intercept <- W2_Stormwater_intercept %>%
  filter(Treatment == "Pre")

w2_pre_intercept <- w2_std_pre_intercept[, -c(1:2)]

# Set Rain_Event_Number as row names - there isn't an elegant way to do this, so we'll manually set it
rownames(w2_pre_intercept) <- c("2022-01", "2022-02", "2022-03", "2022-04", "2022-05", "2022-07", "2022-
                             "2022-09", "2023-01", "2023-02", "2023-03", "2023-04")

# let's also clean up the names while we're at it:
# Rename columns of w2_std_pre_intercept
colnames(w2_pre_intercept) <- gsub("intercept_|_ppm|_uM|_ugL", "", colnames(w2_pre_intercept))

# Check the new column names
colnames(w2_pre_intercept)
```

```
##  [1] "Na"         "Mg"         "Ca"         "Alkalinity" "DOC"
##  [6] "As"         "Ba"         "Cr"         "Li"         "Mn"
## [11] "Si"         "Sr"
```

```r
# Viewing the first few rows of the standardized data
head(w2_pre_intercept) # looks good to me!
```

```
##                  Na         Mg          Ca Alkalinity         DOC           As
```

```
## 2022-01 0.9351360 0.5852743 -0.6169582  0.2803558 -0.1929098 0.12642896
## 2022-02 0.3014065 0.5143555 -0.7784013  0.1540760 -0.9775343 0.06715948
## 2022-03 0.6100230 0.6536671 -0.4159970  0.4020312  0.2875371 0.37738696
## 2022-04 0.7313665 0.7173084 -0.2168643  0.5496577  0.5500840 0.61092541
## 2022-05 1.0400895 0.7708044 -0.2349155  0.5296932  1.8568488 1.34876636
## 2022-07 0.6859073 0.7942569  0.6214634  1.3448539  1.0985922 1.63509912
##                  Ba          Cr         Li          Mn         Si        Sr
## 2022-01  0.08299012 -0.04708965  0.1140722 -0.06089418  0.1529329 0.3182347
## 2022-02 -0.12274372 -0.13898788 -2.0245711 -0.15577320  0.2547796 0.2968901
## 2022-03  0.35546365  0.21636037 -0.9640131 -0.01532663  0.4109237 0.4813263
## 2022-04  0.41258252  0.01587638 -0.9335329 -0.12750849  0.2815999 0.5265089
## 2022-05  0.64151953  0.06296612  0.1140722 -0.50418433  0.6398090 0.6577172
## 2022-07  0.25818260 -1.53744197  0.1140722  0.20441754 -0.3768575 0.9696541
```

```r
# Step 1: PCA using prcomp()
# Reduce dimensionality & ID combinations of variables that drive the greatest variability
pca_result_pre_intercept_nonstd <- prcomp(w2_pre_intercept, center = FALSE, scale. = FALSE)
summary(pca_result_pre_intercept_nonstd)  # View variance explained by each component
```

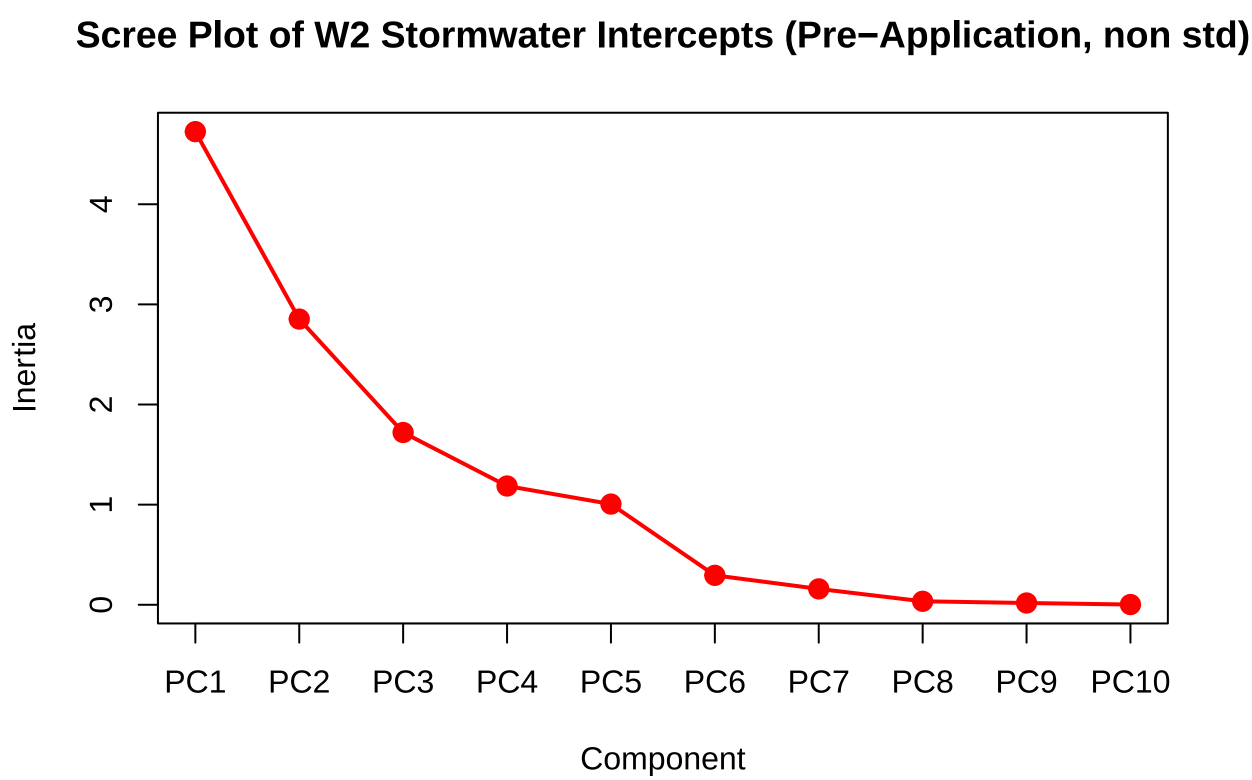
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5    PC6     PC7
## Standard deviation     2.1737 1.6892 1.3117 1.08910 1.00302 0.5422 0.39849
## Proportion of Variance 0.3938 0.2378 0.1434 0.09884 0.08384 0.0245 0.01323
## Cumulative Proportion  0.3938 0.6315 0.7749 0.87376 0.95760 0.9821 0.99534
##                            PC8     PC9    PC10    PC11     PC12
## Standard deviation     0.18698 0.13342 0.05033 0.02606 4.17e-15
## Proportion of Variance 0.00291 0.00148 0.00021 0.00006 0.00e+00
## Cumulative Proportion  0.99825 0.99973 0.99994 1.00000 1.00e+00
```

```r
# View the loadings
loadings_pre_intercept_nonstd <- pca_result_pre_intercept_nonstd$rotation
print(loadings_pre_intercept_nonstd)
```

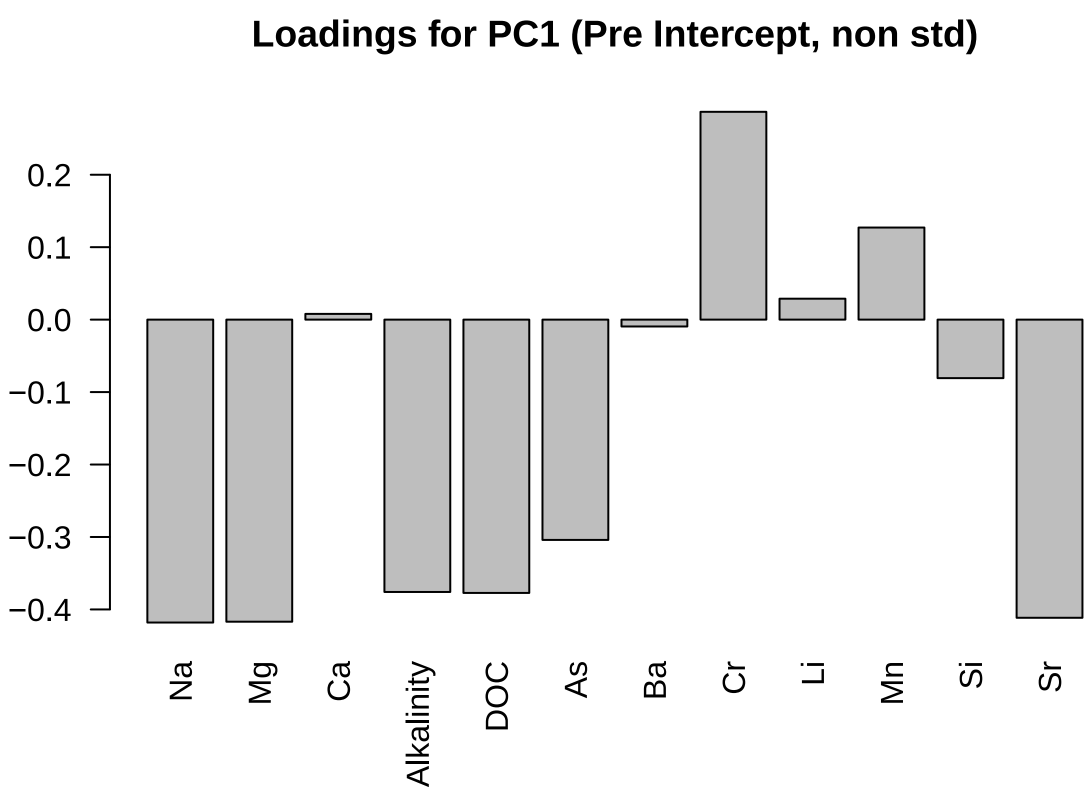
```
##                     PC1         PC2         PC3         PC4         PC5
## Na         -0.418069274 -0.19050276  0.03053481  0.05781612  0.18146001
## Mg         -0.416957071 -0.19269276  0.06001136  0.06646442  0.15873701
## Ca          0.007905703  0.53267543 -0.08037690 -0.35896549 -0.06073986
## Alkalinity -0.375915447 -0.06483593  0.29799406  0.27697188 -0.18997758
## DOC        -0.377239506 -0.15842689  0.06901941 -0.32153265  0.07021764
## As         -0.304053256 -0.07249599 -0.51098255 -0.15691186 -0.20184603
## Ba         -0.009450049 -0.29613193 -0.57333316 -0.35417727  0.14160063
## Cr          0.286847053 -0.29638381  0.04102831  0.12539741  0.52579597
## Li          0.028887666 -0.03207679  0.49150574 -0.67709120 -0.08856162
## Mn          0.127103413 -0.34214878 -0.04939741  0.11215924 -0.74167489
## Si         -0.080770391  0.50172529 -0.23820018  0.20483750  0.02837276
## Sr         -0.411518852  0.25083098  0.05645306  0.06714845  0.03434711
##                    PC6         PC7          PC8         PC9        PC10
## Na          0.20000999 -0.32829850  0.247248046 -0.09037872  0.44186269
## Mg          0.33225299 -0.05685798 -0.256127026 -0.13055969 -0.27825256
## Ca         -0.01751457  0.32922010 -0.036993284 -0.52646805  0.22227167
## Alkalinity -0.24732281  0.28041406  0.433025403 -0.39838155 -0.36301404
## DOC        -0.61147658  0.06060667 -0.515273863  0.13487901 -0.06710309
## As         -0.22859673  0.15654759  0.453179542  0.34779952  0.19884111
```

```
## Ba          0.18978077 -0.08092662  0.002942063 -0.40750990 -0.30202557
## Cr         -0.44293114 -0.04498445  0.124665502 -0.33935608  0.31070708
## Li          0.02095385 -0.37500050  0.306456299  0.01272585 -0.05142793
## Mn         -0.09700306 -0.25808662 -0.230636221 -0.31498710  0.24146493
## Si         -0.31842133 -0.67594929  0.041977477 -0.08425940 -0.26251738
## Sr          0.13609926 -0.03278399 -0.221558604 -0.10975971  0.42951663
##                   PC11        PC12
## Na          0.575694732  0.08929219
## Mg         -0.166690942 -0.67403105
## Ca          0.242304210 -0.28608149
## Alkalinity -0.030893831  0.18473066
## DOC         0.204038065  0.11001454
## As         -0.216717040 -0.29364222
## Ba         -0.085136458  0.36181057
## Cr         -0.254064268 -0.21448309
## Li         -0.222818413 -0.05318183
## Mn         -0.026478899 -0.11934325
## Si         -0.001941932 -0.07851527
## Sr         -0.608585509  0.35109316
```

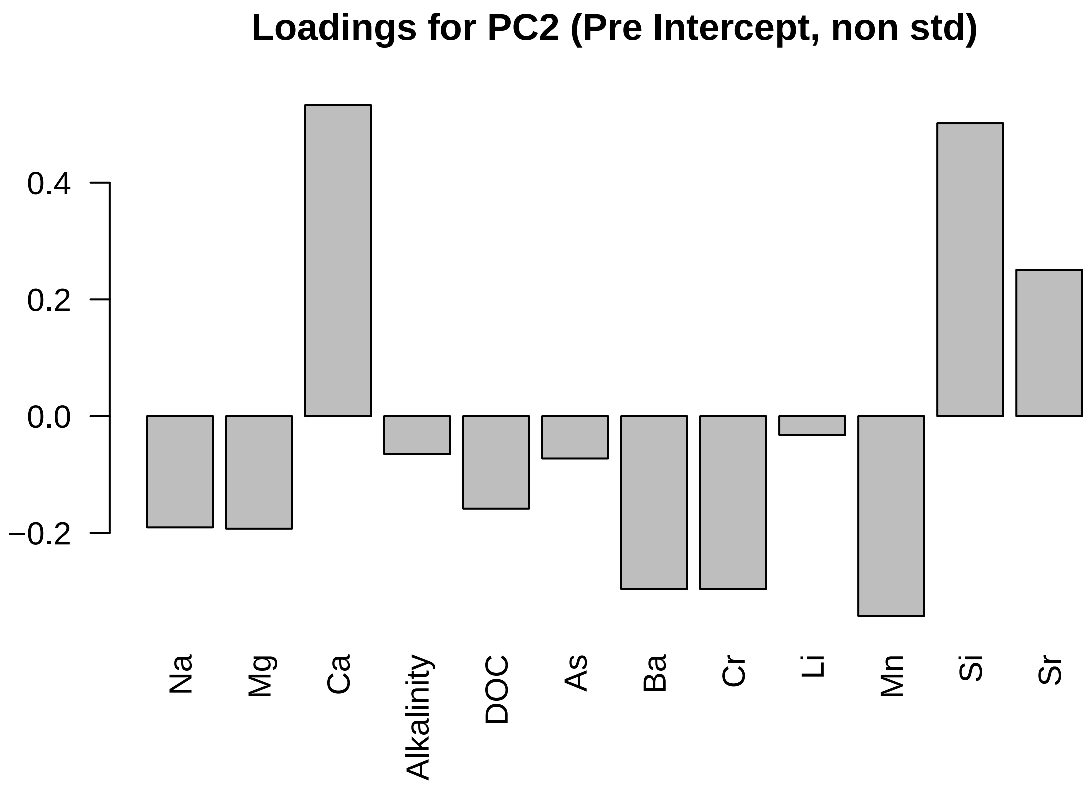
```
# Screeplot to see how many principal components we should use
screeplot(pca_result_pre_intercept_nonstd, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of W2 Stormwater Intercepts (Pre-Application, non std)")
```



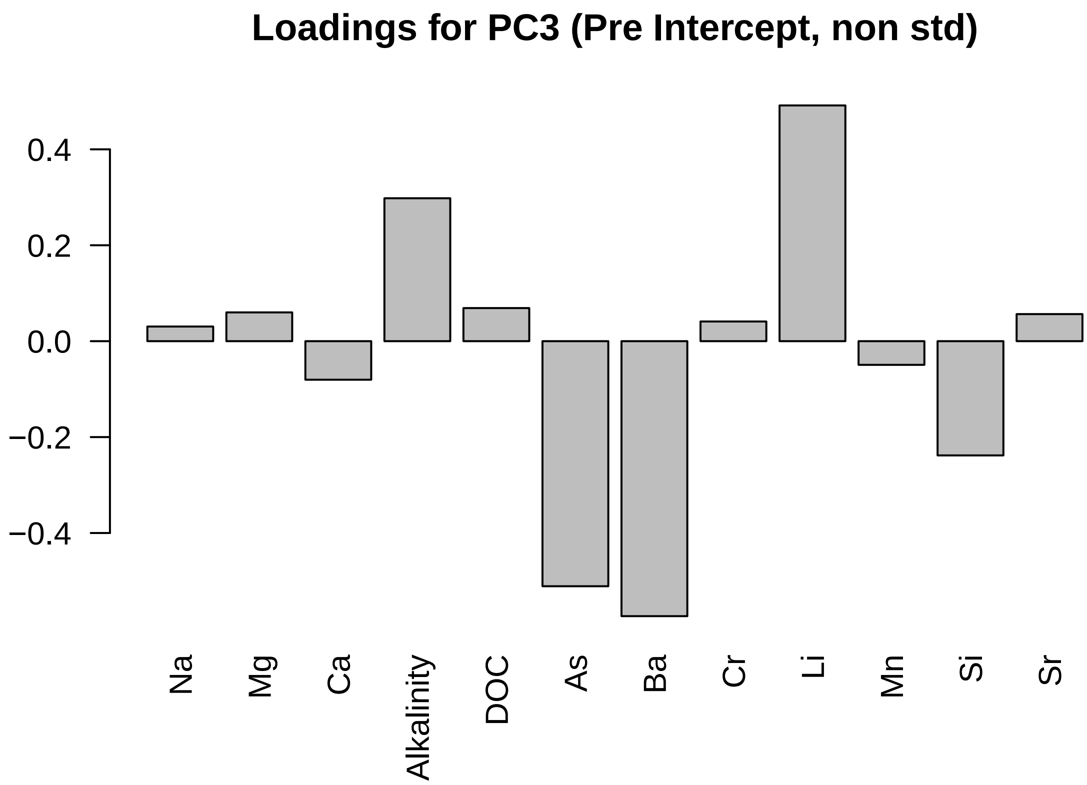
```r
# looks like we should use the first 3!

# Visualize the loadings for the first 2 principal components
barplot(loadings_pre_intercept_nonstd[, 1], main = "Loadings for PC1 (Pre Intercept, non std)", las = 2)
```



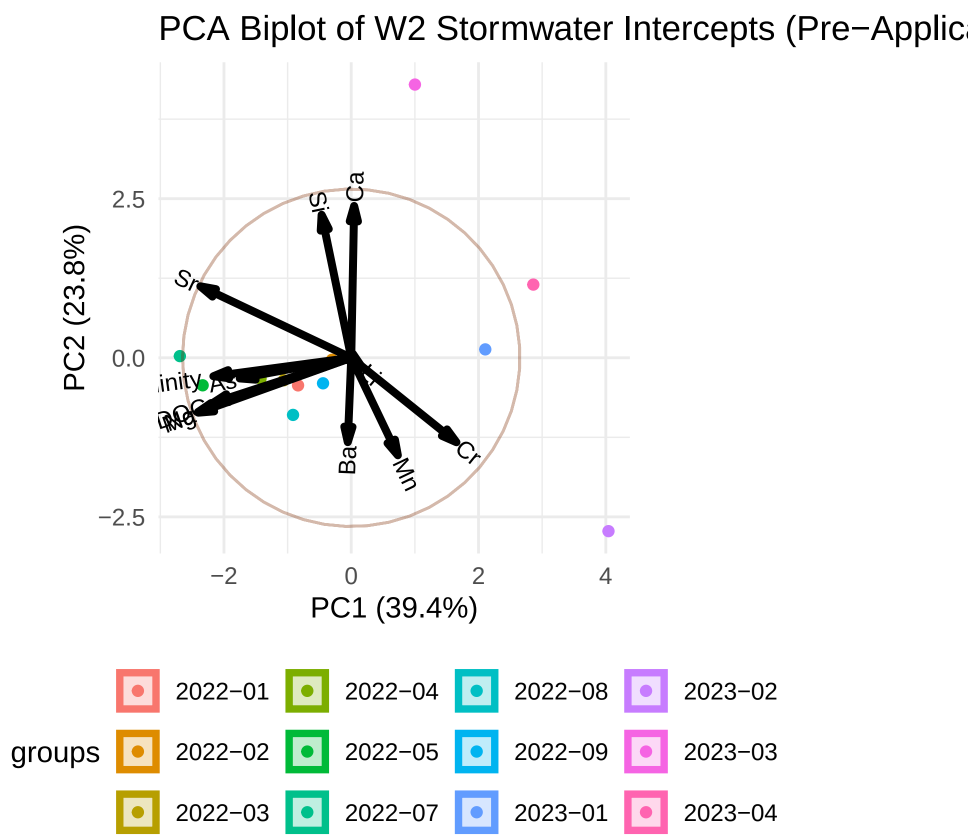

```r
barplot(loadings_pre_intercept_nonstd[, 2], main = "Loadings for PC2 (Pre Intercept, non std)", las = 2)
```



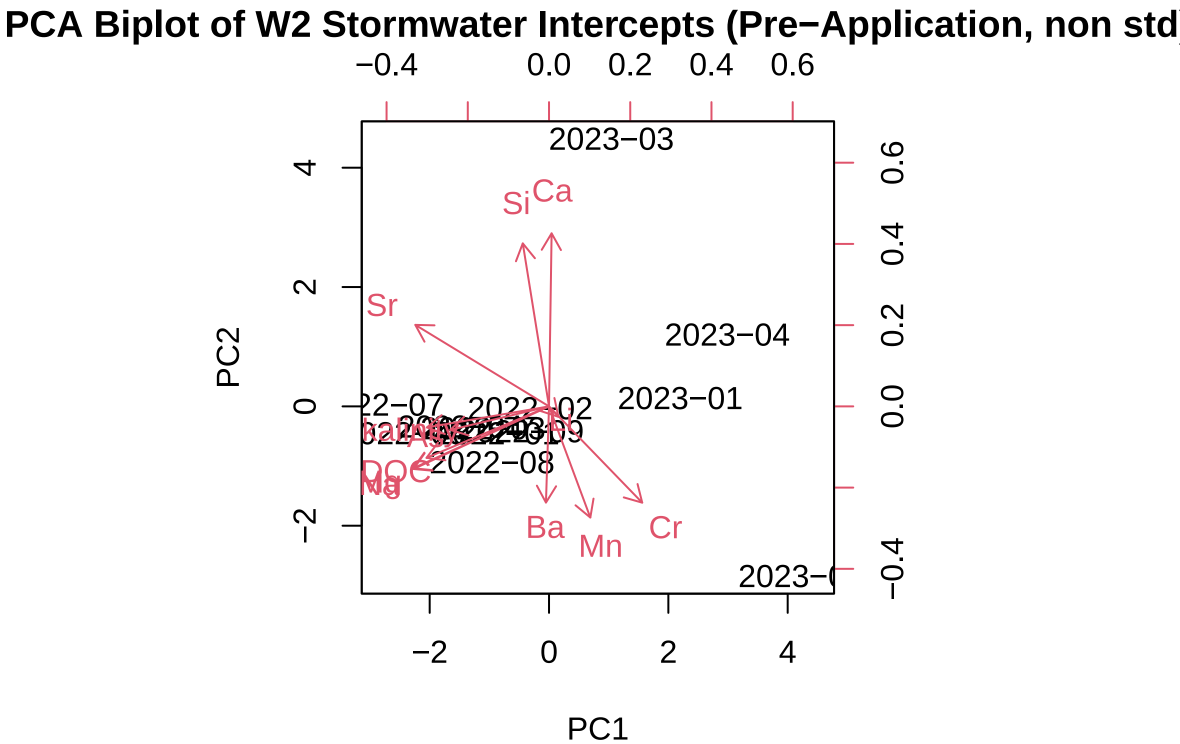

```
barplot(loadings_pre_intercept_nonstd[, 3], main = "Loadings for PC3 (Pre Intercept, non std)", las = 2)
```




```
# Generate the biplot
biplot_pre_intercept_nonstd <- ggbiplot(pca_result_pre_intercept_nonstd,
                             obs.scale = 1,
                             var.scale = 1,
                             groups = rownames(w2_pre_intercept),  # Use Rain_Event_Number as group lab
                             ellipse = TRUE,  # Add confidence ellipses
                             circle = TRUE)   # Add a unit circle

# Customize the biplot
biplot_pre_intercept_nonstd <- biplot_pre_intercept_nonstd +
  ggtitle("PCA Biplot of W2 Stormwater Intercepts (Pre-Application, non std)") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Print the biplot
print(biplot_pre_intercept_nonstd)
```




```r
# Simple base R biplot
biplot(pca_result_pre_intercept_nonstd, scale = 0,
       main = "PCA Biplot of W2 Stormwater Intercepts (Pre-Application, non std)")
```



Part V: Stepwise Multivariate Regression Analysis

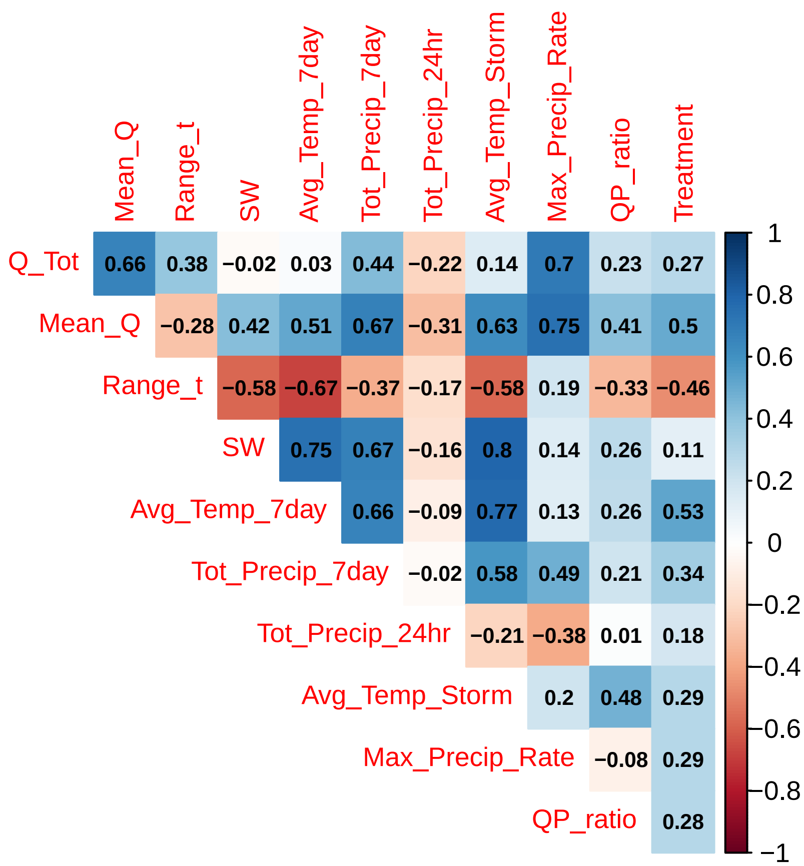
Let's see what the relationship between the environmental controls we calculated above are and the scores of the first 3 PCs from each PCA we calculated above is

```
# First, let's make a quick corrplot to see if there's any major colinearity we need to take care of be



# Calculate the correlation matrix for W2_Stormwater_EnvCont
# Remove the first column from the dataframe since it is non-numeric
W2_Stormwater_EnvCont2 <- W2_Stormwater_EnvCont[, -1]

W2_Stormwater_EnvCont2$Max_Precip_Rate[is.infinite(W2_Stormwater_EnvCont2$Max_Precip_Rate)] <- NA
W2_Stormwater_EnvCont2$QP_ratio[is.infinite(W2_Stormwater_EnvCont2$QP_ratio)] <- NA



cor_matrix <- cor(W2_Stormwater_EnvCont2, use = "complete.obs") # make sure we only use complete observ

# Find highly correlated variables (absolute correlation > 0.9)
high_corr_vars <- findCorrelation(cor_matrix, cutoff = 0.85, verbose = TRUE)
```

```
## Compare row 3  and column  2 with corr  0.999
```

```
##   Means:  0.587 vs 0.436 so flagging column 3
## Compare row 2  and column  4 with corr  0.922
##   Means:  0.545 vs 0.412 so flagging column 2
## Compare row 9  and column  7 with corr  0.856
##   Means:  0.496 vs 0.395 so flagging column 9
## Compare row 1  and column  12 with corr  0.859
##   Means:  0.368 vs 0.38 so flagging column 12
## All correlations <= 0.85
```

```r
# Remove highly correlated variables
filtered_EnvCont <- W2_Stormwater_EnvCont2[, -high_corr_vars]
filtered_EnvCont$Treatment <- c(rep(0, 12), rep(1, 13))

# Generate a correlation matrix for filtered data
filtered_cor_matrix <- cor(filtered_EnvCont, use = "complete.obs")

# Create the correlation plot
corrplot(filtered_cor_matrix,
         method = "color",         # Use color-coded tiles
         tl.cex = 0.8,             # Adjust text size
         na.label = " ",           # Label for missing correlations
         diag = FALSE,             # Exclude 1:1 Diagonal
         type = "upper",           # Exclude lower diagonal
         addCoef.col = "black",    # Add numerical coefficients
         number.cex = 0.65,         # Adjust size of correlation numbers
         sig.level = 0.05)
```



```r
# Now let's get the score data from our PCAs

scores_pre_slope <- as.data.frame(pca_result_pre_slope$x)
scores_pre_intercept <- as.data.frame(pca_result_pre_intercept$x)
scores_post_slope <- as.data.frame(pca_result_post_slope$x)
scores_post_intercept <- as.data.frame(pca_result_post_intercept$x)
scores_all_slope <- as.data.frame(pca_result_all_slope$x)
scores_all_intercept <- as.data.frame(pca_result_all_intercept$x)

# Now we need to merge the scores with our environmental control

# Split filtered_EnvCont into two subsets
env_cont_pre <- filtered_EnvCont[1:12, ]  # Rows 1 to 12
env_cont_post <- filtered_EnvCont[13:25, ]  # Rows 13 to 25

# Combine the split env cont dataframes with pre scores data frames
pre_slope_combined <- cbind(scores_pre_slope, env_cont_pre)
pre_intercept_combined <- cbind(scores_pre_intercept, env_cont_pre)

# Now do the same with post scores data frames
post_slope_combined <- cbind(scores_post_slope, env_cont_post)
post_intercept_combined <- cbind(scores_post_intercept, env_cont_post)

# Output structure of combined data frames for verification
str(pre_slope_combined)
```

```
## 'data.frame':    12 obs. of  23 variables:
##  $ PC1            : num  1.245 1.857 0.845 0.451 0.328 ...
##  $ PC2            : num  0.1689 0.449 0.084 -0.0304 -0.0861 ...
##  $ PC3            : num  0.275 -0.397 -0.336 -0.199 -0.526 ...
##  $ PC4            : num  1.35 -0.421 0.34 0.713 0.45 ...
##  $ PC5            : num  -0.981 -1.219 -0.108 -0.023 0.123 ...
##  $ PC6            : num  -0.7402 -0.1778 -0.0689 0.5183 0.1684 ...
##  $ PC7            : num  -0.822 0.659 -0.462 -0.713 0.381 ...
##  $ PC8            : num  0.2647 0.0878 -0.3017 -0.1158 -0.5948 ...
##  $ PC9            : num  -0.03422 -0.00332 0.31339 -0.18423 -0.11706 ...
##  $ PC10           : num  -0.0966 0.0319 0.0368 0.1343 -0.0934 ...
##  $ PC11           : num  -1.11e-16 -2.64e-16 -1.08e-15 1.28e-15 1.23e-16 ...
##  $ PC12           : num  -1.90e-17 -4.68e-17 -3.09e-16 3.75e-16 -1.08e-16 ...
##  $ Q_Tot          : num  144463 223861 372766 252858 972445 ...
##  $ Mean_Q         : num  3.74 3.09 4.1 2.64 11.47 ...
##  $ Range_t        : num  608 1560 1903 1718 1327 ...
##  $ SW             : num  0.975 0.992 0.975 0.942 0.91 ...
##  $ Avg_Temp_7day  : num  19.8 20.3 18.9 17.2 NaN ...
##  $ Tot_Precip_7day: num  0.693 0.89 0.479 0.397 0.865 ...
##  $ Tot_Precip_24hr: num  0.6496 0.2283 0.0501 0 0.08 ...
##  $ Avg_Temp_Storm : num  22.9 17.8 17.5 NaN NaN ...
##  $ Max_Precip_Rate: num  0 0 0.0801 0.0926 0.1875 ...
##  $ QP_ratio       : num  NA NA 1163749 428056 759145 ...
##  $ Treatment      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
str(post_slope_combined)
```

```
## 'data.frame':	13 obs. of  22 variables:
##  $ PC1            : num  -2.312 3.081 4.853 2.534 0.268 ...
##  $ PC2            : num  0.493 -1.46 3.351 0.116 -0.293 ...
##  $ PC3            : num  -0.923 0.46 0.486 -0.234 -0.391 ...
##  $ PC4            : num  0.292 1.731 -0.376 0.578 -0.346 ...
##  $ PC5            : num  -0.207 -0.258 -0.538 0.686 1.052 ...
##  $ PC6            : num  0.0804 0.2999 0.1537 -0.8193 -0.2578 ...
##  $ PC7            : num  -0.0488 0.2192 -0.0529 -0.0931 -0.0193 ...
##  $ PC8            : num  -0.0139 0.0522 0.0608 -0.2478 0.4049 ...
##  $ PC9            : num  0.000859 0.040446 0.000264 -0.002925 0.036605 ...
##  $ PC10           : num  -2.99e-16 6.84e-16 4.39e-16 4.47e-17 2.06e-17 ...
##  $ PC11           : num  -2.66e-17 1.84e-16 6.61e-16 2.06e-16 -1.31e-16 ...
##  $ Q_Tot          : num  776032 831499 2595354 2776739 4444128 ...
##  $ Mean_Q         : num  20 48.7 88.9 95.2 62.1 ...
##  $ Range_t        : num  631 330 631 631 1365 ...
##  $ SW             : num  0.997 0.986 0.97 0.964 0.953 ...
##  $ Avg_Temp_7day  : num  21.2 20.8 22.3 22.3 22.8 ...
##  $ Tot_Precip_7day: num  3.56 2.5 2.86 4.23 3.93 ...
##  $ Tot_Precip_24hr: num  0.9625 0.0533 0.1025 0.0242 0.0754 ...
##  $ Avg_Temp_Storm : num  19 18.1 21.1 21.4 20.1 ...
##  $ Max_Precip_Rate: num  0.109 0.262 0.333 0.277 0.178 ...
##  $ QP_ratio       : num  1509300 792533 1912333 2503446 2309138 ...
##  $ Treatment      : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
# Now lets do it for all storms

all_slope_combined <- cbind(scores_all_slope, filtered_EnvCont)
all_intercept_combined <- cbind(scores_all_intercept, filtered_EnvCont)
str(all_slope_combined)
```

```
## 'data.frame':	25 obs. of  25 variables:
##  $ PC1            : num  -1.404 -2.065 -1.077 -0.865 -0.645 ...
##  $ PC2            : num  0.188 0.2982 0.085 0.029 -0.0432 ...
##  $ PC3            : num  0.192 -0.577 0.271 0.477 0.425 ...
##  $ PC4            : num  1.138 0.881 0.513 0.319 0.57 ...
##  $ PC5            : num  -1.251 0.0865 -0.4277 -0.5506 -0.1058 ...
##  $ PC6            : num  -0.5882 -0.0641 0.1702 0.5105 0.3449 ...
##  $ PC7            : num  -0.228 -0.352 0.144 0.16 0.342 ...
##  $ PC8            : num  0.379 -0.102 0.105 0.328 -0.528 ...
##  $ PC9            : num  0.0401 0.1104 -0.3526 0.0686 -0.165 ...
##  $ PC10           : num  -0.0201 0.0295 -0.0955 -0.0377 0.0246 ...
##  $ PC11           : num  -0.0468 0.0239 0.1268 0.0093 0.1566 ...
##  $ PC12           : num  -0.0171 -0.0731 -0.032 0.048 0.2308 ...
##  $ PC13           : num  -0.00141 -0.00677 0.06038 0.01535 0.01633 ...
##  $ PC14           : num  3.88e-17 -1.35e-16 5.89e-17 1.22e-16 2.21e-16 ...
##  $ Q_Tot          : num  144463 223861 372766 252858 972445 ...
##  $ Mean_Q         : num  3.74 3.09 4.1 2.64 11.47 ...
##  $ Range_t        : num  608 1560 1903 1718 1327 ...
##  $ SW             : num  0.975 0.992 0.975 0.942 0.91 ...
##  $ Avg_Temp_7day  : num  19.8 20.3 18.9 17.2 NaN ...
##  $ Tot_Precip_7day: num  0.693 0.89 0.479 0.397 0.865 ...
##  $ Tot_Precip_24hr: num  0.6496 0.2283 0.0501 0 0.08 ...
##  $ Avg_Temp_Storm : num  22.9 17.8 17.5 NaN NaN ...
##  $ Max_Precip_Rate: num  0 0 0.0801 0.0926 0.1875 ...
```

```
##  $ QP_ratio       : num  NA NA 1163749 428056 759145 ...
##  $ Treatment      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
# We only want to use the first three PCs from each of these dataframes, let's remove the extra
# Remove PC4 to PC12 from pre_* data frames
pre_slope_combined <- pre_slope_combined %>% select(-PC4:-PC12)
pre_intercept_combined <- pre_intercept_combined %>% select(-PC4:-PC12)

# Remove PC4 to PC11 from post_* data frames
post_slope_combined <- post_slope_combined %>% select(-PC4:-PC11)
post_intercept_combined <- post_intercept_combined %>% select(-PC4:-PC11)

# Remove PC4 to PC14 from all_* data frames
all_slope_combined <- all_slope_combined %>% select(-PC4:-PC14) %>% na.omit(all_slope_combined)
all_intercept_combined <- all_intercept_combined %>% select(-PC4:-PC14) %>% na.omit(all_intercept_combi


# Verify the results
str(pre_slope_combined)
```

```
## 'data.frame':    12 obs. of  14 variables:
##  $ PC1            : num  1.245 1.857 0.845 0.451 0.328 ...
##  $ PC2            : num  0.1689 0.449 0.084 -0.0304 -0.0861 ...
##  $ PC3            : num  0.275 -0.397 -0.336 -0.199 -0.526 ...
##  $ Q_Tot          : num  144463 223861 372766 252858 972445 ...
##  $ Mean_Q         : num  3.74 3.09 4.1 2.64 11.47 ...
##  $ Range_t        : num  608 1560 1903 1718 1327 ...
##  $ SW             : num  0.975 0.992 0.975 0.942 0.91 ...
##  $ Avg_Temp_7day  : num  19.8 20.3 18.9 17.2 NaN ...
##  $ Tot_Precip_7day: num  0.693 0.89 0.479 0.397 0.865 ...
##  $ Tot_Precip_24hr: num  0.6496 0.2283 0.0501 0 0.08 ...
##  $ Avg_Temp_Storm : num  22.9 17.8 17.5 NaN NaN ...
##  $ Max_Precip_Rate: num  0 0 0.0801 0.0926 0.1875 ...
##  $ QP_ratio       : num  NA NA 1163749 428056 759145 ...
##  $ Treatment      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
str(post_slope_combined)
```

```
## 'data.frame':    13 obs. of  14 variables:
##  $ PC1            : num  -2.312 3.081 4.853 2.534 0.268 ...
##  $ PC2            : num  0.493 -1.46 3.351 0.116 -0.293 ...
##  $ PC3            : num  -0.923 0.46 0.486 -0.234 -0.391 ...
##  $ Q_Tot          : num  776032 831499 2595354 2776739 4444128 ...
##  $ Mean_Q         : num  20 48.7 88.9 95.2 62.1 ...
##  $ Range_t        : num  631 330 631 631 1365 ...
##  $ SW             : num  0.997 0.986 0.97 0.964 0.953 ...
##  $ Avg_Temp_7day  : num  21.2 20.8 22.3 22.3 22.8 ...
##  $ Tot_Precip_7day: num  3.56 2.5 2.86 4.23 3.93 ...
##  $ Tot_Precip_24hr: num  0.9625 0.0533 0.1025 0.0242 0.0754 ...
##  $ Avg_Temp_Storm : num  19 18.1 21.1 21.4 20.1 ...
##  $ Max_Precip_Rate: num  0.109 0.262 0.333 0.277 0.178 ...
##  $ QP_ratio       : num  1509300 792533 1912333 2503446 2309138 ...
##  $ Treatment      : num  1 1 1 1 1 1 1 1 1 1 ...
```

```r
str(all_slope_combined)
```

```
## 'data.frame':    20 obs. of  14 variables:
##  $ PC1             : num  -1.08 1.81 1.75 0.68 -2.15 ...
##  $ PC2             : num  0.085 -8.4641 -0.5133 -0.0103 0.318 ...
##  $ PC3             : num  0.271 -0.946 0.56 1.473 -0.276 ...
##  $ Q_Tot           : num  372766 2541009 2255787 1191444 300724 ...
##  $ Mean_Q          : num  4.1 17.4 12.8 10.6 11.7 ...
##  $ Range_t         : num  1903 2766 2676 2139 450 ...
##  $ SW              : num  0.975 0.564 0.433 0.268 0.899 ...
##  $ Avg_Temp_7day   : num  18.91 14.57 7.53 10.15 14.48 ...
##  $ Tot_Precip_7day: num  0.479 0.931 0.2 0.18 0.726 ...
##  $ Tot_Precip_24hr: num  0.0501 0.3602 0 0.1025 0.3163 ...
##  $ Avg_Temp_Storm : num  17.5 14.1 10.8 13.1 19.5 ...
##  $ Max_Precip_Rate: num  0.0801 0.2 0.1851 0.1209 0.0616 ...
##  $ QP_ratio        : num  1163749 1270455 1232352 1360355 1088592 ...
##  $ Treatment       : num  0 0 0 0 0 0 0 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:5] 1 2 4 5 11
##   ..- attr(*, "names")= chr [1:5] "2022-01" "2022-02" "2022-04" "2022-05" ...
```

```r
# before we begin, let's see if we did a good enough job
# removing colinearity problems


step_model_pre_intercept <- lm(PC1 ~ . - PC2 - PC3, data = pre_intercept_combined)  # here we use all E
# Check for aliased coefficients
alias(step_model_pre_intercept)
```

```
## Model :
## PC1 ~ (PC2 + PC3 + Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day +
##     Tot_Precip_7day + Tot_Precip_24hr + Avg_Temp_Storm + Max_Precip_Rate +
##     QP_ratio + Treatment) - PC2 - PC3
##
## Complete :
##                 (Intercept)          Q_Tot                Mean_Q
## Tot_Precip_24hr            421/1165                     0          -1236/52883
## Avg_Temp_Storm         7367165/643408                     0        351169/5440301
## Max_Precip_Rate       -36823/1107606                     0      2897629/611518841
## QP_ratio          120029199284/28889        12933/43024      -2793235337/15658
## Treatment                         0                     0                     0
##                 Range_t              SW                   Avg_Temp_7day
## Tot_Precip_24hr                   0   -10708301/14901731            2188/34101
## Avg_Temp_Storm      -125361/27110989         -40003/50892           26378/29869
## Max_Precip_Rate                   0          17031/412841 17237767/27089075918
## QP_ratio               -1238830/7193     -11272563154/7881       -362894582/4041
## Treatment                         0                     0                     0
##                 Tot_Precip_7day
## Tot_Precip_24hr        95381/3282007
## Avg_Temp_Storm         -466775/99017
## Max_Precip_Rate          -2781/71620
## QP_ratio         662524170283/302430
## Treatment                         0
```

```r
# based on this, we won't be able to run the stepAIC model
# with Q_ratio, Avg_Temp_Storm, Tot_Precip_24hr, or
# Max_Precip_Rate because of the linear dependencies among
# our data. Therefore, we will make a simpler multiple
# regression with less problematic predictors, see below as
# we test

# Simplified model excluding problematic predictors
step_model_pre_intercept_reduced <- lm(PC1 ~ Q_Tot + Mean_Q +
    Range_t + SW + Avg_Temp_7day + Tot_Precip_7day + Treatment,
    data = pre_intercept_combined)

alias(step_model_pre_intercept_reduced)
```

```
## Model :
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
## Complete :
##           (Intercept) Q_Tot Mean_Q Range_t SW Avg_Temp_7day Tot_Precip_7day
## Treatment 0           0     0      0       0  0             0
```

```r
# this works much better we will use the above general
# formula for all regressions going forward: PC1 ~ Q_Tot +
# Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
```

As we saw above, we still need to remove some additional variables to successfully run our analyses - but lets do it now!

```r
## PC1 ## Perform stepwise regression for
## pre_slope_combined
step_model_pre_slope_PC1 <- lm(PC1 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = pre_slope_combined)
stepAIC_model_pre_slope_PC1 <- stepAIC(step_model_pre_slope_PC1,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=-6.71
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=-6.71
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - SW               1    0.0028  1.6765 -8.6934
## - Range_t          1    0.0512  1.7248 -8.3805
## - Mean_Q           1    0.0544  1.7281 -8.3599
## <none>                          1.6737 -6.7116
## - Avg_Temp_7day    1    0.9541  2.6278 -3.7494
## - Tot_Precip_7day  1    0.9974  2.6711 -3.5695
## - Q_Tot            1   10.9729 12.6466 13.5344
```

```
##
## Step:  AIC=-8.69
## PC1 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS      AIC
## - Mean_Q           1    0.0517  1.7281 -10.3596
## - Range_t          1    0.0648  1.7413 -10.2760
## <none>                          1.6765  -8.6934
## + SW               1    0.0028  1.6737  -6.7116
## - Tot_Precip_7day  1    1.0168  2.6933  -5.4786
## - Avg_Temp_7day    1    2.1855  3.8620  -1.5138
## - Q_Tot            1   10.9908 12.6672  11.5523
##
## Step:  AIC=-10.36
## PC1 ~ Q_Tot + Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS      AIC
## - Range_t          1    0.1267  1.8548 -11.5810
## <none>                          1.7281 -10.3596
## + Mean_Q           1    0.0517  1.6765  -8.6934
## + SW               1    0.0001  1.7281  -8.3599
## - Tot_Precip_7day  1    1.0591  2.7872  -7.1013
## - Avg_Temp_7day    1    2.1643  3.8924  -3.4275
## - Q_Tot            1   14.8979 16.6260  12.5438
##
## Step:  AIC=-11.58
## PC1 ~ Q_Tot + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS      AIC
## <none>                          1.855 -11.5810
## + Range_t          1     0.127  1.728 -10.3596
## + Mean_Q           1     0.114  1.741 -10.2760
## + SW               1     0.006  1.849  -9.6143
## - Tot_Precip_7day  1     1.582  3.436  -6.7979
## - Avg_Temp_7day    1     2.166  4.021  -5.0703
## - Q_Tot            1    39.954 41.809  20.6874
```

```
summary(stepAIC_model_pre_slope_PC1)
```

```
##
## Call:
## lm(formula = PC1 ~ Q_Tot + Avg_Temp_7day + Tot_Precip_7day, data = pre_slope_combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78623 -0.25014  0.06169  0.23452  0.77192
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.142e+00  1.141e+00   4.507  0.00277 **
## Q_Tot           -3.477e-06  2.832e-07 -12.279 5.45e-06 ***
## Avg_Temp_7day   -1.910e-01  6.681e-02  -2.859  0.02437 *
## Tot_Precip_7day  6.719e-01  2.750e-01   2.443  0.04455 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5148 on 7 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9727, Adjusted R-squared:  0.961
## F-statistic: 83.24 on 3 and 7 DF,  p-value: 7.711e-06
```

```r
# Perform stepwise regression for post_slope_combined
step_model_post_slope_PC1 <- lm(PC1 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = post_slope_combined)
stepAIC_model_post_slope_PC1 <- stepAIC(step_model_post_slope_PC1,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=19.91
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=19.91
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    0.1949 20.683 18.037
## - Range_t          1    0.4135 20.902 18.173
## - Q_Tot            1    0.5585 21.047 18.263
## - Avg_Temp_7day    1    2.1047 22.593 19.185
## - SW               1    2.1785 22.667 19.227
## <none>                         20.488 19.914
## - Mean_Q           1   11.2742 31.762 23.613
##
## Step:  AIC=18.04
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Range_t          1    0.2483 20.931 16.192
## - Q_Tot            1    0.4546 21.138 16.319
## - SW               1    2.5454 23.228 17.546
## - Avg_Temp_7day    1    2.7302 23.413 17.648
## <none>                         20.683 18.037
## + Tot_Precip_7day  1    0.1949 20.488 19.914
## - Mean_Q           1   11.3231 32.006 21.713
##
## Step:  AIC=16.19
## PC1 ~ Q_Tot + Mean_Q + SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Q_Tot            1    0.3045 21.236 14.380
## - SW               1    2.3693 23.301 15.586
## - Avg_Temp_7day    1    3.4471 24.378 16.174
## <none>                         20.931 16.192
## + Range_t          1    0.2483 20.683 18.037
## + Tot_Precip_7day  1    0.0297 20.902 18.173
## - Mean_Q           1   20.6886 41.620 23.127
##
```

```
## Step:  AIC=14.38
## PC1 ~ Mean_Q + SW + Avg_Temp_7day
##
##                  Df Sum of Sq    RSS    AIC
## - SW              1     2.414 23.650 13.779
## - Avg_Temp_7day   1     3.182 24.417 14.194
## <none>                        21.236 14.380
## + Q_Tot           1     0.305 20.931 16.192
## + Tot_Precip_7day 1     0.186 21.050 16.265
## + Range_t         1     0.098 21.138 16.319
## - Mean_Q          1    35.488 56.724 25.152
##
## Step:  AIC=13.78
## PC1 ~ Mean_Q + Avg_Temp_7day
##
##                  Df Sum of Sq    RSS    AIC
## - Avg_Temp_7day   1     0.985 24.635 12.310
## <none>                        23.650 13.779
## + SW              1     2.414 21.236 14.380
## + Range_t         1     0.421 23.228 15.546
## + Q_Tot           1     0.349 23.301 15.586
## + Tot_Precip_7day 1     0.163 23.487 15.689
## - Mean_Q          1    53.407 77.057 27.135
##
## Step:  AIC=12.31
## PC1 ~ Mean_Q
##
##                  Df Sum of Sq    RSS    AIC
## <none>                        24.635 12.310
## + Avg_Temp_7day   1     0.985 23.650 13.779
## + SW              1     0.217 24.417 14.194
## + Q_Tot           1     0.084 24.551 14.265
## + Tot_Precip_7day 1     0.008 24.627 14.306
## + Range_t         1     0.003 24.632 14.308
## - Mean_Q          1    58.201 82.836 26.075
```

```r
summary(stepAIC_model_post_slope_PC1)
```

```
##
## Call:
## lm(formula = PC1 ~ Mean_Q, data = post_slope_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0222 -0.8103 -0.4904  1.0390  2.8618
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.05588    0.72911  -4.191 0.001508 **
## Mean_Q       0.06725    0.01319   5.098 0.000345 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.497 on 11 degrees of freedom
```

```
## Multiple R-squared:  0.7026, Adjusted R-squared:  0.6756
## F-statistic: 25.99 on 1 and 11 DF,  p-value: 0.0003452
```

```r
# Perform stepwise regression for all_slope_combined
step_model_all_slope_PC1 <- lm(PC1 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = all_slope_combined)
stepAIC_model_all_slope_PC1 <- stepAIC(step_model_all_slope_PC1,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=27.8
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1     0.041 36.120 25.822
## - Treatment        1     0.815 36.893 26.246
## - SW               1     2.247 38.326 27.008
## - Q_Tot            1     3.129 39.207 27.463
## <none>                         36.079 27.799
## - Avg_Temp_7day    1     4.274 40.353 28.039
## - Range_t          1     9.604 45.683 30.520
## - Mean_Q           1    38.388 74.467 40.292
##
## Step:  AIC=25.82
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Treatment        1     0.784 36.904 24.251
## - SW               1     2.726 38.846 25.278
## - Q_Tot            1     3.402 39.522 25.623
## <none>                         36.120 25.822
## - Avg_Temp_7day    1     4.335 40.455 26.089
## + Tot_Precip_7day  1     0.041 36.079 27.799
## - Range_t          1     9.737 45.857 28.596
## - Mean_Q           1    38.363 74.483 38.297
##
## Step:  AIC=24.25
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1     1.972 38.876 23.293
## - Q_Tot            1     2.644 39.548 23.635
## - Avg_Temp_7day    1     3.593 40.497 24.110
## <none>                         36.904 24.251
## + Treatment        1     0.784 36.120 25.822
## + Tot_Precip_7day  1     0.010 36.893 26.246
## - Range_t          1     9.707 46.610 26.922
## - Mean_Q           1    37.796 74.700 36.355
##
## Step:  AIC=23.29
## PC1 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Avg_Temp_7day    1     1.843 40.719 22.219
```

```
## - Q_Tot            1     3.071 41.947 22.813
## <none>                         38.876 23.293
## + SW               1     1.972 36.904 24.251
## + Tot_Precip_7day  1     0.539 38.337 25.013
## + Treatment        1     0.029 38.846 25.278
## - Range_t          1     9.271 48.147 25.571
## - Mean_Q           1    40.400 79.276 35.544
##
## Step:  AIC=22.22
## PC1 ~ Q_Tot + Mean_Q + Range_t
##
##                   Df Sum of Sq    RSS    AIC
## - Q_Tot            1     3.291 44.010 21.774
## <none>                         40.719 22.219
## + Avg_Temp_7day    1     1.843 38.876 23.293
## + SW               1     0.222 40.497 24.110
## + Treatment        1     0.150 40.569 24.145
## + Tot_Precip_7day  1     0.000 40.718 24.219
## - Range_t          1    17.629 58.347 27.414
## - Mean_Q           1    38.560 79.279 33.545
##
## Step:  AIC=21.77
## PC1 ~ Mean_Q + Range_t
##
##                   Df Sum of Sq     RSS    AIC
## <none>                          44.010 21.774
## + Q_Tot            1     3.291  40.719 22.219
## + Avg_Temp_7day    1     2.063  41.947 22.813
## + Treatment        1     1.069  42.941 23.282
## + SW               1     0.313  43.697 23.631
## + Tot_Precip_7day  1     0.298  43.712 23.638
## - Range_t          1    20.688  64.697 27.480
## - Mean_Q           1    89.427 133.437 41.958
```

```
summary(stepAIC_model_all_slope_PC1)
```

```
##
## Call:
## lm(formula = PC1 ~ Mean_Q + Range_t, data = all_slope_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2228 -0.8208 -0.2375  0.6727  3.6386
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.6669967  0.8271593  -4.433 0.000364 ***
## Mean_Q       0.0738920  0.0125723   5.877 1.83e-05 ***
## Range_t      0.0013725  0.0004855   2.827 0.011629 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.609 on 17 degrees of freedom
## Multiple R-squared:  0.6792, Adjusted R-squared:  0.6415
```

```
## F-statistic:    18 on 2 and 17 DF,  p-value: 6.347e-05
```

```r
## PC2 ## Perform stepwise regression for
## pre_slope_combined
step_model_pre_slope_PC2 <- lm(PC2 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = pre_slope_combined)
stepAIC_model_pre_slope_PC2 <- stepAIC(step_model_pre_slope_PC2,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=18.36
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=18.36
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    0.0056 16.354 16.362
## - SW               1    0.7923 17.141 16.879
## - Range_t          1    1.8815 18.230 17.557
## - Mean_Q           1    3.1992 19.548 18.325
## <none>                         16.349 18.359
## - Q_Tot            1    5.7281 22.077 19.663
## - Avg_Temp_7day    1    7.9703 24.319 20.727
##
## Step:  AIC=16.36
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1    0.7869 17.141 14.879
## - Range_t          1    2.3535 18.708 15.841
## <none>                         16.354 16.362
## - Mean_Q           1    4.1493 20.503 16.850
## - Q_Tot            1    6.4621 22.816 18.025
## + Tot_Precip_7day  1    0.0056 16.349 18.359
## - Avg_Temp_7day    1   10.3625 26.717 19.761
##
## Step:  AIC=14.88
## PC2 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Range_t          1    1.7672 18.908 13.959
## - Mean_Q           1    3.4084 20.549 14.874
## <none>                         17.141 14.879
## + SW               1    0.7869 16.354 16.362
## - Q_Tot            1    6.4994 23.640 16.416
## + Tot_Precip_7day  1    0.0001 17.141 16.879
## - Avg_Temp_7day    1   16.7713 33.912 20.385
##
## Step:  AIC=13.96
## PC2 ~ Q_Tot + Mean_Q + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
```

```
## - Mean_Q           1    1.6653 20.574 12.887
## <none>                        18.908 13.959
## + Range_t          1    1.7672 17.141 14.879
## - Q_Tot            1    6.5756 25.484 15.242
## + Tot_Precip_7day  1    0.4350 18.473 15.703
## + SW               1    0.2006 18.708 15.841
## - Avg_Temp_7day    1   15.4703 34.379 18.535
##
## Step:  AIC=12.89
## PC2 ~ Q_Tot + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         20.574 12.887
## - Q_Tot            1    4.9542 25.528 13.261
## + Mean_Q           1    1.6653 18.908 13.959
## + Tot_Precip_7day  1    0.4278 20.146 14.656
## + SW               1    0.0496 20.524 14.861
## + Range_t          1    0.0242 20.549 14.874
## - Avg_Temp_7day    1   14.0915 34.665 16.626
```

```r
summary(stepAIC_model_pre_slope_PC2)
```

```
##
## Call:
## lm(formula = PC2 ~ Q_Tot + Avg_Temp_7day, data = pre_slope_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6715 -0.5891 -0.3045  0.6298  2.3074
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.119e+00  3.190e+00  -2.231   0.0562 .
## Q_Tot          1.168e-06  8.417e-07   1.388   0.2026
## Avg_Temp_7day  3.751e-01  1.602e-01   2.341   0.0474 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.604 on 8 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4133, Adjusted R-squared:  0.2667
## F-statistic: 2.818 on 2 and 8 DF,  p-value: 0.1185
```

```r
# Perform stepwise regression for post_slope_combined
step_model_post_slope_PC2 <- lm(PC2 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = post_slope_combined)
stepAIC_model_post_slope_PC2 <- stepAIC(step_model_post_slope_PC2,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=17.27
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
```

```
##
## Step:  AIC=17.27
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Range_t          1   0.00198 16.721 15.272
## - Mean_Q           1   0.17843 16.898 15.409
## - Q_Tot            1   0.25662 16.976 15.469
## - Tot_Precip_7day  1   0.34208 17.061 15.534
## - SW               1   0.65850 17.378 15.773
## - Avg_Temp_7day    1   2.32461 19.044 16.963
## <none>                         16.719 17.271
##
## Step:  AIC=15.27
## PC2 ~ Q_Tot + Mean_Q + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Mean_Q           1   0.37089 17.092 13.558
## - Tot_Precip_7day  1   0.45699 17.178 13.623
## - Q_Tot            1   1.09278 17.814 14.095
## - SW               1   1.11169 17.833 14.109
## - Avg_Temp_7day    1   2.61989 19.341 15.165
## <none>                         16.721 15.272
## + Range_t          1   0.00198 16.719 17.271
##
## Step:  AIC=13.56
## PC2 ~ Q_Tot + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1   0.36535 17.457 11.832
## - Q_Tot            1   0.73895 17.831 12.108
## - SW               1   0.81855 17.910 12.166
## <none>                         17.092 13.558
## - Avg_Temp_7day    1   3.00497 20.097 13.663
## + Mean_Q           1   0.37089 16.721 15.272
## + Range_t          1   0.19444 16.898 15.409
##
## Step:  AIC=11.83
## PC2 ~ Q_Tot + SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Q_Tot            1    0.3891 17.846 10.119
## - SW               1    0.4669 17.924 10.176
## <none>                         17.457 11.832
## - Avg_Temp_7day    1    3.3722 20.829 12.129
## + Tot_Precip_7day  1    0.3654 17.092 13.558
## + Mean_Q           1    0.2793 17.178 13.623
## + Range_t          1    0.0301 17.427 13.810
##
## Step:  AIC=10.12
## PC2 ~ SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS     AIC
## - SW               1    0.8358 18.682  8.7142
```

```
## <none>                        17.846 10.1191
## - Avg_Temp_7day    1    3.7623 21.609 10.6059
## + Q_Tot            1    0.3891 17.457 11.8325
## + Range_t          1    0.3743 17.472 11.8436
## + Tot_Precip_7day  1    0.0156 17.831 12.1078
## + Mean_Q           1    0.0062 17.840 12.1146
##
## Step:  AIC=8.71
## PC2 ~ Avg_Temp_7day
##
##                   Df Sum of Sq    RSS     AIC
## <none>                         18.682  8.7142
## - Avg_Temp_7day    1    3.5923 22.274  9.0004
## + SW               1    0.8358 17.846 10.1191
## + Q_Tot            1    0.7580 17.924 10.1757
## + Range_t          1    0.2649 18.417 10.5285
## + Mean_Q           1    0.2112 18.471 10.5664
## + Tot_Precip_7day  1    0.2067 18.476 10.5696
```

```
summary(stepAIC_model_post_slope_PC2)
```

```
##
## Call:
## lm(formula = PC2 ~ Avg_Temp_7day, data = post_slope_combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.98691 -0.83832  0.05775  0.67039  2.89070
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.3649     2.3417  -1.437    0.179
## Avg_Temp_7day   0.1714     0.1179   1.454    0.174
##
## Residual standard error: 1.303 on 11 degrees of freedom
## Multiple R-squared:  0.1613, Adjusted R-squared:  0.08502
## F-statistic: 2.115 on 1 and 11 DF,  p-value: 0.1738
```

```
# Perform stepwise regression for all_slope_combined
step_model_all_slope_PC2 <- lm(PC2 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = all_slope_combined)
stepAIC_model_all_slope_PC2 <- stepAIC(step_model_all_slope_PC2,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=31.36
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    0.3965 43.509 29.545
## - SW               1    0.7143 43.827 29.690
## - Mean_Q           1    1.1786 44.291 29.901
## - Q_Tot            1    1.1837 44.296 29.903
```

```
## - Range_t          1    2.8632 45.975 30.648
## - Treatment        1    3.0669 46.179 30.736
## - Avg_Temp_7day    1    4.3876 47.500 31.300
## <none>                        43.112 31.361
##
## Step:  AIC=29.54
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Q_Tot            1    0.8299 44.339 27.922
## - Mean_Q           1    1.1622 44.671 28.072
## - SW               1    1.2207 44.729 28.098
## - Treatment        1    2.8579 46.367 28.817
## - Range_t          1    3.4342 46.943 29.064
## - Avg_Temp_7day    1    4.0039 47.513 29.305
## <none>                        43.509 29.545
## + Tot_Precip_7day  1    0.3965 43.112 31.361
##
## Step:  AIC=27.92
## PC2 ~ Mean_Q + Range_t + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Mean_Q           1    0.3389 44.678 26.075
## - SW               1    0.9065 45.245 26.327
## - Treatment        1    2.0931 46.432 26.845
## - Avg_Temp_7day    1    3.5826 47.921 27.477
## <none>                        44.339 27.923
## + Q_Tot            1    0.8299 43.509 29.545
## + Tot_Precip_7day  1    0.0427 44.296 29.903
## - Range_t          1   15.7002 60.039 31.985
##
## Step:  AIC=26.07
## PC2 ~ Range_t + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1    1.4859 46.164 24.729
## - Avg_Temp_7day    1    3.5182 48.196 25.591
## - Treatment        1    3.7181 48.396 25.674
## <none>                        44.678 26.075
## + Mean_Q           1    0.3389 44.339 27.922
## + Tot_Precip_7day  1    0.2045 44.473 27.983
## + Q_Tot            1    0.0067 44.671 28.072
## - Range_t          1   15.5590 60.237 30.051
##
## Step:  AIC=24.73
## PC2 ~ Range_t + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Avg_Temp_7day    1    2.0661 48.230 23.605
## - Treatment        1    2.2797 48.443 23.693
## <none>                        46.164 24.729
## + SW               1    1.4859 44.678 26.075
## + Tot_Precip_7day  1    0.9459 45.218 26.315
## + Mean_Q           1    0.9183 45.245 26.327
```

```
## + Q_Tot            1    0.1468 46.017 26.666
## - Range_t          1   19.9281 66.092 29.906
##
## Step:  AIC=23.6
## PC2 ~ Range_t + Treatment
##
##                  Df Sum of Sq    RSS    AIC
## - Treatment       1    1.2048 49.435 22.098
## <none>                        48.230 23.605
## + Avg_Temp_7day   1    2.0661 46.164 24.729
## + Mean_Q          1    0.1384 48.091 25.547
## + SW              1    0.0339 48.196 25.591
## + Q_Tot           1    0.0034 48.226 25.603
## + Tot_Precip_7day 1    0.0012 48.228 25.604
## - Range_t         1   19.6994 67.929 28.455
##
## Step:  AIC=22.1
## PC2 ~ Range_t
##
##                  Df Sum of Sq    RSS    AIC
## <none>                        49.435 22.098
## + Treatment       1    1.2048 48.230 23.605
## + Avg_Temp_7day   1    0.9913 48.443 23.693
## + Mean_Q          1    0.6606 48.774 23.829
## + Q_Tot           1    0.2996 49.135 23.977
## + SW              1    0.1713 49.263 24.029
## + Tot_Precip_7day 1    0.0385 49.396 24.083
## - Range_t         1   31.0348 80.469 29.843
```

```
summary(stepAIC_model_all_slope_PC2)
```

```
##
## Call:
## lm(formula = PC2 ~ Range_t, data = all_slope_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6085 -0.5310  0.0109  0.7740  2.1972
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.6024322  0.6135036   2.612  0.01765 *
## Range_t     -0.0016117  0.0004795  -3.362  0.00347 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.657 on 18 degrees of freedom
## Multiple R-squared:  0.3857, Adjusted R-squared:  0.3515
## F-statistic:  11.3 on 1 and 18 DF,  p-value: 0.003475
```

```
## PC3 ## Perform stepwise regression for
## pre_slope_combined
step_model_pre_slope_PC3 <- lm(PC3 ~ Q_Tot + Mean_Q + Range_t +
```

```r
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = pre_slope_combined)
stepAIC_model_pre_slope_PC3 <- stepAIC(step_model_pre_slope_PC3,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=-17.14
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=-17.14
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS      AIC
## <none>                          0.6485 -17.1402
## - SW               1    0.7582  1.4068 -10.6226
## - Range_t          1    1.2365  1.8850  -7.4037
## - Q_Tot            1    1.5024  2.1510  -5.9517
## - Avg_Temp_7day    1    2.9281  3.5767  -0.3580
## - Tot_Precip_7day  1    4.0273  4.6758   2.5896
## - Mean_Q           1   11.1931 11.8416  12.8110
```

```r
summary(stepAIC_model_pre_slope_PC3)
```

```
##
## Call:
## lm(formula = PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day +
##     Tot_Precip_7day, data = pre_slope_combined)
##
## Residuals:
##  2022-01  2022-02  2022-03  2022-04  2022-07  2022-08  2022-09  2023-01
## -0.21957 -0.39108  0.43498 -0.08240  0.21121 -0.30520  0.06721  0.16803
##  2023-02  2023-03  2023-04
##  0.25473 -0.01202 -0.12589
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.573e+00  9.642e-01   4.743  0.00902 **
## Q_Tot            1.349e-06  4.432e-07   3.044  0.03825 *
## Mean_Q          -1.894e-01  2.280e-02  -8.309  0.00115 **
## Range_t         -1.076e-03  3.898e-04  -2.762  0.05077 .
## SW               2.933e+00  1.356e+00   2.163  0.09662 .
## Avg_Temp_7day   -3.642e-01  8.570e-02  -4.250  0.01316 *
## Tot_Precip_7day  2.096e+00  4.205e-01   4.984  0.00758 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4027 on 4 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9654, Adjusted R-squared:  0.9134
## F-statistic: 18.59 on 6 and 4 DF,  p-value: 0.006867
```

```r
# Perform stepwise regression for post_slope_combined
step_model_post_slope_PC3 <- lm(PC3 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = post_slope_combined)
stepAIC_model_post_slope_PC3 <- stepAIC(step_model_post_slope_PC3,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=9.31
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=9.31
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - Avg_Temp_7day    1   0.01607  9.0774  7.3310
## - Mean_Q           1   0.77803  9.8394  8.3788
## - Q_Tot            1   1.09772 10.1591  8.7944
## <none>                          9.0614  9.3079
## - Tot_Precip_7day  1   2.08062 11.1420  9.9950
## - Range_t          1   2.14064 11.2020 10.0649
## - SW               1   2.45192 11.5133 10.4212
##
## Step:  AIC=7.33
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS    AIC
## - Mean_Q           1   0.85046  9.9279 6.4952
## - Q_Tot            1   1.21816 10.2956 6.9680
## <none>                          9.0774 7.3310
## - Tot_Precip_7day  1   2.39416 11.4716 8.3740
## - SW               1   2.43844 11.5159 8.4241
## - Range_t          1   2.62163 11.6991 8.6293
## + Avg_Temp_7day    1   0.01607  9.0614 9.3079
##
## Step:  AIC=6.5
## PC3 ~ Q_Tot + Range_t + SW + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS    AIC
## - Q_Tot            1   0.37675 10.3046 4.9794
## <none>                          9.9279 6.4952
## - SW               1   1.77336 11.7013 6.6317
## - Tot_Precip_7day  1   1.87878 11.8067 6.7483
## - Range_t          1   2.02113 11.9490 6.9041
## + Mean_Q           1   0.85046  9.0774 7.3310
## + Avg_Temp_7day    1   0.08850  9.8394 8.3788
##
## Step:  AIC=4.98
## PC3 ~ Range_t + SW + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS    AIC
## - SW               1   1.40348 11.7081 4.6393
## <none>                         10.3046 4.9794
```

```
## - Range_t           1   2.00513 12.3098 5.2908
## - Tot_Precip_7day   1   2.15935 12.4640 5.4526
## + Q_Tot             1   0.37675  9.9279 6.4952
## + Avg_Temp_7day     1   0.13782 10.1668 6.8044
## + Mean_Q            1   0.00905 10.2956 6.9680
##
## Step:  AIC=4.64
## PC3 ~ Range_t + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Range_t          1   0.67220 12.380 3.3651
## - Tot_Precip_7day  1   0.78479 12.493 3.4828
## <none>                         11.708 4.6393
## + SW               1   1.40348 10.305 4.9794
## + Mean_Q           1   0.02044 11.688 6.6166
## + Q_Tot            1   0.00687 11.701 6.6317
## + Avg_Temp_7day    1   0.00456 11.704 6.6343
##
## Step:  AIC=3.37
## PC3 ~ Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1   0.71812 13.098 2.0981
## <none>                         12.380 3.3651
## + Range_t          1   0.67220 11.708 4.6393
## + Avg_Temp_7day    1   0.31690 12.063 5.0280
## + Q_Tot            1   0.25415 12.126 5.0954
## + SW               1   0.07055 12.310 5.2908
## + Mean_Q           1   0.00604 12.374 5.3587
##
## Step:  AIC=2.1
## PC3 ~ 1
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         13.098 2.0981
## + Avg_Temp_7day    1   0.91013 12.188 3.1619
## + Tot_Precip_7day  1   0.71812 12.380 3.3651
## + Range_t          1   0.60553 12.493 3.4828
## + SW               1   0.22827 12.870 3.8695
## + Mean_Q           1   0.22013 12.878 3.8777
## + Q_Tot            1   0.00759 13.091 4.0905
```

```r
summary(stepAIC_model_post_slope_PC3)
```

```
##
## Call:
## lm(formula = PC3 ~ 1, data = post_slope_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1177 -0.9230 -0.2341  0.4861  2.3865
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 1.427e-16  2.898e-01       0       1
##
## Residual standard error: 1.045 on 12 degrees of freedom
```

```
# Perform stepwise regression for all_slope_combined
step_model_all_slope_PC3 <- lm(PC3 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = all_slope_combined)
stepAIC_model_all_slope_PC3 <- stepAIC(step_model_all_slope_PC3,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=25.87
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1   0.02716 32.784 23.884
## - Mean_Q           1   0.07195 32.829 23.912
## - Range_t          1   0.11410 32.871 23.937
## - Q_Tot            1   0.17832 32.935 23.976
## - Treatment        1   0.39584 33.153 24.108
## - SW               1   0.51391 33.271 24.179
## - Avg_Temp_7day    1   3.04210 35.799 25.644
## <none>                         32.757 25.868
##
## Step:  AIC=23.88
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Mean_Q           1    0.0709 32.855 21.927
## - Range_t          1    0.0969 32.881 21.943
## - Q_Tot            1    0.1512 32.935 21.976
## - Treatment        1    0.4253 33.209 22.142
## - SW               1    0.4914 33.276 22.182
## - Avg_Temp_7day    1    3.3961 36.180 23.856
## <none>                         32.784 23.884
## + Tot_Precip_7day  1    0.0272 32.757 25.868
##
## Step:  AIC=21.93
## PC3 ~ Q_Tot + Range_t + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Range_t          1    0.0414 32.896 19.953
## - Q_Tot            1    0.0859 32.941 19.980
## - Treatment        1    0.4513 33.306 20.200
## - SW               1    0.4741 33.329 20.214
## <none>                         32.855 21.927
## - Avg_Temp_7day    1    3.5843 36.439 21.998
## + Mean_Q           1    0.0709 32.784 23.884
## + Tot_Precip_7day  1    0.0261 32.829 23.912
##
## Step:  AIC=19.95
## PC3 ~ Q_Tot + SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
```

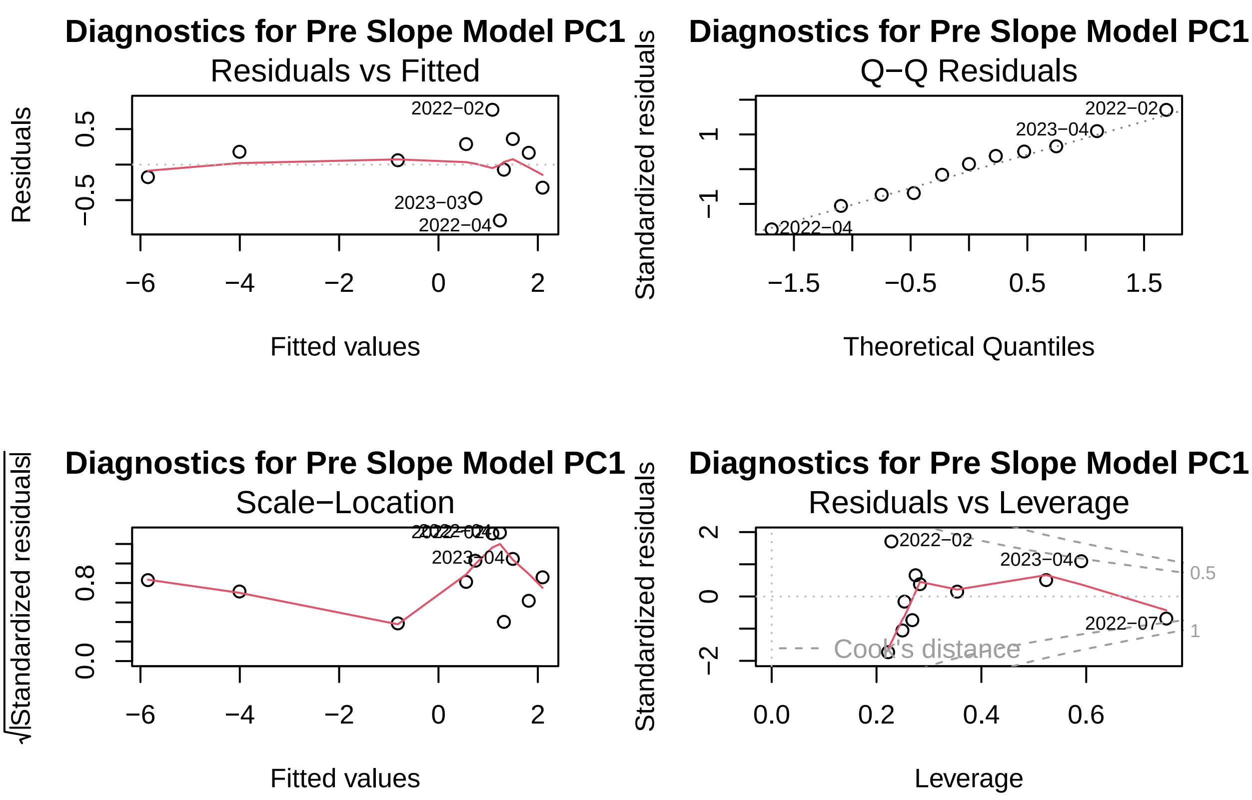
```
## - Q_Tot            1    0.0444 32.941 17.980
## - SW               1    0.7598 33.656 18.409
## - Treatment        1    0.9150 33.812 18.501
## <none>                         32.896 19.953
## - Avg_Temp_7day    1    3.5440 36.441 19.999
## + Range_t          1    0.0414 32.855 21.927
## + Mean_Q           1    0.0154 32.881 21.943
## + Tot_Precip_7day  1    0.0126 32.884 21.945
##
## Step:  AIC=17.98
## PC3 ~ SW + Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1    0.8080 33.749 16.464
## - Treatment        1    1.1590 34.100 16.671
## <none>                         32.941 17.980
## - Avg_Temp_7day    1    3.7854 36.726 18.155
## + Q_Tot            1    0.0444 32.896 19.953
## + Mean_Q           1    0.0051 32.936 19.977
## + Tot_Precip_7day  1    0.0014 32.939 19.979
## + Range_t          1    0.0000 32.941 19.980
##
## Step:  AIC=16.46
## PC3 ~ Avg_Temp_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Treatment        1    0.5274 34.276 14.774
## <none>                         33.749 16.464
## - Avg_Temp_7day    1    4.3688 38.118 16.899
## + SW               1    0.8080 32.941 17.980
## + Tot_Precip_7day  1    0.1617 33.587 18.368
## + Mean_Q           1    0.0993 33.650 18.405
## + Q_Tot            1    0.0926 33.656 18.409
## + Range_t          1    0.0645 33.684 18.426
##
## Step:  AIC=14.77
## PC3 ~ Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         34.276 14.774
## - Avg_Temp_7day    1    4.0351 38.311 15.000
## + Treatment        1    0.5274 33.749 16.464
## + Mean_Q           1    0.2804 33.996 16.610
## + Q_Tot            1    0.2597 34.017 16.622
## + SW               1    0.1764 34.100 16.671
## + Tot_Precip_7day  1    0.1568 34.120 16.683
## + Range_t          1    0.1391 34.137 16.693
```

```r
summary(stepAIC_model_all_slope_PC3)
```

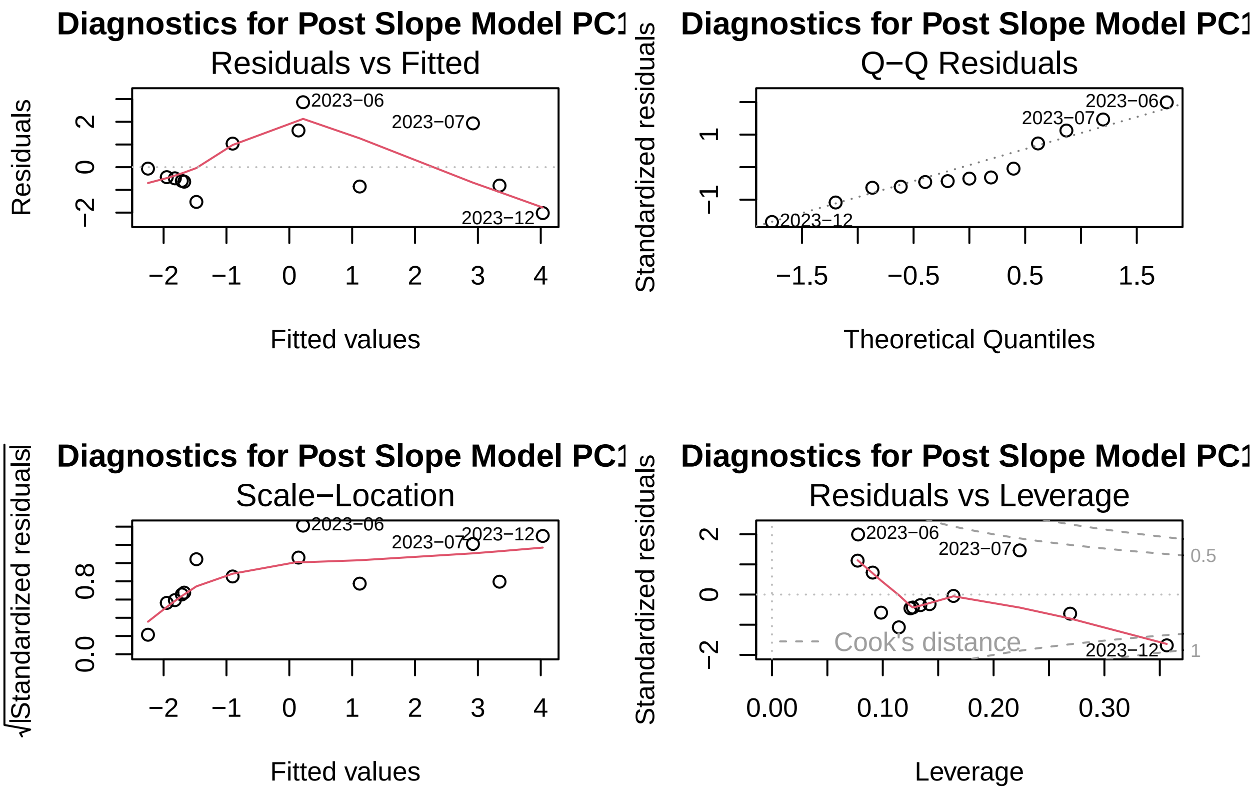
```
##
## Call:
## lm(formula = PC3 ~ Avg_Temp_7day, data = all_slope_combined)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.6944 -0.5391 -0.1300  0.6794  2.6152 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.87376    1.33858   1.400    0.179
## Avg_Temp_7day -0.10542    0.07242  -1.456    0.163
## 
## Residual standard error: 1.38 on 18 degrees of freedom
## Multiple R-squared:  0.1053, Adjusted R-squared:  0.05562 
## F-statistic: 2.119 on 1 and 18 DF,  p-value: 0.1627
```

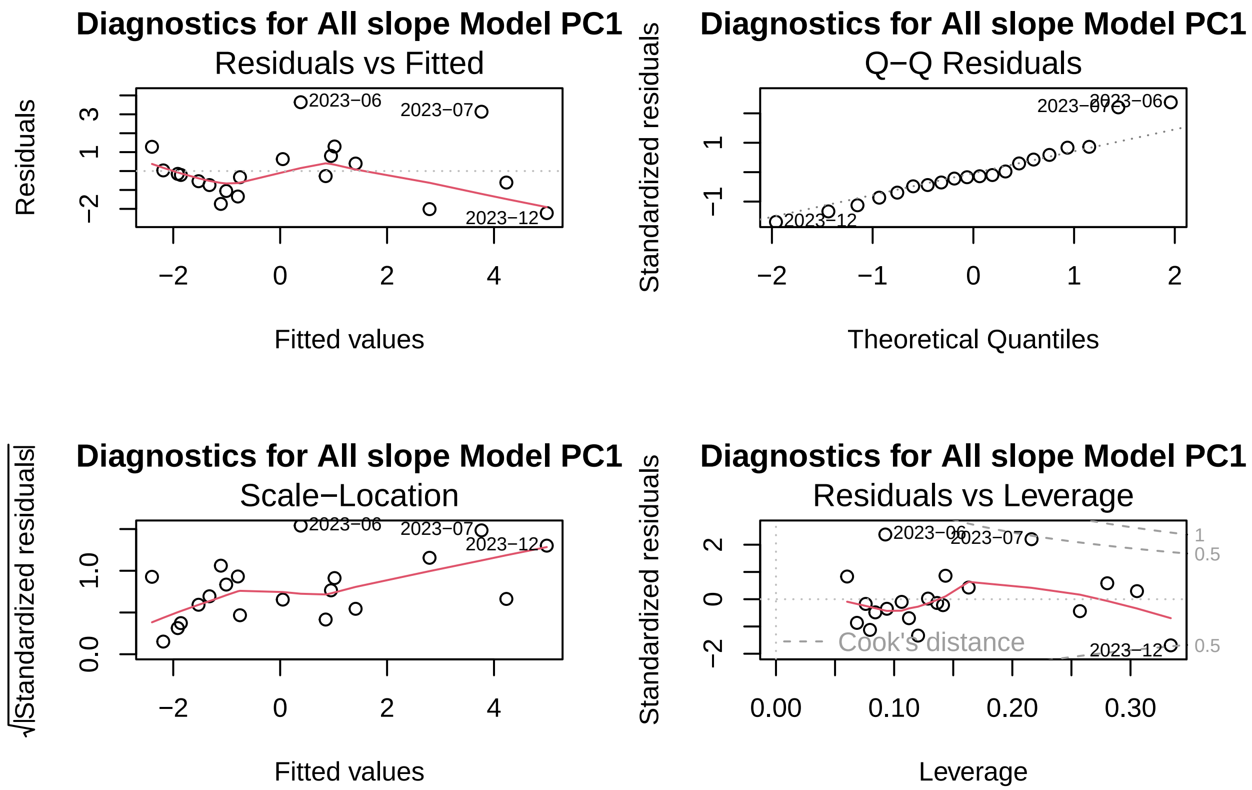
```r
## PC1 ##

# Diagnostic plots for the optimized models
par(mfrow = c(2, 2))  # Set layout for multiple plots

# Pre-slope
plot(stepAIC_model_pre_slope_PC1, main = "Diagnostics for Pre Slope Model PC1")
```



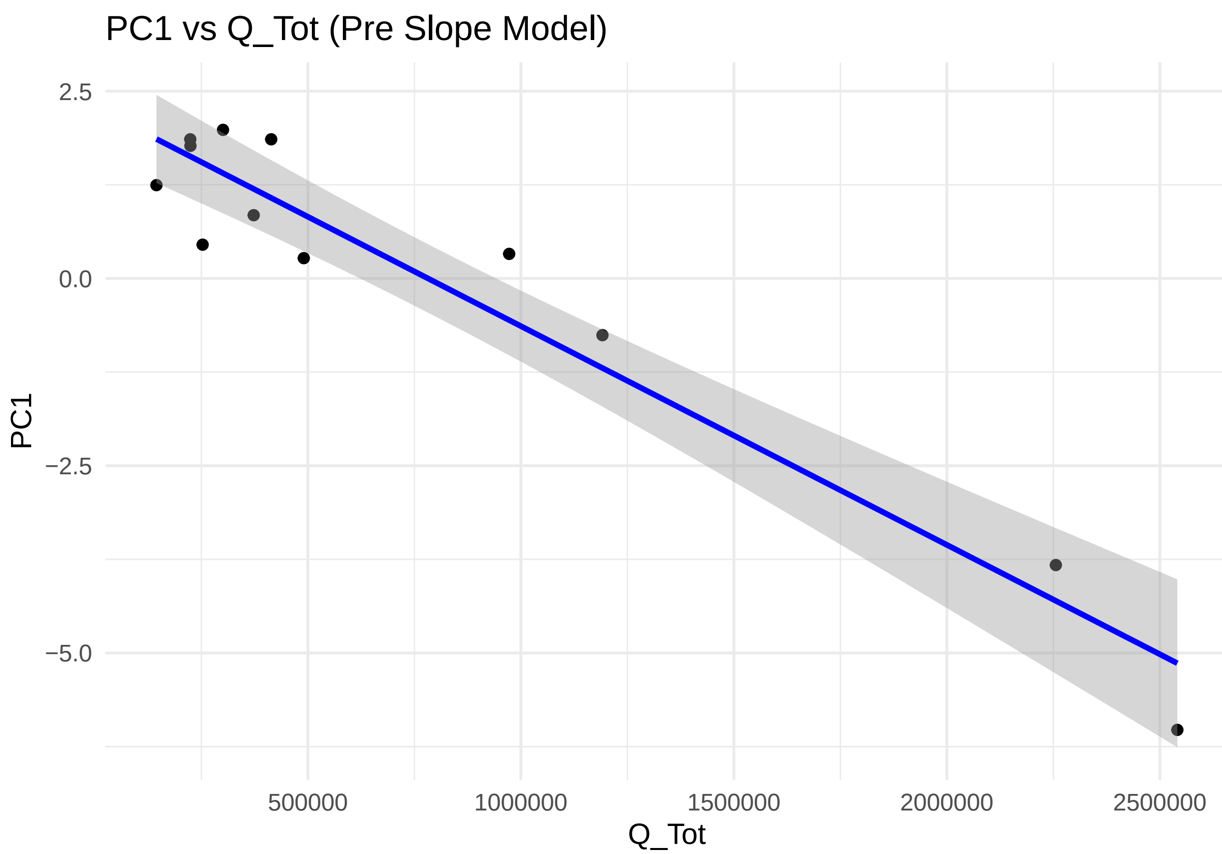

```r
# Post-slope
plot(stepAIC_model_post_slope_PC1, main = "Diagnostics for Post Slope Model PC1")
```



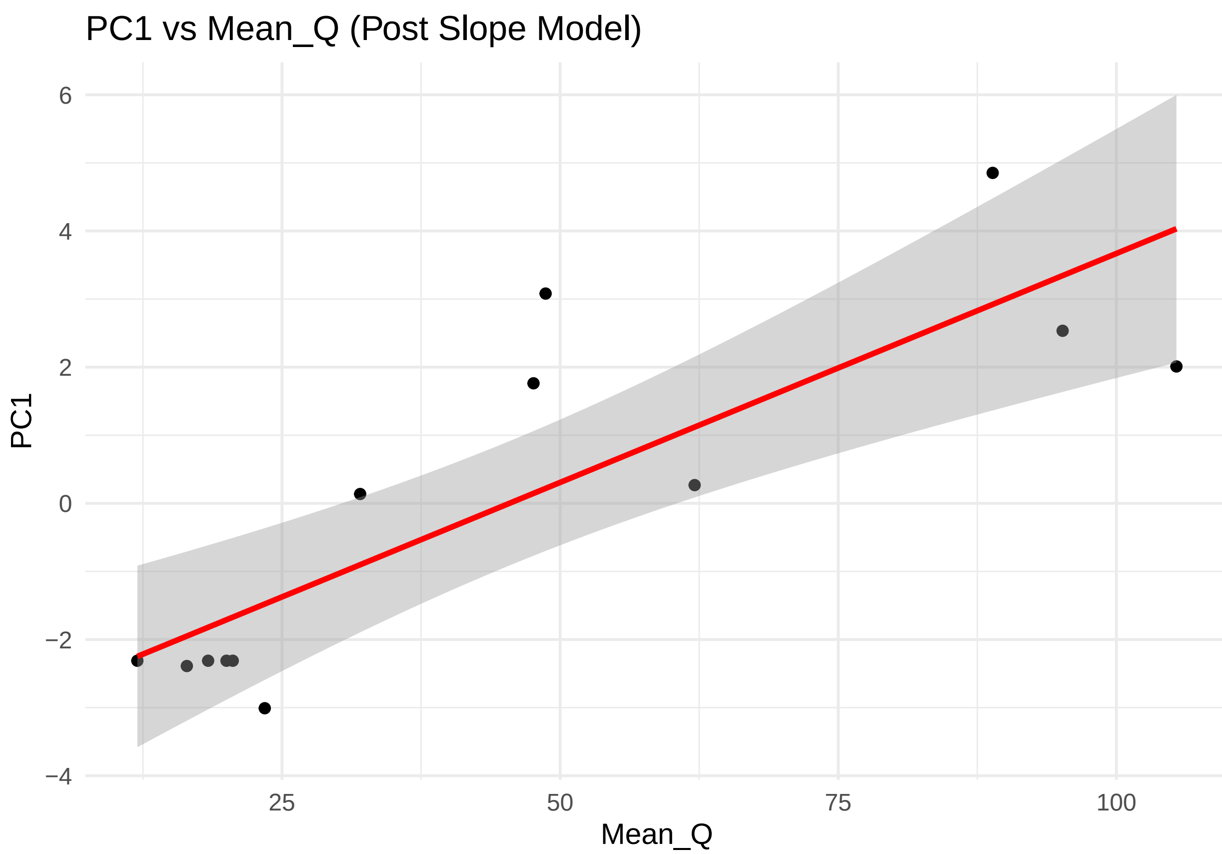

```r
# All-slope
plot(stepAIC_model_all_slope_PC1, main = "Diagnostics for All slope Model PC1")
```



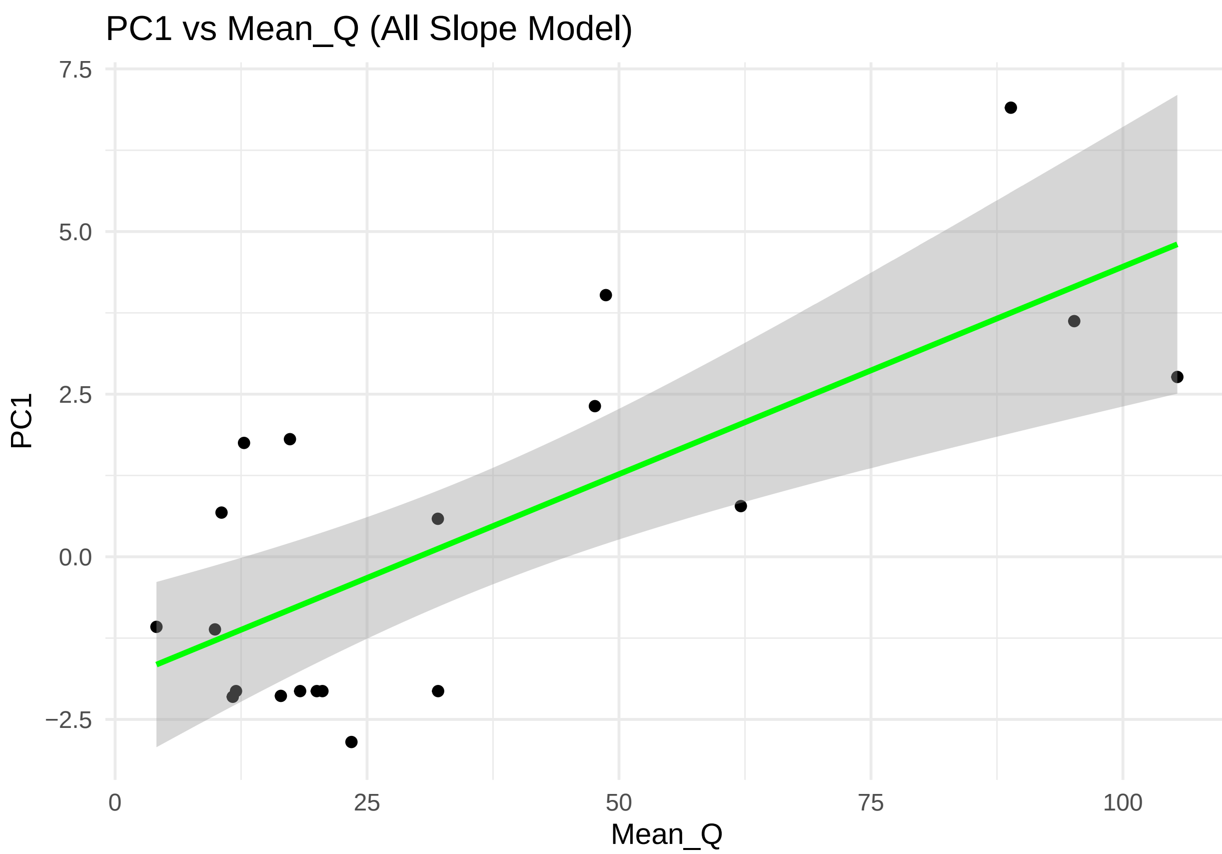

```r
# Extract significant predictors from pre-slope model
predictors_pre_PC1 <- names(coef(stepAIC_model_pre_slope_PC1))[-1]  # Exclude slope

# Example scatter plot for the first significant predictor
# in pre-slope model
ggplot(pre_slope_combined, aes_string(x = predictors_pre_PC1[1],
    y = "PC1")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "blue") + ggtitle(paste("PC1 vs", predictors_pre_PC1[1],
    "(Pre Slope Model)")) + theme_minimal()
```



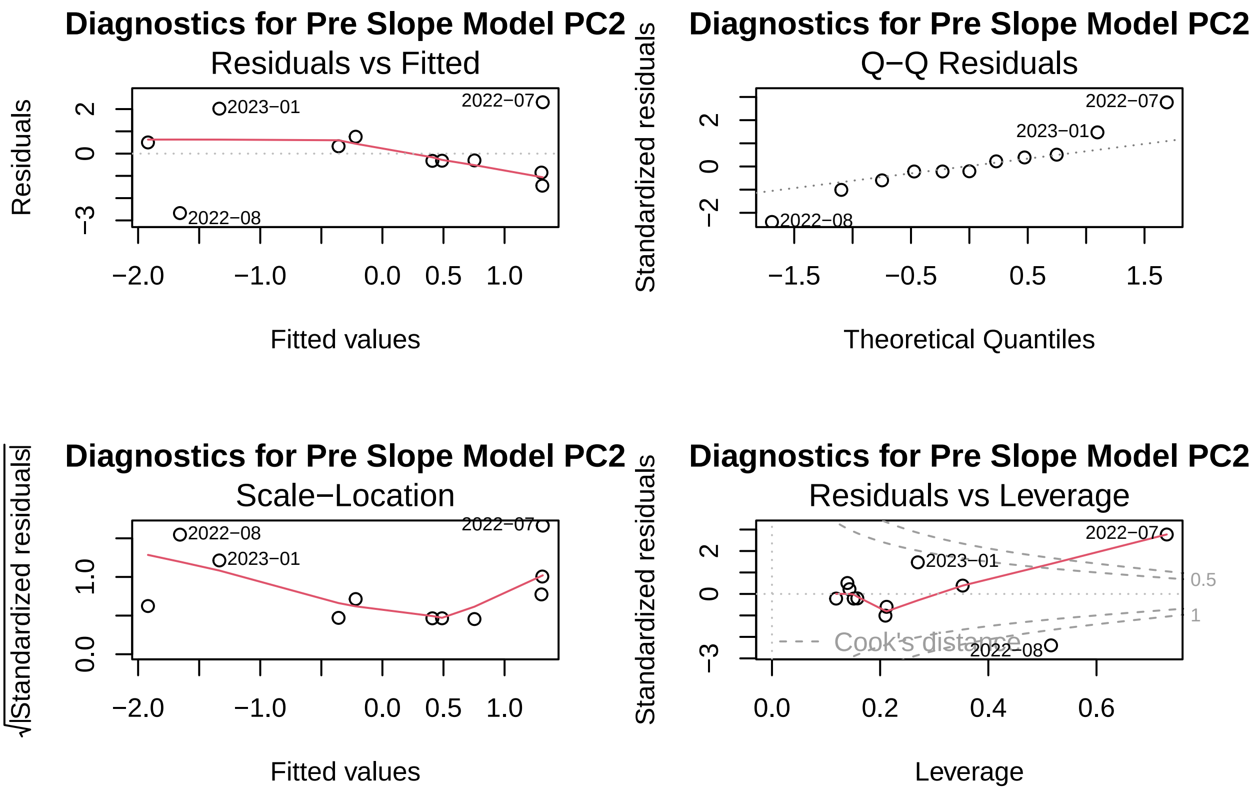

```r
# Repeat for post-slope model
predictors_post_PC1 <- names(coef(stepAIC_model_post_slope_PC1))[-1]
ggplot(post_slope_combined, aes_string(x = predictors_post_PC1[1],
    y = "PC1")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "red") + ggtitle(paste("PC1 vs", predictors_post_PC1[1],
    "(Post Slope Model)")) + theme_minimal()
```



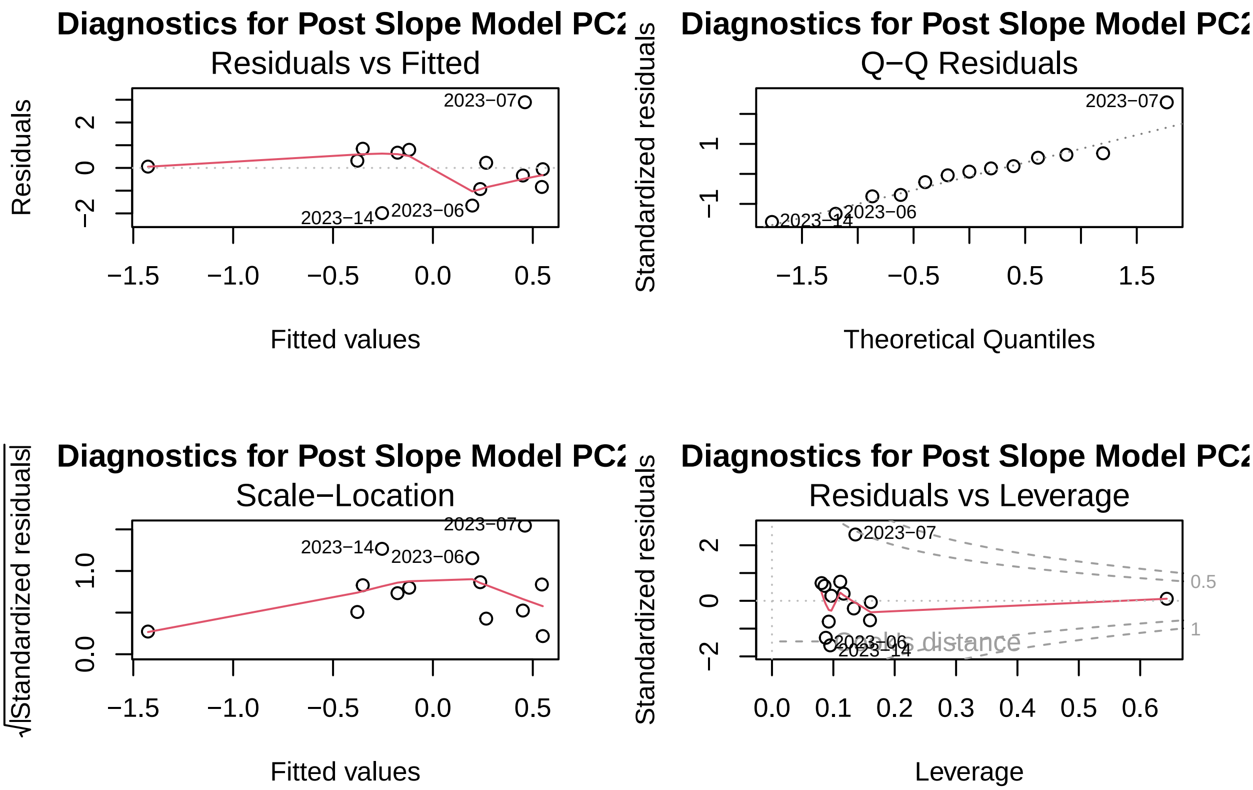

```
# Repeat for all-slope model
predictors_all_PC1 <- names(coef(stepAIC_model_all_slope_PC1))[-1]
ggplot(all_slope_combined, aes_string(x = predictors_all_PC1[1],
    y = "PC1")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "green") + ggtitle(paste("PC1 vs", predictors_all_PC1[1],
    "(All Slope Model)")) + theme_minimal()
```



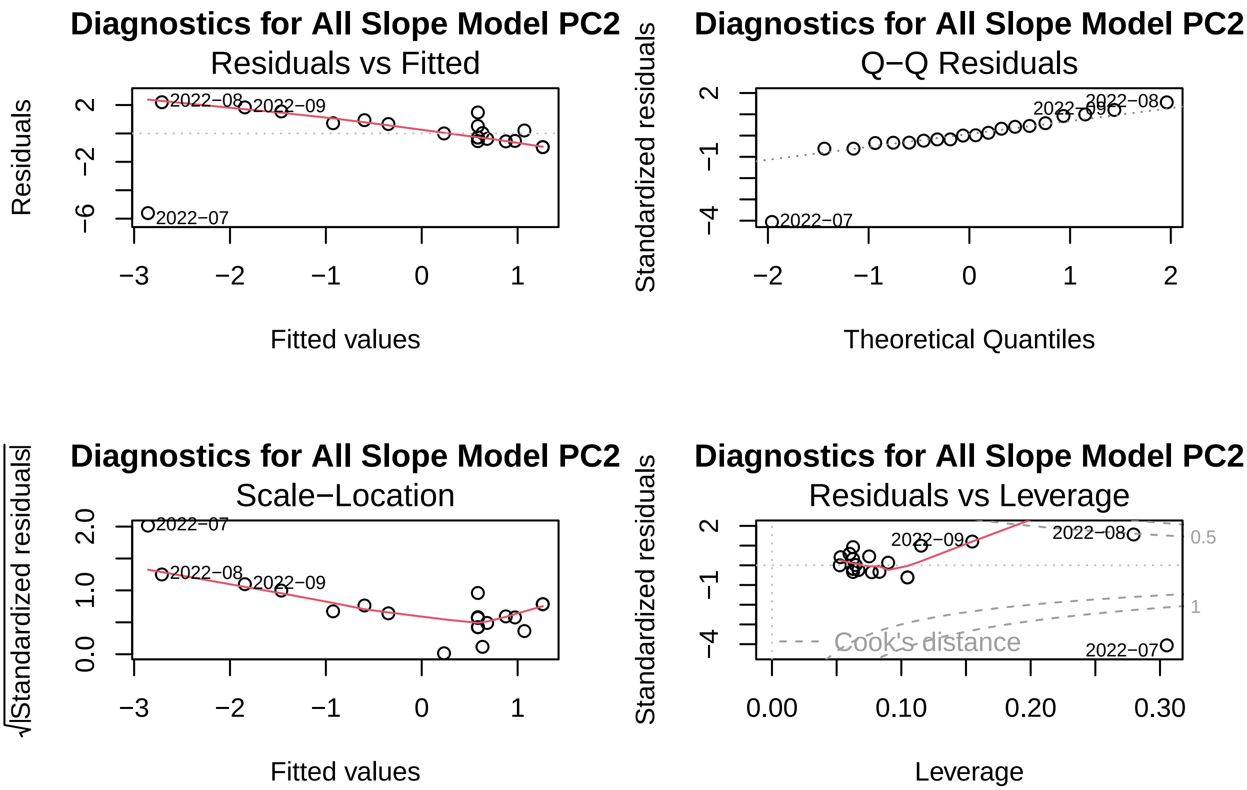

```r
## PC2 ##

# Diagnostic plots for the optimized models
par(mfrow = c(2, 2))  # Set layout for multiple plots

# Pre-slope
plot(stepAIC_model_pre_slope_PC2, main = "Diagnostics for Pre Slope Model PC2")
```



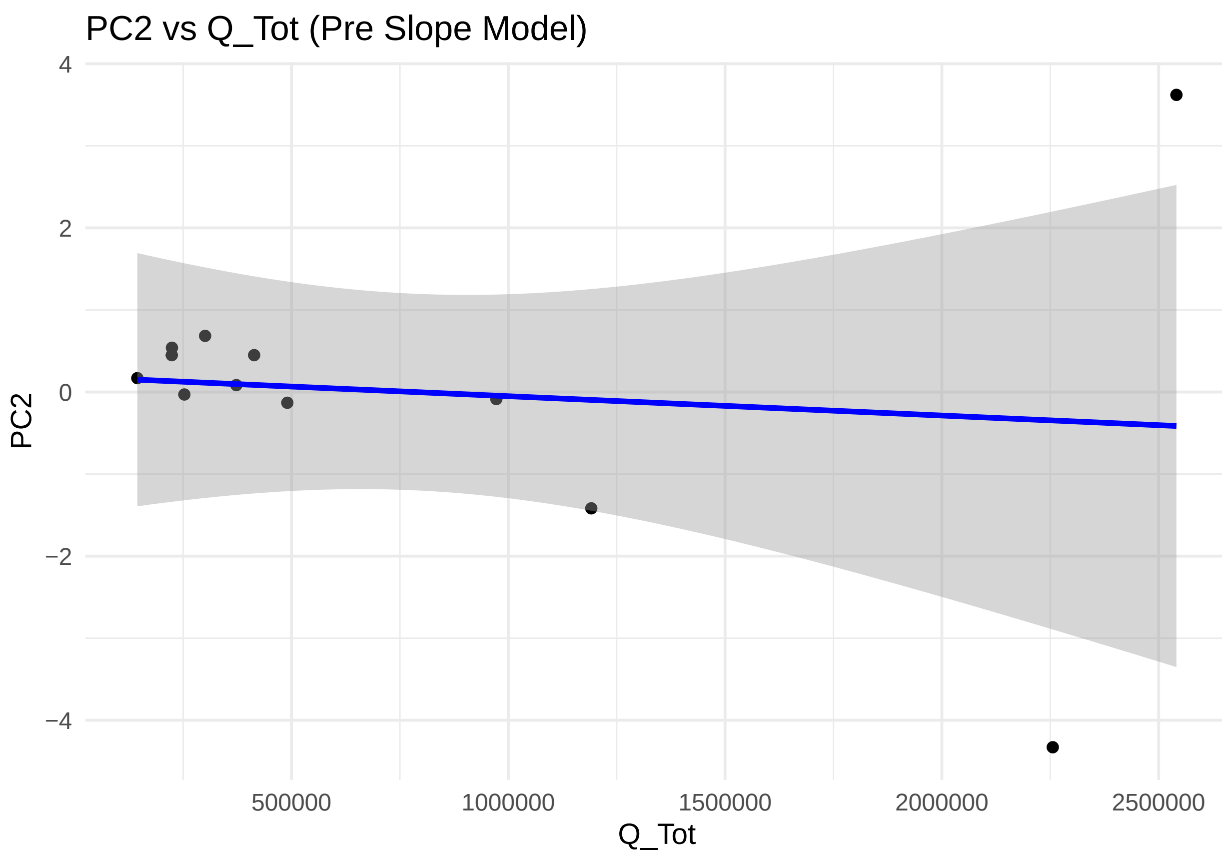

```r
# Post-slope
plot(stepAIC_model_post_slope_PC2, main = "Diagnostics for Post Slope Model PC2")
```



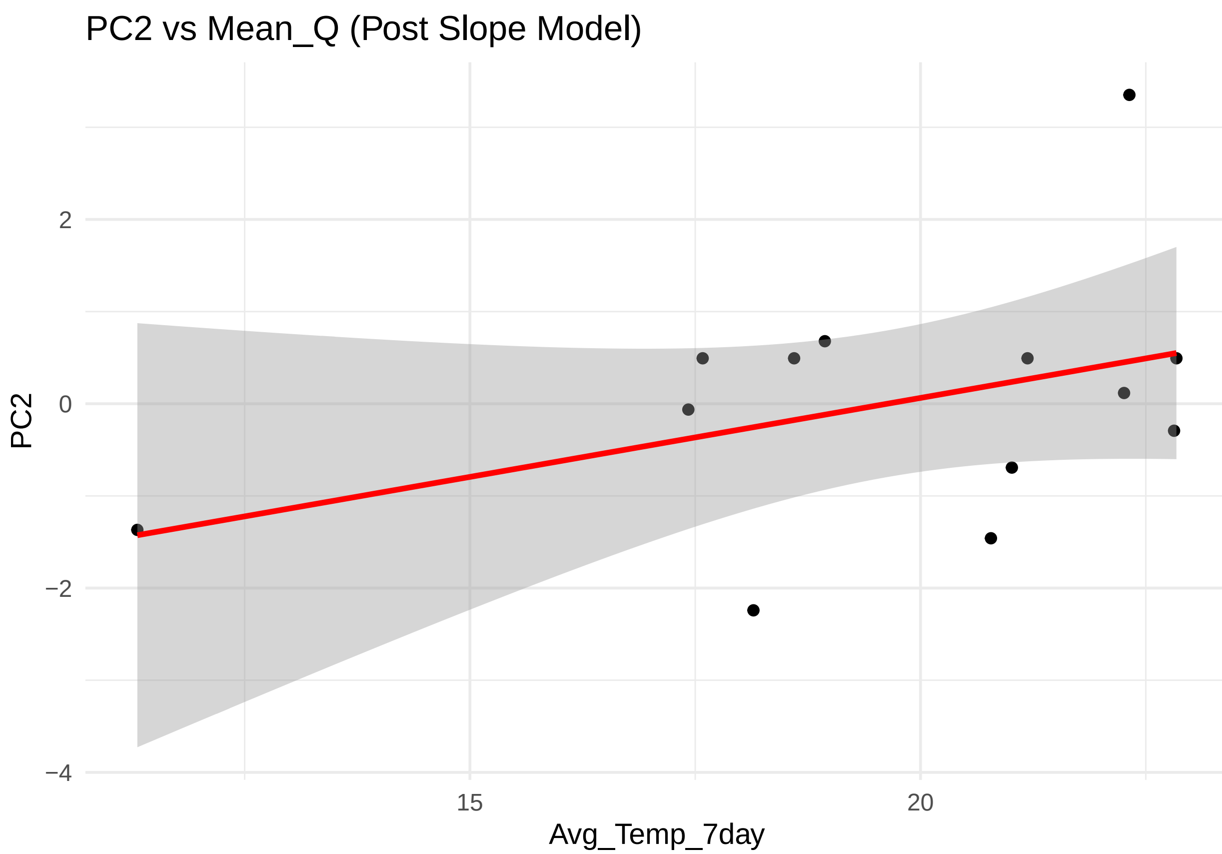

```r
# All-slope
plot(stepAIC_model_all_slope_PC2, main = "Diagnostics for All Slope Model PC2")
```



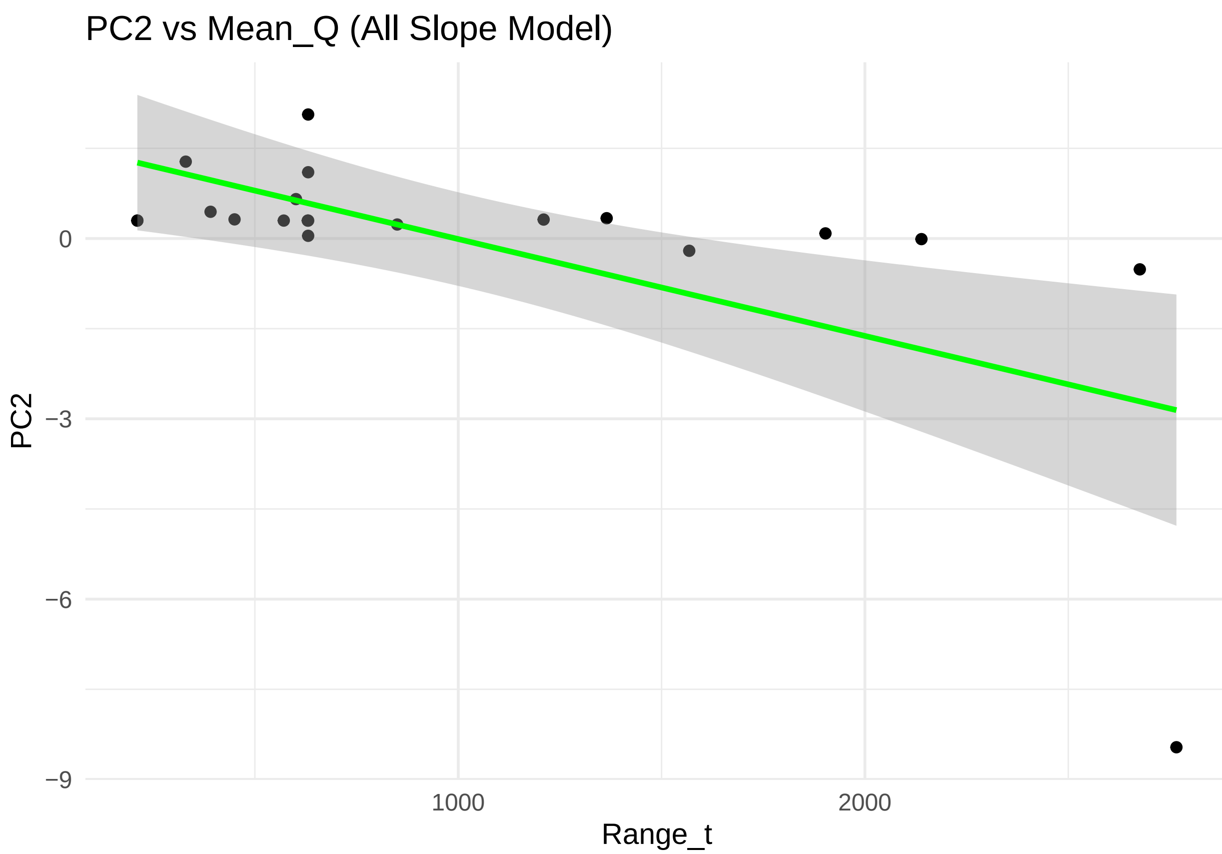

```
# Extract significant predictors from pre-slope model
predictors_pre_PC2 <- names(coef(stepAIC_model_pre_slope_PC2))[-1]  # Exclude slope

# Example scatter plot for the first significant predictor
# in pre-slope model
ggplot(pre_slope_combined, aes_string(x = predictors_pre_PC2[1],
    y = "PC2")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "blue") + ggtitle(paste("PC2 vs", predictors_pre_PC1[1],
    "(Pre Slope Model)")) + theme_minimal()
```



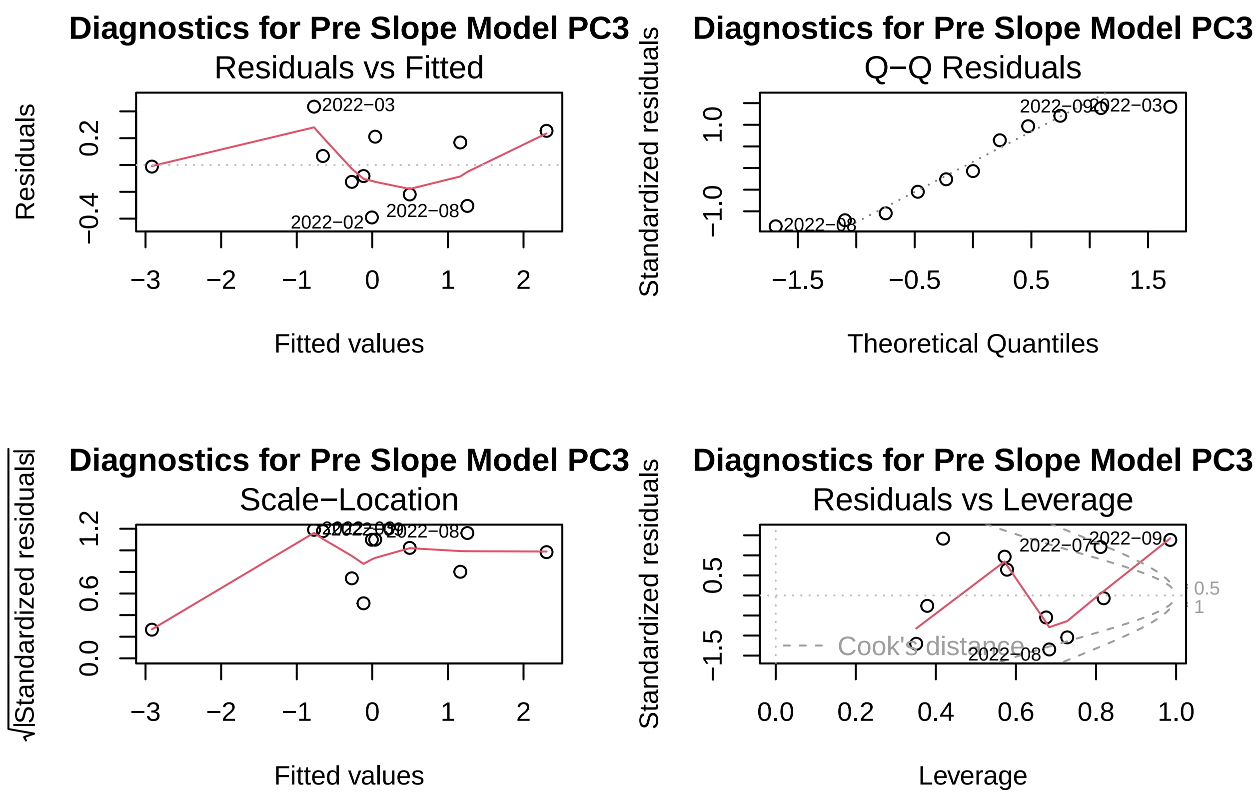

```
# Repeat for post-slope model
predictors_post_PC2 <- names(coef(stepAIC_model_post_slope_PC2))[-1]
ggplot(post_slope_combined, aes_string(x = predictors_post_PC2[1],
    y = "PC2")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "red") + ggtitle(paste("PC2 vs", predictors_post_PC1[1],
    "(Post Slope Model)")) + theme_minimal()
```



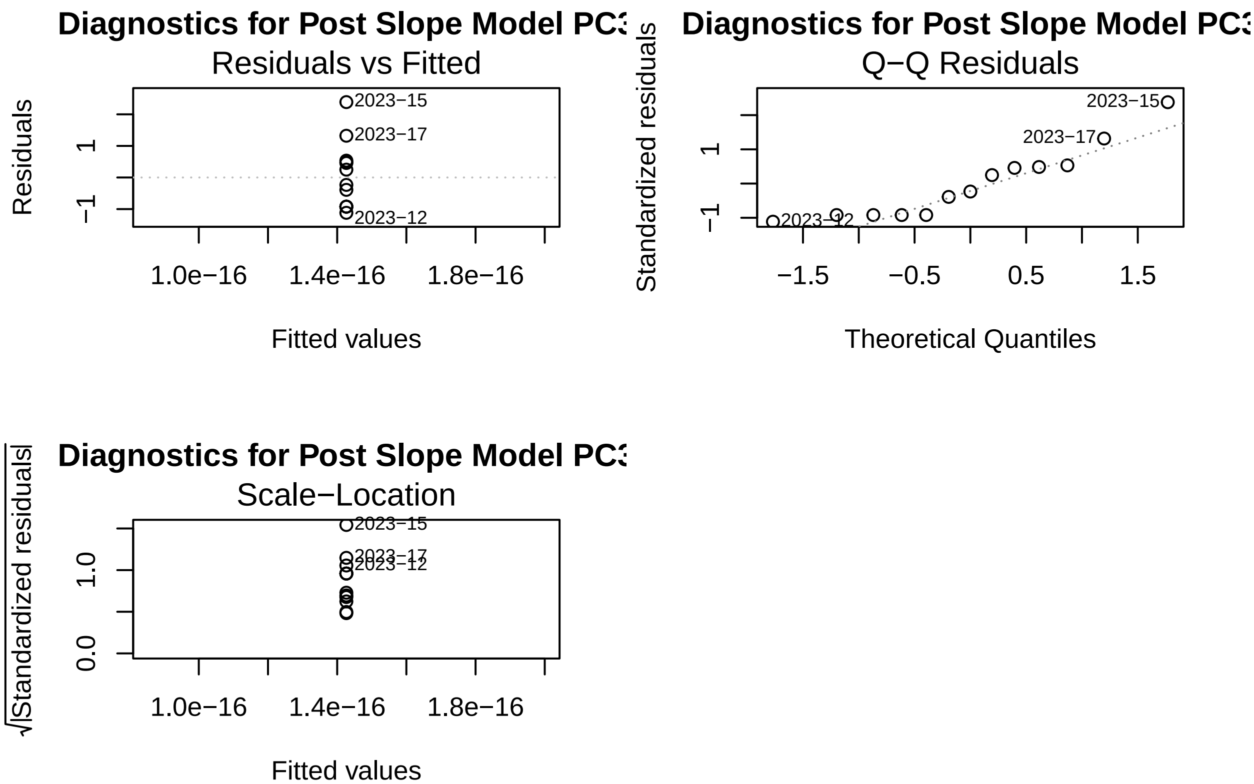

```r
# Repeat for all-slope model
predictors_all_PC2 <- names(coef(stepAIC_model_all_slope_PC2))[-1]
ggplot(all_slope_combined, aes_string(x = predictors_all_PC2[1],
    y = "PC2")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "green") + ggtitle(paste("PC2 vs", predictors_all_PC1[1],
    "(All Slope Model)")) + theme_minimal()
```



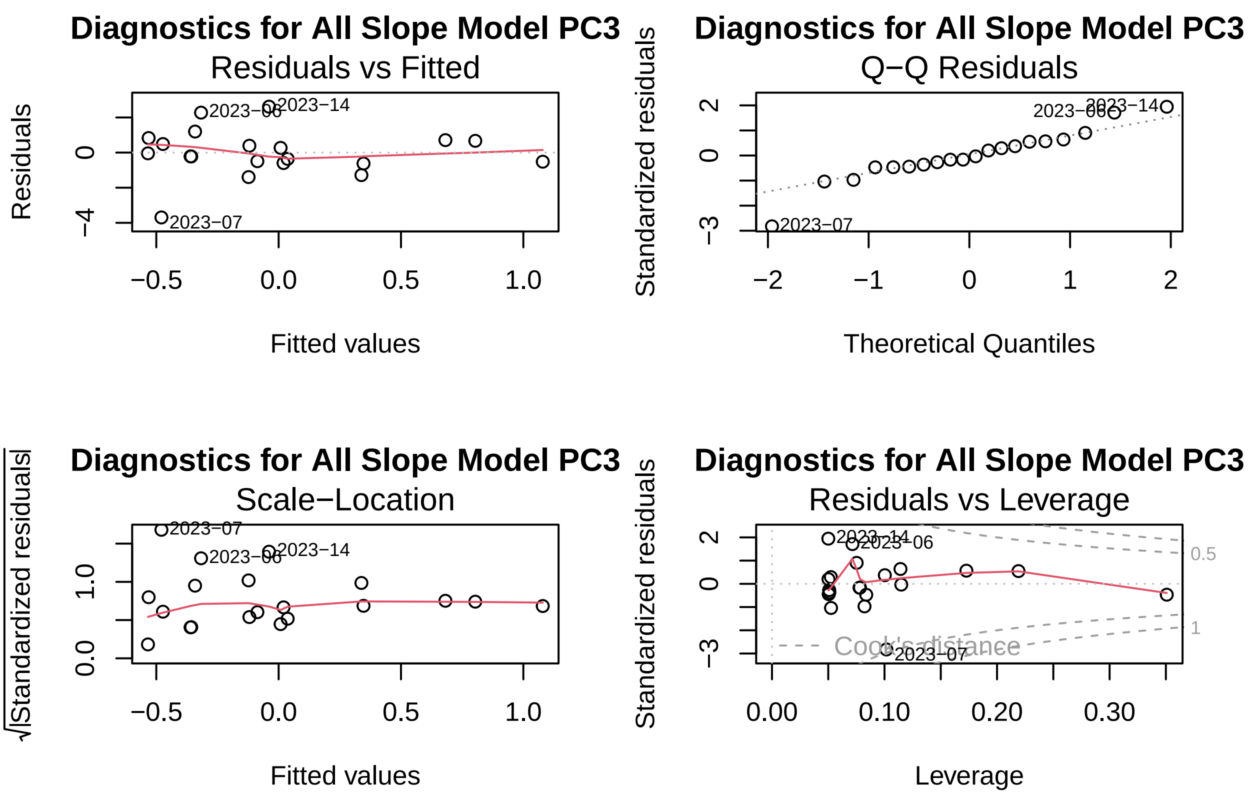

```r
## PC3 ##

# Diagnostic plots for the optimized models
par(mfrow = c(2, 2))  # Set layout for multiple plots

# Pre-slope
plot(stepAIC_model_pre_slope_PC3, main = "Diagnostics for Pre Slope Model PC3")
```



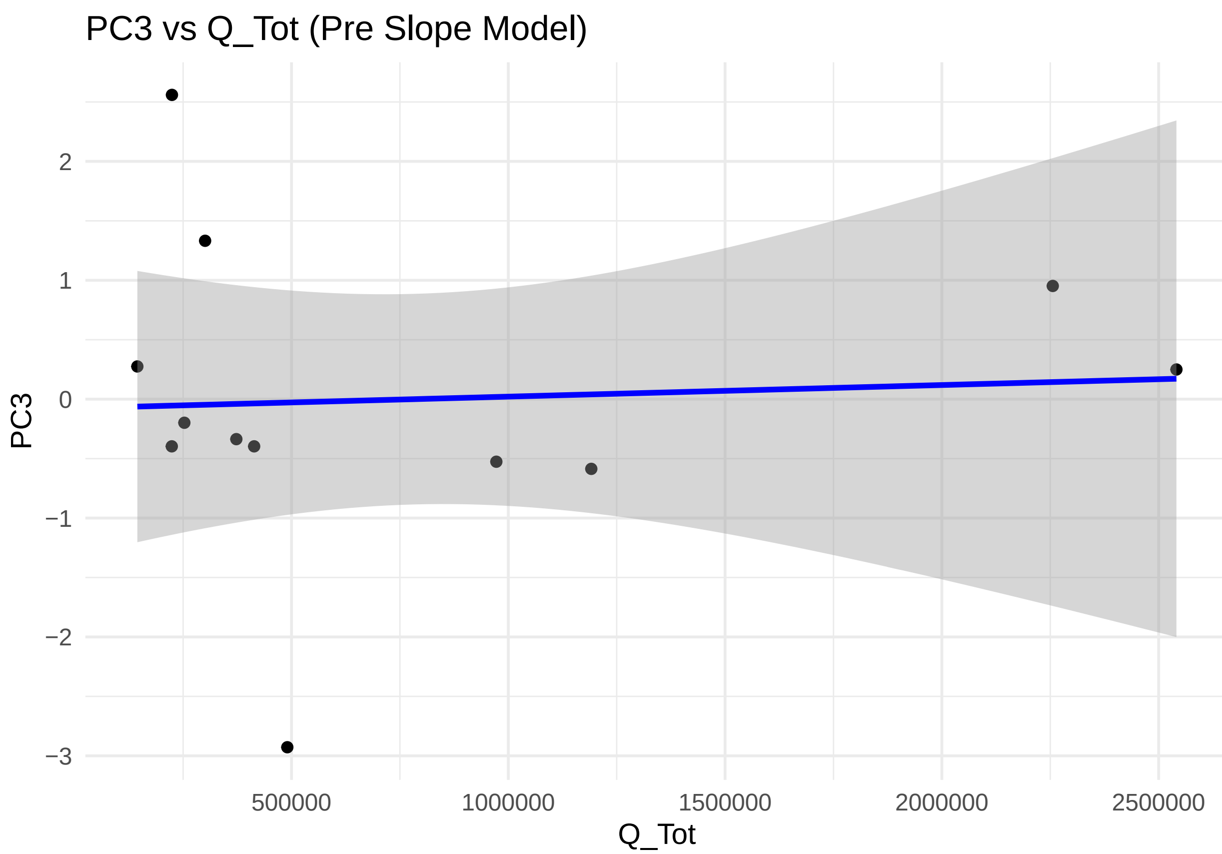

```
# Post-slope
plot(stepAIC_model_post_slope_PC3, main = "Diagnostics for Post Slope Model PC3")
```



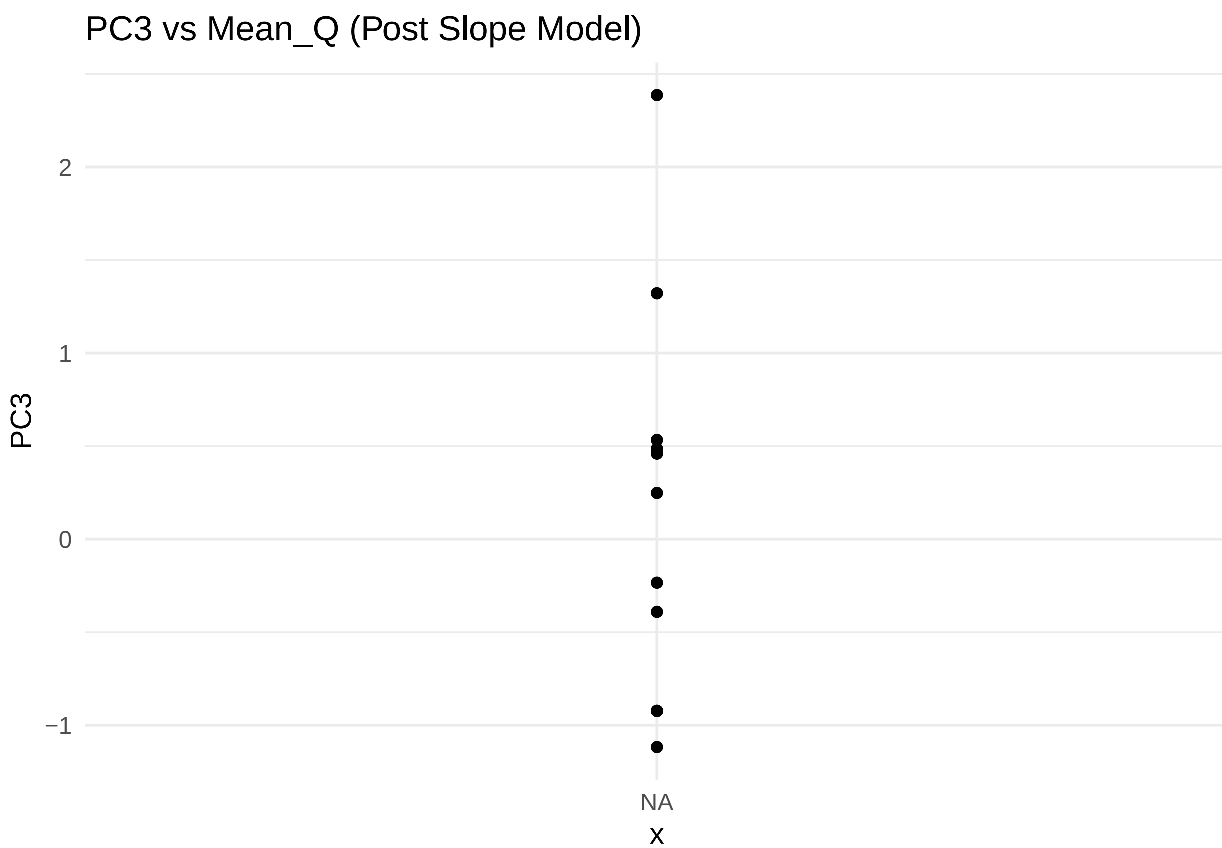

```
# All-slope
plot(stepAIC_model_all_slope_PC3, main = "Diagnostics for All Slope Model PC3")
```



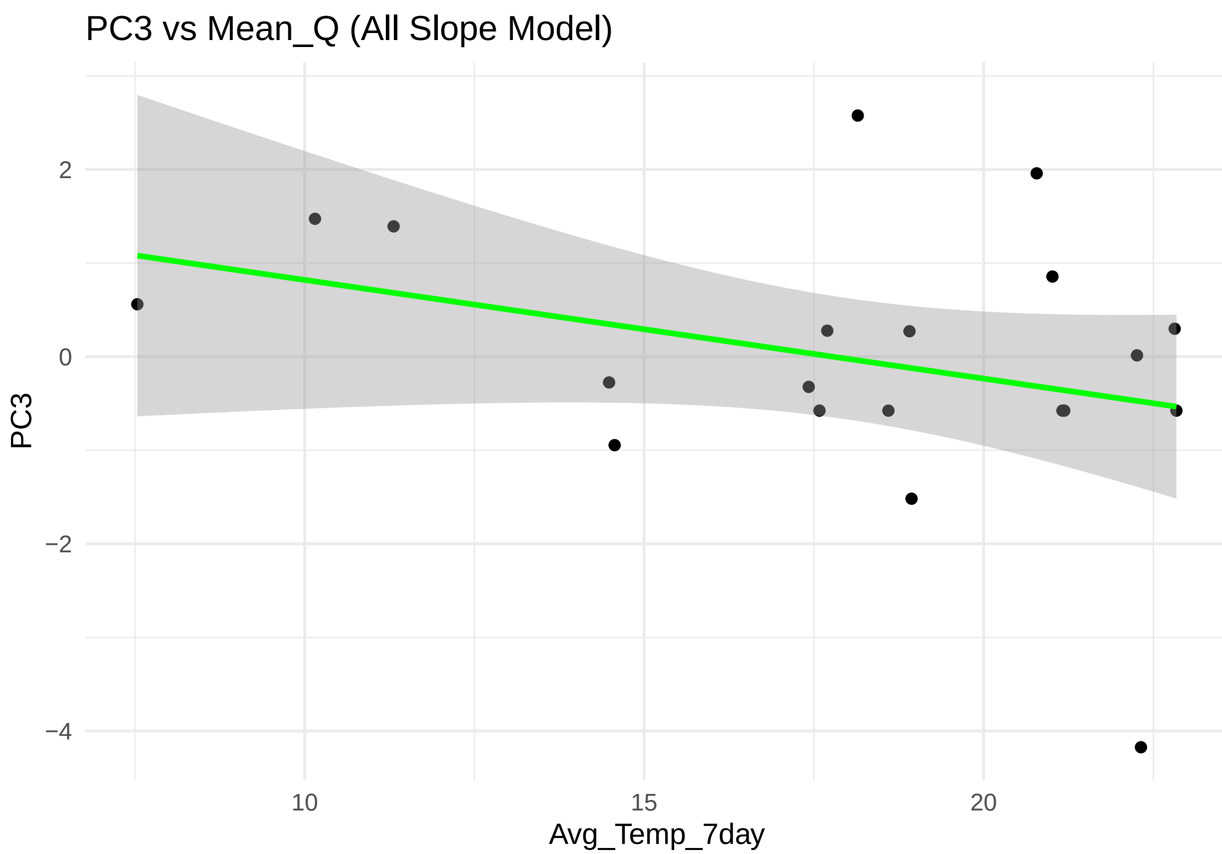

```r
# Extract significant predictors from pre-slope model
predictors_pre_PC3 <- names(coef(stepAIC_model_pre_slope_PC3))[-1]  # Exclude slope

# Example scatter plot for the first significant predictor
# in pre-slope model
ggplot(pre_slope_combined, aes_string(x = predictors_pre_PC3[1],
    y = "PC3")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "blue") + ggtitle(paste("PC3 vs", predictors_pre_PC1[1],
    "(Pre Slope Model)")) + theme_minimal()
```




```r
# Repeat for post-slope model
predictors_post_PC3 <- names(coef(stepAIC_model_post_slope_PC3))[-1]
ggplot(post_slope_combined, aes_string(x = predictors_post_PC3[1],
    y = "PC3")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "red") + ggtitle(paste("PC3 vs", predictors_post_PC1[1],
    "(Post Slope Model)")) + theme_minimal()
```




```r
# Repeat for all-slope model
predictors_all_PC3 <- names(coef(stepAIC_model_all_slope_PC3))[-1]
ggplot(all_slope_combined, aes_string(x = predictors_all_PC3[1],
    y = "PC3")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "green") + ggtitle(paste("PC3 vs", predictors_all_PC1[1],
    "(All Slope Model)")) + theme_minimal()
```




```r
## PC1 ## Perform stepwise regression for
## pre_intercept_combined
step_model_pre_intercept_PC1 <- lm(PC1 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = pre_intercept_combined)
stepAIC_model_pre_intercept_PC1 <- stepAIC(step_model_pre_intercept_PC1,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=-3.92
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=-3.92
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - Range_t          1    0.2046  2.3617 -4.9235
## - SW               1    0.2224  2.3795 -4.8409
## <none>                          2.1572 -3.9201
## - Mean_Q           1    0.8471  3.0042 -2.2766
## - Q_Tot            1    3.2998  5.4570  4.2890
## - Avg_Temp_7day    1    8.2442 10.4013 11.3844
## - Tot_Precip_7day  1    9.5768 11.7339 12.7105
##
## Step:  AIC=-4.92
## PC1 ~ Q_Tot + Mean_Q + SW + Avg_Temp_7day + Tot_Precip_7day
```

```
##
##                   Df Sum of Sq     RSS     AIC
## - SW               1    0.4111  2.7729 -5.1581
## <none>                          2.3617 -4.9235
## - Mean_Q           1    0.6511  3.0128 -4.2451
## + Range_t          1    0.2046  2.1572 -3.9201
## - Avg_Temp_7day    1   12.7007 15.0624 13.4574
## - Q_Tot            1   14.1834 16.5451 14.4902
## - Tot_Precip_7day  1   14.5357 16.8974 14.7219
##
## Step:  AIC=-5.16
## PC1 ~ Q_Tot + Mean_Q + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## <none>                          2.7729 -5.1581
## + SW               1    0.4111  2.3617 -4.9235
## + Range_t          1    0.3934  2.3795 -4.8409
## - Mean_Q           1    0.8401  3.6130 -4.2470
## - Tot_Precip_7day  1   14.9323 17.7052 13.2356
## - Avg_Temp_7day    1   18.8294 21.6023 15.4239
## - Q_Tot            1   19.3320 22.1049 15.6770
```

```r
summary(stepAIC_model_pre_intercept_PC1)
```

```
##
## Call:
## lm(formula = PC1 ~ Q_Tot + Mean_Q + Avg_Temp_7day + Tot_Precip_7day,
##     data = pre_intercept_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6520 -0.3847 -0.1524  0.3116  0.9504
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -8.925e+00  1.518e+00  -5.878 0.001074 **
## Q_Tot            2.570e-06  3.974e-07   6.468 0.000648 ***
## Mean_Q           4.702e-02  3.488e-02   1.348 0.226243
## Avg_Temp_7day    5.764e-01  9.030e-02   6.383 0.000695 ***
## Tot_Precip_7day -3.503e+00  6.163e-01  -5.684 0.001278 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6798 on 6 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.943,  Adjusted R-squared:  0.9051
## F-statistic: 24.83 on 4 and 6 DF,  p-value: 0.0007079
```

```r
# Perform stepwise regression for post_intercept_combined
step_model_post_intercept_PC1 <- lm(PC1 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = post_intercept_combined)
stepAIC_model_post_intercept_PC1 <- stepAIC(step_model_post_intercept_PC1,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=6.89
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=6.89
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - Q_Tot            1    0.0508  7.5759  4.9804
## - Range_t          1    0.1019  7.6270  5.0677
## - Tot_Precip_7day  1    0.1049  7.6300  5.0729
## - SW               1    0.4031  7.9282  5.5712
## - Avg_Temp_7day    1    0.7178  8.2429  6.0772
## <none>                          7.5251  6.8928
## - Mean_Q           1    5.4382 12.9633 11.9632
##
## Step:  AIC=4.98
## PC1 ~ Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS     AIC
## - Range_t          1    0.0692  7.645  3.0986
## - Tot_Precip_7day  1    0.1407  7.717  3.2195
## - SW               1    0.6603  8.236  4.0667
## - Avg_Temp_7day    1    0.6683  8.244  4.0794
## <none>                          7.576  4.9804
## + Q_Tot            1    0.0508  7.525  6.8928
## - Mean_Q           1   28.7358 36.312 23.3535
##
## Step:  AIC=3.1
## PC1 ~ Mean_Q + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS     AIC
## - Tot_Precip_7day  1    0.4738  8.119  1.8803
## - Avg_Temp_7day    1    0.6013  8.246  2.0828
## <none>                          7.645  3.0986
## - SW               1    1.2976  8.943  3.1367
## + Range_t          1    0.0692  7.576  4.9804
## + Q_Tot            1    0.0182  7.627  5.0677
## - Mean_Q           1   28.7066 36.352 21.3678
##
## Step:  AIC=1.88
## PC1 ~ Mean_Q + SW + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS     AIC
## - Avg_Temp_7day    1    0.6434  8.762  0.8717
## - SW               1    0.8426  8.962  1.1640
## <none>                          8.119  1.8803
## + Tot_Precip_7day  1    0.4738  7.645  3.0986
## + Range_t          1    0.4024  7.717  3.2195
## + Q_Tot            1    0.2199  7.899  3.5233
## - Mean_Q           1   29.8221 37.941 19.9241
##
## Step:  AIC=0.87
```

```
## PC1 ~ Mean_Q + SW
##
##                   Df Sum of Sq    RSS     AIC
## - SW               1    0.2585  9.021 -0.7504
## <none>                          8.762  0.8717
## + Avg_Temp_7day    1    0.6434  8.119  1.8803
## + Tot_Precip_7day  1    0.5159  8.246  2.0828
## + Range_t          1    0.1469  8.615  2.6519
## + Q_Tot            1    0.0933  8.669  2.7325
## - Mean_Q           1   29.8312 38.594 18.1458
##
## Step:  AIC=-0.75
## PC1 ~ Mean_Q
##
##                   Df Sum of Sq    RSS     AIC
## <none>                          9.021 -0.7504
## + Range_t          1     0.339  8.682  0.7518
## + SW               1     0.258  8.762  0.8717
## + Q_Tot            1     0.164  8.857  1.0117
## + Avg_Temp_7day    1     0.059  8.962  1.1640
## + Tot_Precip_7day  1     0.054  8.967  1.1716
## - Mean_Q           1    49.698 58.719 21.6015
```

```r
summary(stepAIC_model_post_intercept_PC1)
```

```
##
## Call:
## lm(formula = PC1 ~ Mean_Q, data = post_intercept_combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64187 -0.50641 -0.00316  0.29510  1.56431
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.823835   0.441207  -6.400 5.08e-05 ***
## Mean_Q       0.062146   0.007983   7.785 8.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9056 on 11 degrees of freedom
## Multiple R-squared:  0.8464, Adjusted R-squared:  0.8324
## F-statistic:  60.6 on 1 and 11 DF,  p-value: 8.458e-06
```

```r
# Perform stepwise regression for all_intercept_combined
step_model_all_intercept_PC1 <- lm(PC1 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = all_intercept_combined)
stepAIC_model_all_intercept_PC1 <- stepAIC(step_model_all_intercept_PC1,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=7.3
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
```

```
##
##                  Df Sum of Sq    RSS     AIC
## - Avg_Temp_7day   1    0.2970 13.241  5.7522
## <none>                        12.944  7.2986
## - Q_Tot           1    1.5100 14.454  7.5053
## - Treatment       1    1.7184 14.663  7.7917
## - Tot_Precip_7day 1    2.1393 15.084  8.3577
## - SW              1    4.5757 17.520 11.3524
## - Range_t         1    9.4350 22.379 16.2482
## - Mean_Q          1   27.1379 40.082 27.9040
##
## Step:  AIC=5.75
## PC1 ~ Q_Tot + Mean_Q + Range_t + SW + Tot_Precip_7day + Treatment
##
##                  Df Sum of Sq    RSS     AIC
## - Q_Tot           1    1.2898 14.531  5.6113
## <none>                        13.241  5.7522
## - Treatment       1    1.4463 14.688  5.8255
## + Avg_Temp_7day   1    0.2970 12.944  7.2986
## - Tot_Precip_7day 1    2.7215 15.963  7.4907
## - SW              1    4.7978 18.039  9.9362
## - Range_t         1    9.3174 22.559 14.4078
## - Mean_Q          1   26.8565 40.098 25.9118
##
## Step:  AIC=5.61
## PC1 ~ Mean_Q + Range_t + SW + Tot_Precip_7day + Treatment
##
##                  Df Sum of Sq    RSS     AIC
## - Treatment       1     0.735 15.266  4.5976
## <none>                        14.531  5.6113
## + Q_Tot           1     1.290 13.241  5.7522
## + Avg_Temp_7day   1     0.077 14.454  7.5053
## - Tot_Precip_7day 1     4.966 19.497  9.4907
## - SW              1     5.416 19.947  9.9473
## - Range_t         1    11.897 26.428 15.5742
## - Mean_Q          1    40.744 55.275 30.3317
##
## Step:  AIC=4.6
## PC1 ~ Mean_Q + Range_t + SW + Tot_Precip_7day
##
##                  Df Sum of Sq    RSS    AIC
## <none>                        15.266  4.598
## + Treatment       1     0.735 14.531  5.611
## + Q_Tot           1     0.578 14.688  5.826
## + Avg_Temp_7day   1     0.028 15.237  6.560
## - Tot_Precip_7day 1     4.465 19.731  7.729
## - SW              1     4.714 19.980  7.980
## - Range_t         1    12.677 27.943 14.688
## - Mean_Q          1    53.874 69.140 32.808
```

```r
summary(stepAIC_model_all_intercept_PC1)
```

```
##
## Call:
```

```
## lm(formula = PC1 ~ Mean_Q + Range_t + SW + Tot_Precip_7day, data = all_intercept_combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94219 -0.38020 -0.06864  0.48877  1.99560
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.4434368  1.2439402  -4.376 0.000542 ***
## Mean_Q           0.0748781  0.0102915   7.276 2.72e-06 ***
## Range_t          0.0012703  0.0003599   3.529 0.003035 **
## SW               3.2678655  1.5183952   2.152 0.048075 *
## Tot_Precip_7day -0.6265272  0.2991240  -2.095 0.053603 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.009 on 15 degrees of freedom
## Multiple R-squared:  0.8348, Adjusted R-squared:  0.7907
## F-statistic: 18.95 on 4 and 15 DF,  p-value: 9.91e-06
```

```
## PC2 ## Perform stepwise regression for
## pre_intercept_combined
step_model_pre_intercept_PC2 <- lm(PC2 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = pre_intercept_combined)
stepAIC_model_pre_intercept_PC2 <- stepAIC(step_model_pre_intercept_PC2,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=-4.73
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=-4.73
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - SW               1    0.0000  2.0040 -6.7300
## - Range_t          1    0.0004  2.0044 -6.7280
## - Q_Tot            1    0.0059  2.0099 -6.6978
## <none>                          2.0040 -4.7300
## - Avg_Temp_7day    1    2.3289  4.3329  1.7518
## - Tot_Precip_7day  1    6.2952  8.2992  8.9010
## - Mean_Q           1   15.8526 17.8567 17.3293
##
## Step:  AIC=-6.73
## PC2 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - Range_t          1    0.0004  2.0045 -8.7278
## - Q_Tot            1    0.0059  2.0099 -8.6978
## <none>                          2.0040 -6.7300
## + SW               1    0.0000  2.0040 -4.7300
## - Avg_Temp_7day    1    4.9429  6.9469  4.9444
## - Tot_Precip_7day  1    6.3537  8.3577  6.9782
```

```
## - Mean_Q           1   16.9698 18.9739 15.9968
##
## Step:  AIC=-8.73
## PC2 ~ Q_Tot + Mean_Q + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS      AIC
## - Q_Tot            1    0.0126  2.0170 -10.6589
## <none>                          2.0045  -8.7278
## + Range_t          1    0.0004  2.0040  -6.7300
## + SW               1    0.0000  2.0044  -6.7280
## - Avg_Temp_7day    1    5.9386  7.9431   4.4185
## - Tot_Precip_7day  1    8.2870 10.2914   7.2676
## - Mean_Q           1   19.3755 21.3800  15.3102
##
## Step:  AIC=-10.66
## PC2 ~ Mean_Q + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS      AIC
## <none>                          2.0170 -10.6589
## + Q_Tot            1    0.0126  2.0045  -8.7278
## + Range_t          1    0.0071  2.0099  -8.6978
## + SW               1    0.0015  2.0155  -8.6672
## - Tot_Precip_7day  1    8.4533 10.4703   5.4571
## - Avg_Temp_7day    1    8.5452 10.5623   5.5533
## - Mean_Q           1   22.2550 24.2721  14.7057
```

```r
summary(stepAIC_model_pre_intercept_PC2)
```

```
##
## Call:
## lm(formula = PC2 ~ Mean_Q + Avg_Temp_7day + Tot_Precip_7day,
##     data = pre_intercept_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7810 -0.2258  0.0322  0.3610  0.6552
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.61069    0.83502   6.719 0.000273 ***
## Mean_Q          -0.22777    0.02592  -8.788 4.98e-05 ***
## Avg_Temp_7day   -0.31497    0.05784  -5.446 0.000960 ***
## Tot_Precip_7day  2.62081    0.48387   5.416 0.000991 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5368 on 7 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9358, Adjusted R-squared:  0.9083
## F-statistic:    34 on 3 and 7 DF,  p-value: 0.0001523
```

```r
# Perform stepwise regression for post_intercept_combined
step_model_post_intercept_PC2 <- lm(PC2 ~ Q_Tot + Mean_Q + Range_t +
```

```r
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = post_intercept_combined)
stepAIC_model_post_intercept_PC2 <- stepAIC(step_model_post_intercept_PC2,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=15.96
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=15.96
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1    0.0157 15.135 13.977
## - Tot_Precip_7day  1    0.5425 15.662 14.422
## - Avg_Temp_7day    1    1.8993 17.019 15.502
## <none>                         15.120 15.964
## - Mean_Q           1    3.5474 18.667 16.704
## - Q_Tot            1    6.7773 21.897 18.778
## - Range_t          1    7.7641 22.884 19.351
##
## Step:  AIC=13.98
## PC2 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    1.2361 16.372 12.998
## - Avg_Temp_7day    1    1.9638 17.099 13.563
## <none>                         15.135 13.977
## - Mean_Q           1    3.7462 18.882 14.852
## + SW               1    0.0157 15.120 15.964
## - Q_Tot            1    7.9955 23.131 17.491
## - Range_t          1   11.6165 26.752 19.382
##
## Step:  AIC=13
## PC2 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         16.372 12.998
## + Tot_Precip_7day  1    1.2361 15.135 13.977
## - Mean_Q           1    4.2272 20.599 13.984
## + SW               1    0.7093 15.662 14.422
## - Avg_Temp_7day    1    5.1443 21.516 14.550
## - Q_Tot            1    7.2599 23.631 15.769
## - Range_t          1   12.1651 28.537 18.221
```

```r
summary(stepAIC_model_post_intercept_PC2)
```

```
##
## Call:
## lm(formula = PC2 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day,
##     data = post_intercept_combined)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5030 -0.6074  0.1819  0.9351  1.5566
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -9.563e+00  4.676e+00  -2.045   0.0751 .
## Q_Tot         -2.132e-06  1.132e-06  -1.884   0.0964 .
## Mean_Q         4.779e-02  3.325e-02   1.437   0.1886
## Range_t        6.928e-03  2.842e-03   2.438   0.0407 *
## Avg_Temp_7day  2.947e-01  1.859e-01   1.585   0.1515
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.431 on 8 degrees of freedom
## Multiple R-squared:  0.4512, Adjusted R-squared:  0.1768
## F-statistic: 1.644 on 4 and 8 DF,  p-value: 0.2545
```

```
# Perform stepwise regression for all_intercept_combined
step_model_all_intercept_PC2 <- lm(PC2 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = all_intercept_combined)
stepAIC_model_all_intercept_PC2 <- stepAIC(step_model_all_intercept_PC2,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=29.04
## PC2 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Mean_Q           1    0.0488 38.437 27.066
## - Range_t          1    0.3292 38.717 27.211
## - Avg_Temp_7day    1    0.3853 38.773 27.240
## - Q_Tot            1    0.5116 38.900 27.305
## - SW               1    0.7656 39.154 27.435
## - Treatment        1    2.2624 40.651 28.186
## <none>                         38.388 29.040
## - Tot_Precip_7day  1    5.3721 43.760 29.660
##
## Step:  AIC=27.07
## PC2 ~ Q_Tot + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Range_t          1    0.2868 38.724 25.214
## - Avg_Temp_7day    1    0.3568 38.794 25.250
## - Q_Tot            1    0.6664 39.103 25.410
## - SW               1    0.7479 39.185 25.451
## - Treatment        1    2.3194 40.756 26.238
## <none>                         38.437 27.066
## - Tot_Precip_7day  1    5.3604 43.797 27.677
## + Mean_Q           1    0.0488 38.388 29.040
##
## Step:  AIC=25.21
## PC2 ~ Q_Tot + SW + Avg_Temp_7day + Tot_Precip_7day + Treatment
```

```
##
##                   Df Sum of Sq    RSS    AIC
## - Avg_Temp_7day    1    0.3762 39.100 23.408
## - Q_Tot            1    0.3799 39.104 23.410
## - SW               1    1.2441 39.968 23.847
## <none>                         38.724 25.214
## - Treatment        1    5.0107 43.735 25.648
## - Tot_Precip_7day  1    5.0740 43.798 25.677
## + Range_t          1    0.2868 38.437 27.066
## + Mean_Q           1    0.0064 38.717 27.211
##
## Step:  AIC=23.41
## PC2 ~ Q_Tot + SW + Tot_Precip_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Q_Tot            1    0.2110 39.311 21.515
## - SW               1    3.4080 42.508 23.079
## <none>                         39.100 23.408
## - Tot_Precip_7day  1    4.6984 43.798 23.677
## + Avg_Temp_7day    1    0.3762 38.724 25.214
## + Range_t          1    0.3063 38.794 25.250
## + Mean_Q           1    0.0245 39.076 25.395
## - Treatment        1   12.0676 51.168 26.788
##
## Step:  AIC=21.52
## PC2 ~ SW + Tot_Precip_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1    3.3994 42.710 21.174
## <none>                         39.311 21.515
## - Tot_Precip_7day  1    5.3968 44.708 22.088
## + Q_Tot            1    0.2110 39.100 23.408
## + Avg_Temp_7day    1    0.2073 39.104 23.410
## + Mean_Q           1    0.1674 39.144 23.430
## + Range_t          1    0.0064 39.305 23.512
## - Treatment        1   12.3540 51.665 24.981
##
## Step:  AIC=21.17
## PC2 ~ Tot_Precip_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    2.1265 44.837 20.146
## <none>                         42.710 21.174
## + SW               1    3.3994 39.311 21.515
## + Avg_Temp_7day    1    2.5689 40.142 21.934
## + Range_t          1    1.3716 41.339 22.521
## + Q_Tot            1    0.2024 42.508 23.079
## + Mean_Q           1    0.1775 42.533 23.091
## - Treatment        1   10.5495 53.260 23.589
##
## Step:  AIC=20.15
## PC2 ~ Treatment
##
##                   Df Sum of Sq    RSS    AIC
```

```
## <none>                        44.837 20.146
## + Tot_Precip_7day  1    2.1265 42.710 21.174
## - Treatment        1    8.5563 53.393 21.639
## + Q_Tot            1    0.9450 43.892 21.720
## + Range_t          1    0.5811 44.256 21.885
## + Mean_Q           1    0.3238 44.513 22.001
## + Avg_Temp_7day    1    0.1593 44.678 22.075
## + SW               1    0.1291 44.708 22.088
```

```r
summary(stepAIC_model_all_intercept_PC2)
```

```
##
## Call:
## lm(formula = PC2 ~ Treatment, data = all_intercept_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8815 -0.7577 -0.2728  0.3941  5.7862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.8272     0.5965  -1.387   0.1825
## Treatment     1.3713     0.7399   1.853   0.0803 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.578 on 18 degrees of freedom
## Multiple R-squared:  0.1603, Adjusted R-squared:  0.1136
## F-statistic: 3.435 on 1 and 18 DF,  p-value: 0.0803
```

```r
## PC3 ## Perform stepwise regression for
## pre_intercept_combined
step_model_pre_intercept_PC3 <- lm(PC3 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = pre_intercept_combined)
stepAIC_model_pre_intercept_PC3 <- stepAIC(step_model_pre_intercept_PC3,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=14.55
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=14.55
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1    0.0382 11.605 12.589
## - Tot_Precip_7day  1    0.3944 11.961 12.922
## - Avg_Temp_7day    1    0.5881 12.155 13.098
## <none>                         11.567 14.553
## - Range_t          1    2.4689 14.036 14.681
## - Mean_Q           1    4.2404 15.807 15.989
## - Q_Tot            1    6.9506 18.518 17.729
```

```
##
## Step:  AIC=12.59
## PC3 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    0.4221 12.027 10.982
## - Avg_Temp_7day    1    0.8279 12.433 11.347
## <none>                         11.605 12.589
## - Range_t          1    2.5213 14.127 12.752
## - Mean_Q           1    4.3212 15.927 14.071
## + SW               1    0.0382 11.567 14.553
## - Q_Tot            1    6.9276 18.533 15.738
##
## Step:  AIC=10.98
## PC3 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         12.027 10.982
## - Avg_Temp_7day    1    2.8978 14.925 11.357
## - Mean_Q           1    4.0790 16.106 12.194
## + Tot_Precip_7day  1    0.4221 11.605 12.589
## - Range_t          1    4.7738 16.801 12.659
## + SW               1    0.0659 11.961 12.922
## - Q_Tot            1    9.4949 21.522 15.383
```

```r
summary(stepAIC_model_pre_intercept_PC3)
```

```
##
## Call:
## lm(formula = PC3 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day,
##     data = pre_intercept_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5331 -0.6394 -0.1690  0.5022  1.8863
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.919e+00  3.077e+00   0.624   0.5558
## Q_Tot         -3.157e-06  1.450e-06  -2.176   0.0724 .
## Mean_Q         9.633e-02  6.753e-02   1.426   0.2036
## Range_t        1.756e-03  1.138e-03   1.543   0.1737
## Avg_Temp_7day -1.929e-01  1.605e-01  -1.202   0.2745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.416 on 6 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4626, Adjusted R-squared:  0.1044
## F-statistic: 1.291 on 4 and 6 DF,  p-value: 0.3705
```

```r
# Perform stepwise regression for post_intercept_combined
step_model_post_intercept_PC3 <- lm(PC3 ~ Q_Tot + Mean_Q + Range_t +
```

```r
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = post_intercept_combined)
stepAIC_model_post_intercept_PC3 <- stepAIC(step_model_post_intercept_PC3,
    direction = "both", trace = TRUE)
```

```
## Start:  AIC=7.52
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##
## Step:  AIC=7.52
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## - Range_t          1    0.0154  7.9135  5.5471
## - SW               1    0.7584  8.6565  6.7137
## - Mean_Q           1    0.9263  8.8244  6.9635
## <none>                          7.8981  7.5218
## - Q_Tot            1    1.6272  9.5253  7.9570
## - Avg_Temp_7day    1    1.6779  9.5760  8.0260
## - Tot_Precip_7day  1    5.7616 13.6597 12.6435
##
## Step:  AIC=5.55
## PC3 ~ Q_Tot + Mean_Q + SW + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq     RSS     AIC
## <none>                          7.9135  5.5471
## - SW               1    1.4156  9.3291  5.6865
## - Mean_Q           1    1.8496  9.7632  6.2777
## - Avg_Temp_7day    1    2.0682  9.9818  6.5656
## + Range_t          1    0.0154  7.8981  7.5218
## - Q_Tot            1    6.8012 14.7147 11.6107
## - Tot_Precip_7day  1    7.5362 15.4497 12.2443
```

```r
summary(stepAIC_model_post_intercept_PC3)
```

```
##
## Call:
## lm(formula = PC3 ~ Q_Tot + Mean_Q + SW + Avg_Temp_7day + Tot_Precip_7day,
##     data = post_intercept_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0356 -0.6228 -0.2219  0.4235  1.5100
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.244e+00  2.425e+00  -2.163   0.0673 .
## Q_Tot            1.033e-06  4.212e-07   2.453   0.0439 *
## Mean_Q          -2.123e-02  1.660e-02  -1.279   0.2416
## SW               3.304e+00  2.953e+00   1.119   0.3001
## Avg_Temp_7day    2.028e-01  1.499e-01   1.353   0.2183
## Tot_Precip_7day -1.110e+00  4.298e-01  -2.582   0.0364 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.063 on 7 degrees of freedom
## Multiple R-squared:  0.5601, Adjusted R-squared:  0.2459
## F-statistic: 1.783 on 5 and 7 DF,  p-value: 0.235
```

```r
# Perform stepwise regression for all_intercept_combined
step_model_all_intercept_PC3 <- lm(PC3 ~ Q_Tot + Mean_Q + Range_t +
    SW + Avg_Temp_7day + Tot_Precip_7day + Treatment, data = all_intercept_combined)
stepAIC_model_all_intercept_PC3 <- stepAIC(step_model_all_intercept_PC3,
    direction = "both", trace = TRUE)
```

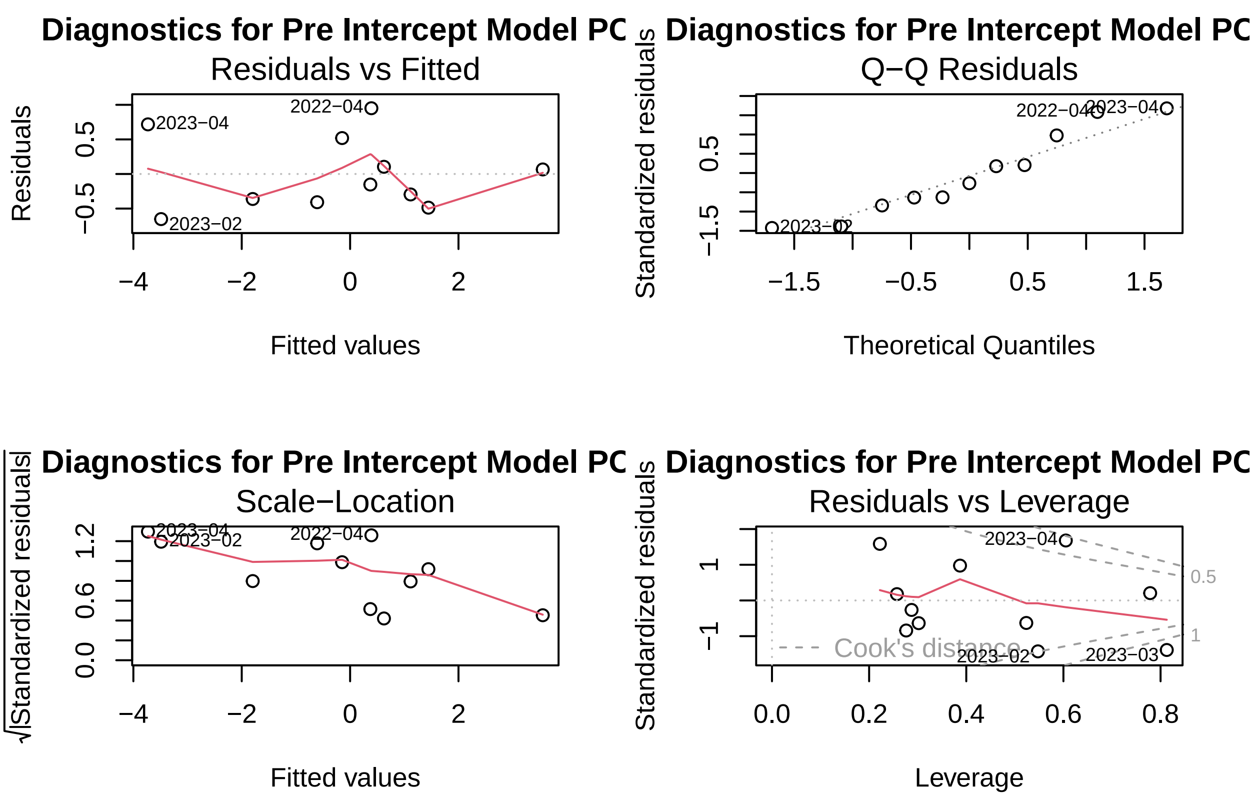
```
## Start:  AIC=24.61
## PC3 ~ Q_Tot + Mean_Q + Range_t + SW + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - SW               1   0.16148 30.916 22.711
## - Mean_Q           1   0.17326 30.928 22.718
## - Treatment        1   0.45619 31.211 22.901
## - Q_Tot            1   0.56902 31.323 22.973
## - Tot_Precip_7day  1   0.57757 31.332 22.978
## - Range_t          1   1.06804 31.822 23.289
## - Avg_Temp_7day    1   1.24929 32.004 23.402
## <none>                         30.754 24.606
##
## Step:  AIC=22.71
## PC3 ~ Q_Tot + Mean_Q + Range_t + Avg_Temp_7day + Tot_Precip_7day +
##     Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Mean_Q           1   0.19131 31.107 20.834
## - Treatment        1   0.29633 31.212 20.902
## - Q_Tot            1   0.55071 31.467 21.064
## - Tot_Precip_7day  1   0.87977 31.796 21.272
## - Avg_Temp_7day    1   1.14419 32.060 21.438
## - Range_t          1   1.40476 32.321 21.599
## <none>                         30.916 22.711
## + SW               1   0.16148 30.754 24.606
##
## Step:  AIC=20.83
## PC3 ~ Q_Tot + Range_t + Avg_Temp_7day + Tot_Precip_7day + Treatment
##
##                   Df Sum of Sq    RSS    AIC
## - Treatment        1   0.24713 31.354 18.992
## - Q_Tot            1   0.38649 31.494 19.081
## - Tot_Precip_7day  1   0.88341 31.991 19.394
## - Avg_Temp_7day    1   1.01212 32.119 19.474
## - Range_t          1   2.85050 33.958 20.588
## <none>                         31.107 20.834
## + Mean_Q           1   0.19131 30.916 22.711
## + SW               1   0.17952 30.928 22.718
##
```

```
## Step:  AIC=18.99
## PC3 ~ Q_Tot + Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Q_Tot            1    0.1690 31.523 17.100
## - Tot_Precip_7day  1    0.6634 32.018 17.411
## - Avg_Temp_7day    1    0.8052 32.160 17.500
## <none>                         31.354 18.992
## - Range_t          1    4.9401 36.294 19.919
## + Treatment        1    0.2471 31.107 20.834
## + Mean_Q           1    0.1421 31.212 20.902
## + SW               1    0.0064 31.348 20.988
##
## Step:  AIC=17.1
## PC3 ~ Range_t + Avg_Temp_7day + Tot_Precip_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Tot_Precip_7day  1    0.5008 32.024 15.415
## - Avg_Temp_7day    1    0.8151 32.338 15.611
## <none>                         31.523 17.100
## + Q_Tot            1    0.1690 31.354 18.992
## + SW               1    0.0353 31.488 19.078
## + Treatment        1    0.0296 31.494 19.081
## + Mean_Q           1    0.0072 31.516 19.095
## - Range_t          1    8.8235 40.347 20.036
##
## Step:  AIC=15.42
## PC3 ~ Range_t + Avg_Temp_7day
##
##                   Df Sum of Sq    RSS    AIC
## - Avg_Temp_7day    1    0.3568 32.381 13.637
## <none>                         32.024 15.415
## + Tot_Precip_7day  1    0.5008 31.523 17.100
## + SW               1    0.2010 31.823 17.289
## + Mean_Q           1    0.0849 31.939 17.362
## + Treatment        1    0.0334 31.991 17.394
## + Q_Tot            1    0.0064 32.018 17.411
## - Range_t          1    8.4139 40.438 18.081
##
## Step:  AIC=13.64
## PC3 ~ Range_t
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         32.381 13.637
## + Avg_Temp_7day    1    0.3568 32.024 15.415
## + Tot_Precip_7day  1    0.0424 32.338 15.611
## + Q_Tot            1    0.0303 32.351 15.618
## + Treatment        1    0.0006 32.380 15.636
## + Mean_Q           1    0.0001 32.381 15.637
## + SW               1    0.0001 32.381 15.637
## - Range_t          1   11.4264 43.807 17.681
```

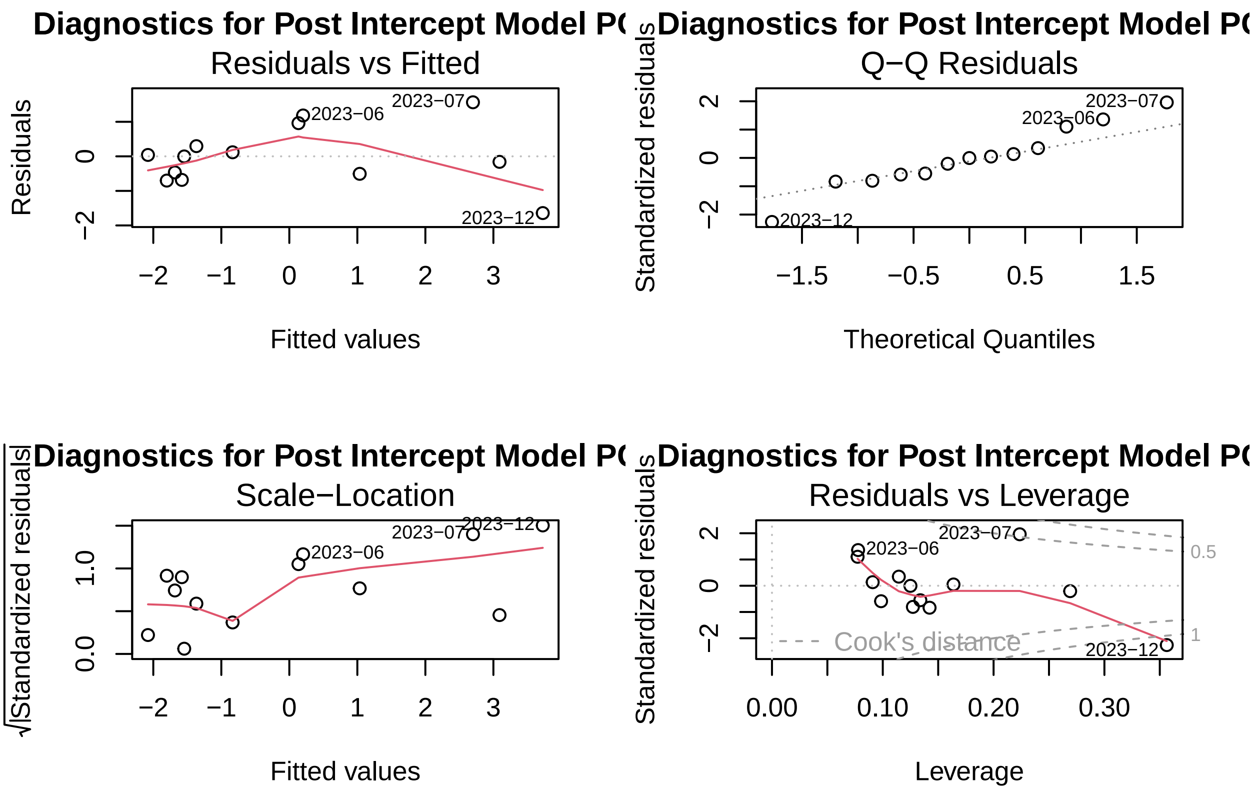
```r
summary(stepAIC_model_all_intercept_PC3)
```

```
##
## Call:
## lm(formula = PC3 ~ Range_t, data = all_intercept_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3132 -0.5499  0.0093  0.6029  2.1904
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.922162   0.496531   1.857   0.0797 .
## Range_t     -0.000978   0.000388  -2.520   0.0214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.341 on 18 degrees of freedom
## Multiple R-squared:  0.2608, Adjusted R-squared:  0.2198
## F-statistic: 6.352 on 1 and 18 DF,  p-value: 0.02139
```

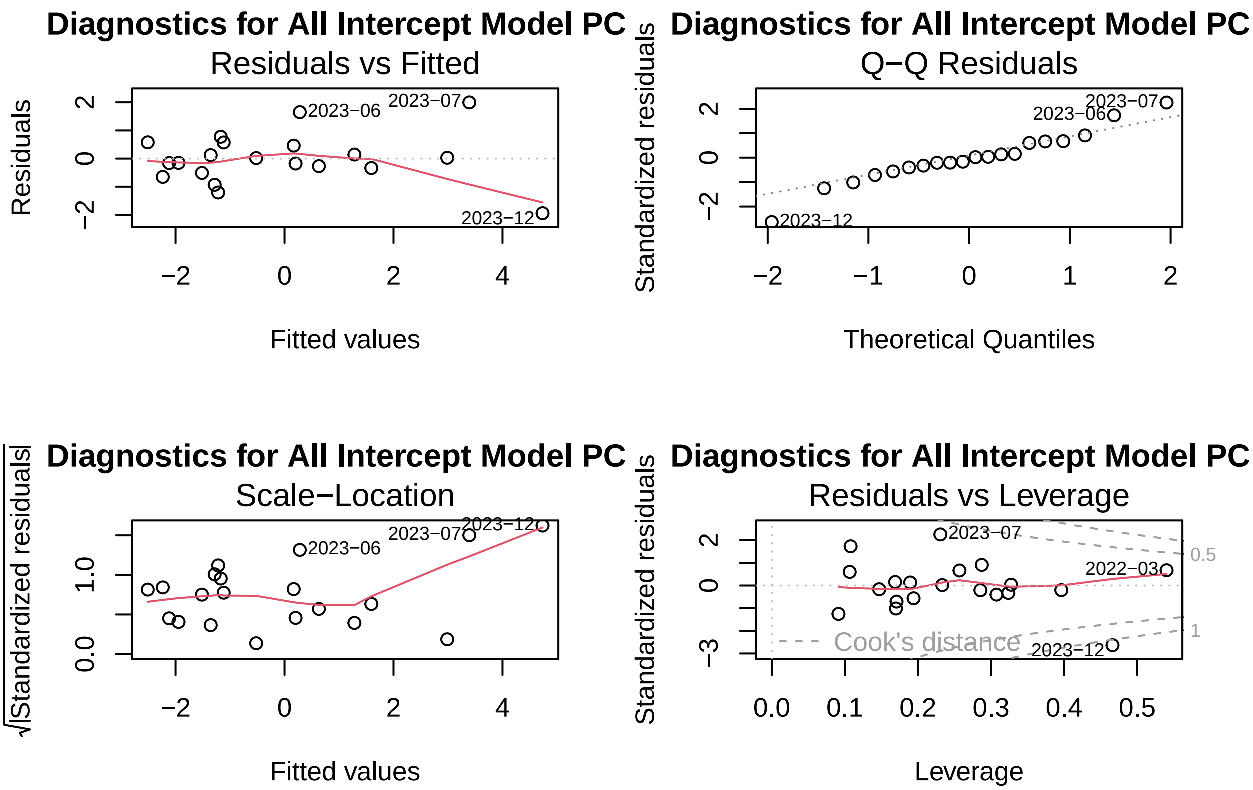
```r
## PC1 ##

# Diagnostic plots for the optimized models
par(mfrow = c(2, 2))  # Set layout for multiple plots

# Pre-intercept
plot(stepAIC_model_pre_intercept_PC1, main = "Diagnostics for Pre Intercept Model PC1")
```



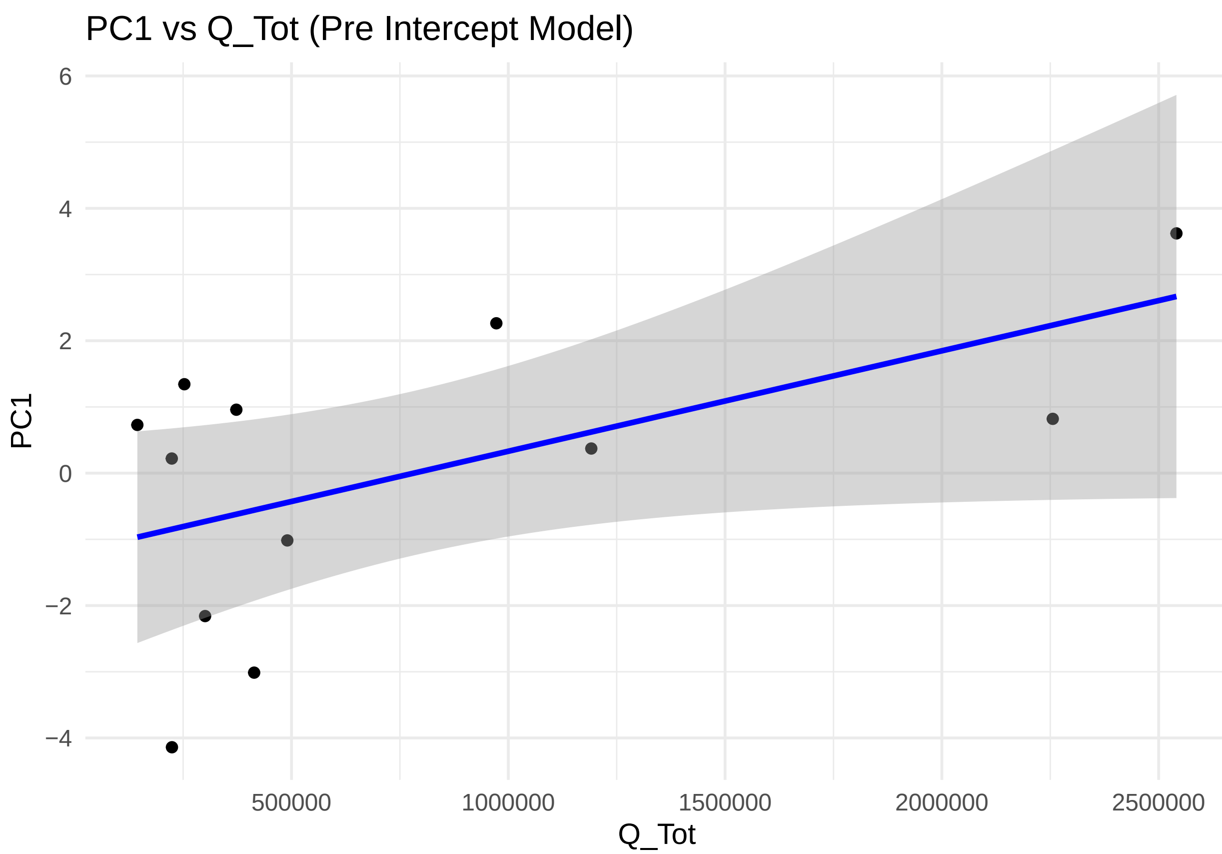
```r
# Post-intercept
plot(stepAIC_model_post_intercept_PC1, main = "Diagnostics for Post Intercept Model PC1")
```



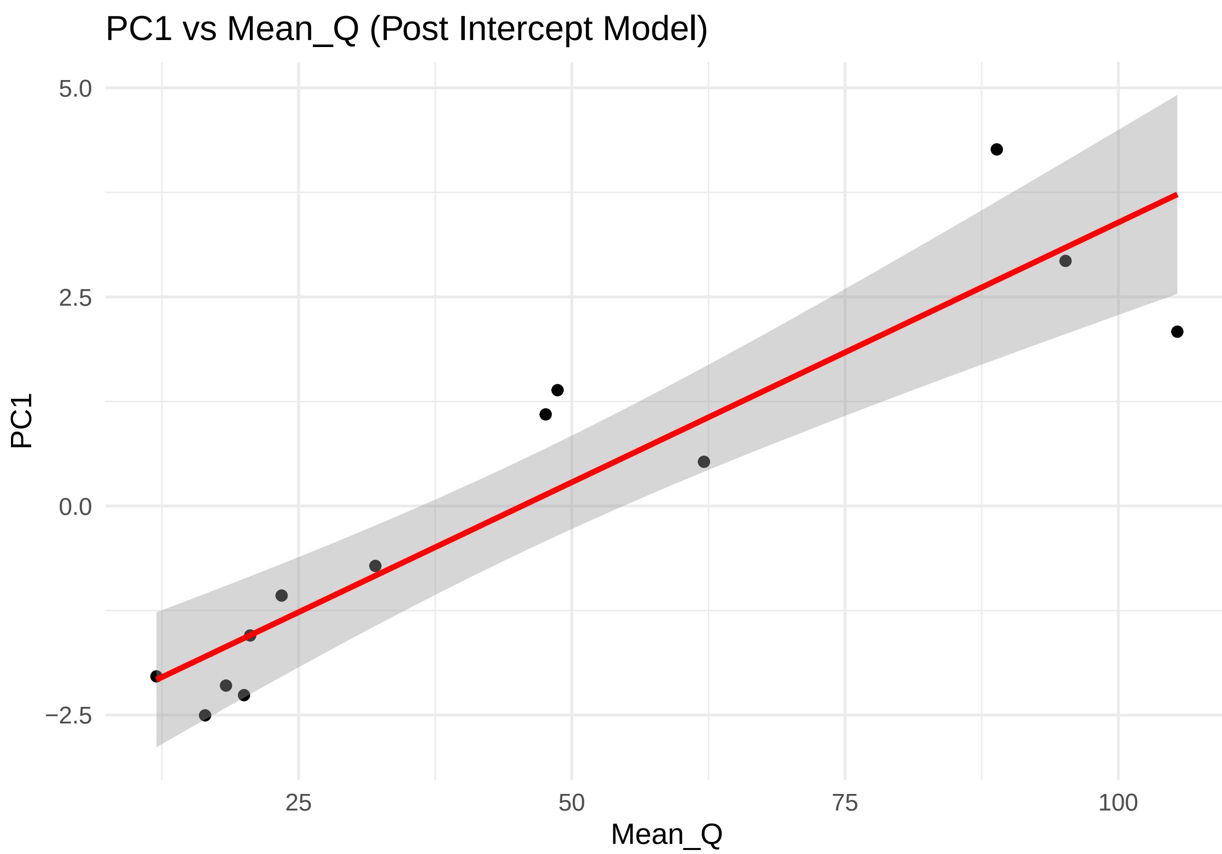

```r
# All-intercept
plot(stepAIC_model_all_intercept_PC1, main = "Diagnostics for All Intercept Model PC1")
```



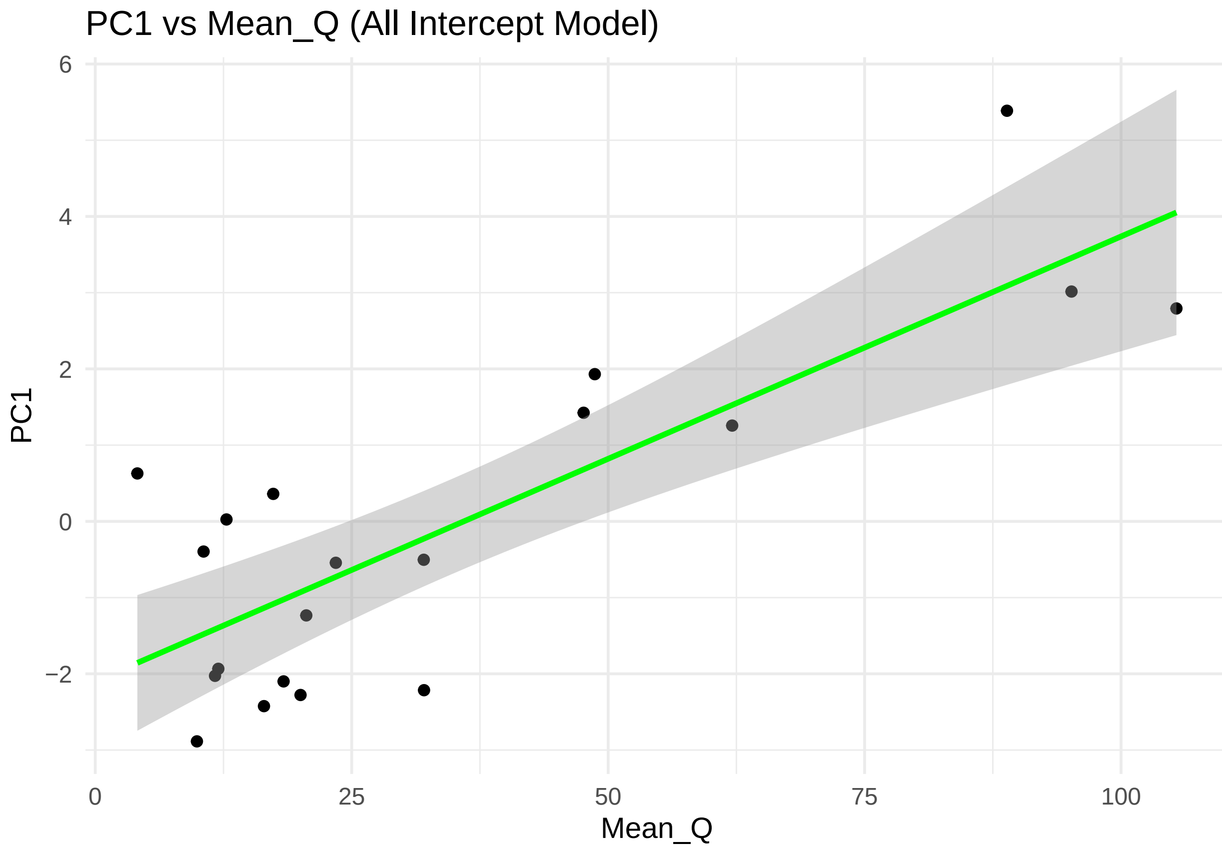

```r
# Extract significant predictors from pre-intercept model
predictors_pre_PC1 <- names(coef(stepAIC_model_pre_intercept_PC1))[-1]  # Exclude intercept

# Example scatter plot for the first significant predictor
# in pre-intercept model
ggplot(pre_intercept_combined, aes_string(x = predictors_pre_PC1[1],
    y = "PC1")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "blue") + ggtitle(paste("PC1 vs", predictors_pre_PC1[1],
    "(Pre Intercept Model)")) + theme_minimal()
```



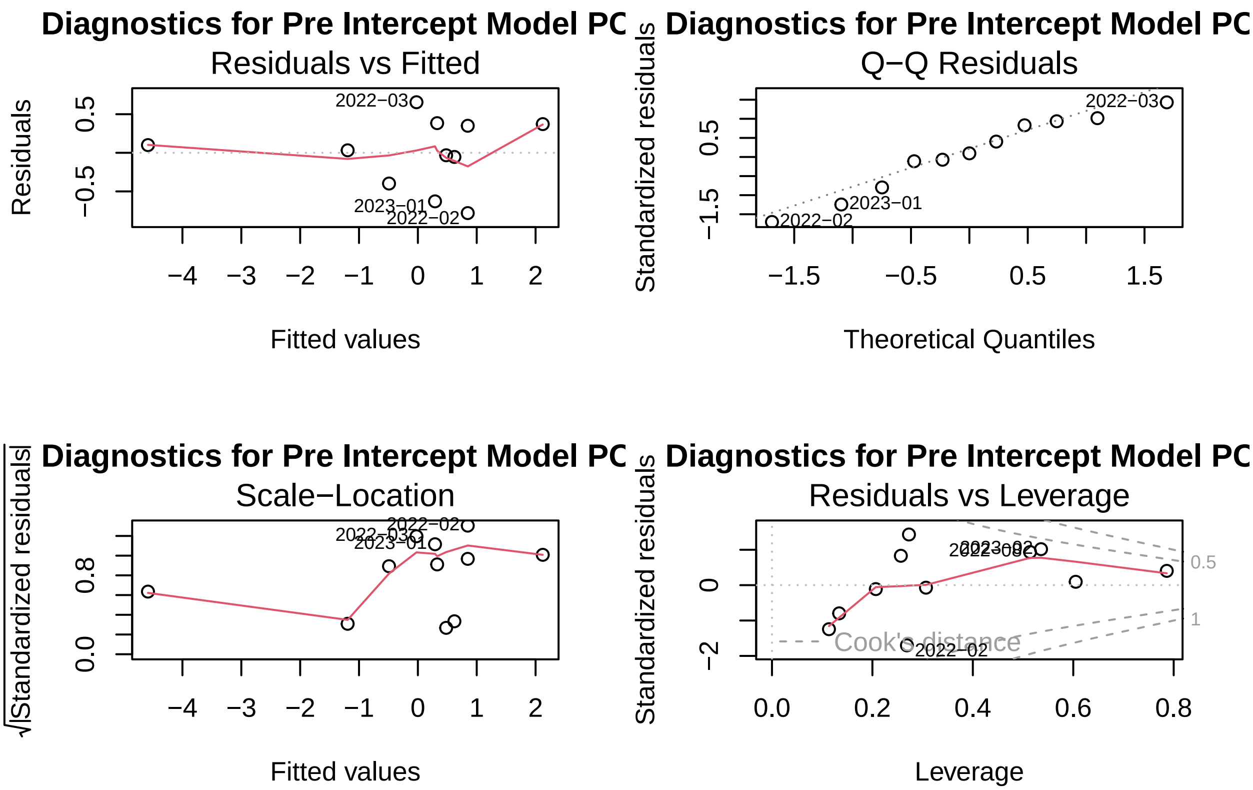

```r
# Repeat for post-intercept model
predictors_post_PC1 <- names(coef(stepAIC_model_post_intercept_PC1))[-1]
ggplot(post_intercept_combined, aes_string(x = predictors_post_PC1[1],
    y = "PC1")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "red") + ggtitle(paste("PC1 vs", predictors_post_PC1[1],
    "(Post Intercept Model)")) + theme_minimal()
```



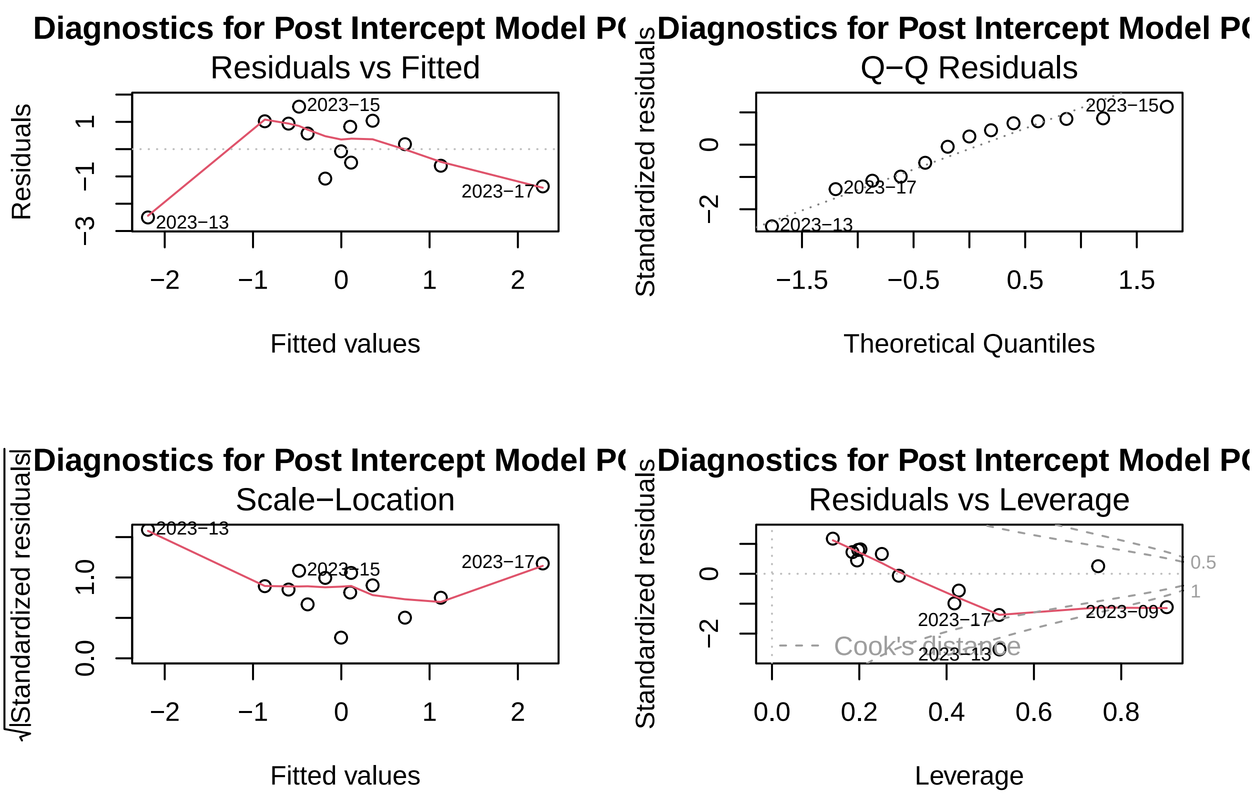

```r
# Repeat for all-intercept model
predictors_all_PC1 <- names(coef(stepAIC_model_all_intercept_PC1))[-1]
ggplot(all_intercept_combined, aes_string(x = predictors_all_PC1[1],
    y = "PC1")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "green") + ggtitle(paste("PC1 vs", predictors_all_PC1[1],
    "(All Intercept Model)")) + theme_minimal()
```



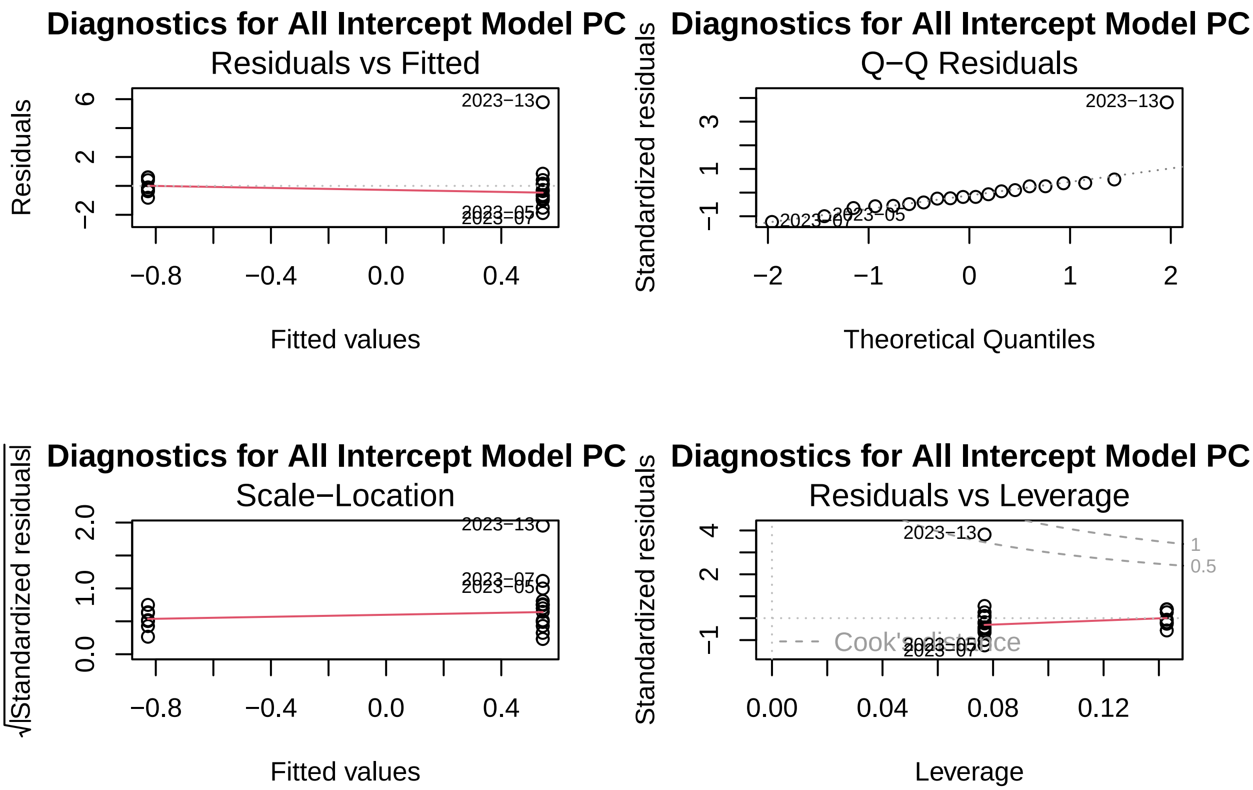

```r
## PC2 ##

# Diagnostic plots for the optimized models
par(mfrow = c(2, 2))  # Set layout for multiple plots

# Pre-intercept
plot(stepAIC_model_pre_intercept_PC2, main = "Diagnostics for Pre Intercept Model PC2")
```



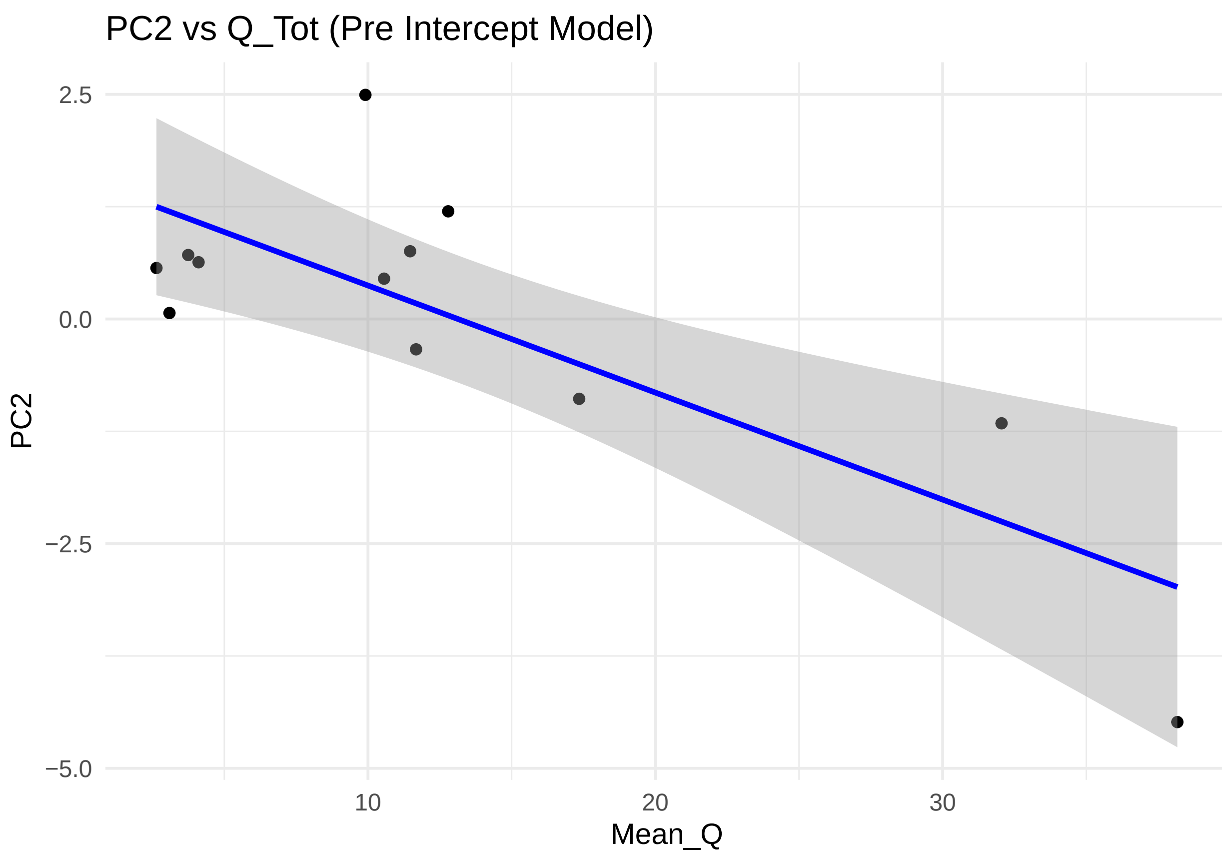

```r
# Post-intercept
plot(stepAIC_model_post_intercept_PC2, main = "Diagnostics for Post Intercept Model PC2")
```



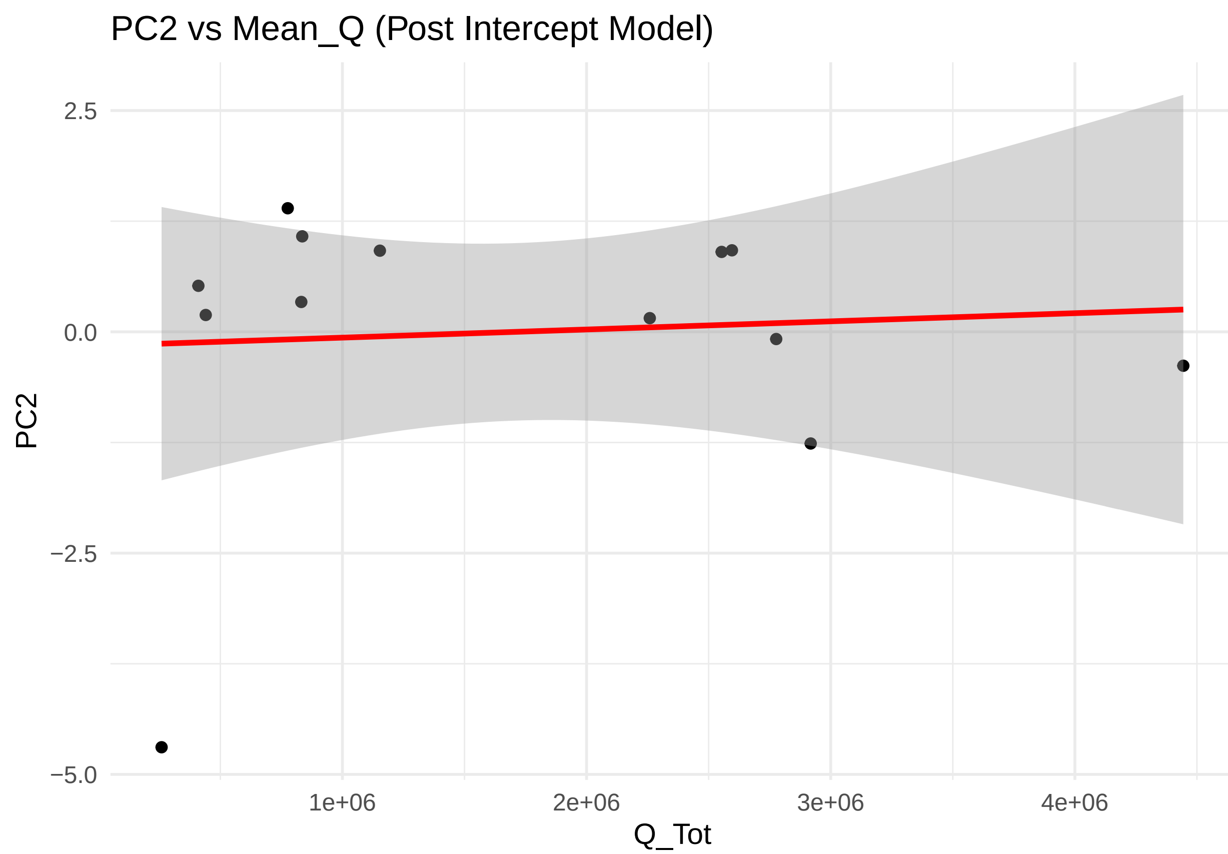

```r
# All-intercept
plot(stepAIC_model_all_intercept_PC2, main = "Diagnostics for All Intercept Model PC2")
```



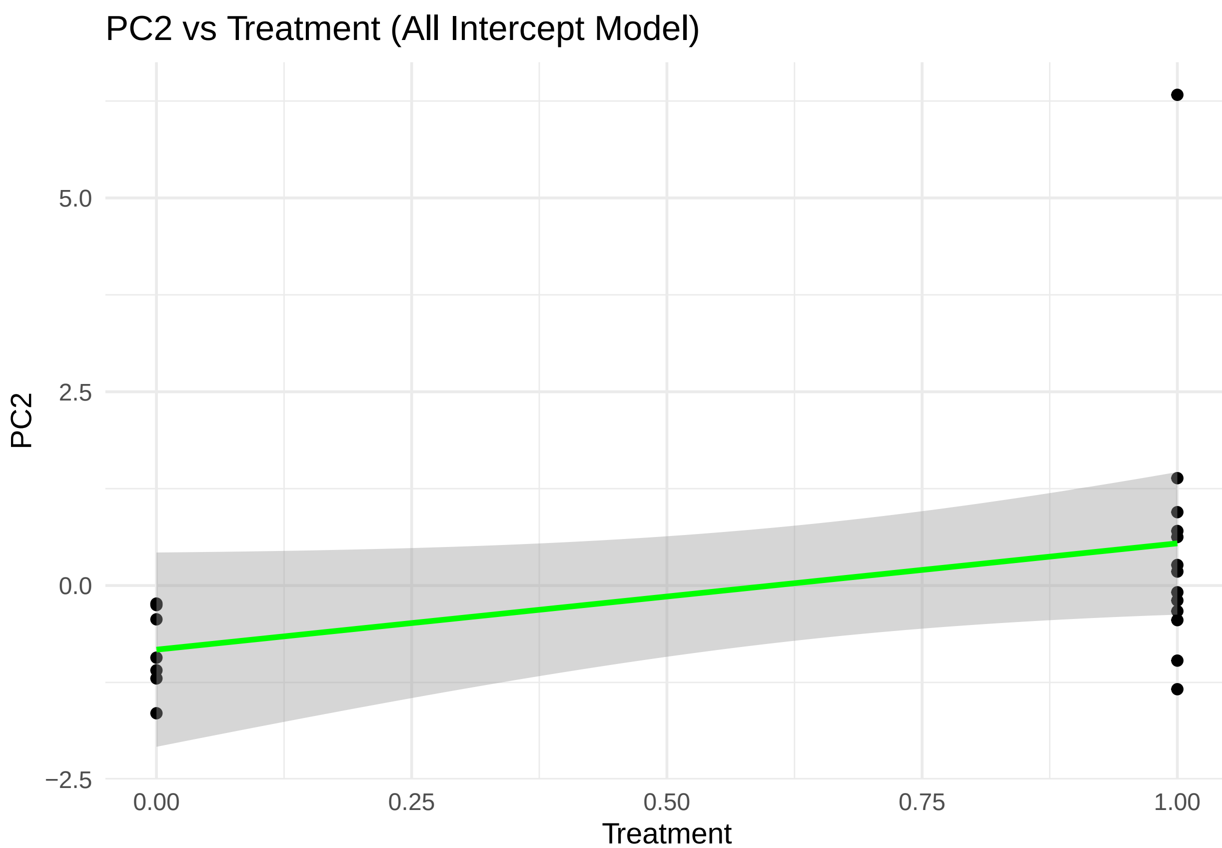

```r
# Extract significant predictors from pre-intercept model
predictors_pre_PC2 <- names(coef(stepAIC_model_pre_intercept_PC2))[-1]  # Exclude intercept

# Example scatter plot for the first significant predictor
# in pre-intercept model
ggplot(pre_intercept_combined, aes_string(x = predictors_pre_PC2[1],
    y = "PC2")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "blue") + ggtitle(paste("PC2 vs", predictors_pre_PC1[1],
    "(Pre Intercept Model)")) + theme_minimal()
```



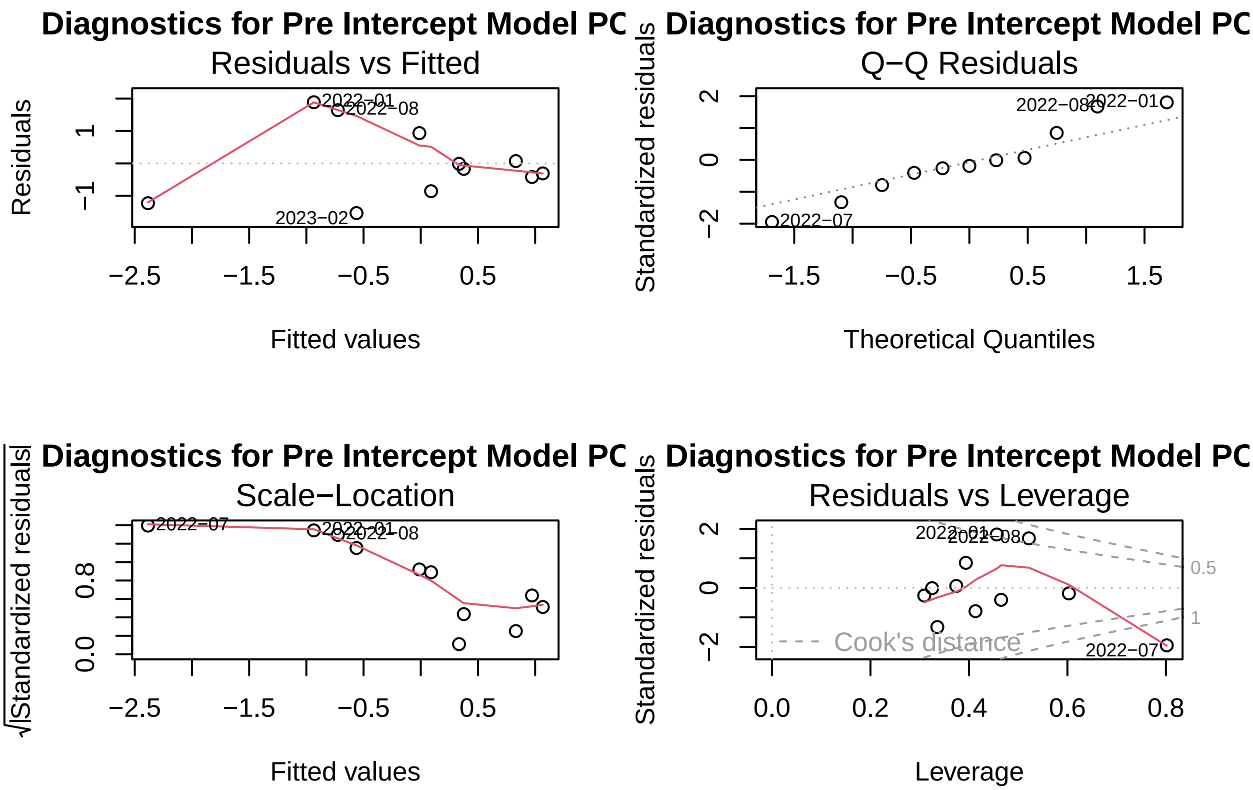

```r
# Repeat for post-intercept model
predictors_post_PC2 <- names(coef(stepAIC_model_post_intercept_PC2))[-1]
ggplot(post_intercept_combined, aes_string(x = predictors_post_PC2[1],
    y = "PC2")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "red") + ggtitle(paste("PC2 vs", predictors_post_PC1[1],
    "(Post Intercept Model)")) + theme_minimal()
```



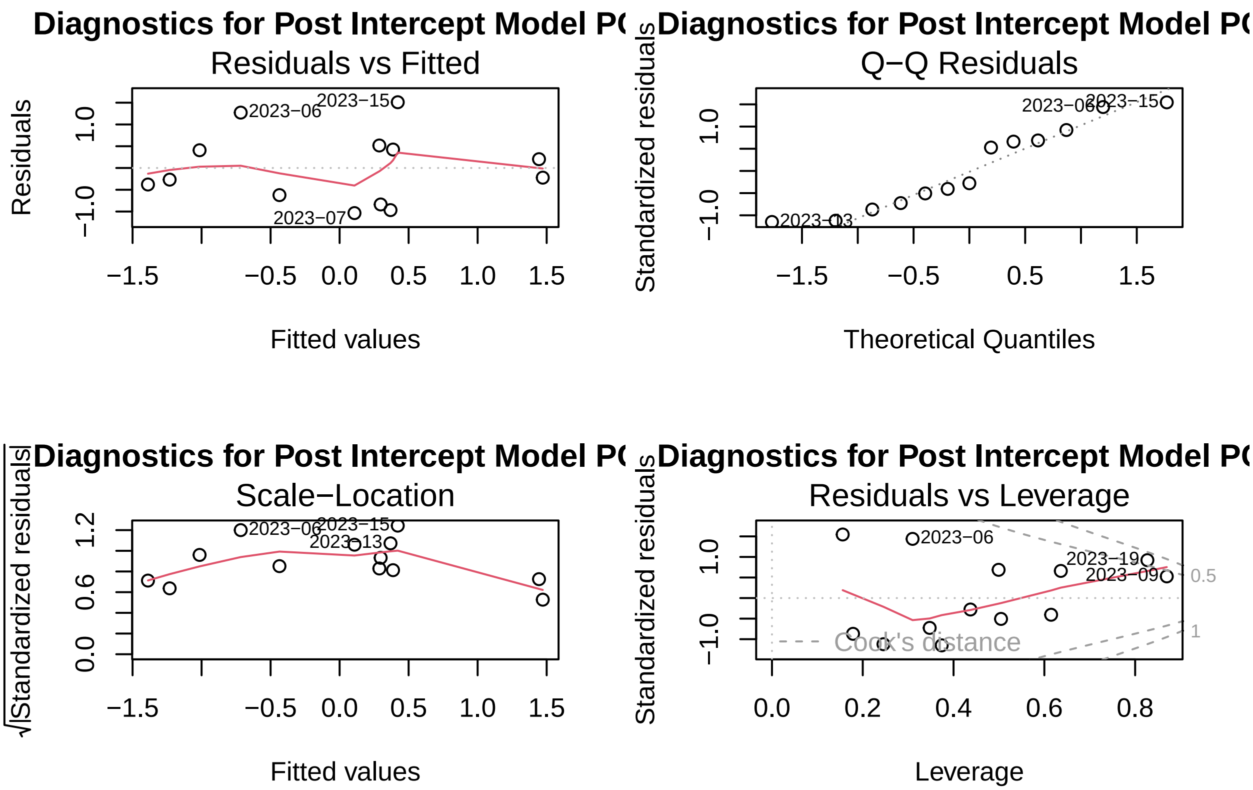

```
# Repeat for all-intercept model
predictors_all_PC2 <- names(coef(stepAIC_model_all_intercept_PC2))[-1]
ggplot(all_intercept_combined, aes_string(x = predictors_all_PC2[1],
    y = "PC2")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "green") + ggtitle(paste("PC2 vs", predictors_all_PC2[1],
    "(All Intercept Model)")) + theme_minimal()
```



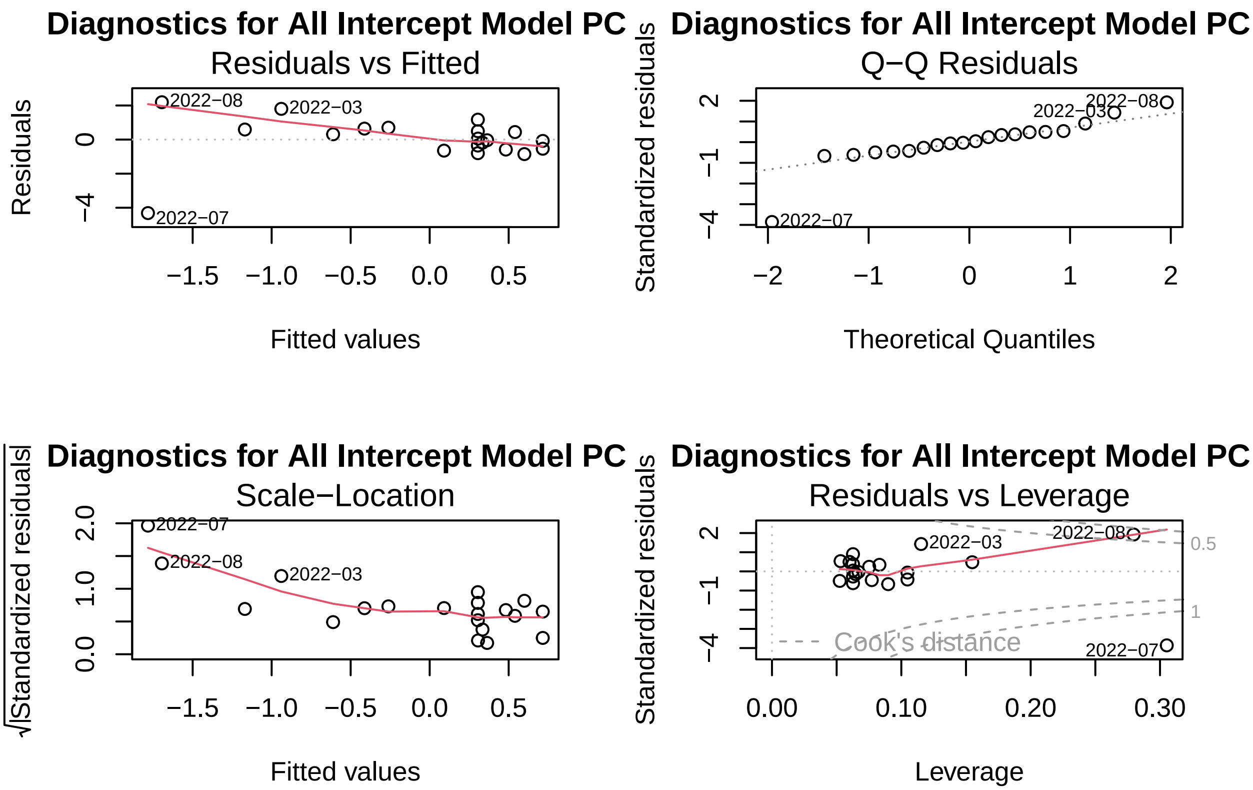

```
## PC3 ##

# Diagnostic plots for the optimized models
par(mfrow = c(2, 2))  # Set layout for multiple plots

# Pre-intercept
plot(stepAIC_model_pre_intercept_PC3, main = "Diagnostics for Pre Intercept Model PC3")
```



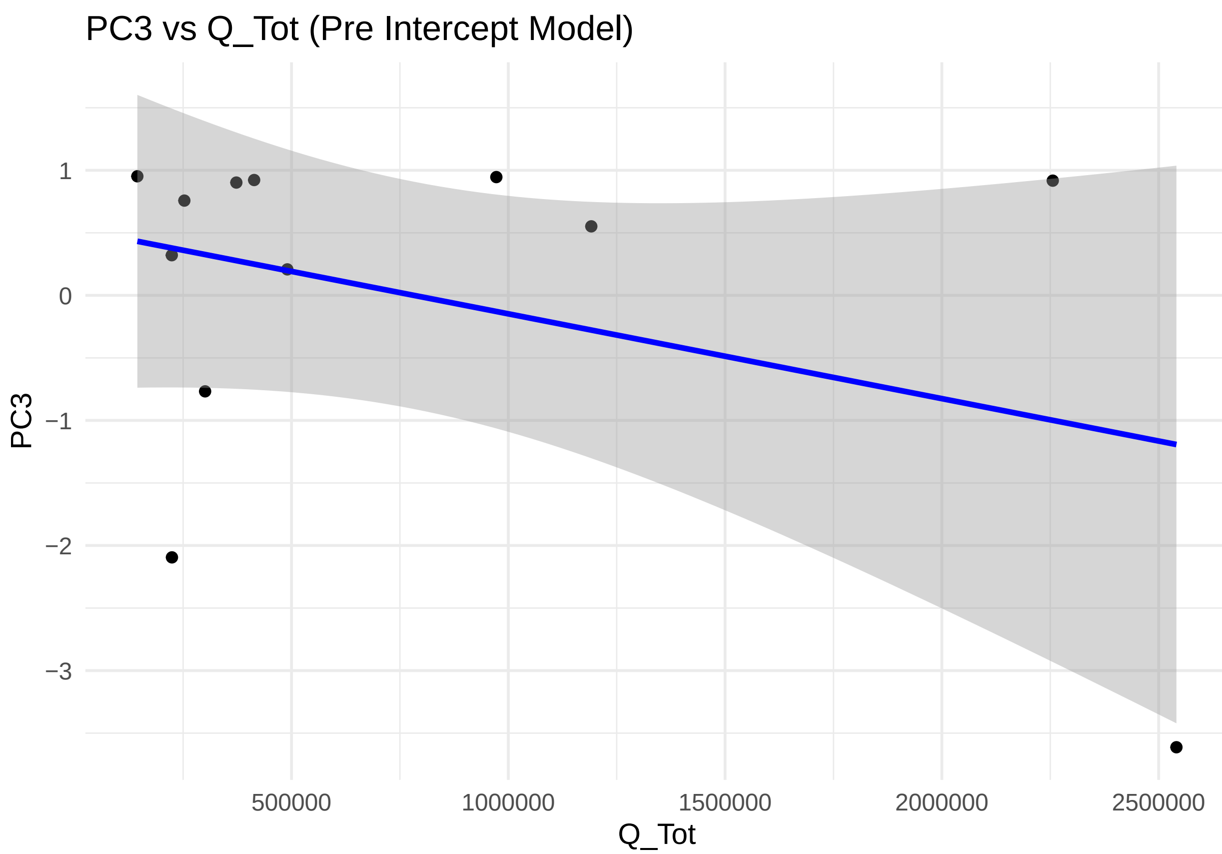

```r
# Post-intercept
plot(stepAIC_model_post_intercept_PC3, main = "Diagnostics for Post Intercept Model PC3")
```



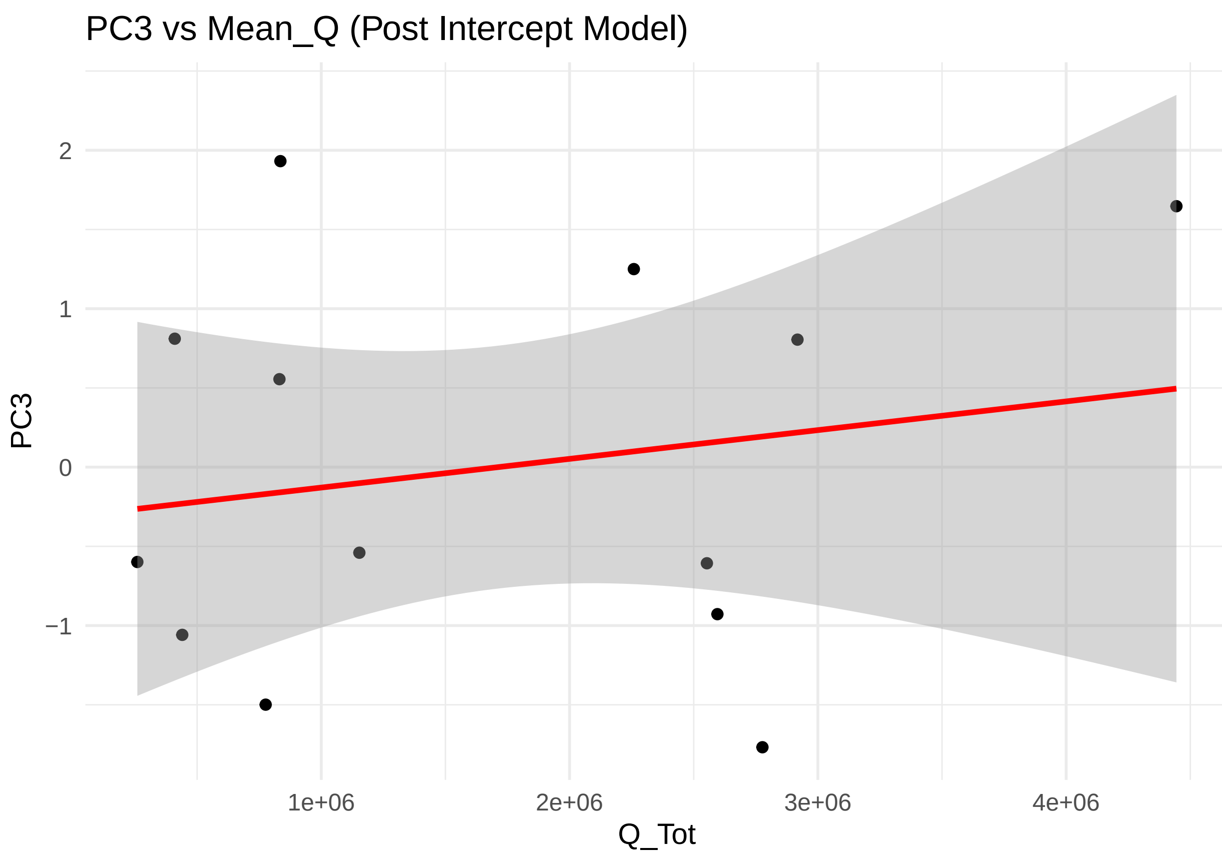

```r
# All-intercept
plot(stepAIC_model_all_intercept_PC3, main = "Diagnostics for All Intercept Model PC3")
```



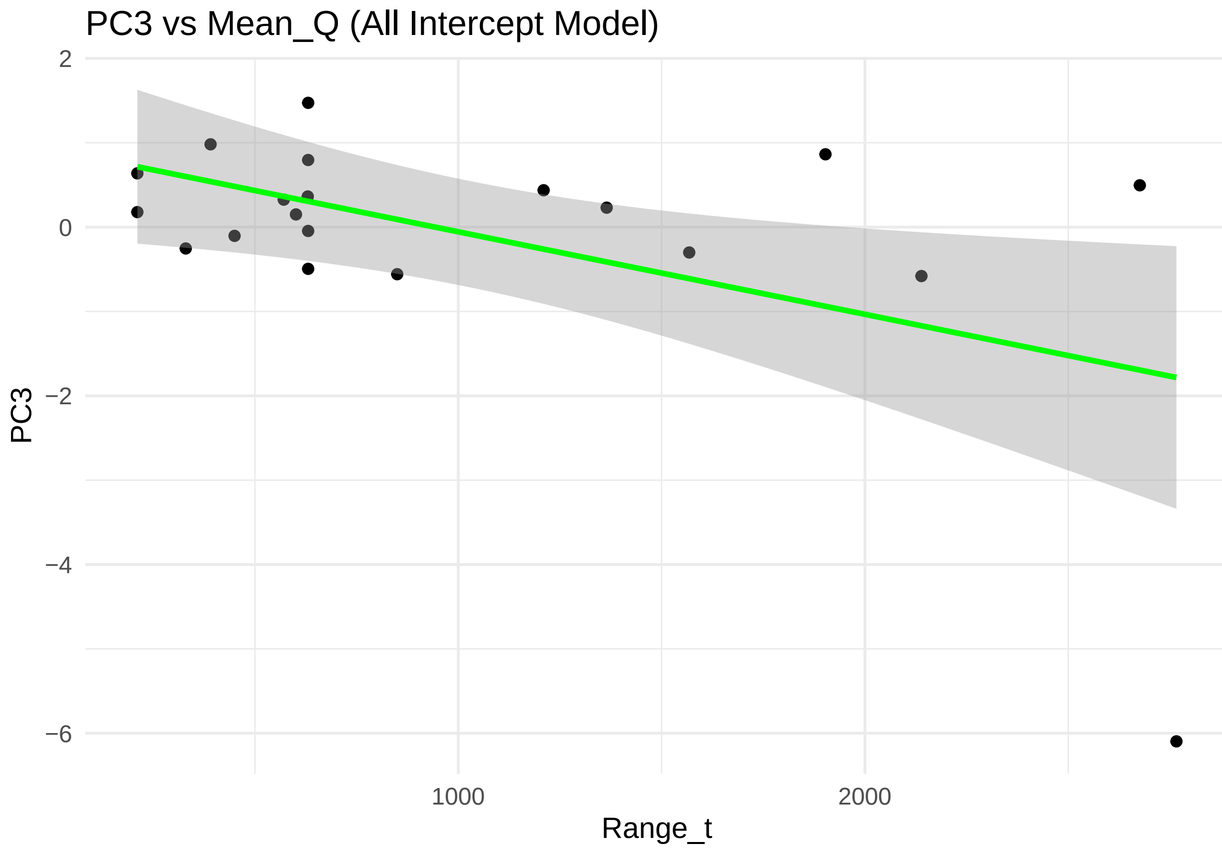

```
# Extract significant predictors from pre-intercept model
predictors_pre_PC3 <- names(coef(stepAIC_model_pre_intercept_PC3))[-1]  # Exclude intercept

# Example scatter plot for the first significant predictor
# in pre-intercept model
ggplot(pre_intercept_combined, aes_string(x = predictors_pre_PC3[1],
    y = "PC3")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "blue") + ggtitle(paste("PC3 vs", predictors_pre_PC1[1],
    "(Pre Intercept Model)")) + theme_minimal()
```




```r
# Repeat for post-intercept model
predictors_post_PC3 <- names(coef(stepAIC_model_post_intercept_PC3))[-1]
ggplot(post_intercept_combined, aes_string(x = predictors_post_PC3[1],
    y = "PC3")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "red") + ggtitle(paste("PC3 vs", predictors_post_PC1[1],
    "(Post Intercept Model)")) + theme_minimal()
```



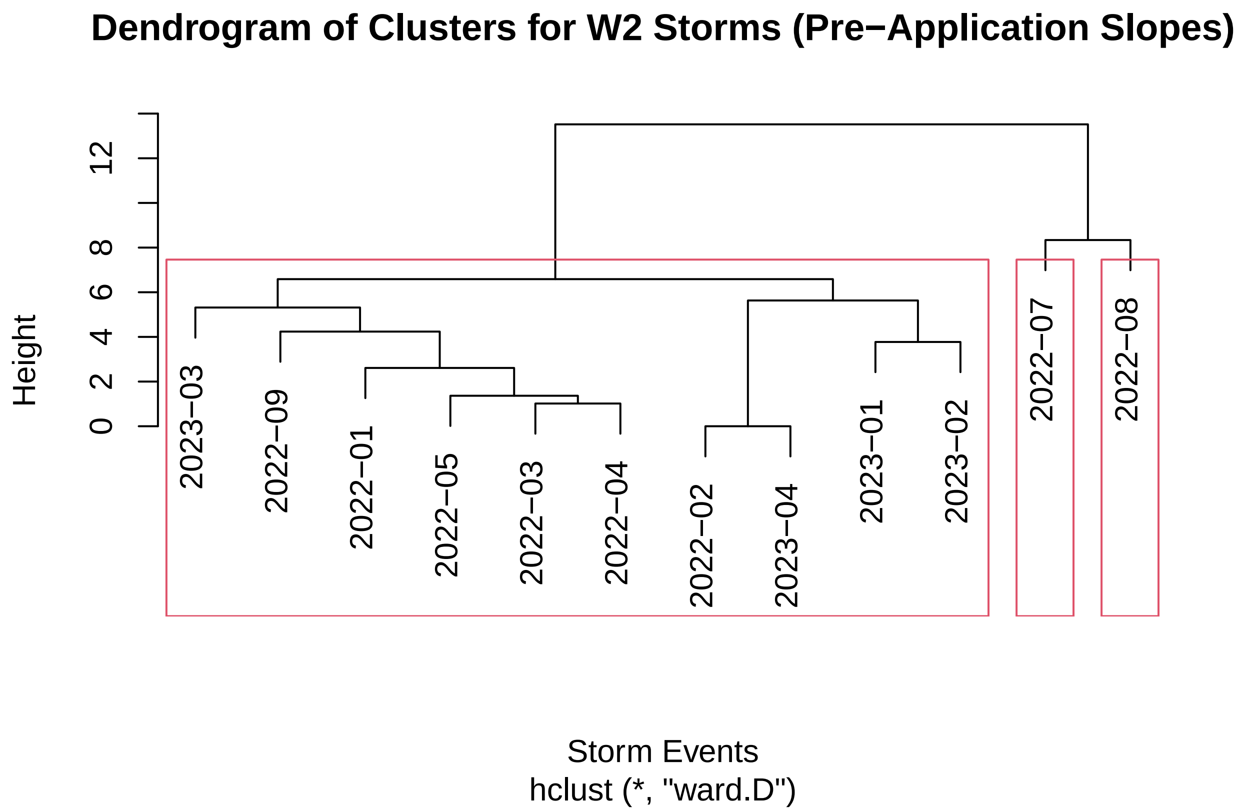

```r
# Repeat for all-intercept model
predictors_all_PC3 <- names(coef(stepAIC_model_all_intercept_PC3))[-1]
ggplot(all_intercept_combined, aes_string(x = predictors_all_PC3[1],
    y = "PC3")) + geom_point() + geom_smooth(method = "lm", formula = y ~
    x, col = "green") + ggtitle(paste("PC3 vs", predictors_all_PC1[1],
    "(All Intercept Model)")) + theme_minimal()
```



maybe i'll make some more plots here. Leaving it for now, though.

Part VI: HCA

```
# Hierarchical Clustering
dist_matrix_pre_slope <- dist(w2_std_pre_slope, method = "euclidean")
cluster_result_pre_slope <- hclust(dist_matrix_pre_slope, method = "ward.D")

# Plot dendrogram for visualization
plot(cluster_result_pre_slope, labels = rownames(w2_std_pre_slope), main = "Dendrogram of Clusters for W
     xlab = "Storm Events")
rect.hclust(cluster_result_pre_slope, k = 3)
```

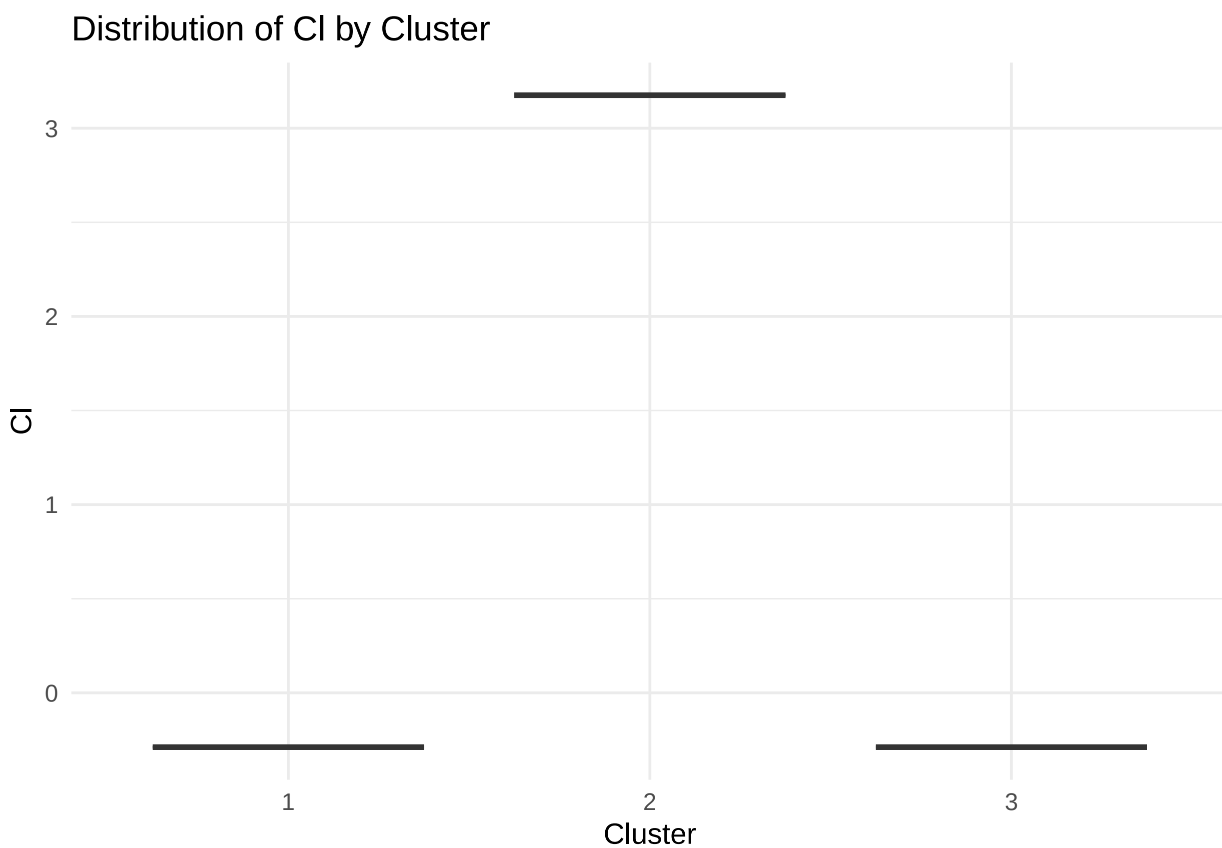


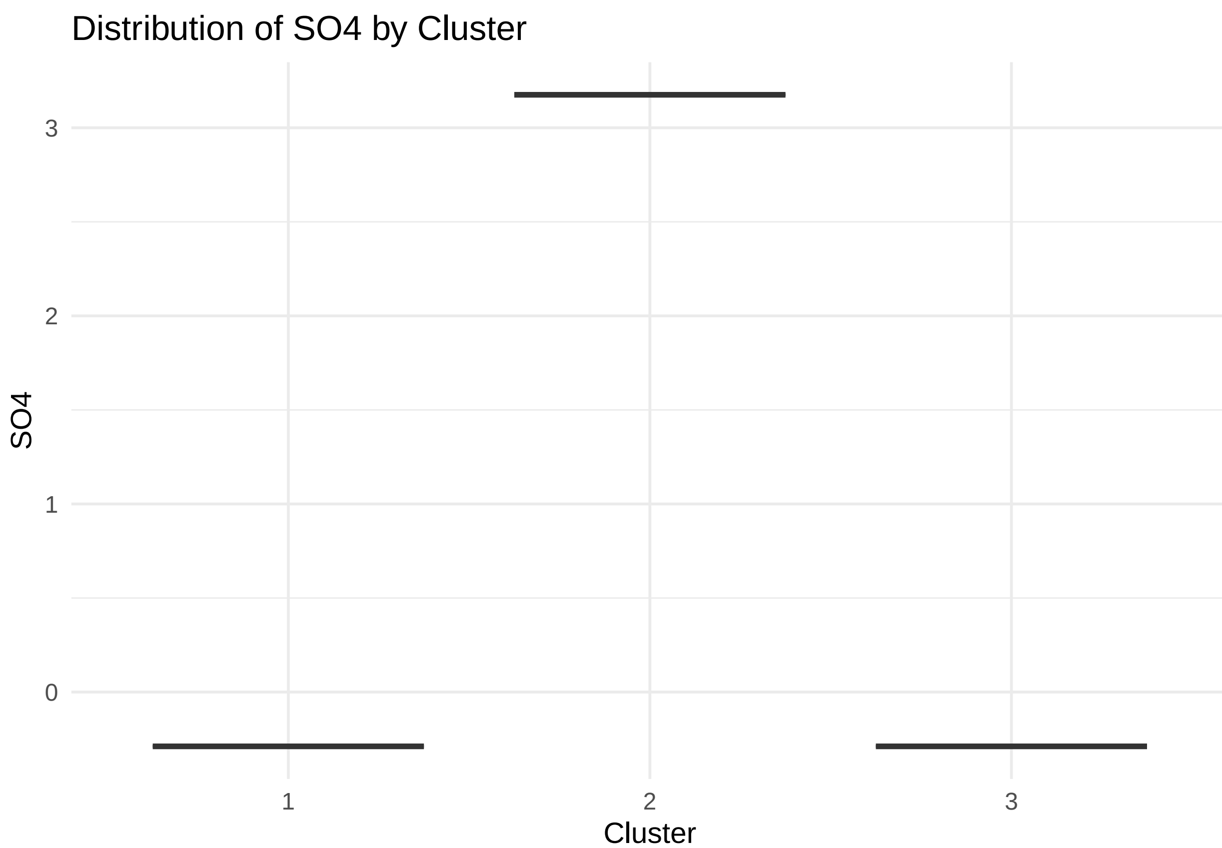

```
# Cut the dendrogram to create 3 clusters
cuts_pre_slope <- cutree(cluster_result_pre_slope, k = 3)
w2_std_pre_slope$cluster <- as.factor(cuts_pre_slope)

# Cluster Profiling
# Aggregate the original variables by cluster to identify differences
cluster_summary_pre_slope <- aggregate(w2_std_pre_slope, by = list(cluster = cuts_pre_slope), FUN = mean
print(cluster_summary_pre_slope)
```

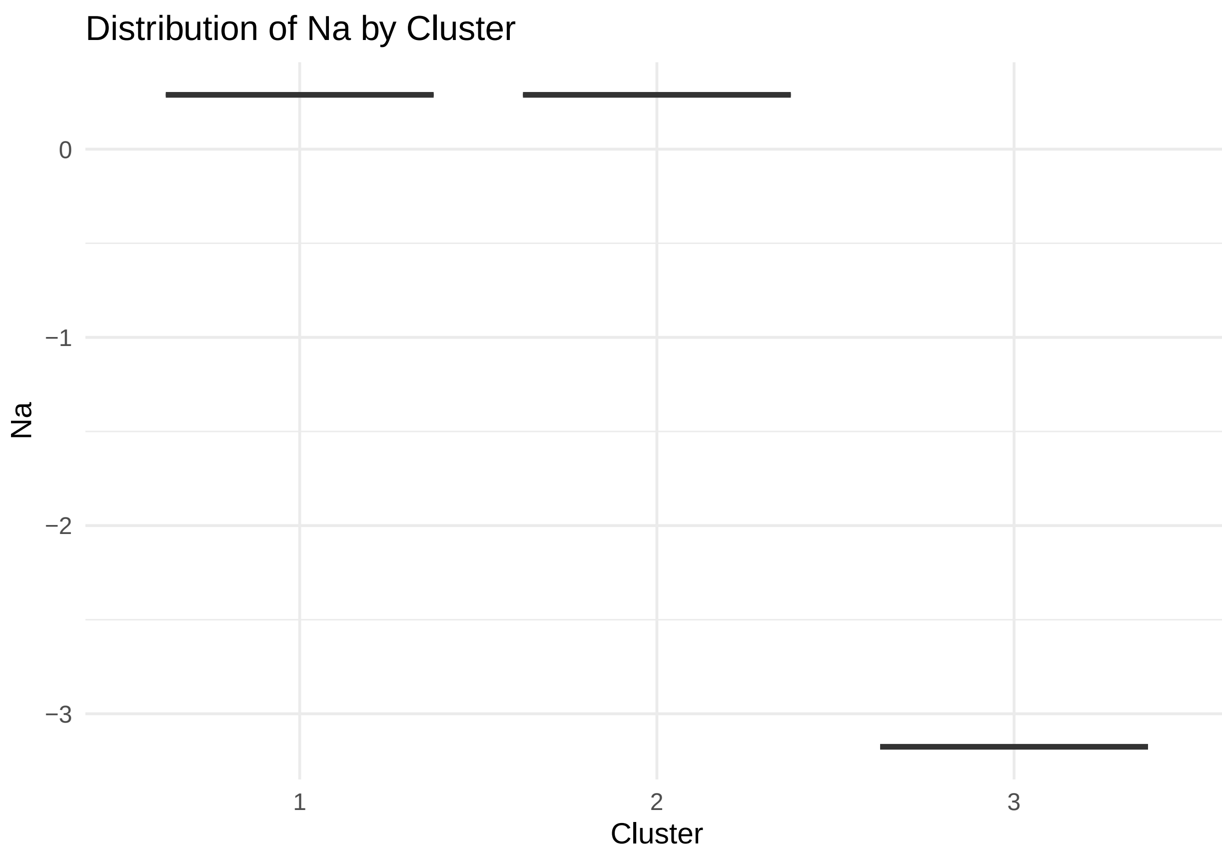
```
##   cluster         Cl        SO4         Na         Mg         Ca Alkalinity
## 1       1 -0.2886751 -0.2886751  0.2886751  0.4271718  0.2886751  0.3368037
## 2       2  3.1754265  3.1754265  0.2886751 -1.9754599  0.2886751 -1.8404610
## 3       3 -0.2886751 -0.2886751 -3.1754265 -2.2962580 -3.1754265 -1.5275761
##          DOC         As          Ba         Cr         Li           Mn
## 1 -0.1967628  0.1180551  0.06293511 -0.3586376  0.3234972  0.004214911
## 2  0.9972309 -2.5070167 -0.26807136  3.0743486 -1.6331340 -0.089941076
## 3  0.9703970  1.3264653 -0.36127978  0.5120272 -1.6018384  0.047791967
##          Si         Sr cluster
## 1  0.208595  0.2079061      NA
## 2 -1.110204 -1.0866443      NA
## 3 -0.975746 -0.9924168      NA
```

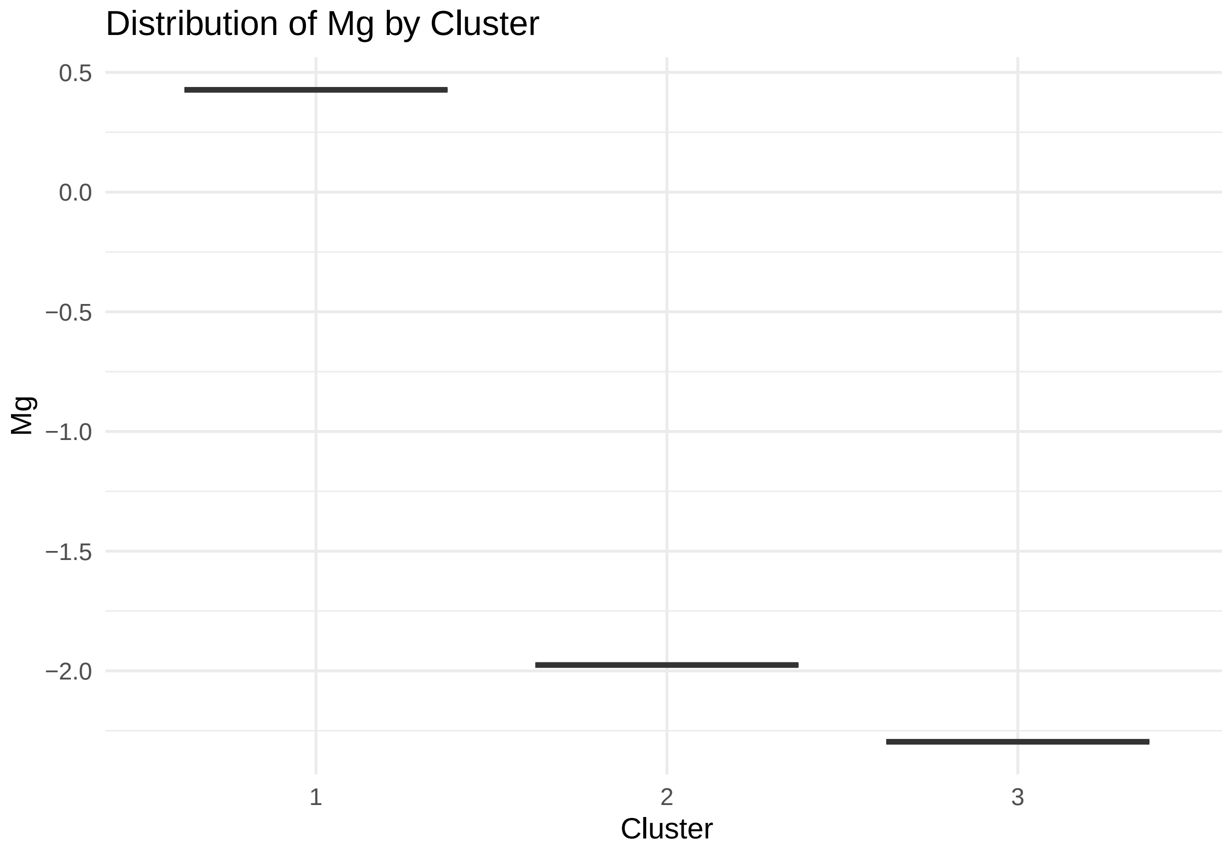
```
# Boxplots to visualize differences in key variables across clusters
for (variable in colnames(w2_std_pre_slope)) {
  ggplot(w2_std_pre_slope, aes(x = as.factor(cuts_pre_slope), y = .data[[variable]])) +
```

```r
    geom_boxplot() +
    labs(title = paste("Distribution of", variable, "by Cluster"),
         x = "Cluster",
         y = variable) +
    theme_minimal() -> plot
  print(plot)
}
```

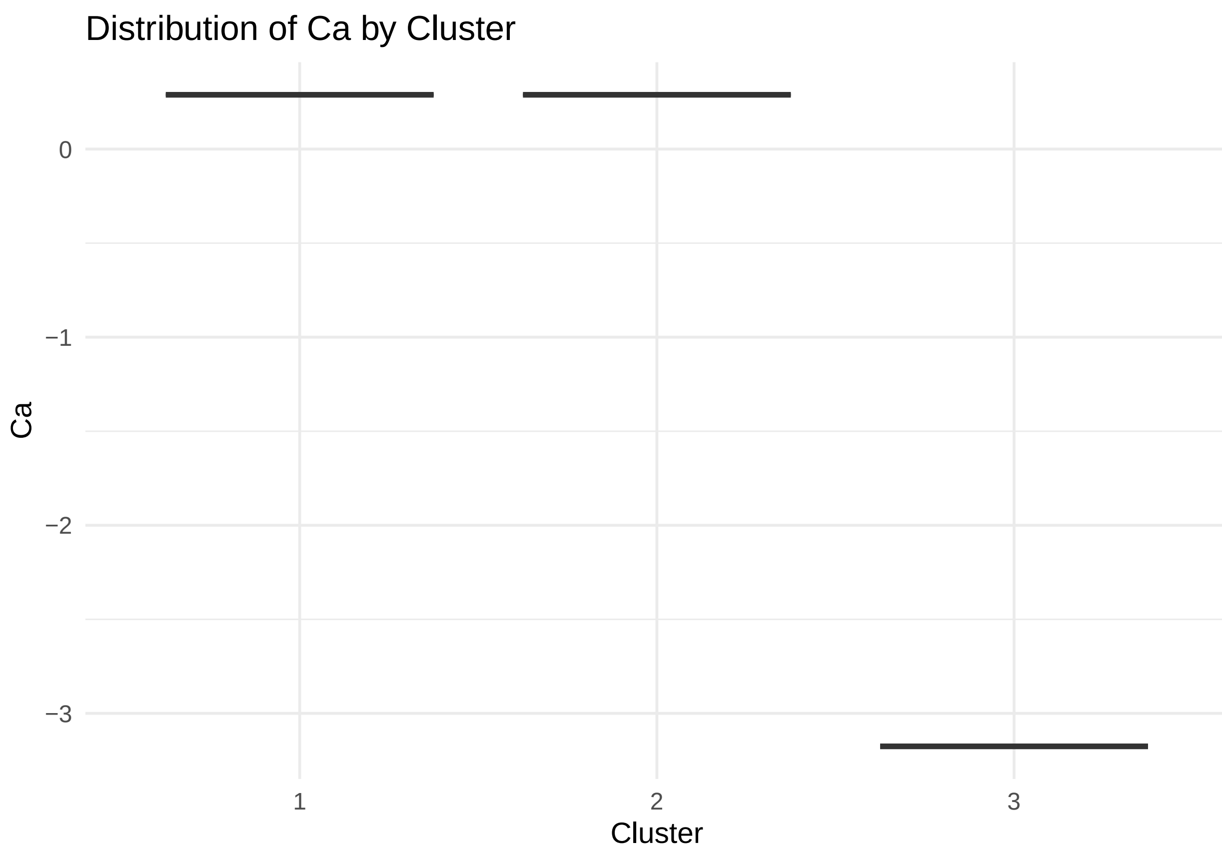


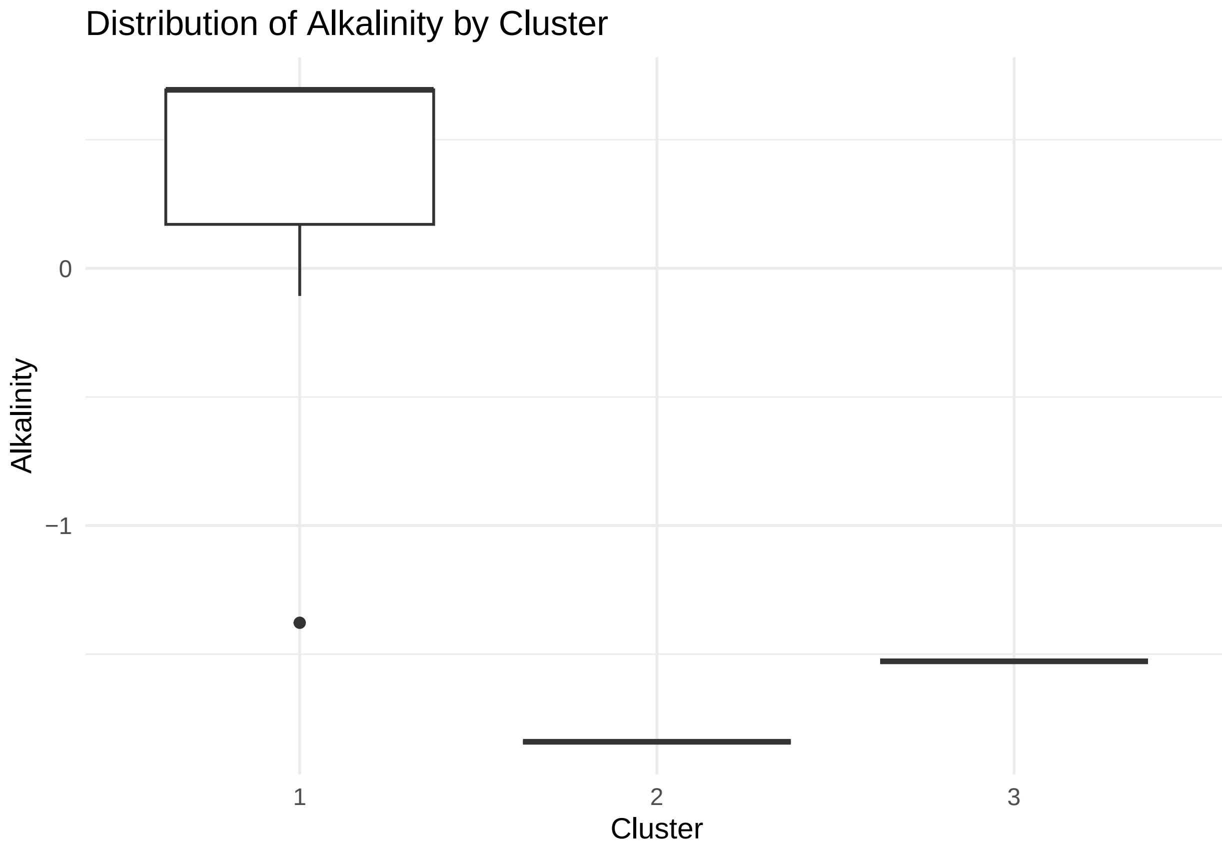

Distribution of SO4 by Cluster
3
2
1
0
SO4
1
2
3
Cluster

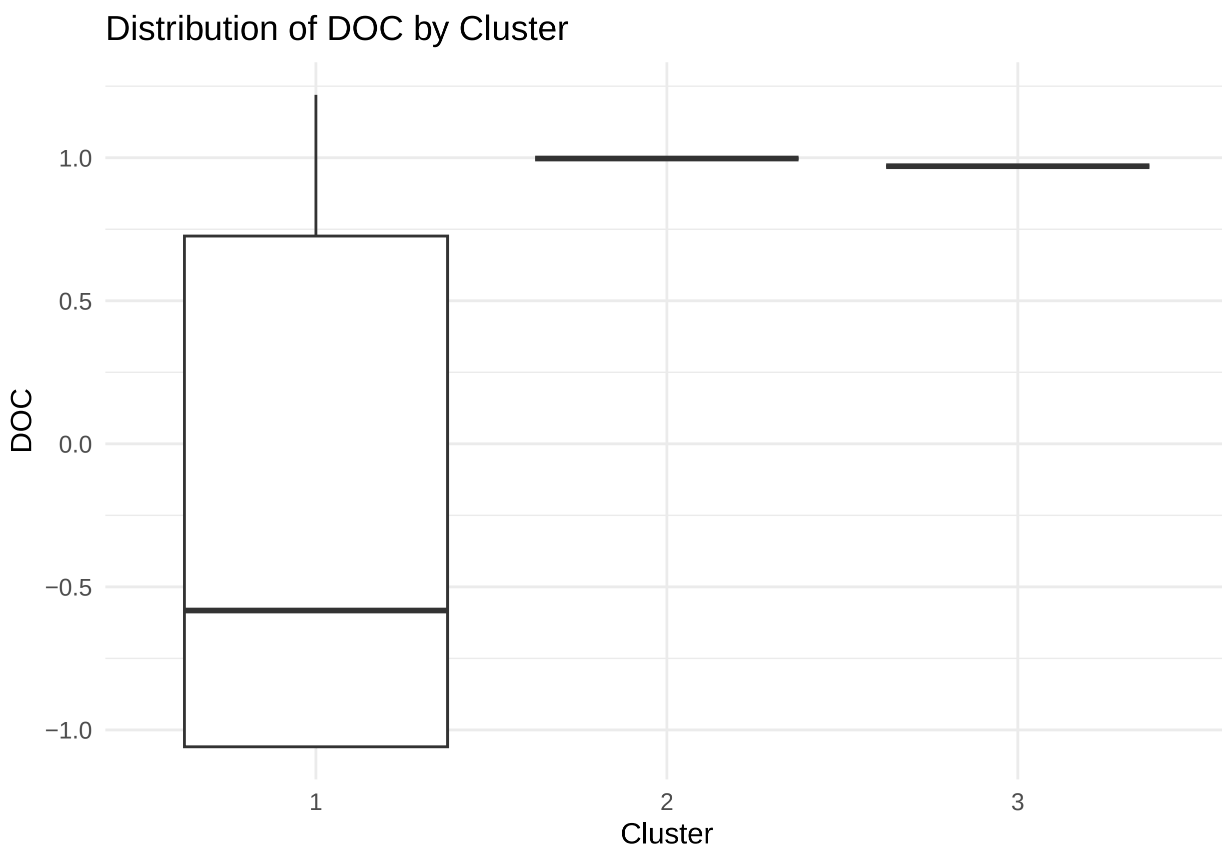

Distribution of Na by Cluster
0
−1
−2
−3
Na
1
2
3
Cluster

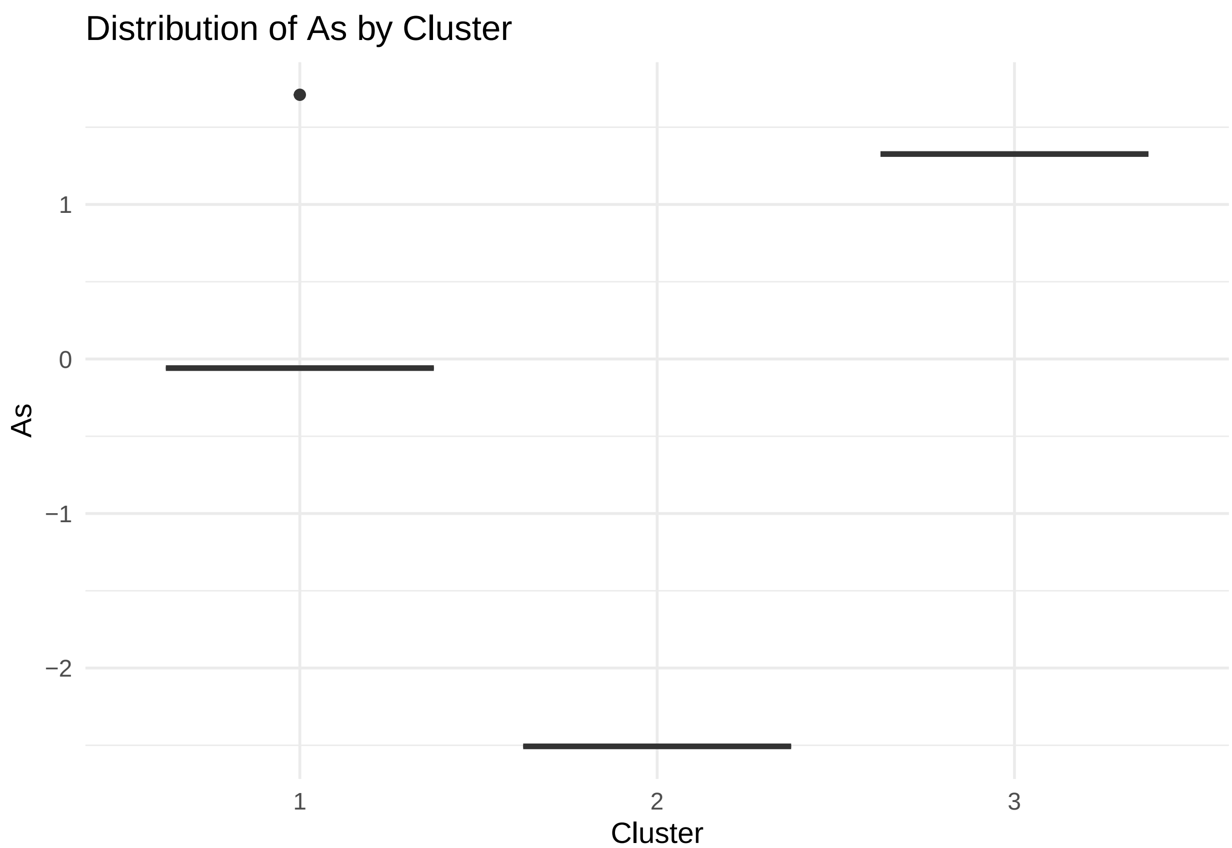


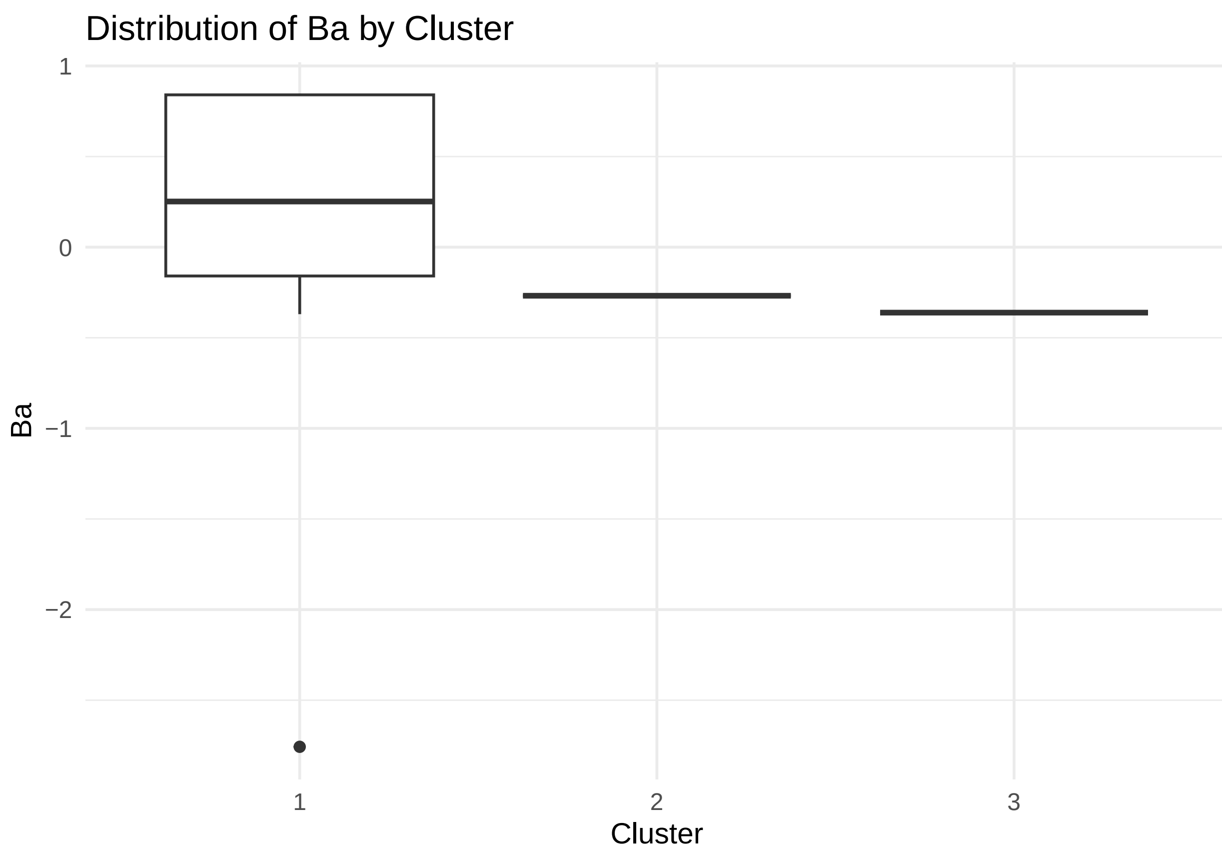
Distribution of Mg by Cluster
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Mg
1
2
3
Cluster

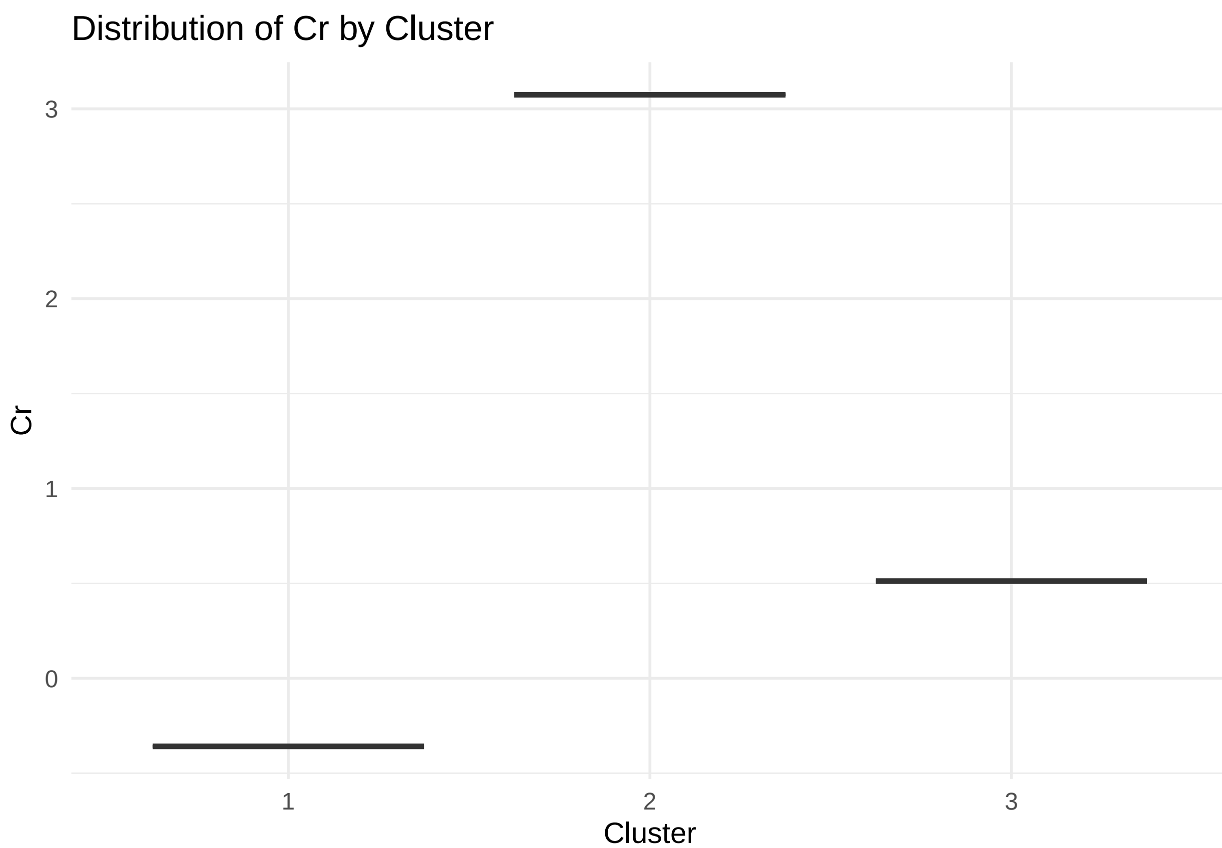

Distribution of Ca by Cluster
0
−1
−2
−3
Ca
1
2
3
Cluster

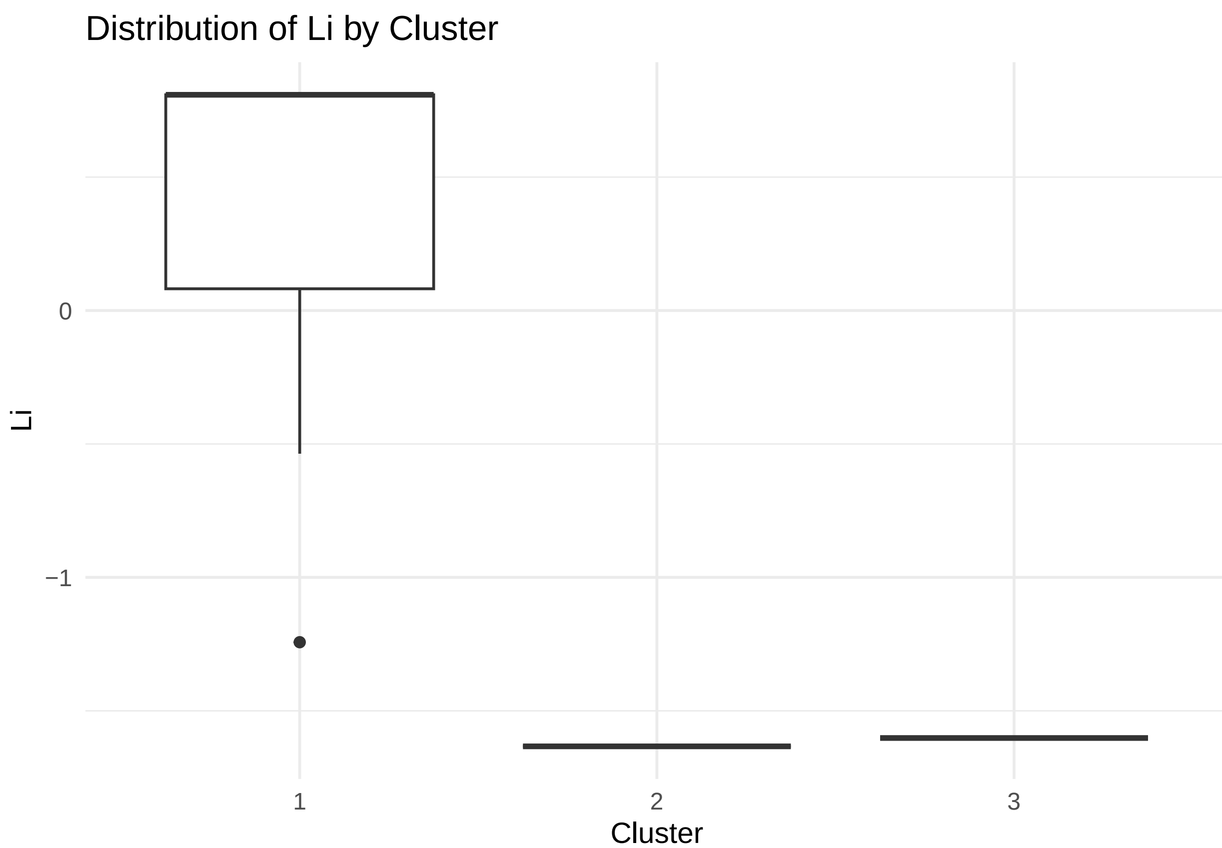

Distribution of Alkalinity by Cluster
0
−1
Alkalinity
1
2
3
Cluster

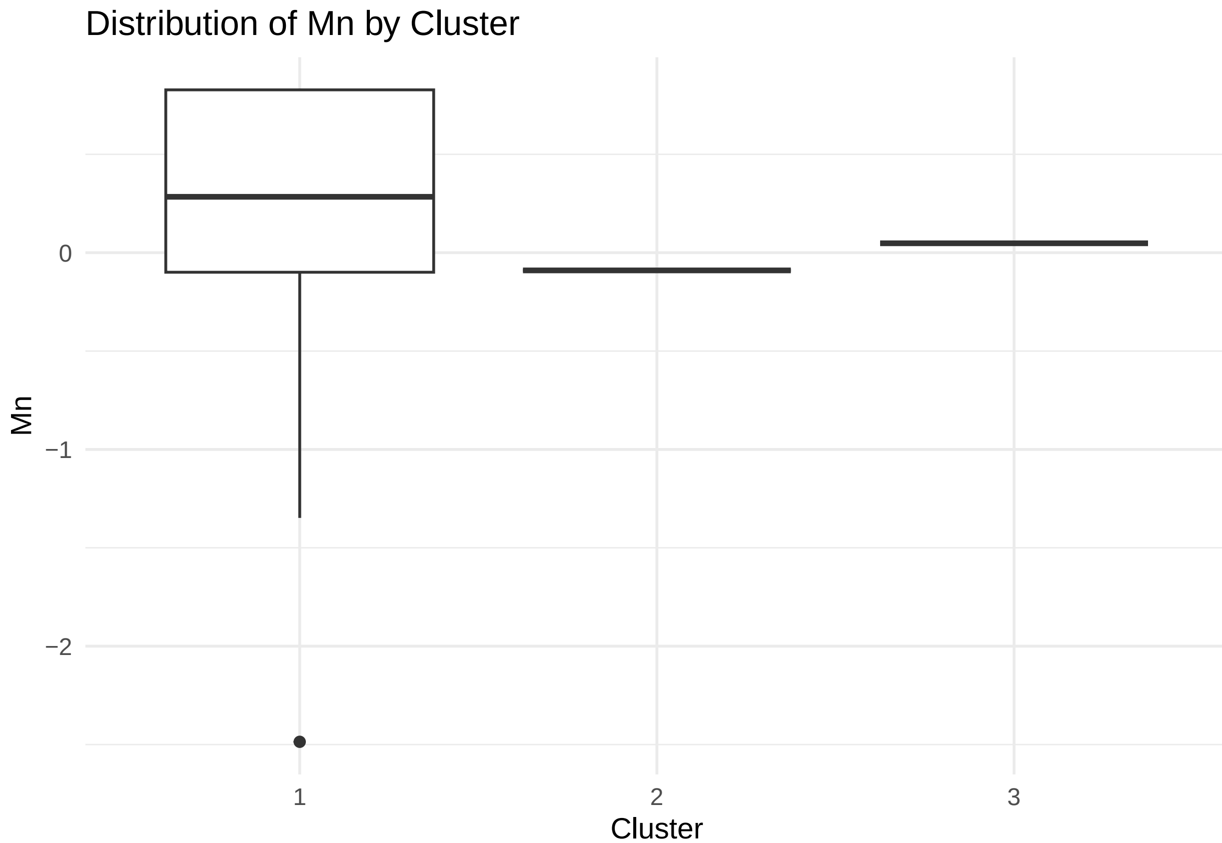

Distribution of DOC by Cluster
DOC
1.0
0.5
0.0
−0.5
−1.0
1
2
3
Cluster

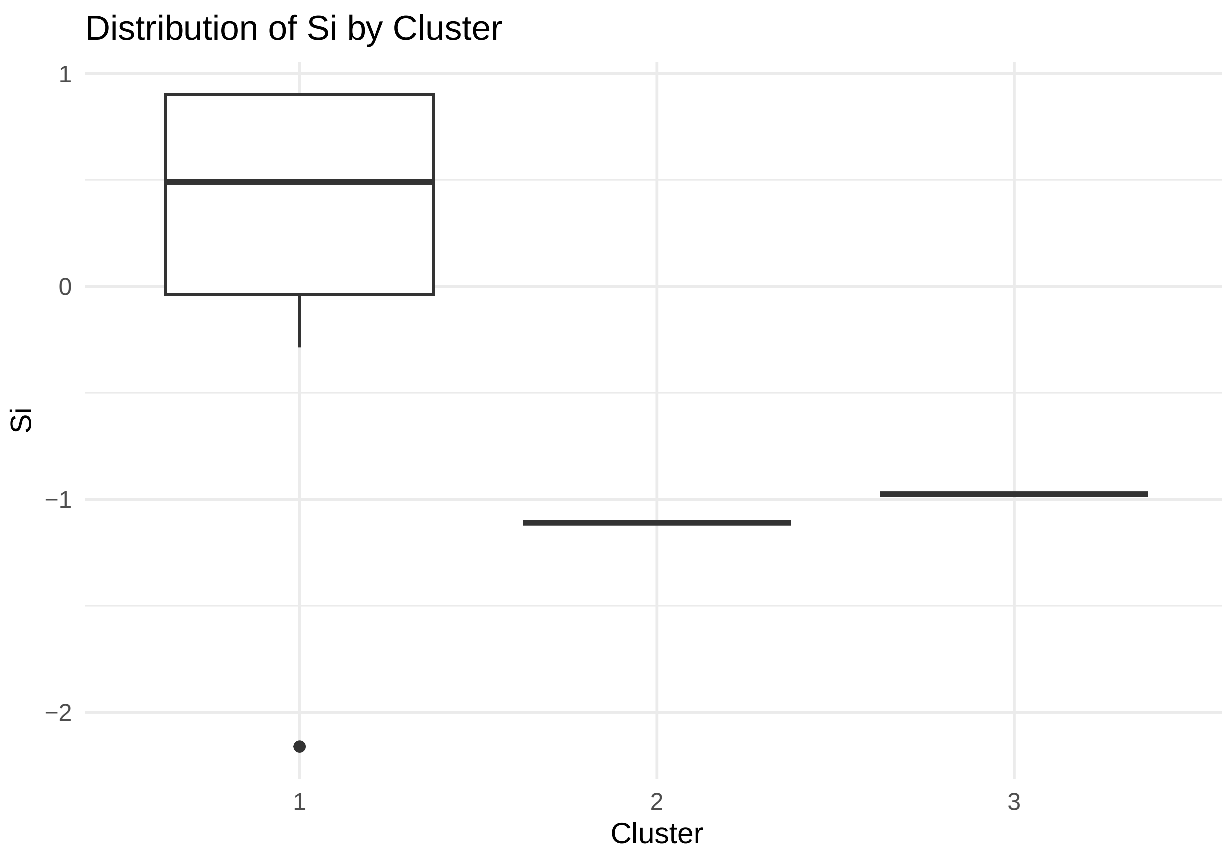

Distribution of As by Cluster
As
1
0
−1
−2
1
2
3
Cluster

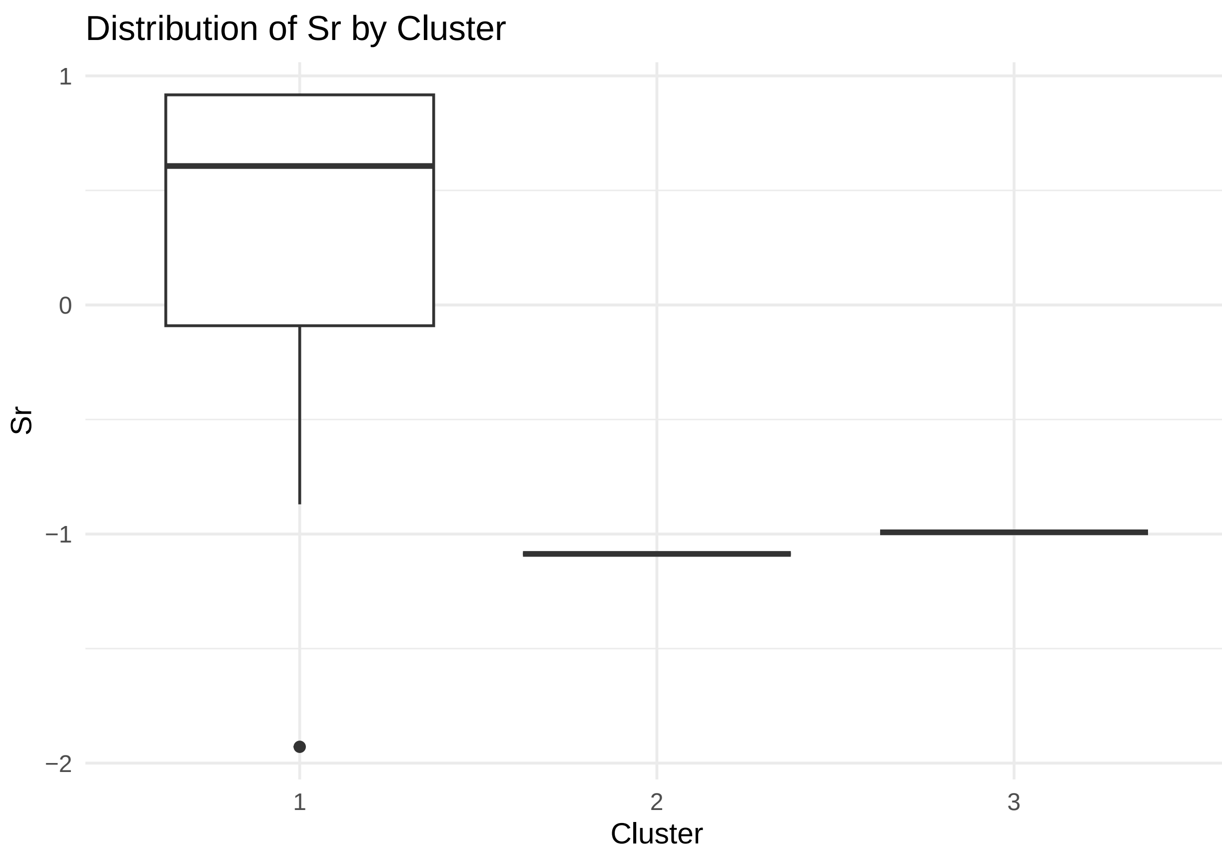

Distribution of Ba by Cluster
1
0
−1
−2
Ba
1
2
3
Cluster

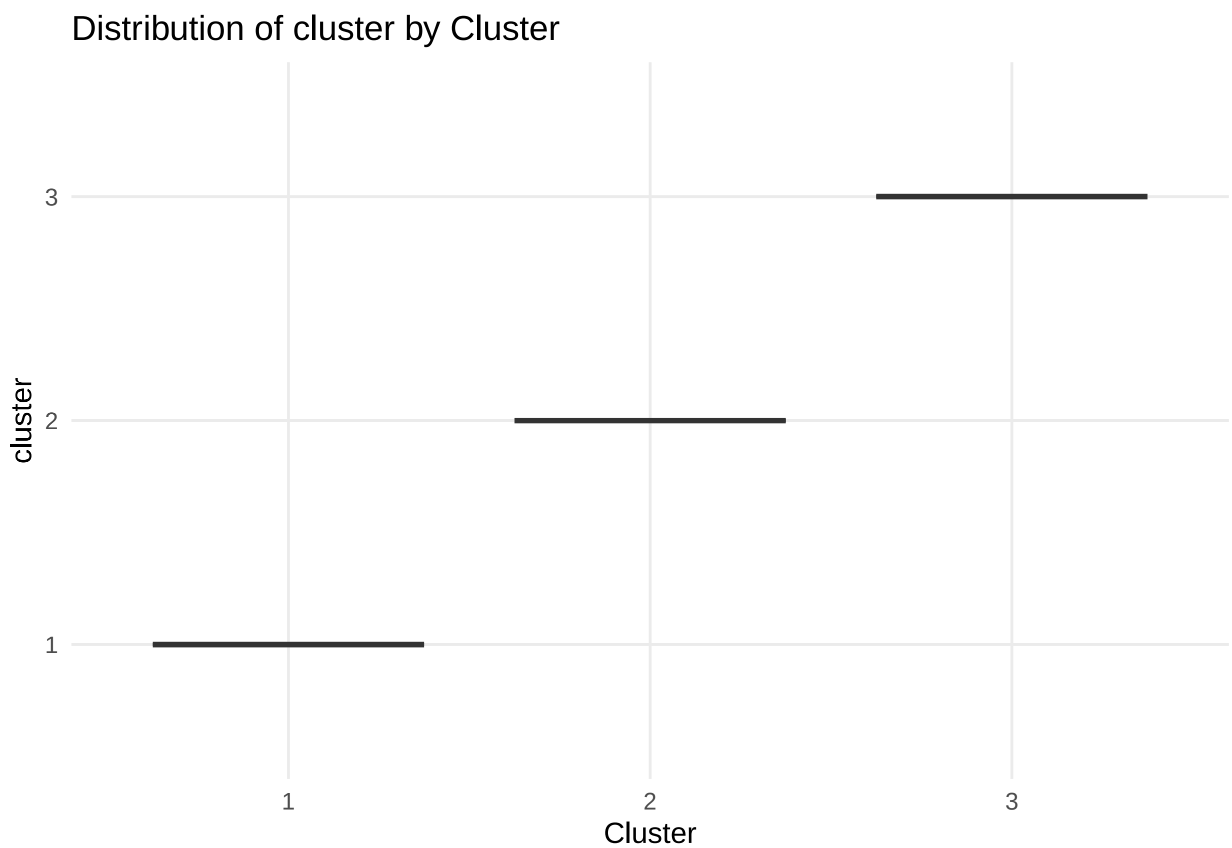

Distribution of Cr by Cluster
Cr
3
2
1
0
1
2
3
Cluster


Distribution of Li by Cluster
Li
0
−1
1
2
3
Cluster

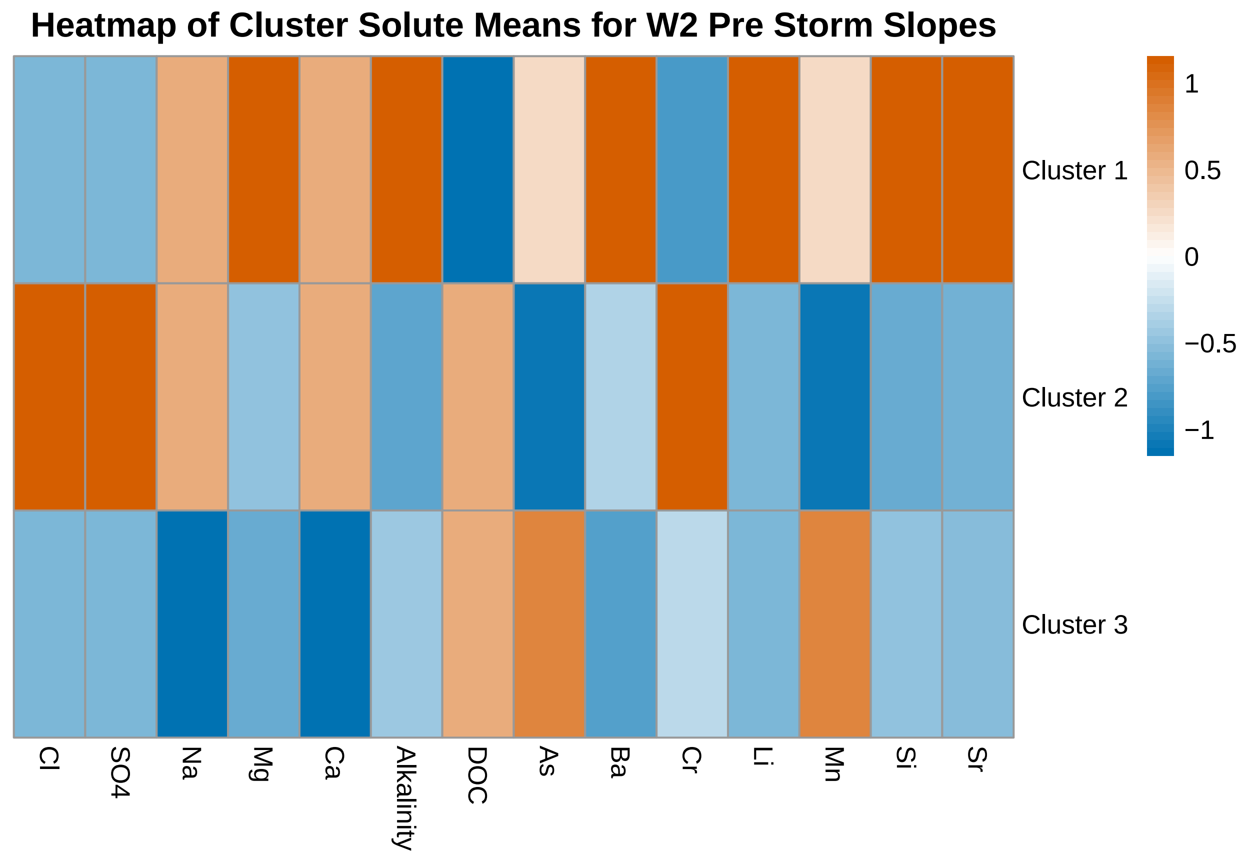

Distribution of Mn by Cluster
0
−1
−2
Mn
1
2
3
Cluster


Distribution of Si by Cluster
1
0
−1
−2
Si
1
2
3
Cluster

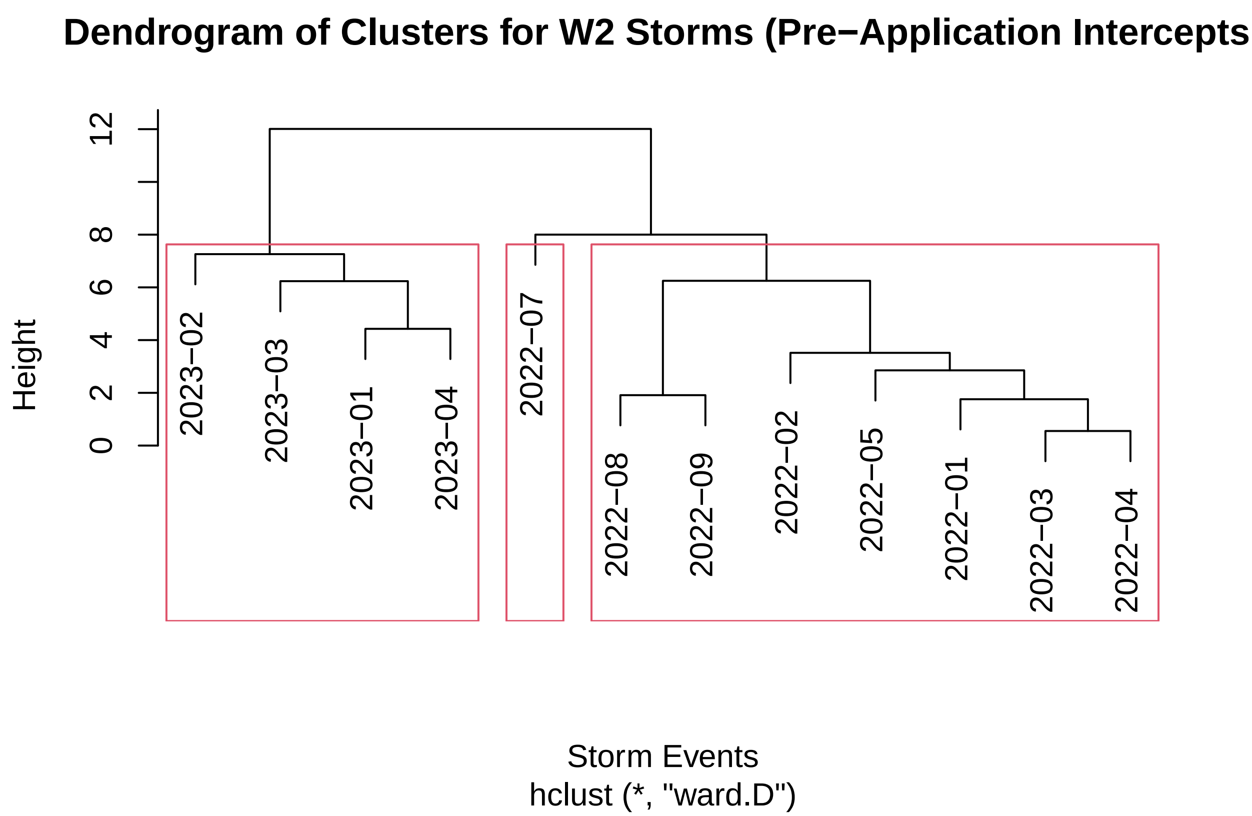

Distribution of Sr by Cluster
1
0
−1
−2
Sr
1
2
3
Cluster




```r
# Let's make some heat maps!
# Cluster Profiling: Aggregate the original variables by cluster
cluster_summary_pre_slope <- aggregate(. ~ cluster, data = w2_std_pre_slope, FUN = mean)

# Remove the cluster column for visualization
heatmap_data_pre_slope <- cluster_summary_pre_slope[, -1]

# Set row names as cluster identifiers
rownames(heatmap_data_pre_slope) <- paste("Cluster", cluster_summary_pre_slope$cluster)

# Create a heatmap
pheatmap(
  mat = as.matrix(heatmap_data_pre_slope),
  main = "Heatmap of Cluster Solute Means for W2 Pre Storm Slopes",
  cluster_rows = FALSE,  # Don't re-cluster rows (clusters are predefined)
  cluster_cols = FALSE,   # Cluster variables
  scale = "column",      # Scale variables to make them comparable
  color = colorRampPalette(c("#0072B2", "white", "#D55E00"))(50) # Gradient color scheme
)
```




```
# Hierarchical Clustering
dist_matrix_pre_intercept <- dist(w2_std_pre_intercept, method = "euclidean")
cluster_result_pre_intercept <- hclust(dist_matrix_pre_intercept, method = "ward.D")

# Plot dendrogram for visualization
plot(cluster_result_pre_intercept, labels = rownames(w2_std_pre_intercept), main = "Dendrogram of Clust
     xlab = "Storm Events")
rect.hclust(cluster_result_pre_intercept, k = 3)
```

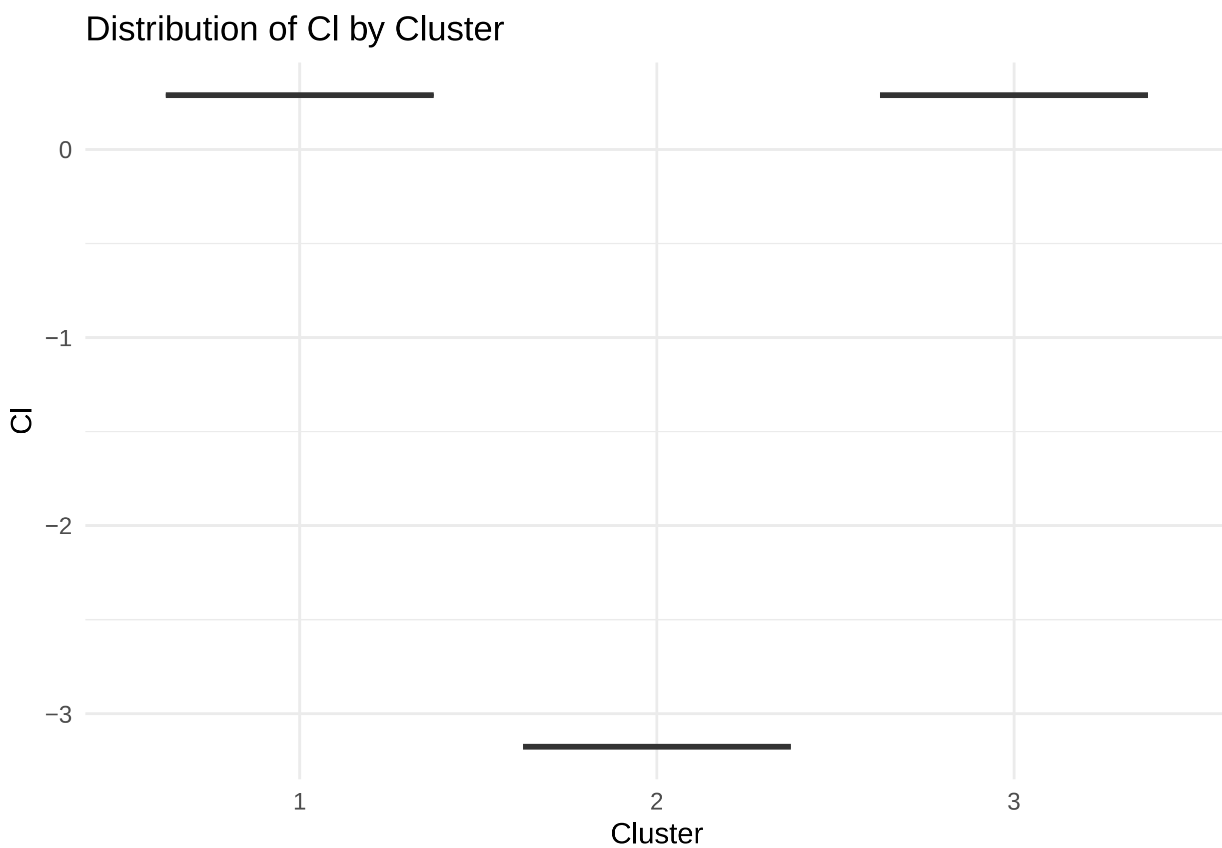


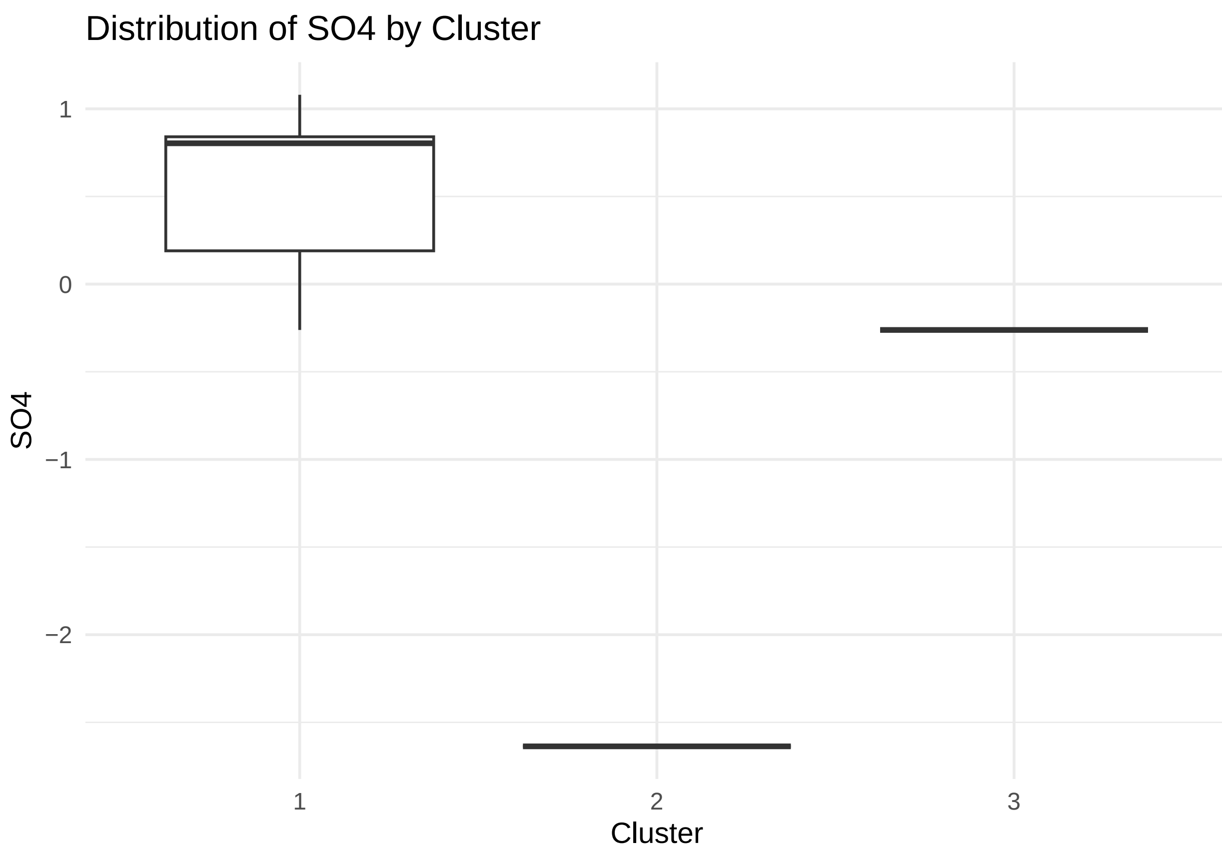

```r
# Cut the dendrogram to create 3 clusters
cuts_pre_intercept <- cutree(cluster_result_pre_intercept, k = 3)
w2_std_pre_intercept$cluster <- as.factor(cuts_pre_intercept)

# Cluster Profiling
# Aggregate the original variables by cluster to identify differences
cluster_summary_pre_intercept <- aggregate(w2_std_pre_intercept, by = list(cluster = cuts_pre_intercept)
print(cluster_summary_pre_intercept)
```

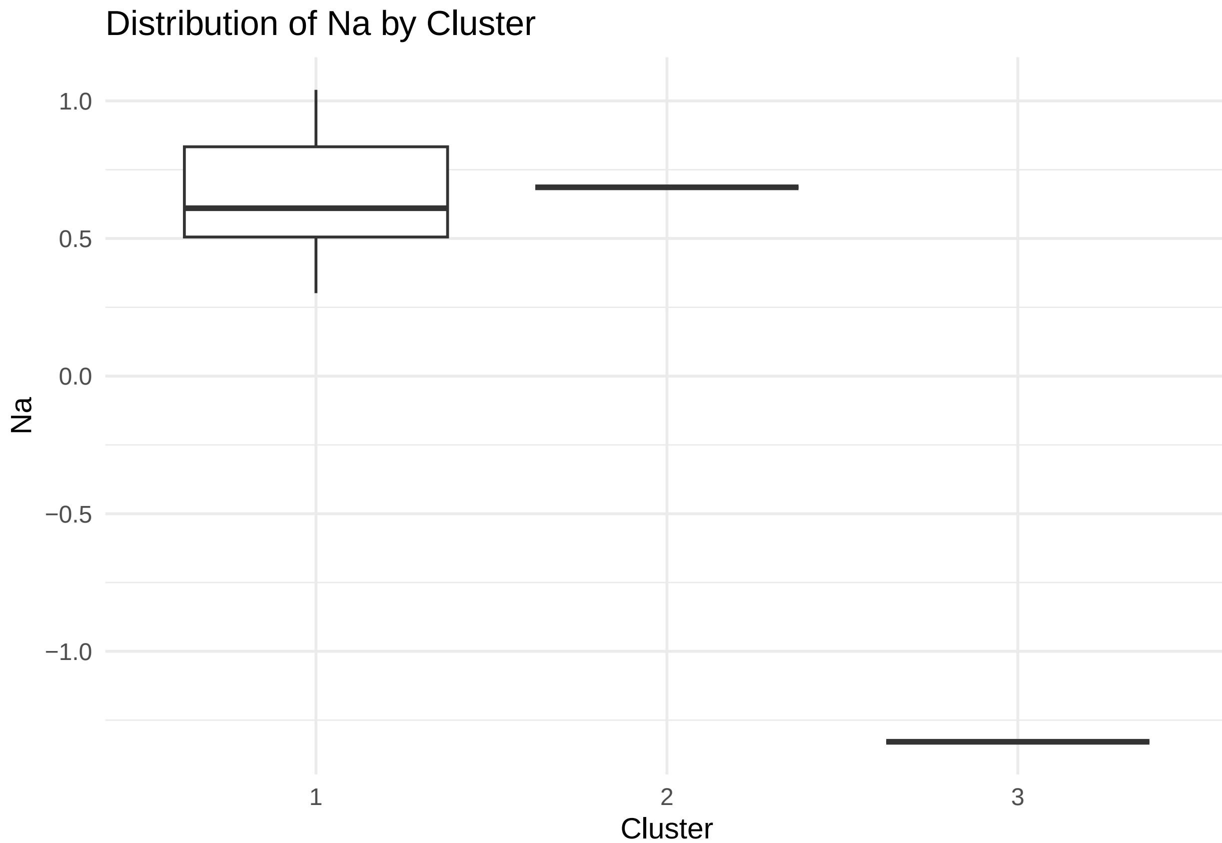
```
##   cluster         Cl        SO4         Na         Mg         Ca Alkalinity
## 1       1  0.2886751  0.5261217  0.6612486  0.6581381 -0.3827192  0.3918194
## 2       2 -3.1754265 -2.6369583  0.6859073  0.7942569  0.6214634  1.3448539
## 3       3  0.2886751 -0.2614734 -1.3286619 -1.3503060  0.5143928 -1.0218975
##          DOC         As          Ba          Cr          Li         Mn
## 1  0.4016493  0.1417884  0.06178705 -0.03883597 -0.08148011 -0.1702212
## 2  1.0985922  1.6350991  0.25818260 -1.53744197  0.11407215  0.2044175
## 3 -0.9775343 -0.6569045 -0.17267299  0.45232345  0.11407215  0.2467827
##             Si         Sr cluster
## 1 -0.05508394  0.4001253      NA
## 2 -0.37685749  0.9696541      NA
## 3  0.19061127 -0.9426328      NA
```

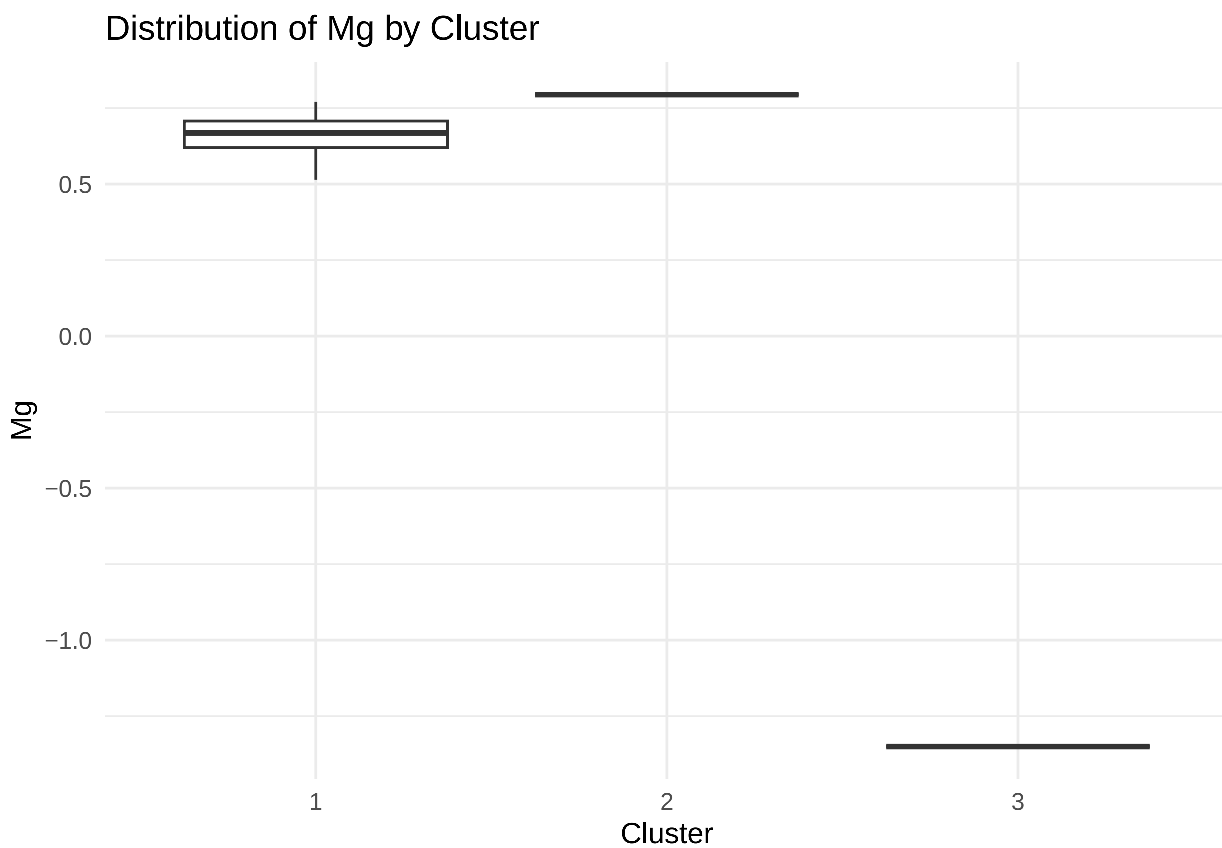
```r
# Boxplots to visualize differences in key variables across clusters
for (variable in colnames(w2_std_pre_intercept)) {
  ggplot(w2_std_pre_intercept, aes(x = as.factor(cuts_pre_intercept), y = .data[[variable]])) +
```

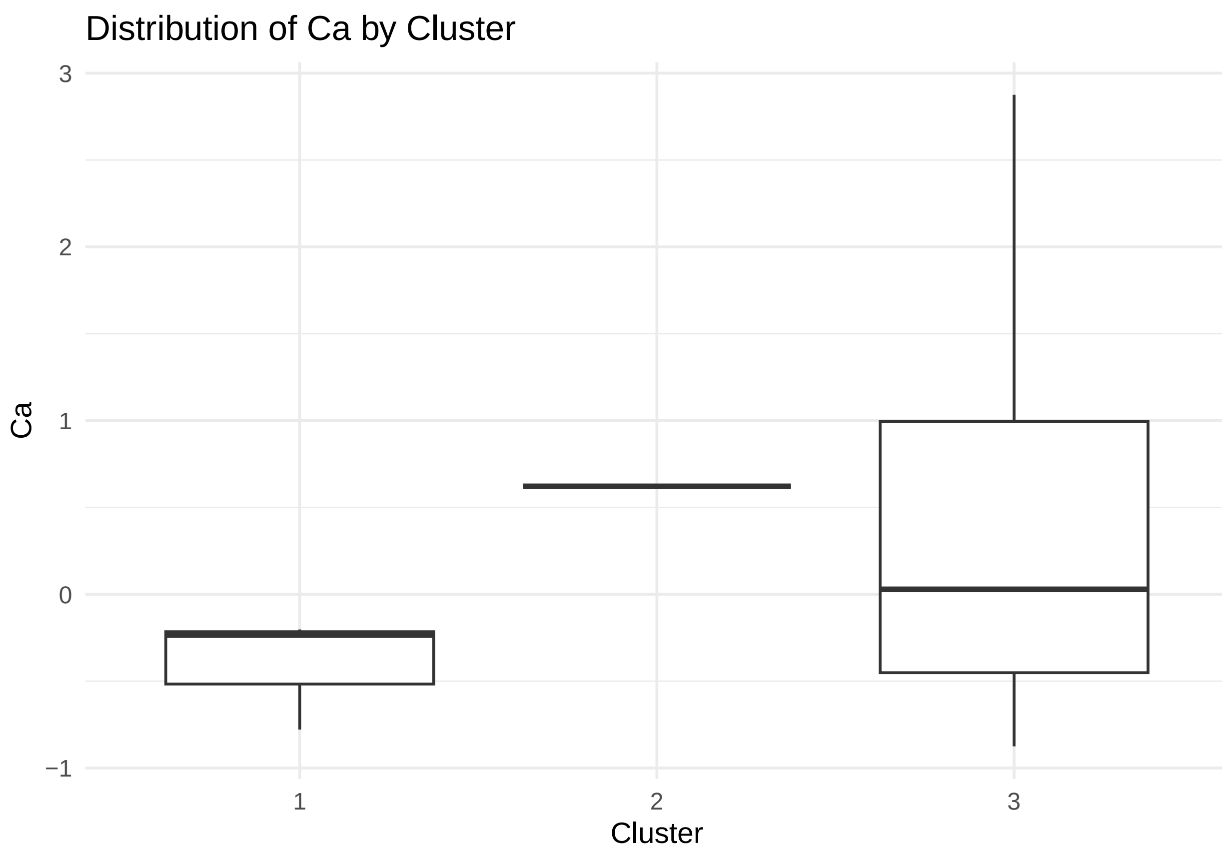
```r
    geom_boxplot() +
    labs(title = paste("Distribution of", variable, "by Cluster"),
         x = "Cluster",
         y = variable) +
    theme_minimal() -> plot
  print(plot)
}
```

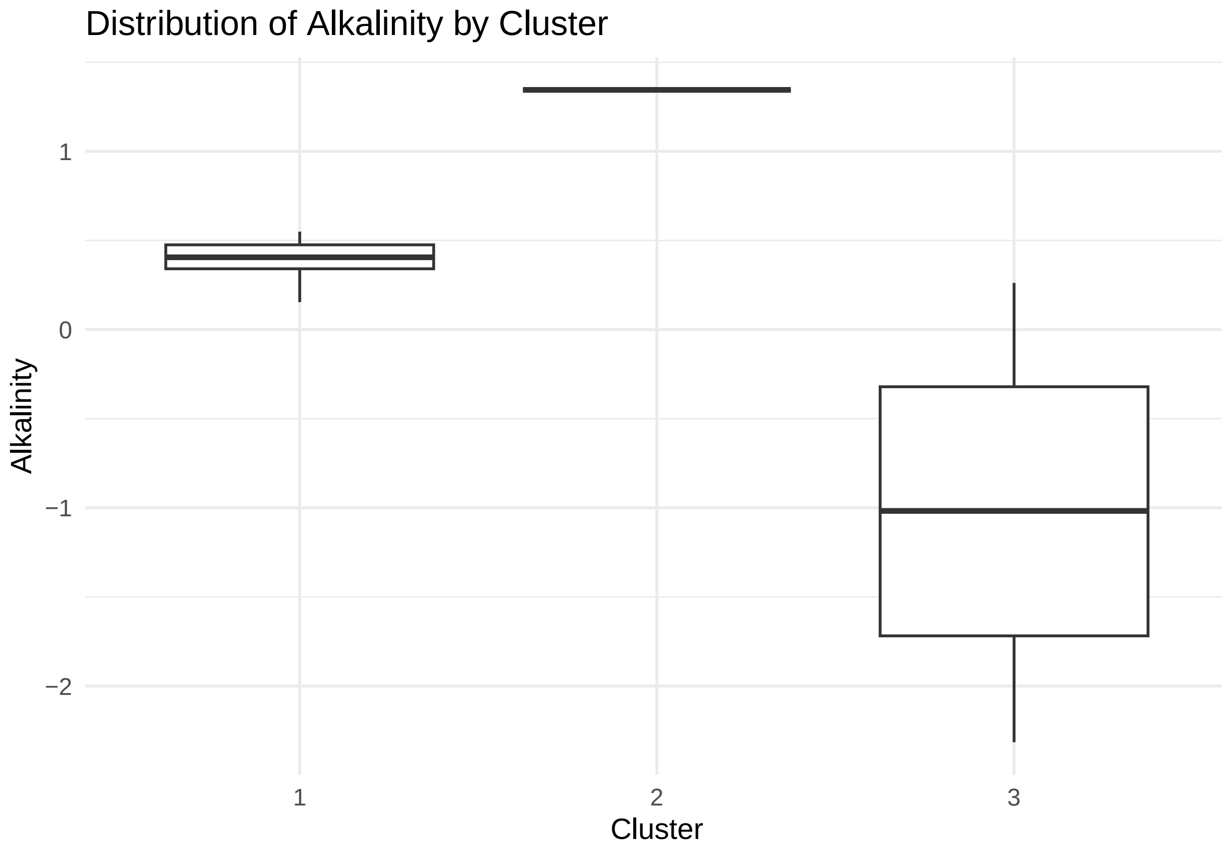


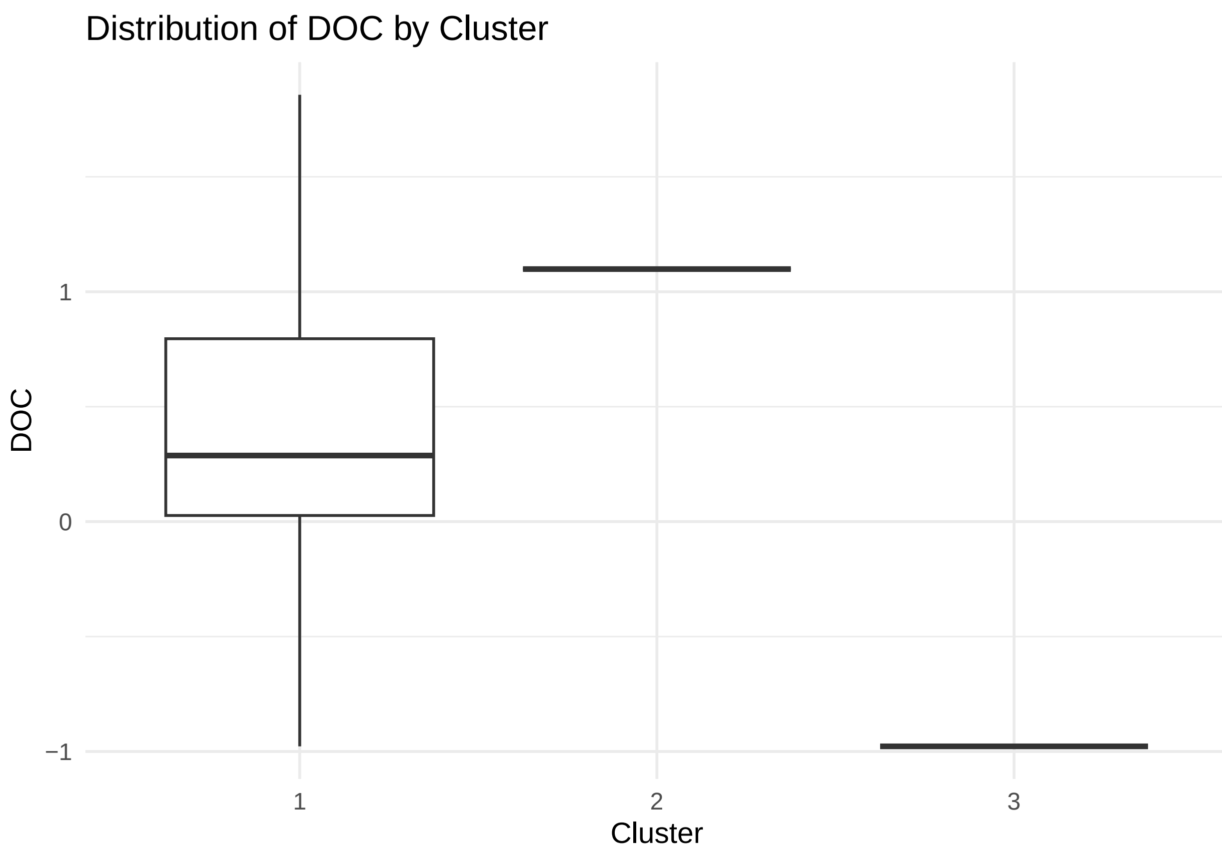

Distribution of SO4 by Cluster
1
0
−1
−2
SO4
1
2
3
Cluster

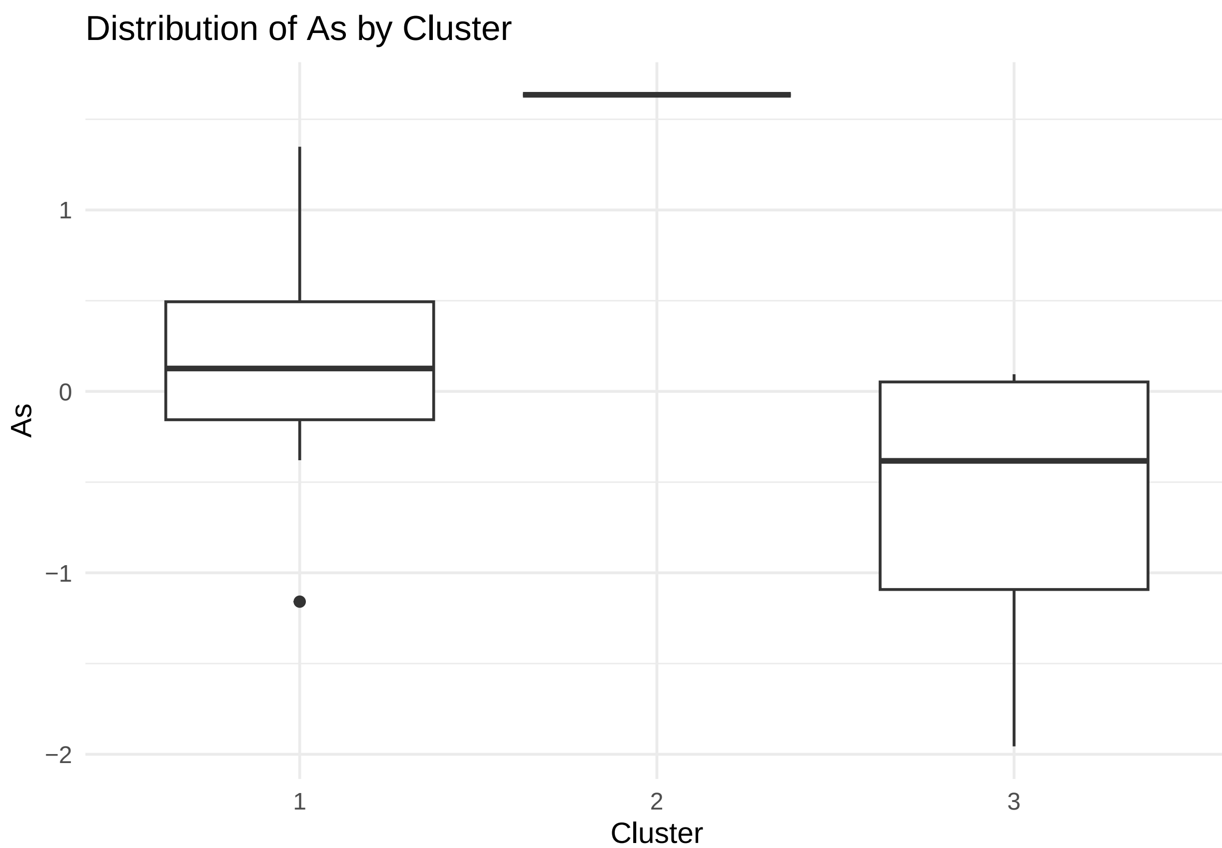

Distribution of Na by Cluster
Na
1.0
0.5
0.0
−0.5
−1.0
1
2
3
Cluster


Distribution of Mg by Cluster
Mg
0.5
0.0
−0.5
−1.0
1
2
3
Cluster

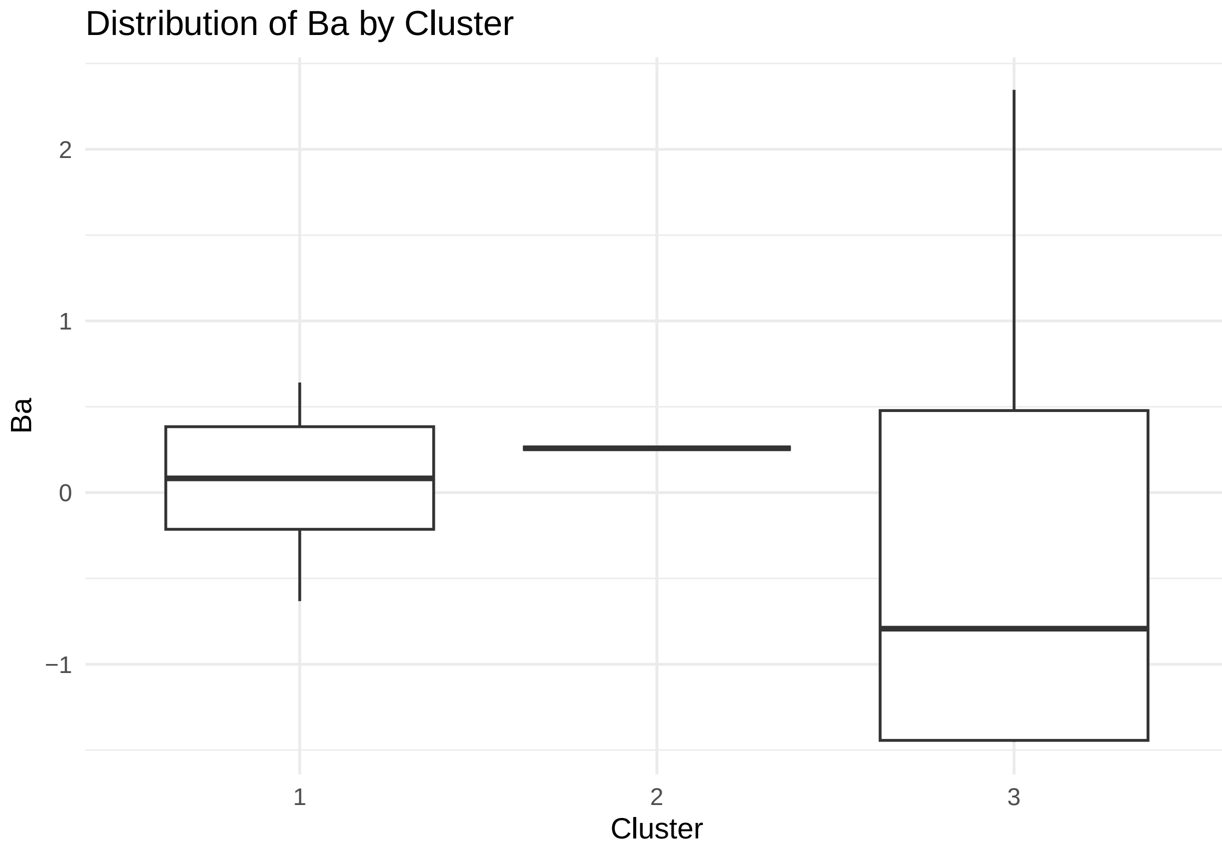

Distribution of Ca by Cluster
Ca
3
2
1
0
−1
1
2
3
Cluster

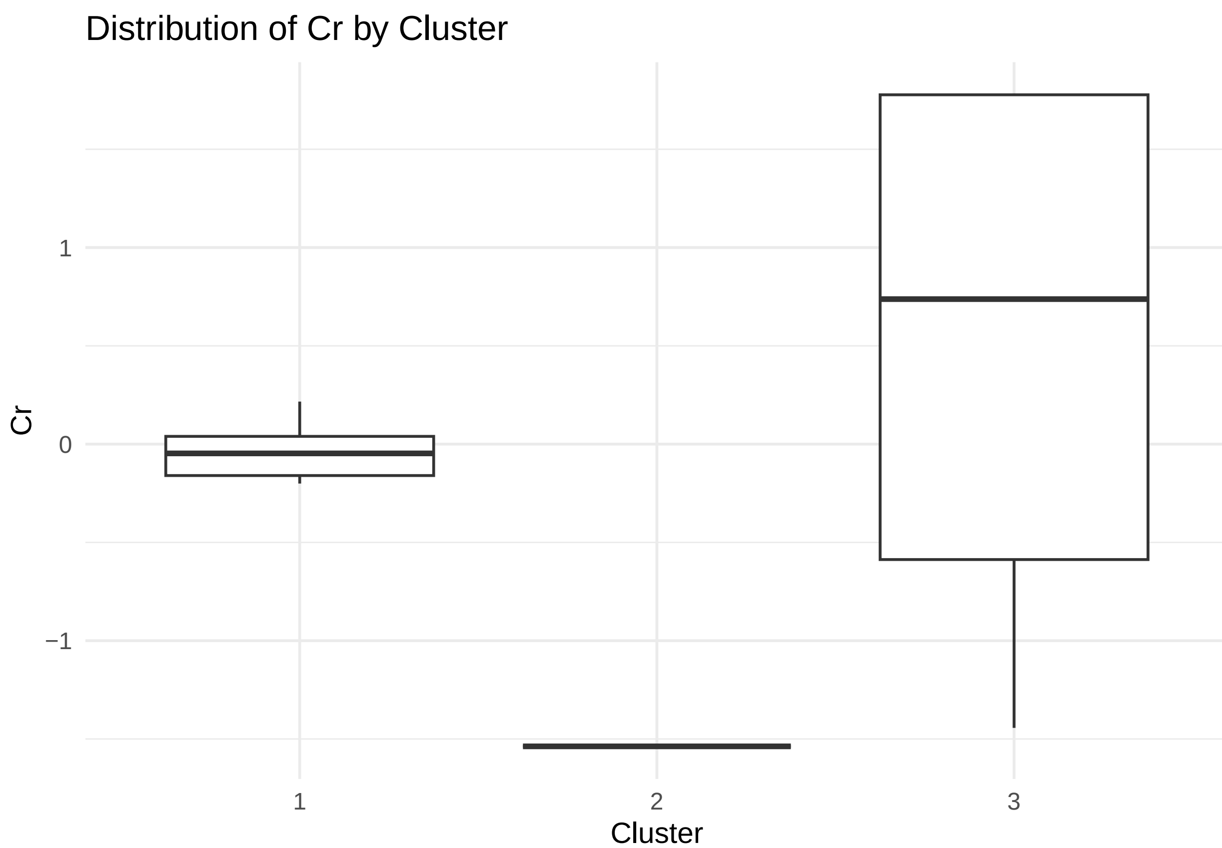

Distribution of Alkalinity by Cluster
1
0
−1
−2
Alkalinity
1
2
3
Cluster

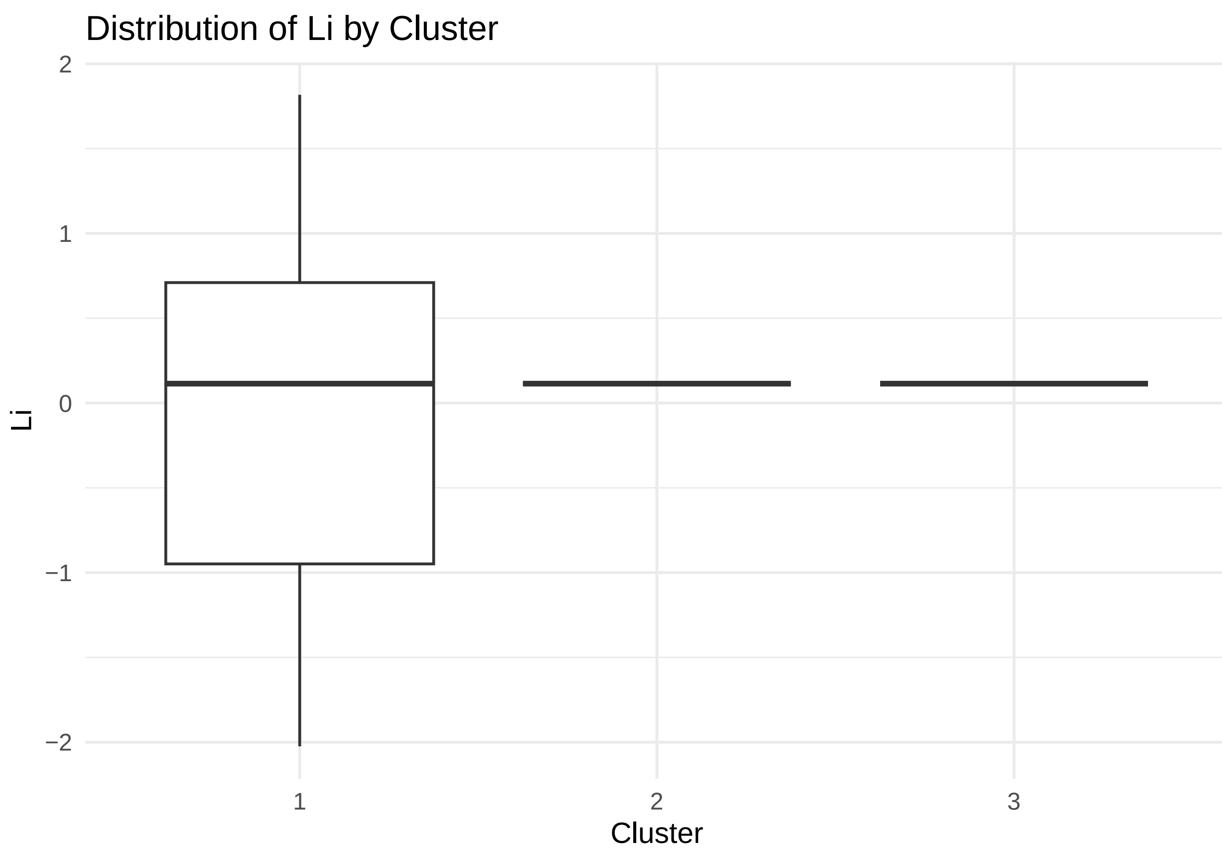

Distribution of DOC by Cluster
DOC
1
0
−1
1
2
3
Cluster

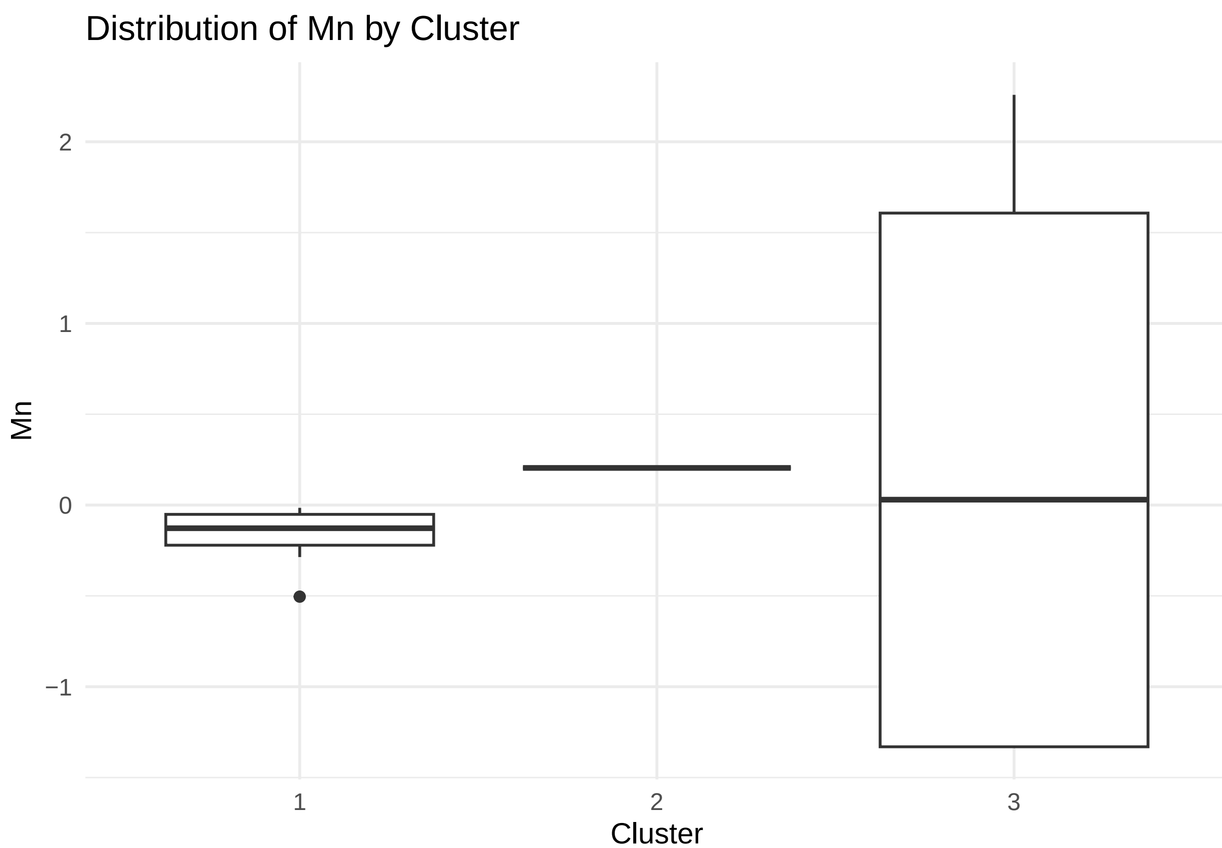


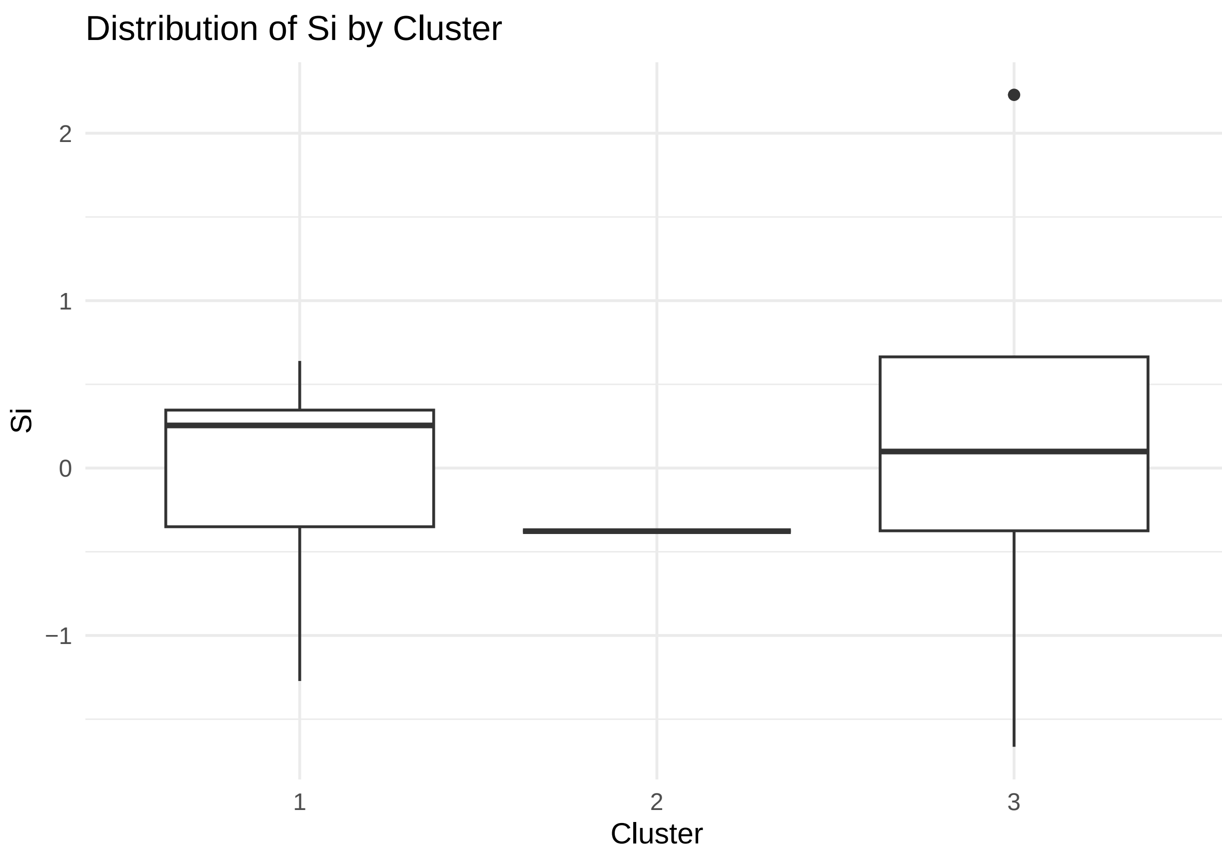
Distribution of As by Cluster
As
1
0
−1
−2
1
2
3
Cluster

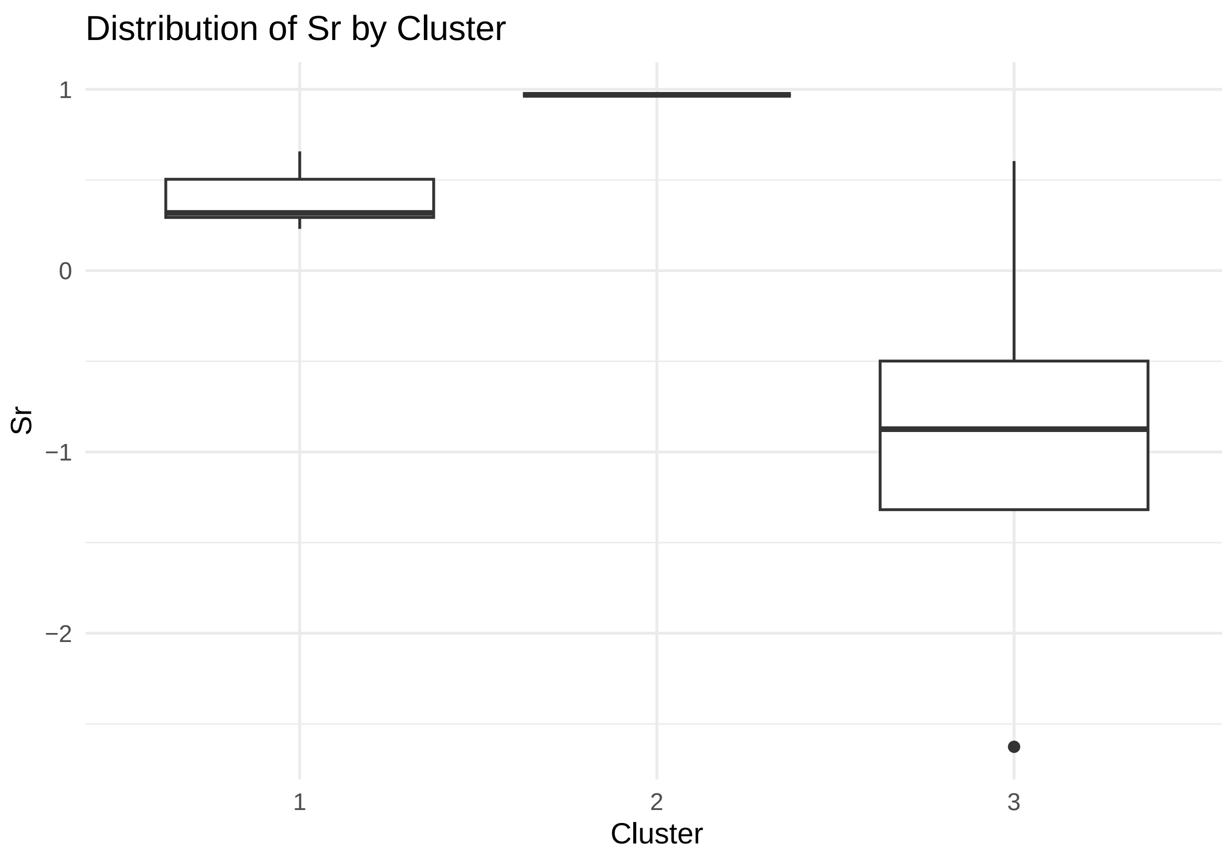

Distribution of Ba by Cluster
2
1
0
−1
Ba
1
2
3
Cluster

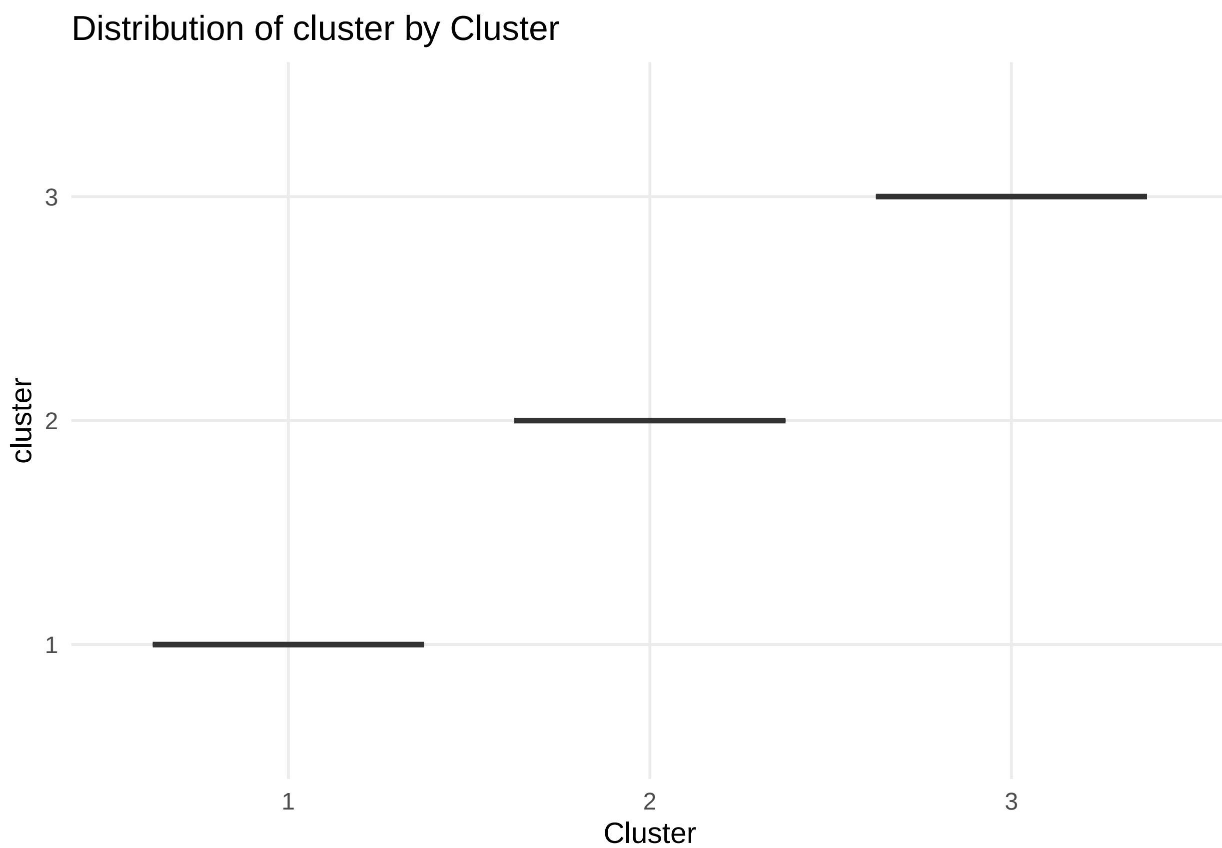

Distribution of Cr by Cluster
Cr
1
0
−1
1
2
3
Cluster


Distribution of Li by Cluster
2
1
0
−1
−2
Li
1
2
3
Cluster

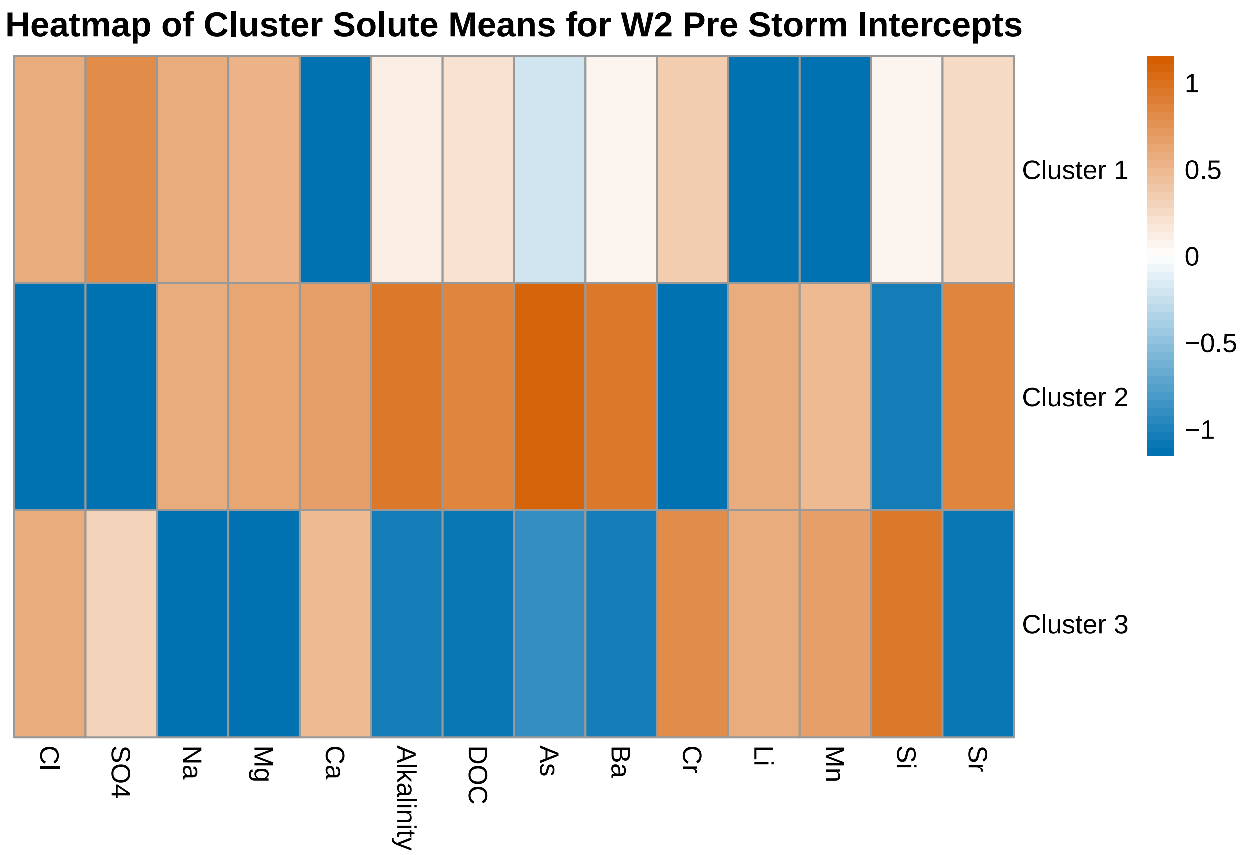

Distribution of Mn by Cluster
Mn
2
1
0
−1
1
2
3
Cluster


Distribution of Si by Cluster
Si
2
1
0
−1
1
2
3
Cluster

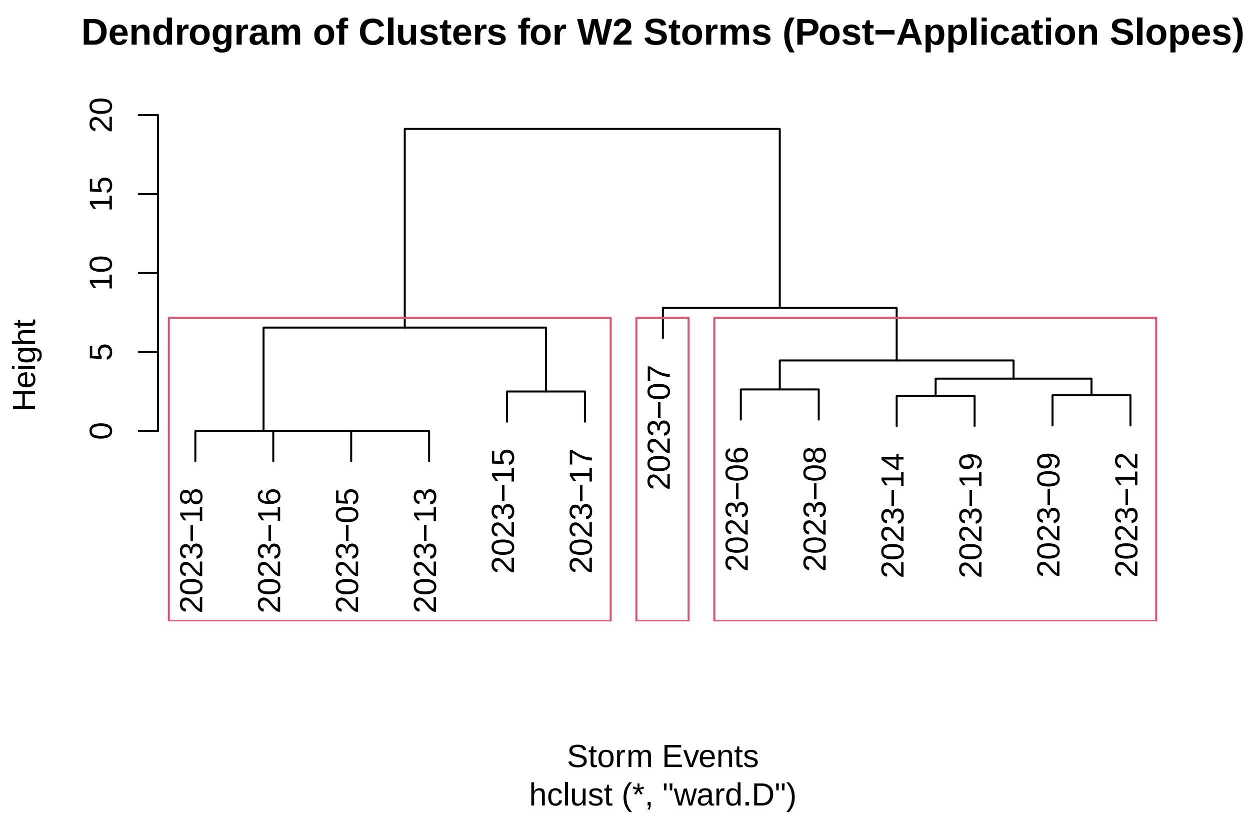

Distribution of Sr by Cluster
1
0
−1
−2
Sr
1
2
3
Cluster




```r
# Let's make some heat maps!
# Cluster Profiling: Aggregate the original variables by cluster
cluster_summary_pre_intercept <- aggregate(. ~ cluster, data = w2_std_pre_intercept, FUN = mean)

# Remove the cluster column for visualization
heatmap_data_pre_intercept <- cluster_summary_pre_intercept[, -1]

# Set row names as cluster identifiers
rownames(heatmap_data_pre_intercept) <- paste("Cluster", cluster_summary_pre_intercept$cluster)

# Create a heatmap
pheatmap(
  mat = as.matrix(heatmap_data_pre_intercept),
  main = "Heatmap of Cluster Solute Means for W2 Pre Storm Intercepts",
  cluster_rows = FALSE,  # Don't re-cluster rows (clusters are predefined)
  cluster_cols = FALSE,   # Cluster variables
  scale = "column",       # Scale variables to make them comparable
  color = colorRampPalette(c("#0072B2", "white", "#D55E00"))(50) # Gradient color scheme
)
```




```r
# Hierarchical Clustering
dist_matrix_post_slope <- dist(w2_std_post_slope, method = "euclidean")
cluster_result_post_slope <- hclust(dist_matrix_post_slope, method = "ward.D")

# Plot dendrogram for visualization
plot(cluster_result_post_slope, labels = rownames(w2_std_post_slope), main = "Dendrogram of Clusters fo
     xlab = "Storm Events")
rect.hclust(cluster_result_post_slope, k = 3)
```

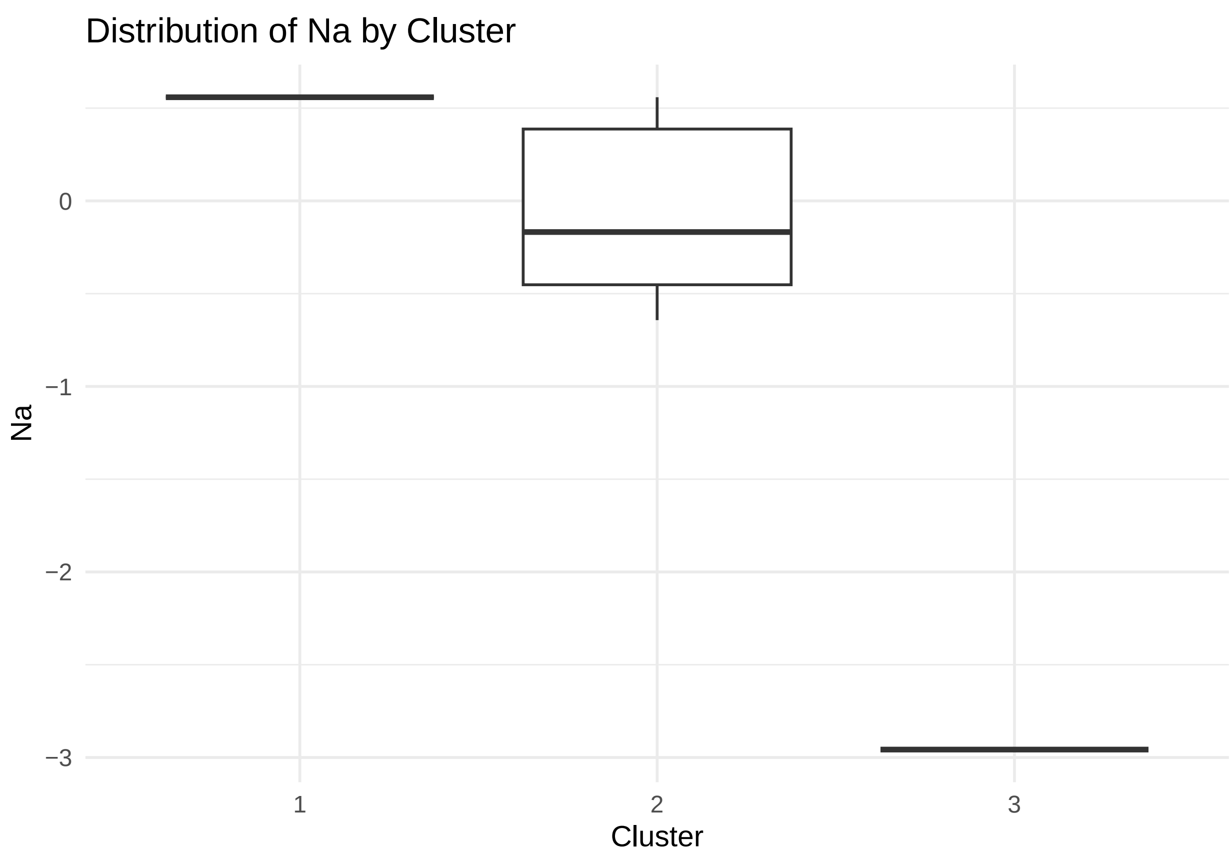


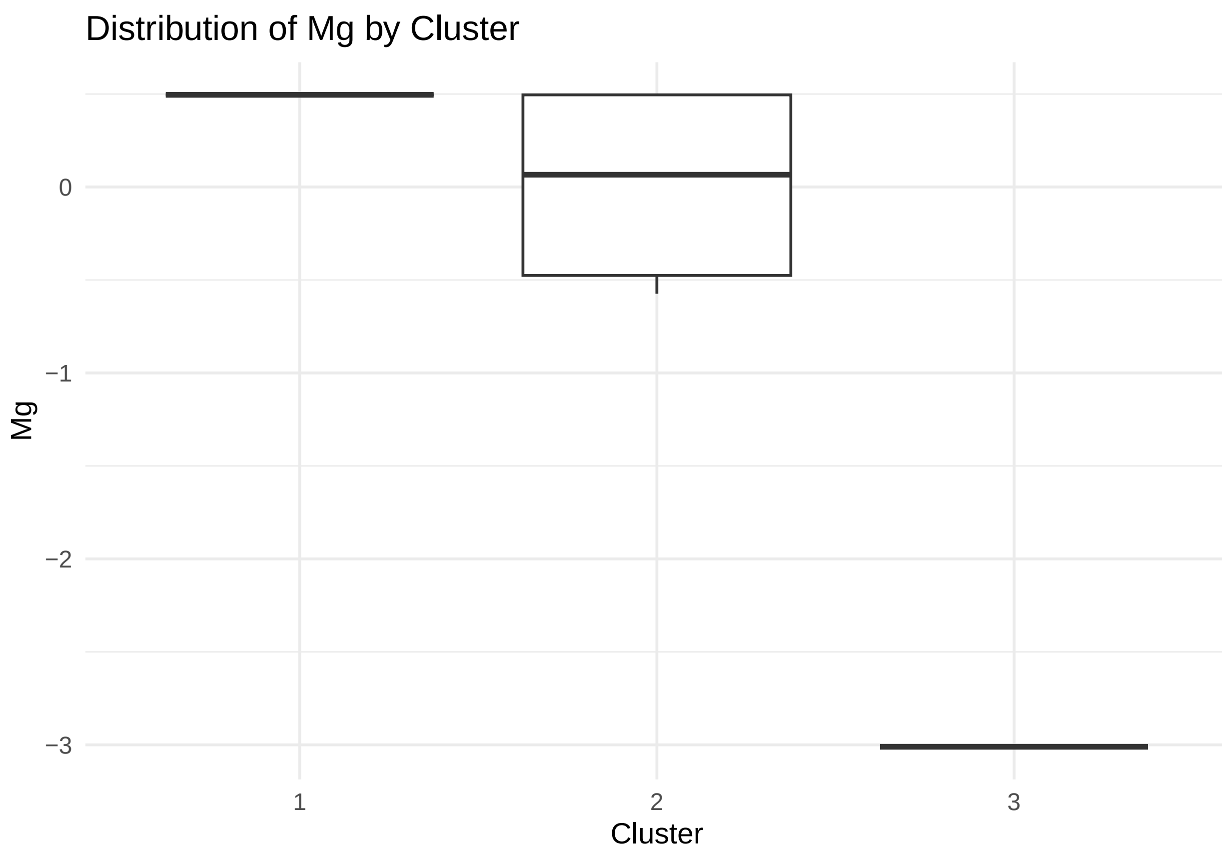

```r
# Cut the dendrogram to create 3 clusters
cuts_post_slope <- cutree(cluster_result_post_slope, k = 3)
w2_std_post_slope$cluster <- as.factor(cuts_post_slope)

# Cluster Profiling
# Aggregate the original variables by cluster to identify differences
cluster_summary_post_slope <- aggregate(w2_std_post_slope, by = list(cluster = cuts_post_slope), FUN = r
print(cluster_summary_post_slope)
```

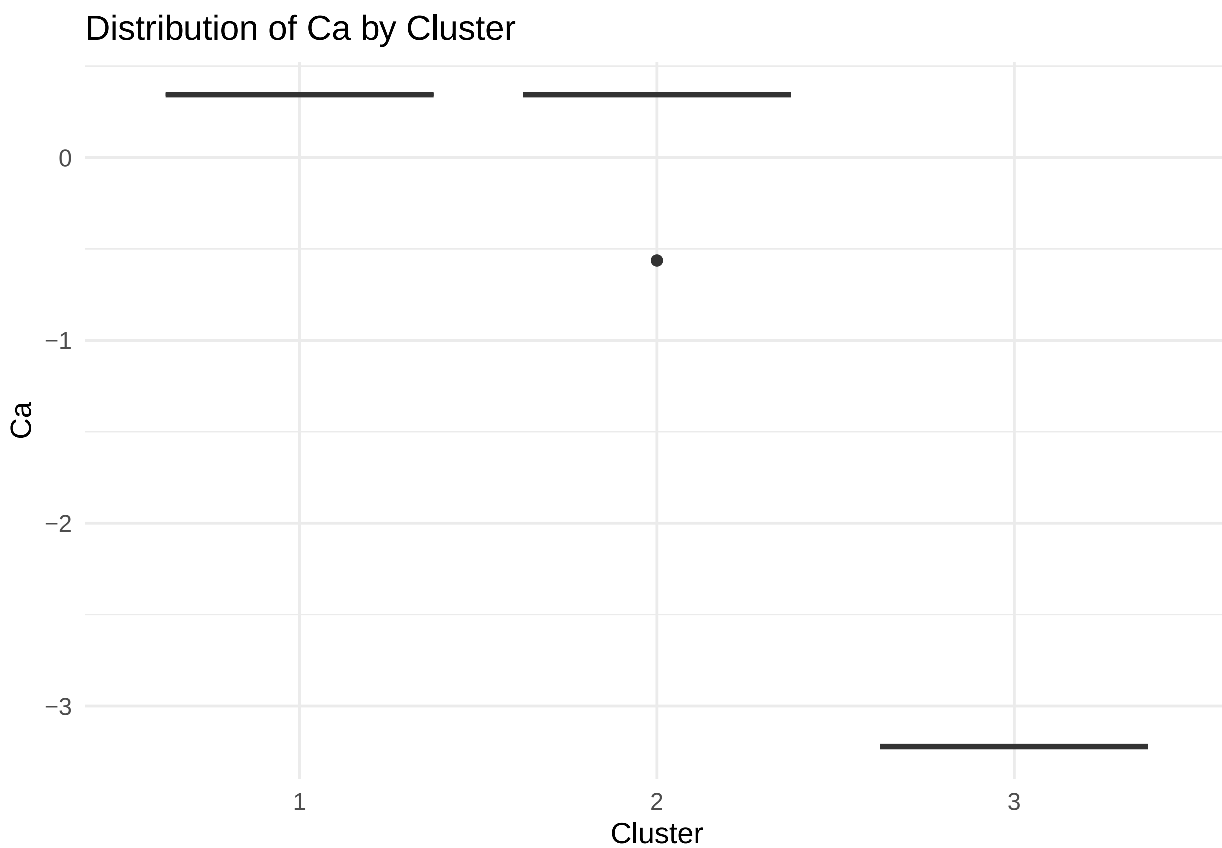
```
##   cluster         Na           Mg         Ca Alkalinity        DOC         As
## 1       1  0.5586417  0.495696619  0.3441058  0.9802014 -0.6943076 -0.5498953
## 2       2 -0.0657379  0.006068799  0.1928649 -0.7751613  0.6338358  0.4741518
## 3       3 -2.9574229 -3.010592509 -3.2218242 -1.2302407  0.3628312  0.4544612
##           Ba         Li          Mn         Si         Sr cluster
## 1  0.9737287  0.9252347  0.03598909  1.0059068  0.9699187      NA
## 2 -0.7968495 -0.7715833 -0.08505238 -0.7897407 -0.7533974      NA
## 3 -1.0612752 -0.9219088  0.29437973 -1.2969968 -1.2991275      NA
```

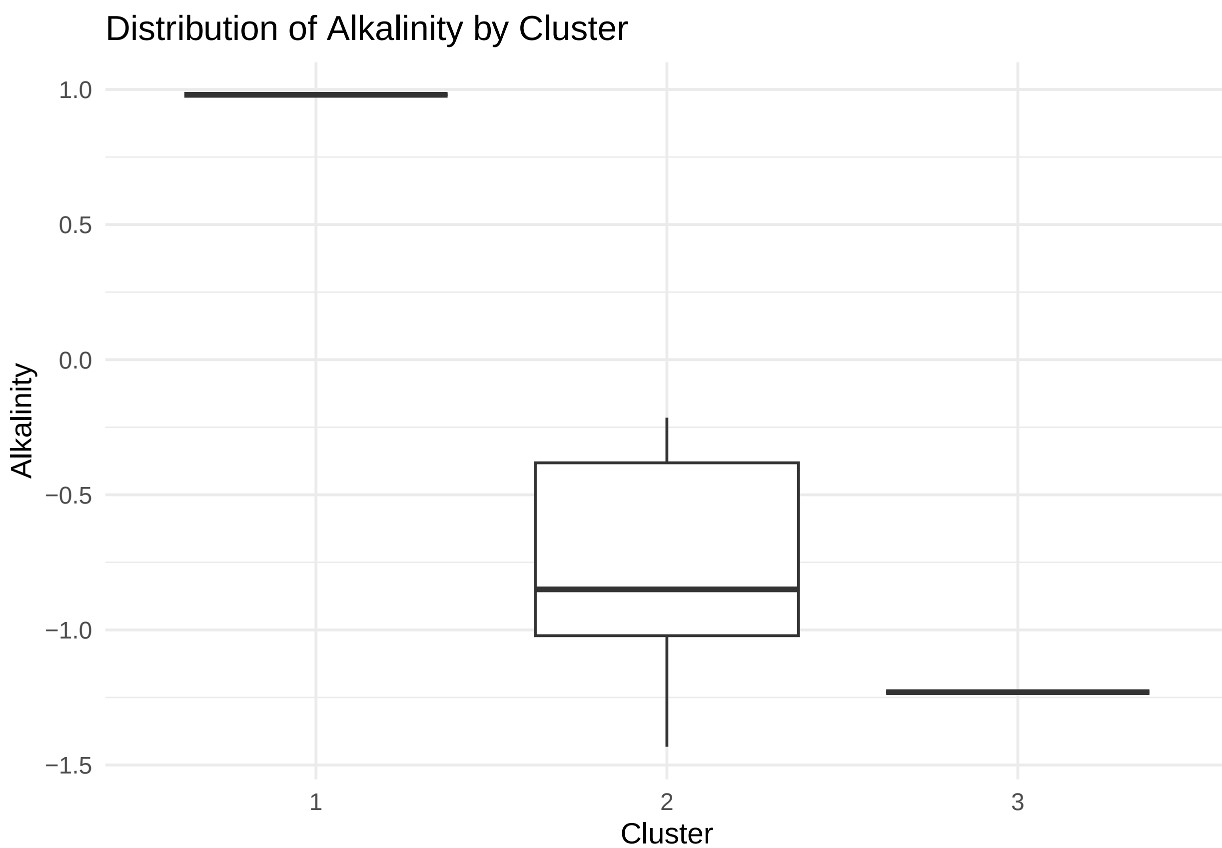
```r
# Boxplots to visualize differences in key variables across clusters
for (variable in colnames(w2_std_post_slope)) {
  ggplot(w2_std_post_slope, aes(x = as.factor(cuts_post_slope), y = .data[[variable]])) +
    geom_boxplot() +
    labs(title = paste("Distribution of", variable, "by Cluster"),
         x = "Cluster",
         y = variable) +
```

```
    theme_minimal() -> plot
  print(plot)
}
```

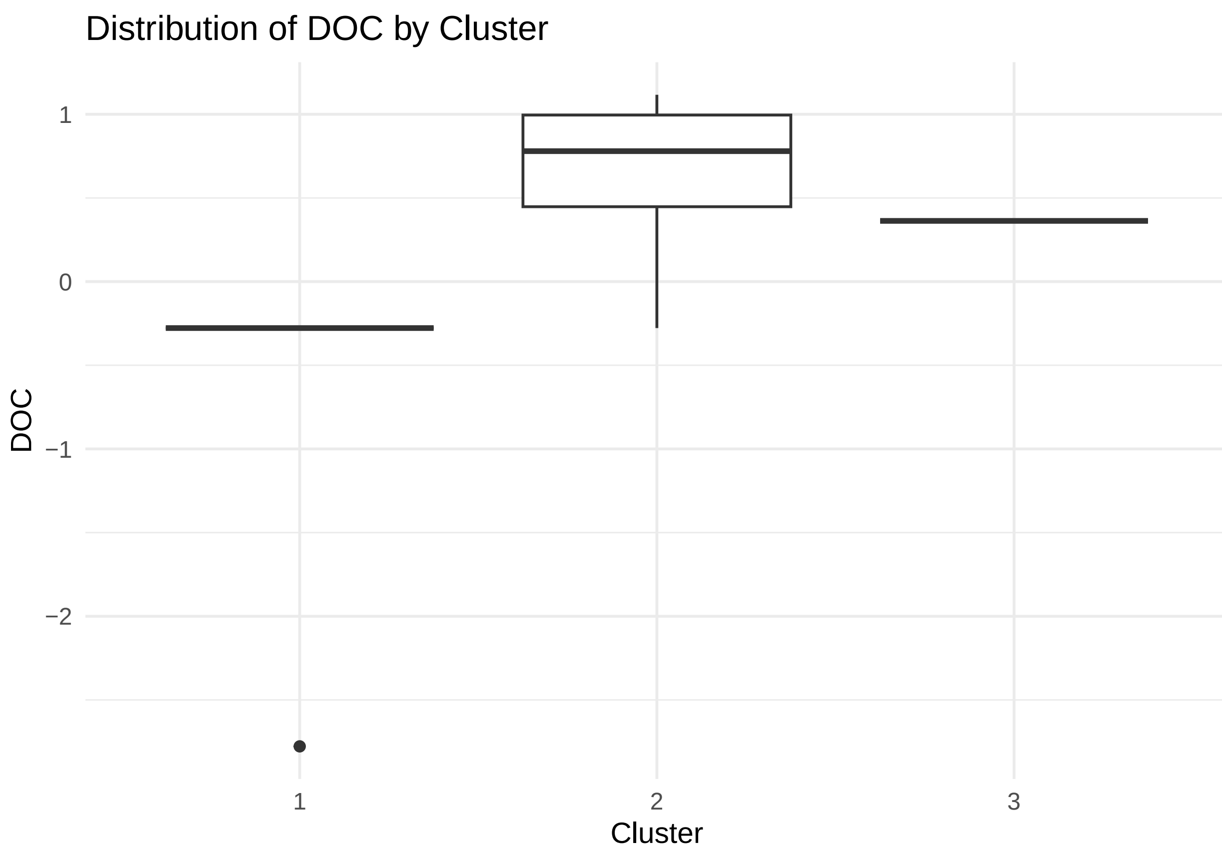



Distribution of Mg by Cluster
0
−1
−2
−3
Mg
1
2
3
Cluster

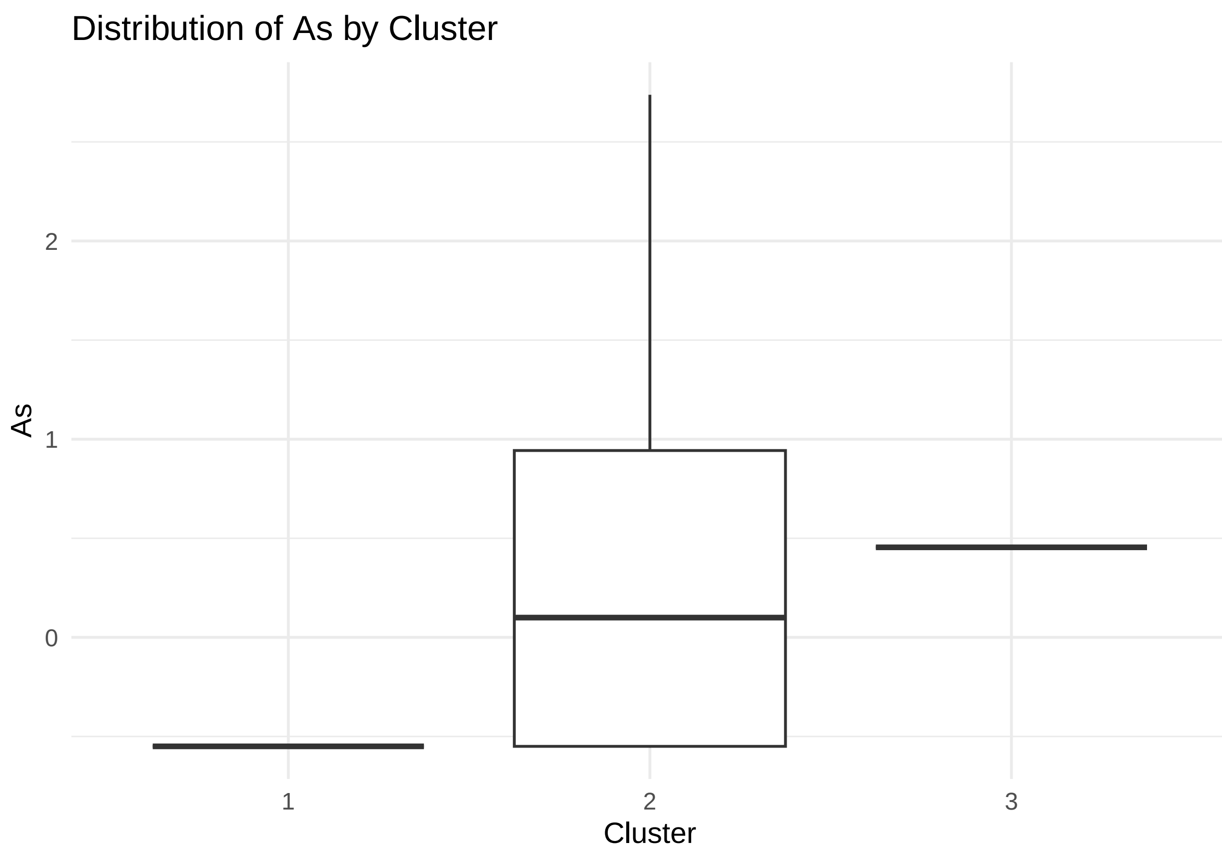

Distribution of Ca by Cluster
0
−1
−2
−3
Ca
1
2
3
Cluster

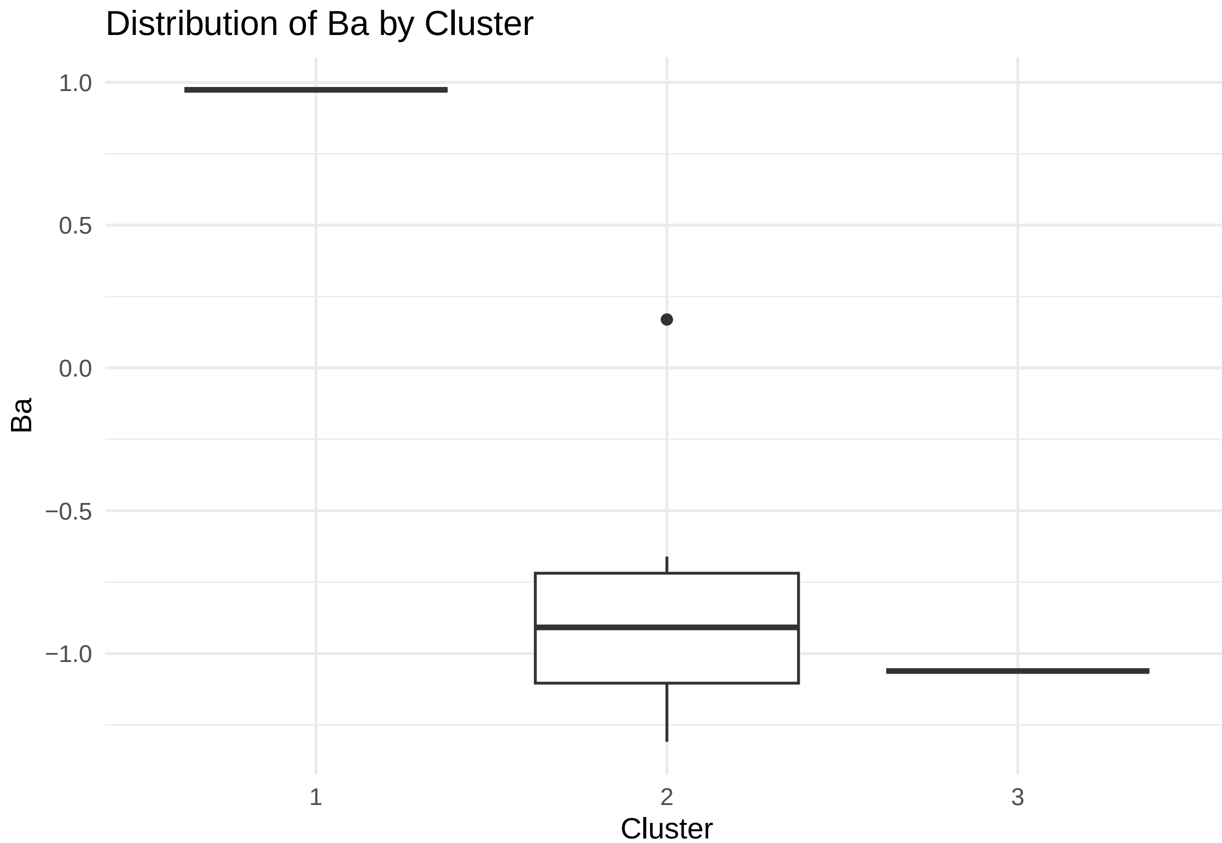


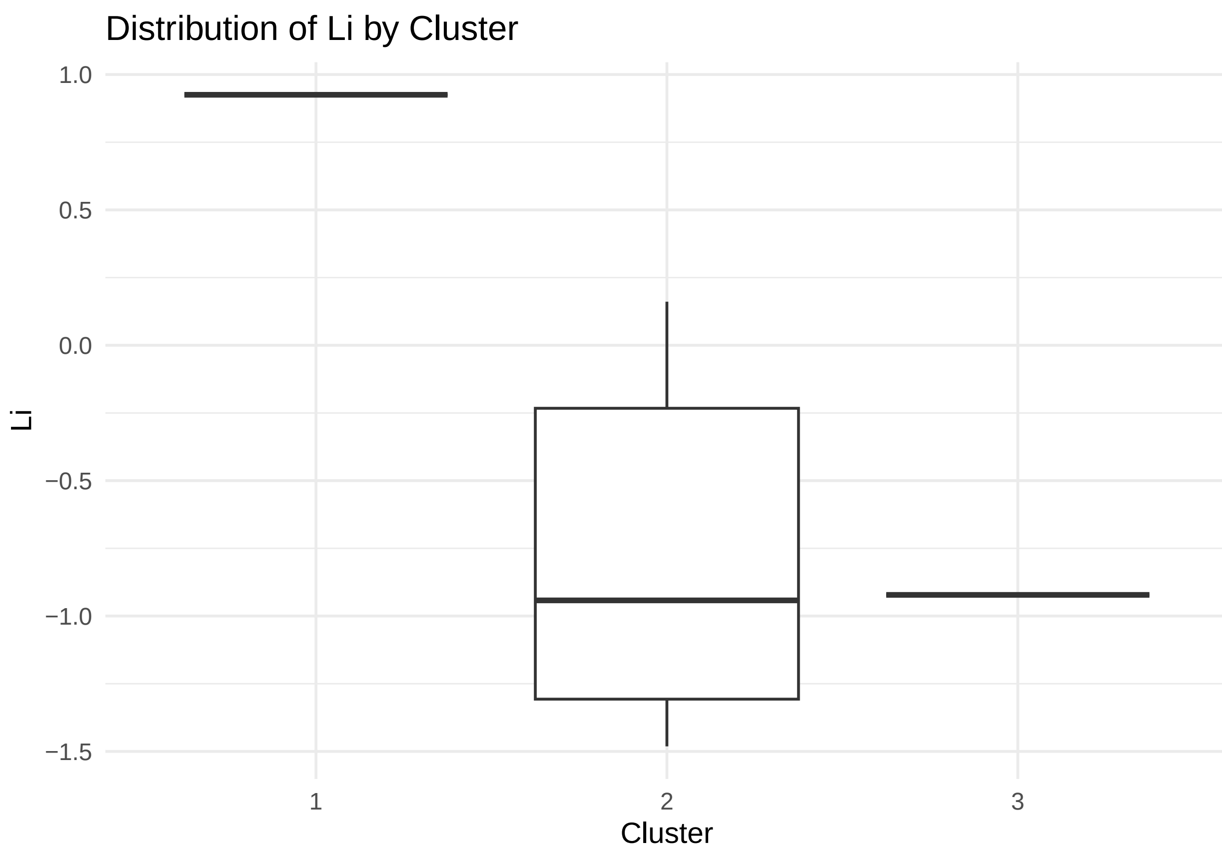
Distribution of Alkalinity by Cluster
1.0
0.5
0.0
−0.5
−1.0
−1.5
Alkalinity
1
2
3
Cluster

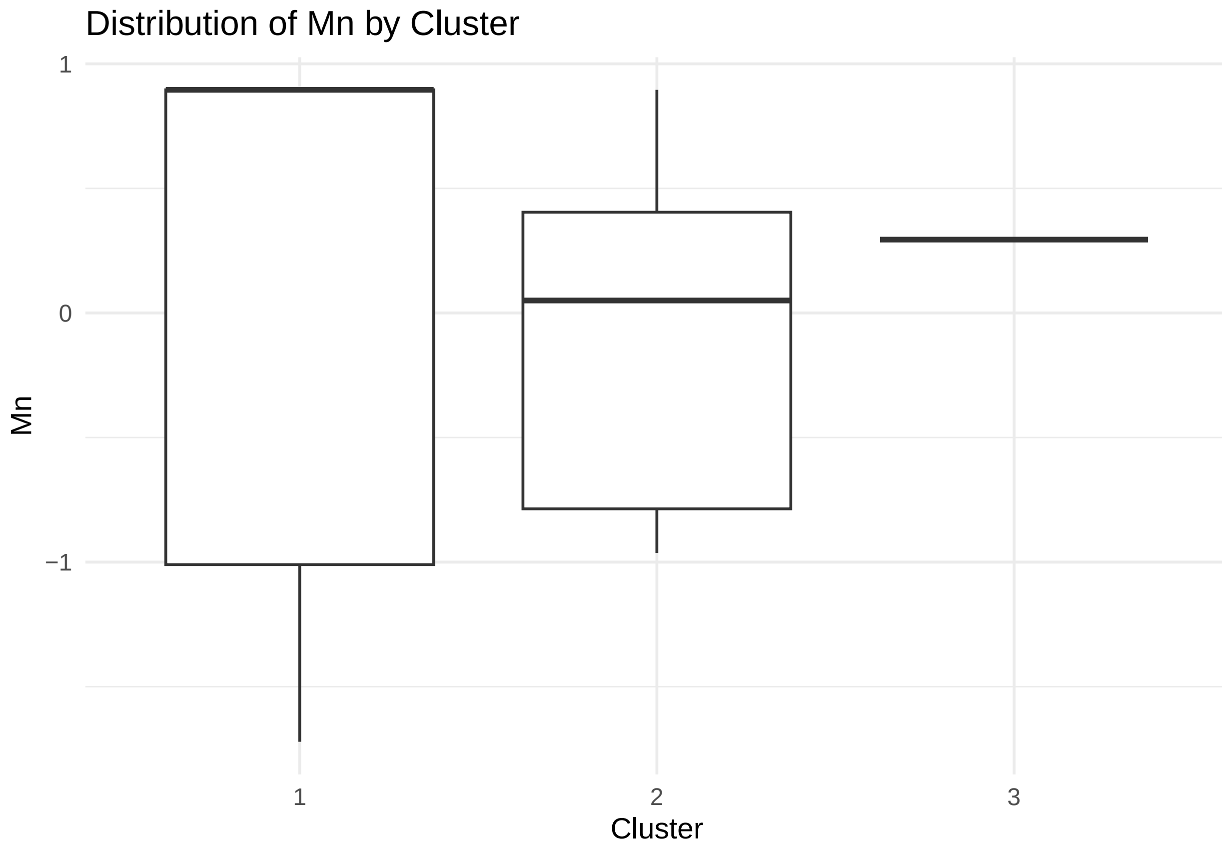


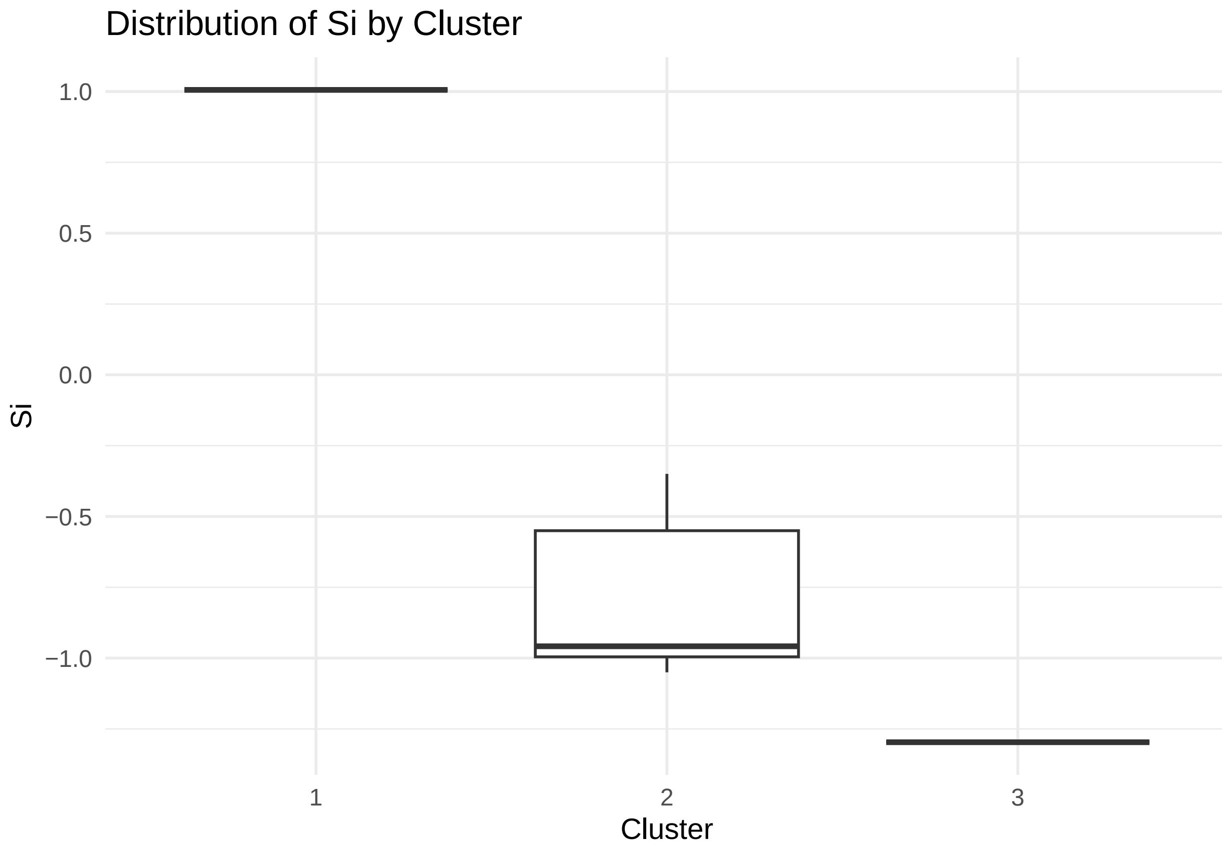
Distribution of DOC by Cluster
1
0
−1
−2
DOC
1
2
3
Cluster

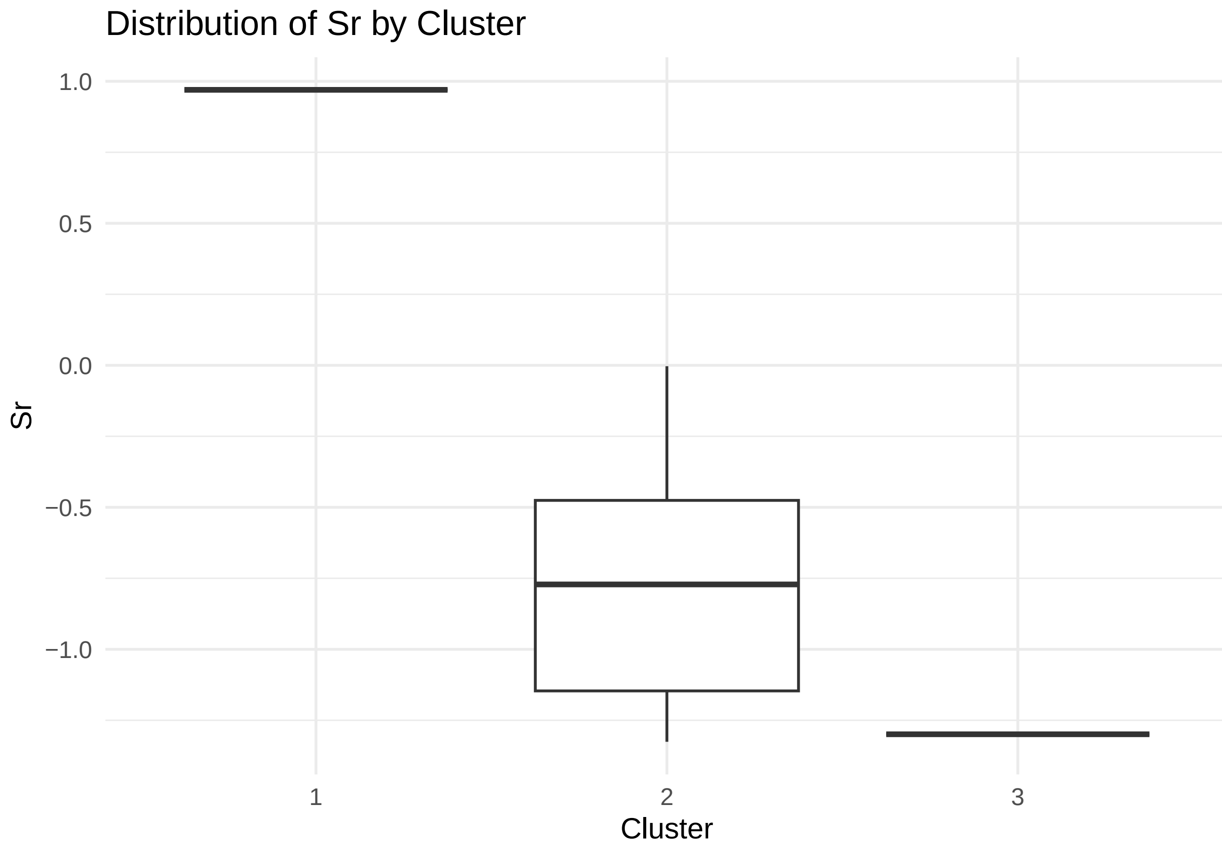

Distribution of As by Cluster
2
1
0
As
1
2
3
Cluster


Distribution of Ba by Cluster
1.0
0.5
0.0
−0.5
−1.0
Ba
1
2
3
Cluster

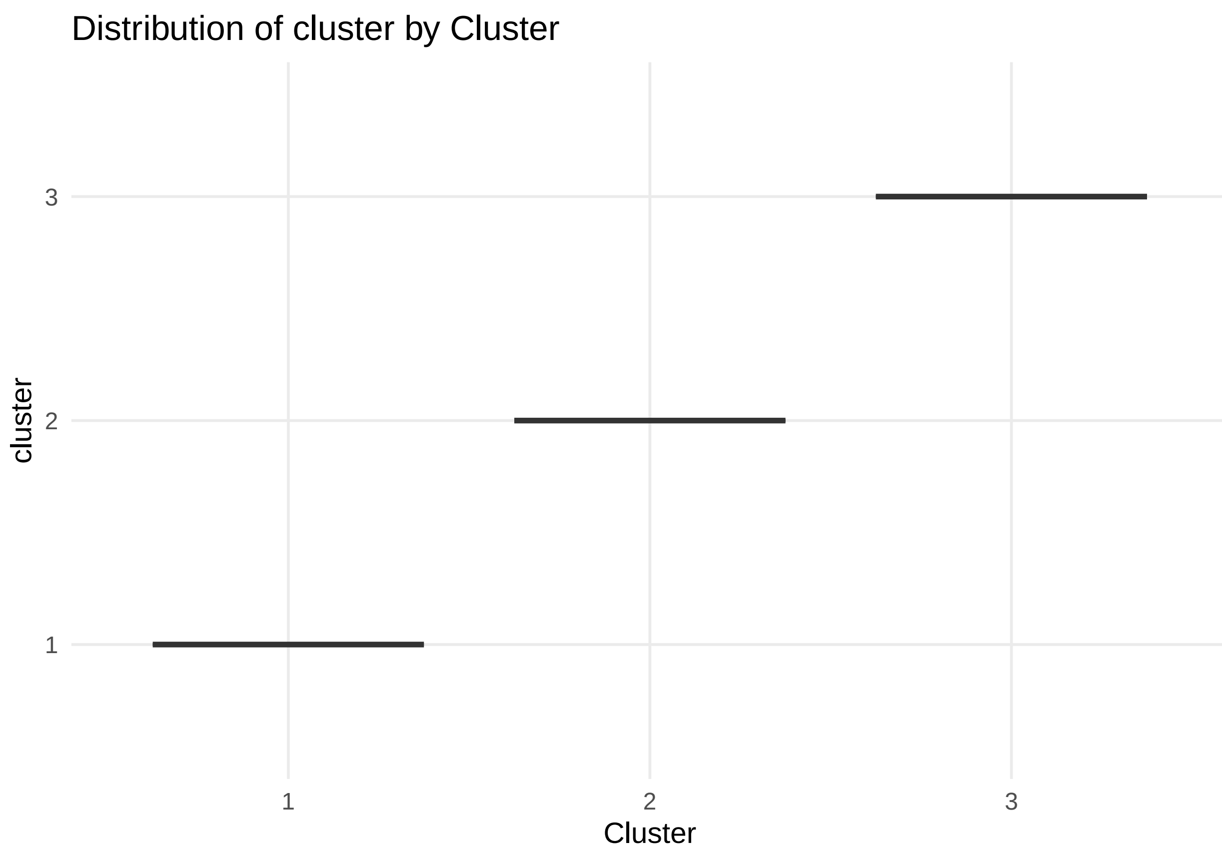

Distribution of Li by Cluster
1.0
0.5
0.0
−0.5
−1.0
−1.5
Li
1
2
3
Cluster


Distribution of Mn by Cluster
1
0
−1
Mn
1
2
3
Cluster

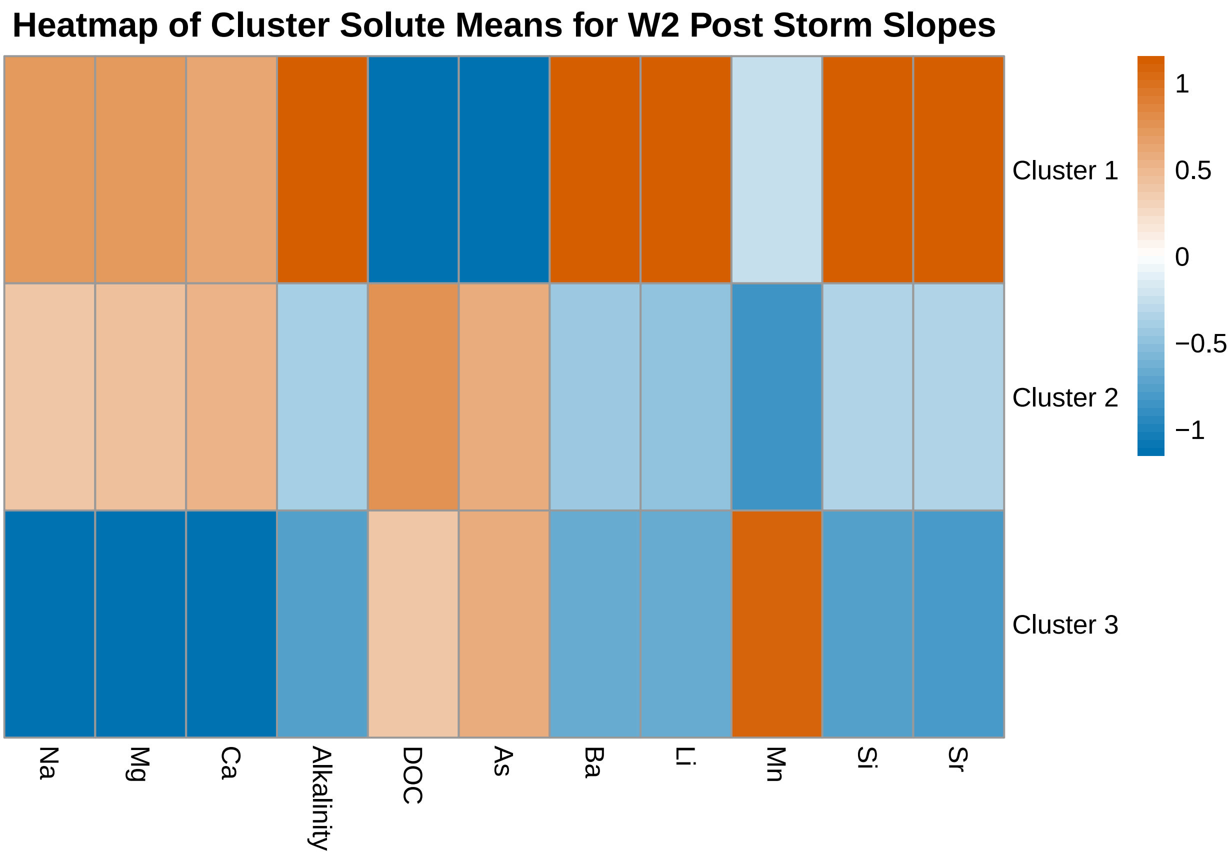

Distribution of Si by Cluster
1.0
0.5
0.0
−0.5
−1.0
Si
1
2
3
Cluster



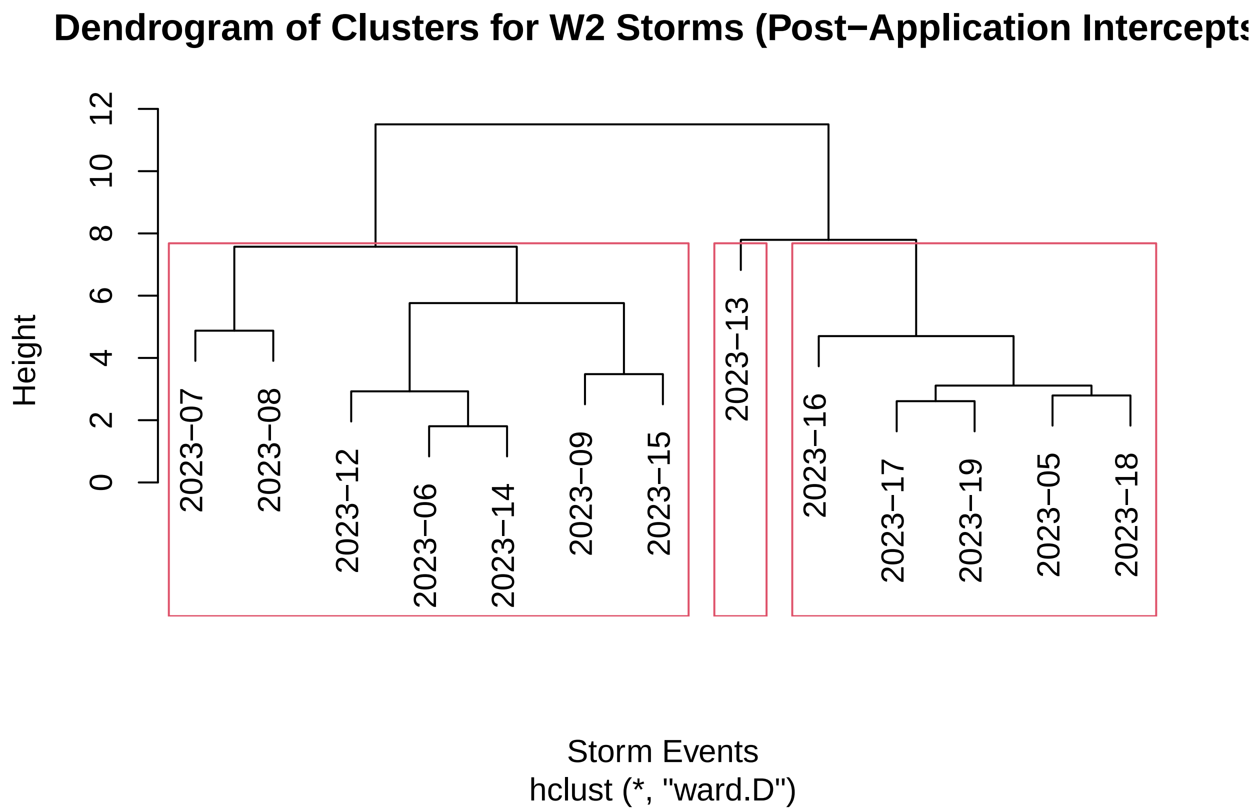
Distribution of Sr by Cluster
1.0
0.5
0.0
−0.5
−1.0
Sr
1
2
3
Cluster




```
# Cluster Profiling: Aggregate the original variables by cluster
cluster_summary_post_slope <- aggregate(. ~ cluster, data = w2_std_post_slope, FUN = mean)

# Remove the cluster column for visualization
heatmap_data_post_slope <- cluster_summary_post_slope[, -1]

# Set row names as cluster identifiers
rownames(heatmap_data_post_slope) <- paste("Cluster", cluster_summary_post_slope$cluster)

# Create a heatmap
pheatmap(
  mat = as.matrix(heatmap_data_post_slope),
  main = "Heatmap of Cluster Solute Means for W2 Post Storm Slopes",
  cluster_rows = FALSE,  # Don't re-cluster rows (clusters are predefined)
  cluster_cols = FALSE,   # Cluster variables
  scale = "column",       # Scale variables to make them comparable
  color = colorRampPalette(c("#0072B2", "white", "#D55E00"))(50) # Gradient color scheme
)
```




```r
# Hierarchical Clustering
dist_matrix_post_intercept <- dist(w2_std_post_intercept, method = "euclidean")
cluster_result_post_intercept <- hclust(dist_matrix_post_intercept, method = "ward.D")

# Plot dendrogram for visualization
plot(cluster_result_post_intercept, labels = rownames(w2_std_post_intercept), main = "Dendrogram of Clu
     xlab = "Storm Events")
rect.hclust(cluster_result_post_intercept, k = 3)
```

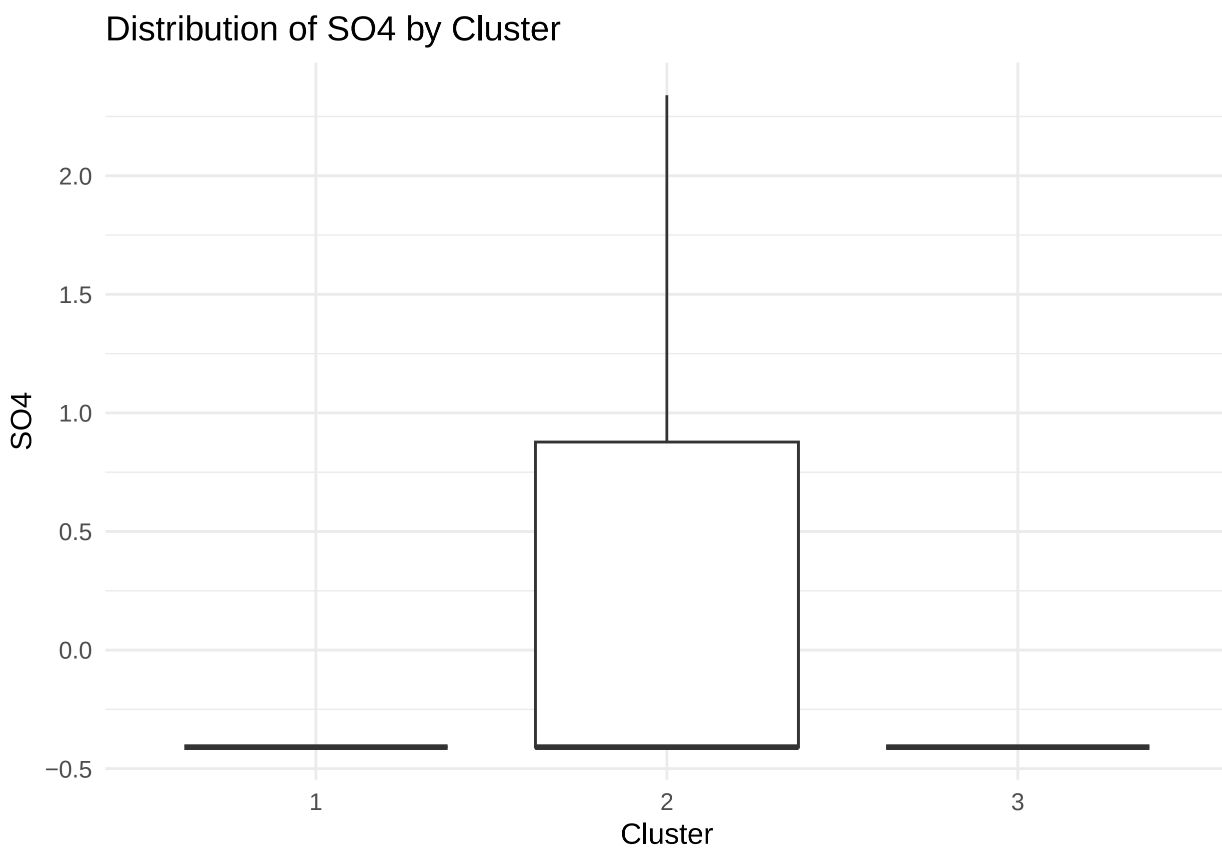


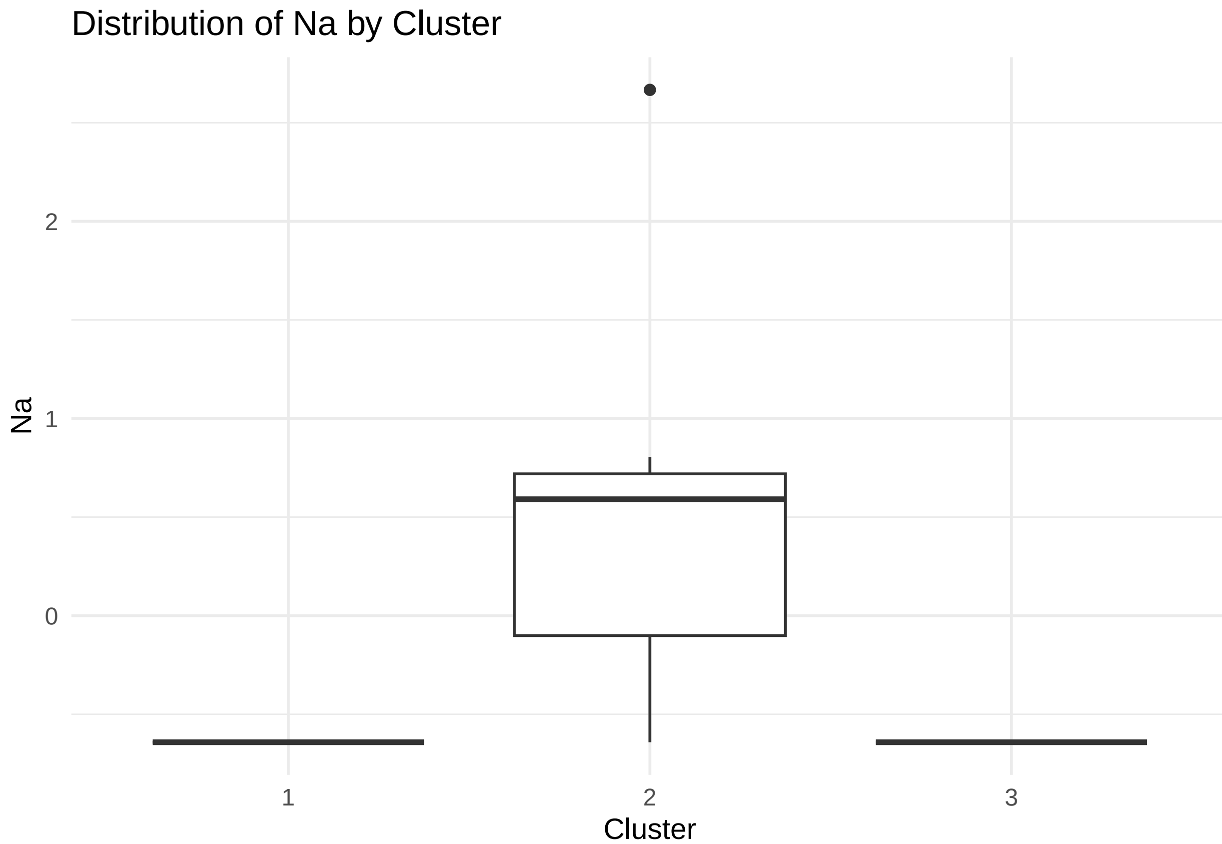

```r
# Cut the dendrogram to create 3 clusters
cuts_post_intercept <- cutree(cluster_result_post_intercept, k = 3)
w2_std_post_intercept$cluster <- as.factor(cuts_post_intercept)

# Cluster Profiling
# Aggregate the original variables by cluster to identify differences
cluster_summary_post_intercept <- aggregate(w2_std_post_intercept, by = list(cluster = cuts_post_interc
print(cluster_summary_post_intercept)
```

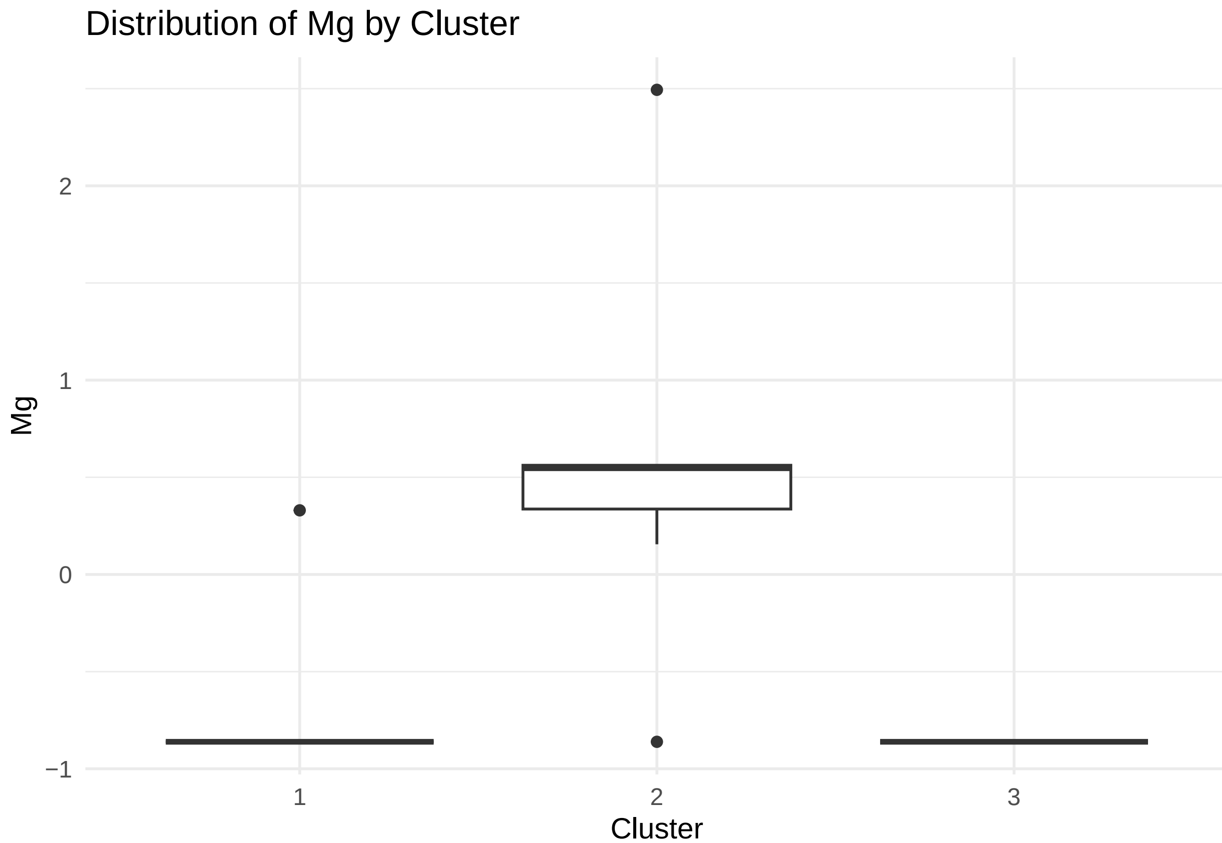
```
##   cluster        SO4         Na         Mg          Ca Alkalinity        DOC
## 1       1 -0.4094091 -0.6419600 -0.6229443 -0.07862202 -0.9199903 -0.3785330
## 2       2  0.3509221  0.5502514  0.5679847  0.44778761  0.3757806  0.3518001
## 3       3 -0.4094091 -0.6419600 -0.8611717 -2.74140316  1.9694875 -0.5699354
##           As         Ba         Cr         Li           Mn         Si
## 1  0.2651359 -0.8976544  0.2676093  0.2773501  0.283947289 -1.0073745
## 2 -0.4874962  0.7488700 -0.3383770 -0.2377287 -0.006738802  0.4283736
## 3  2.0867940 -0.7538182  1.0305922  0.2773501 -1.372564828  2.0382576
##           Sr cluster
## 1 -0.8937512      NA
## 2  0.7327263      NA
## 3 -0.6603285      NA
```

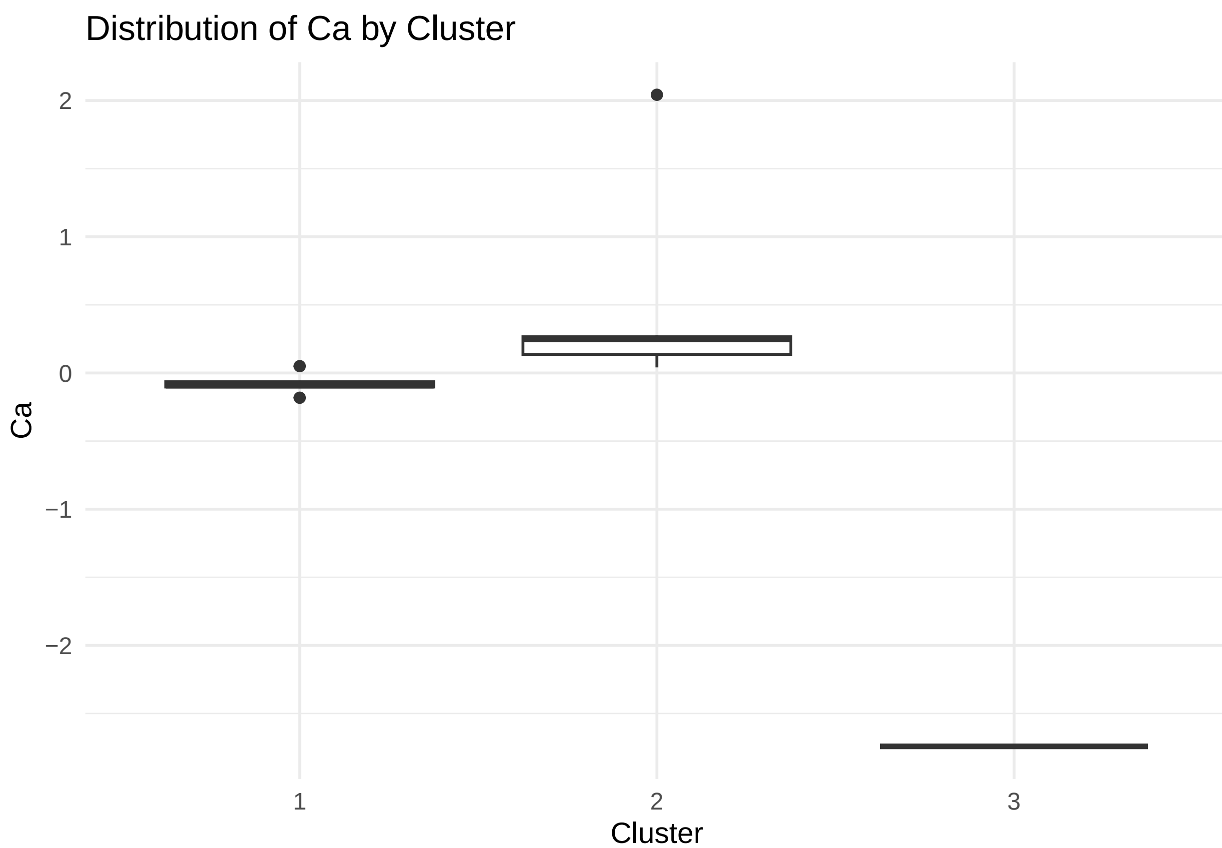
```r
# Boxplots to visualize differences in key variables across clusters
for (variable in colnames(w2_std_post_intercept)) {
  ggplot(w2_std_post_intercept, aes(x = as.factor(cuts_post_intercept), y = .data[[variable]])) +
```

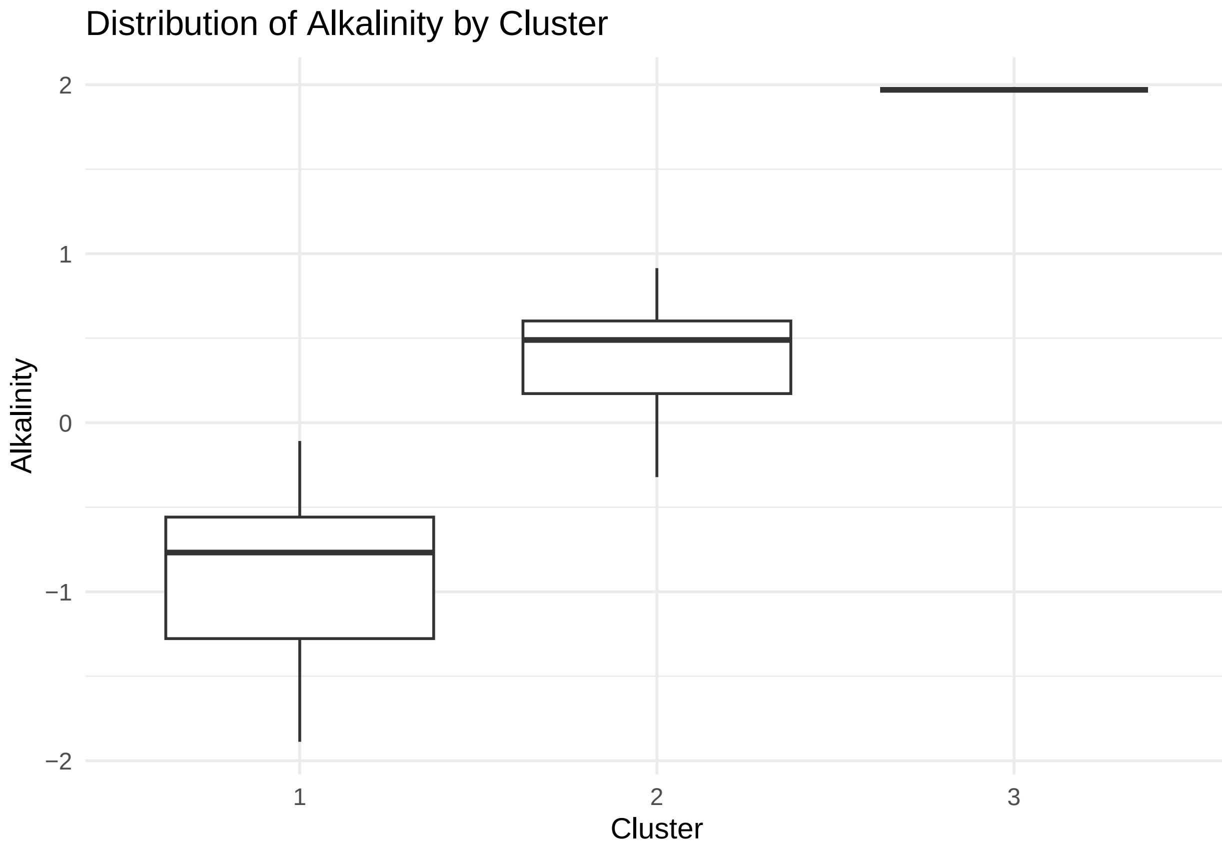
```r
    geom_boxplot() +
    labs(title = paste("Distribution of", variable, "by Cluster"),
         x = "Cluster",
         y = variable) +
    theme_minimal() -> plot
  print(plot)
}
```

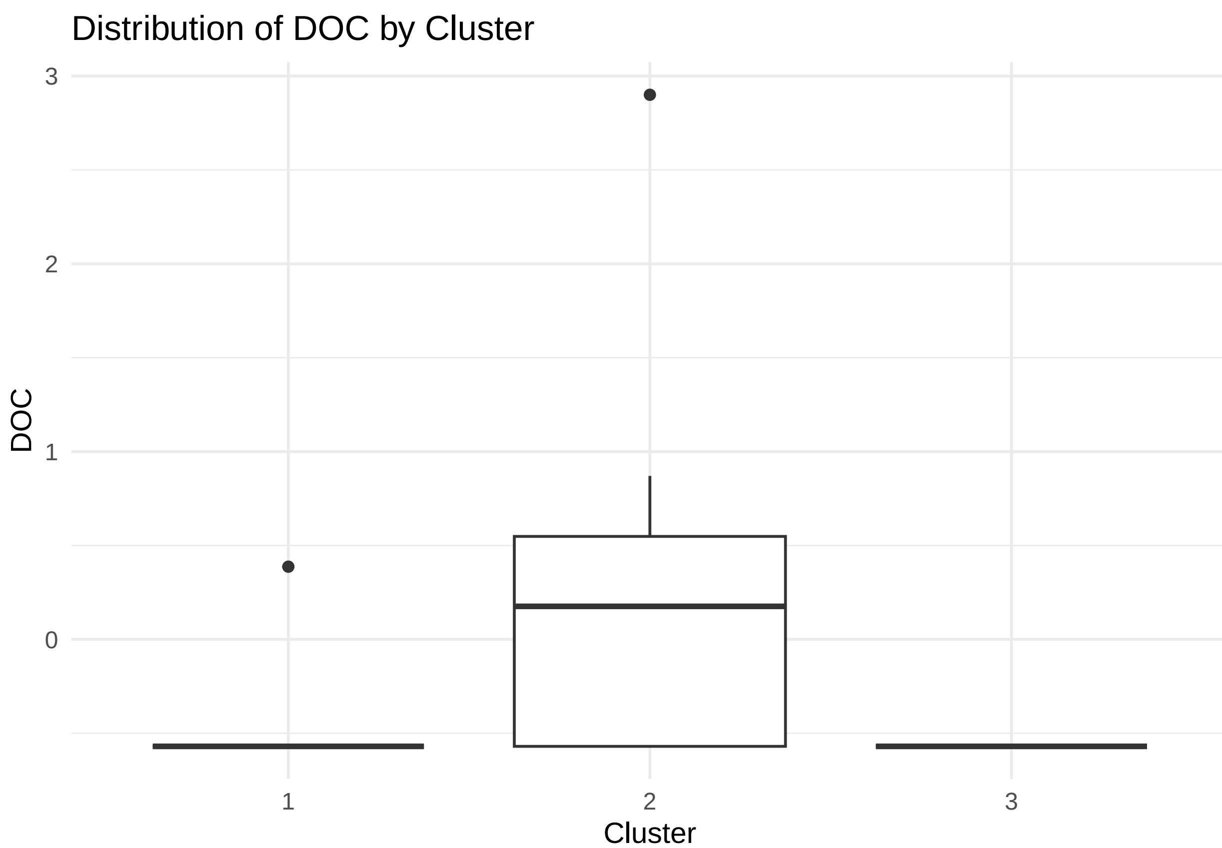


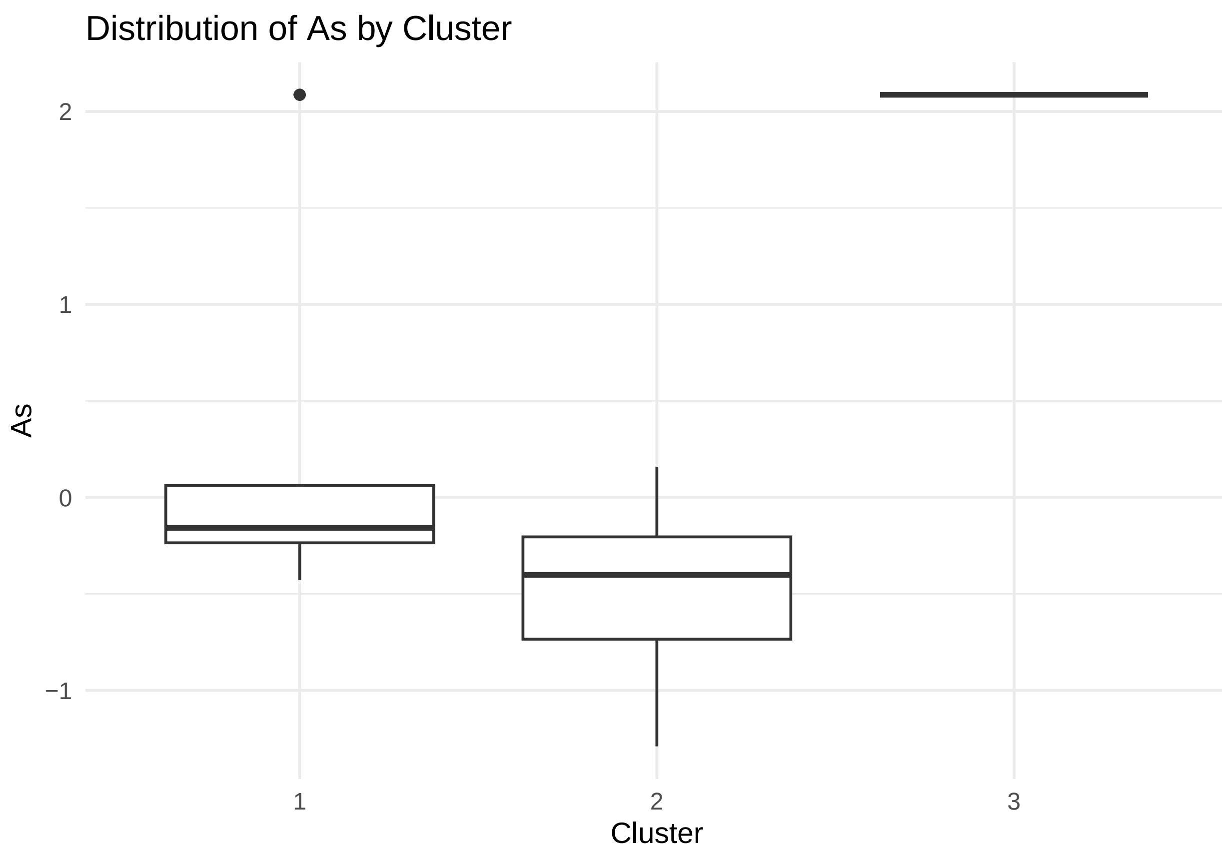

Distribution of Na by Cluster
Na
2
1
0
1
2
3
Cluster

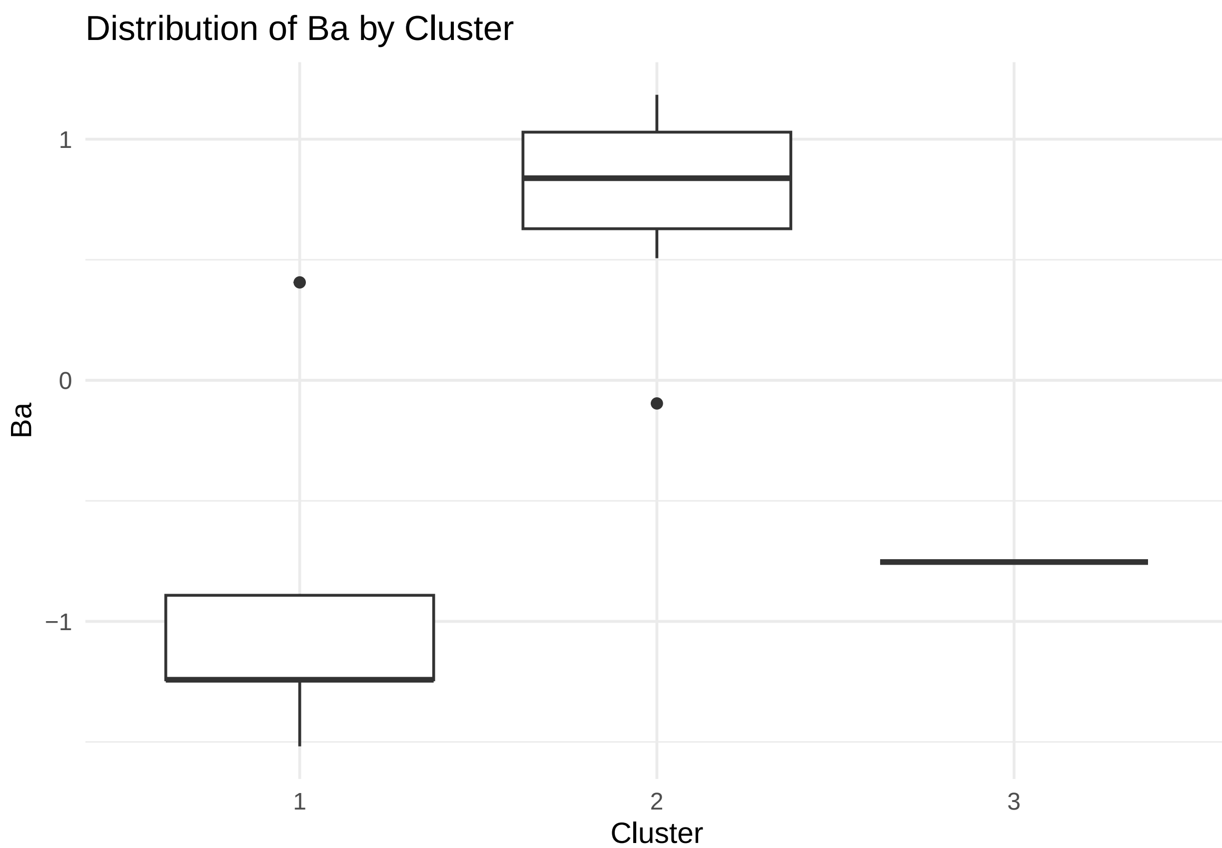

Distribution of Mg by Cluster
Mg
2
1
0
−1
1
2
3
Cluster

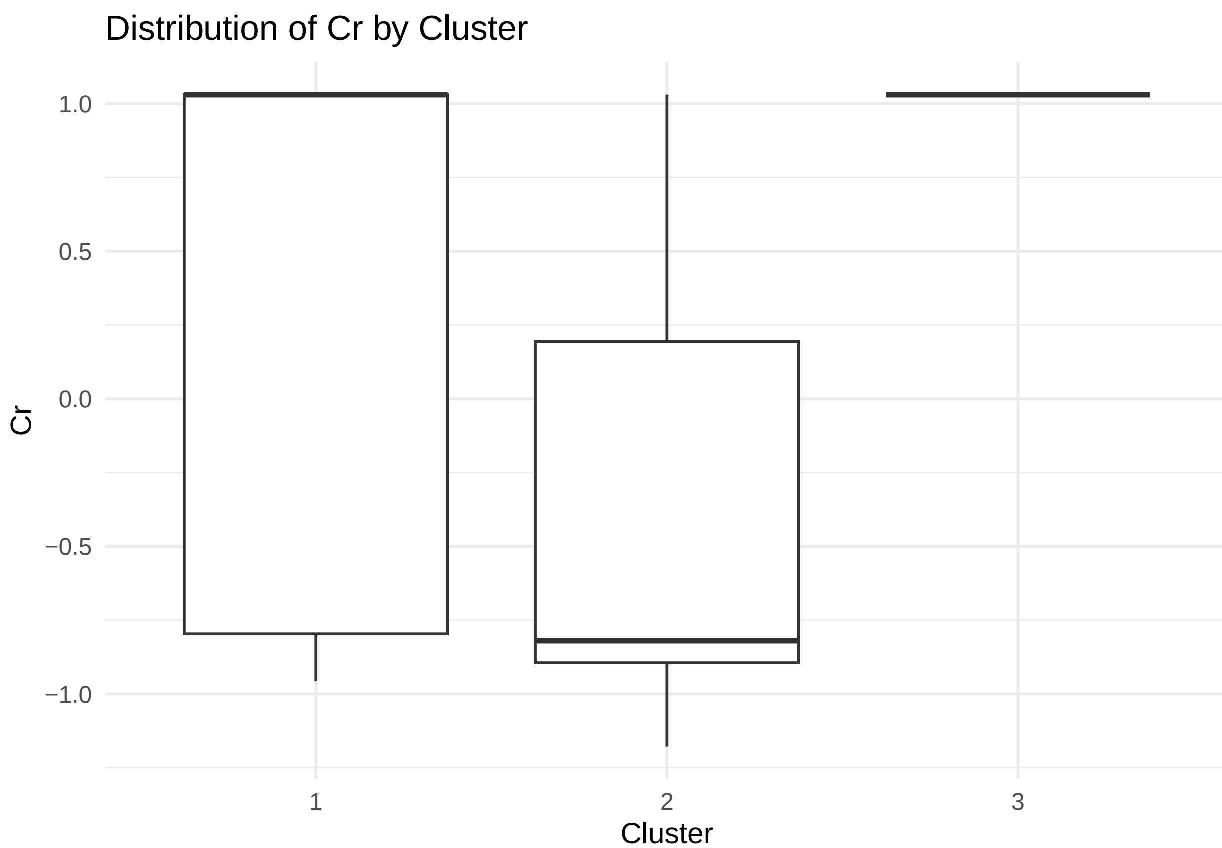

Distribution of Ca by Cluster
2
1
0
−1
−2
Ca
1
2
3
Cluster

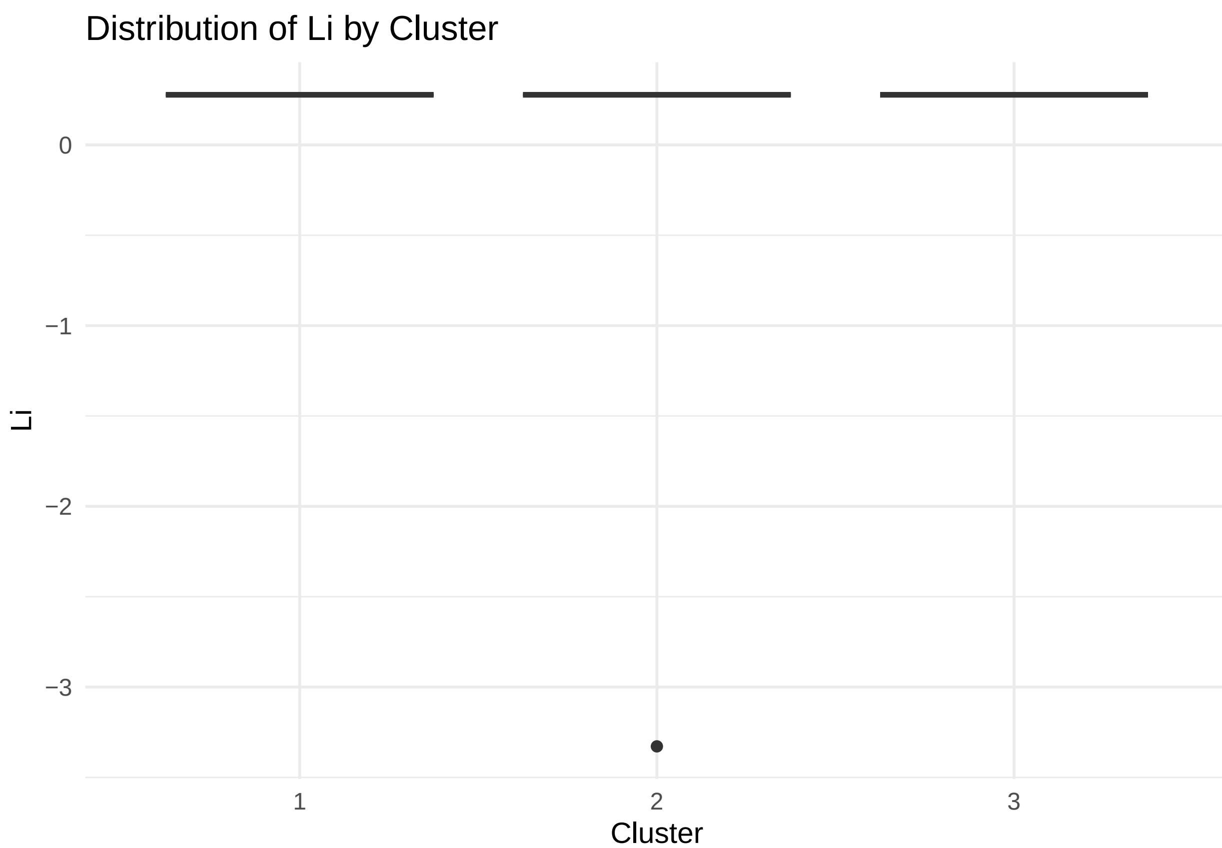

Distribution of Alkalinity by Cluster
Alkalinity
Cluster
2
1
0
−1
−2
1
2
3

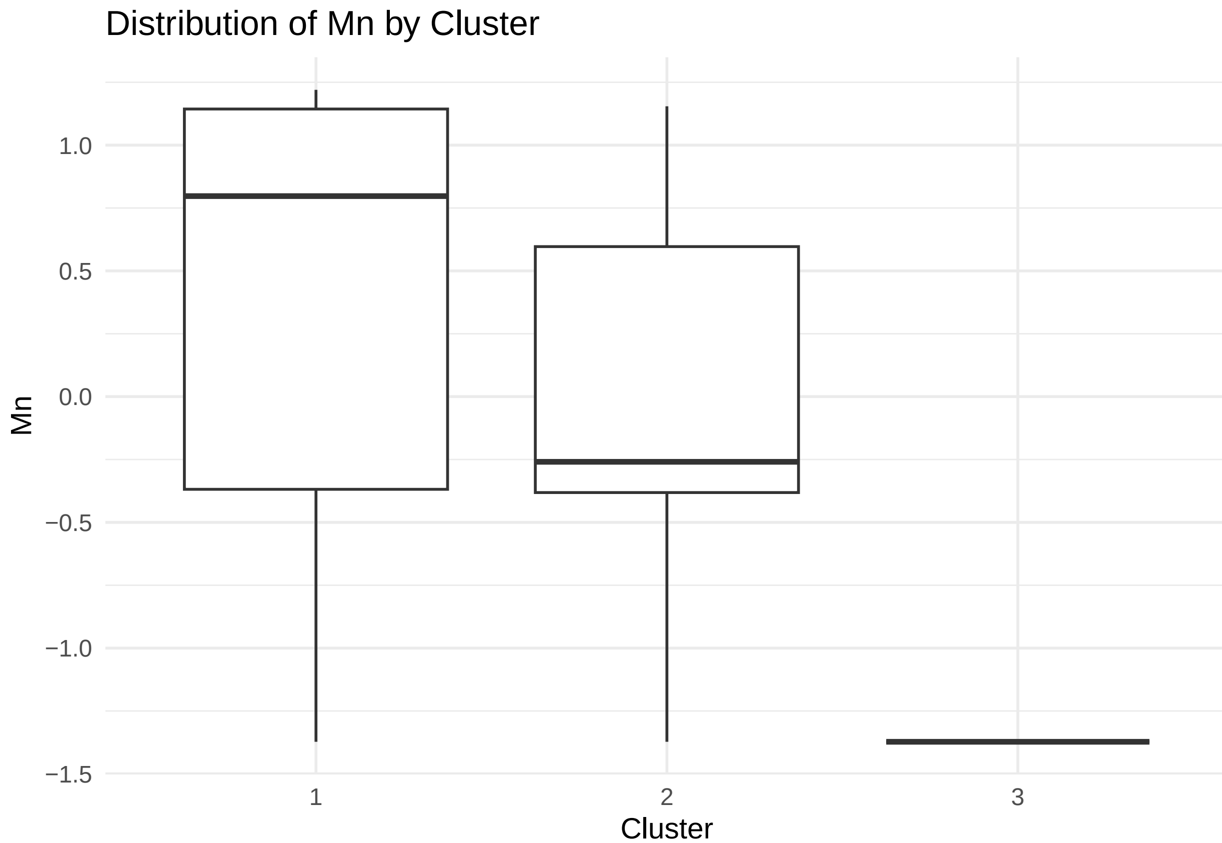

Distribution of DOC by Cluster
3
2
1
0
DOC
1
2
3
Cluster

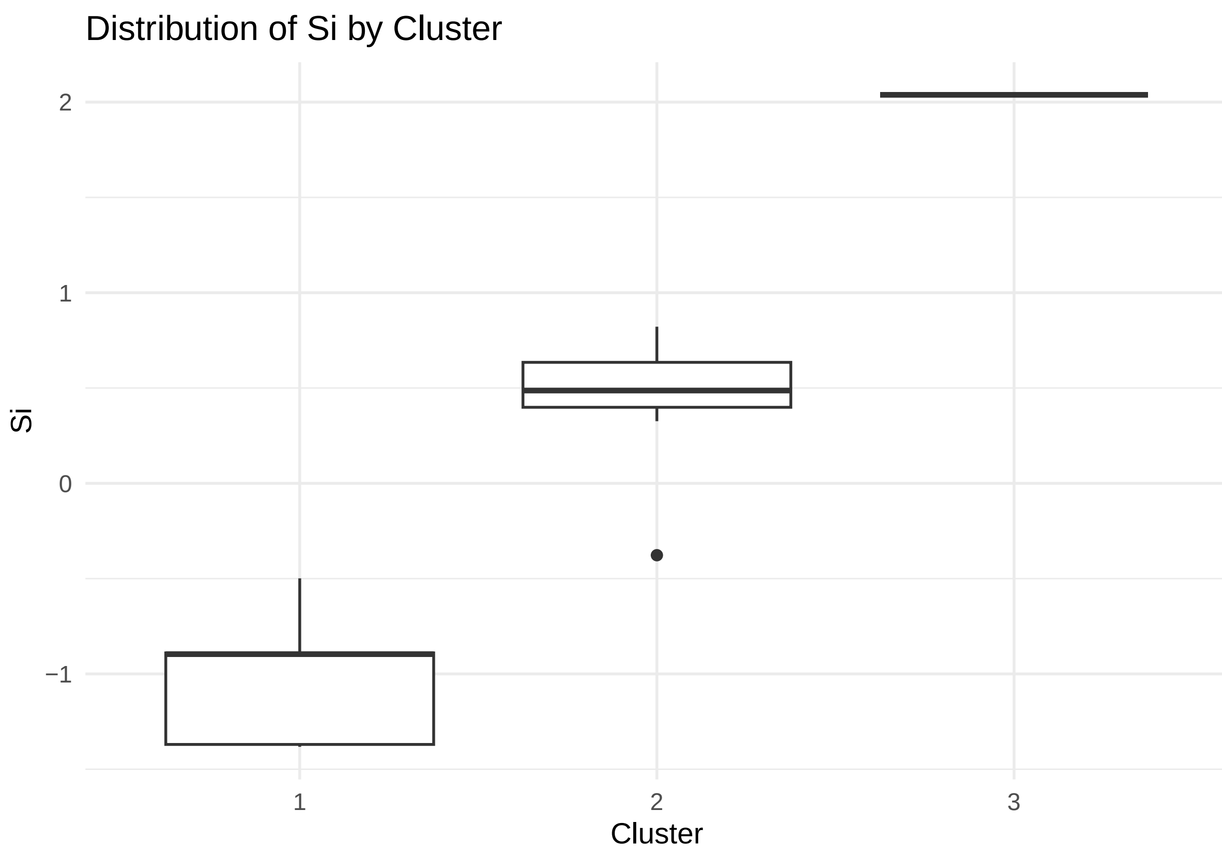

Distribution of As by Cluster
As
2
1
0
−1
1
2
3
Cluster

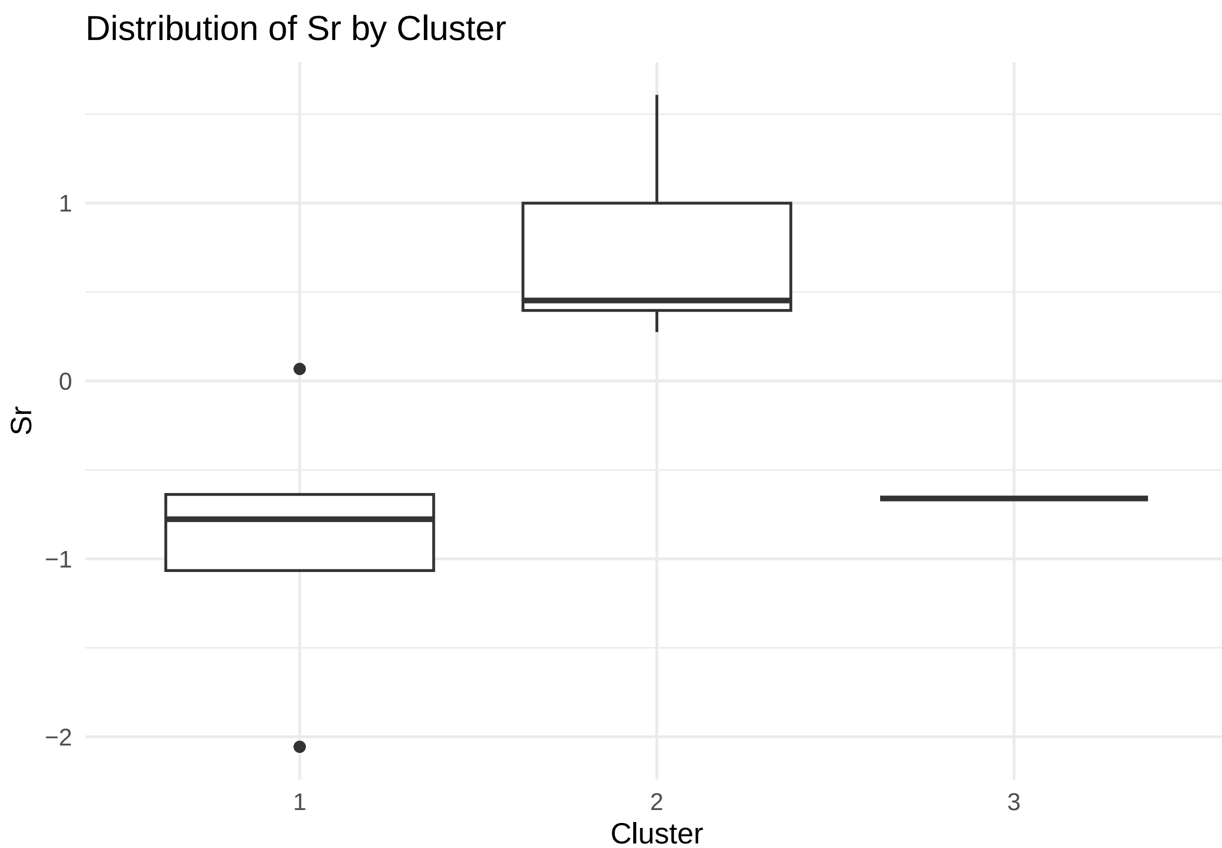

Distribution of Ba by Cluster
Ba
1
0
−1
1
2
3
Cluster

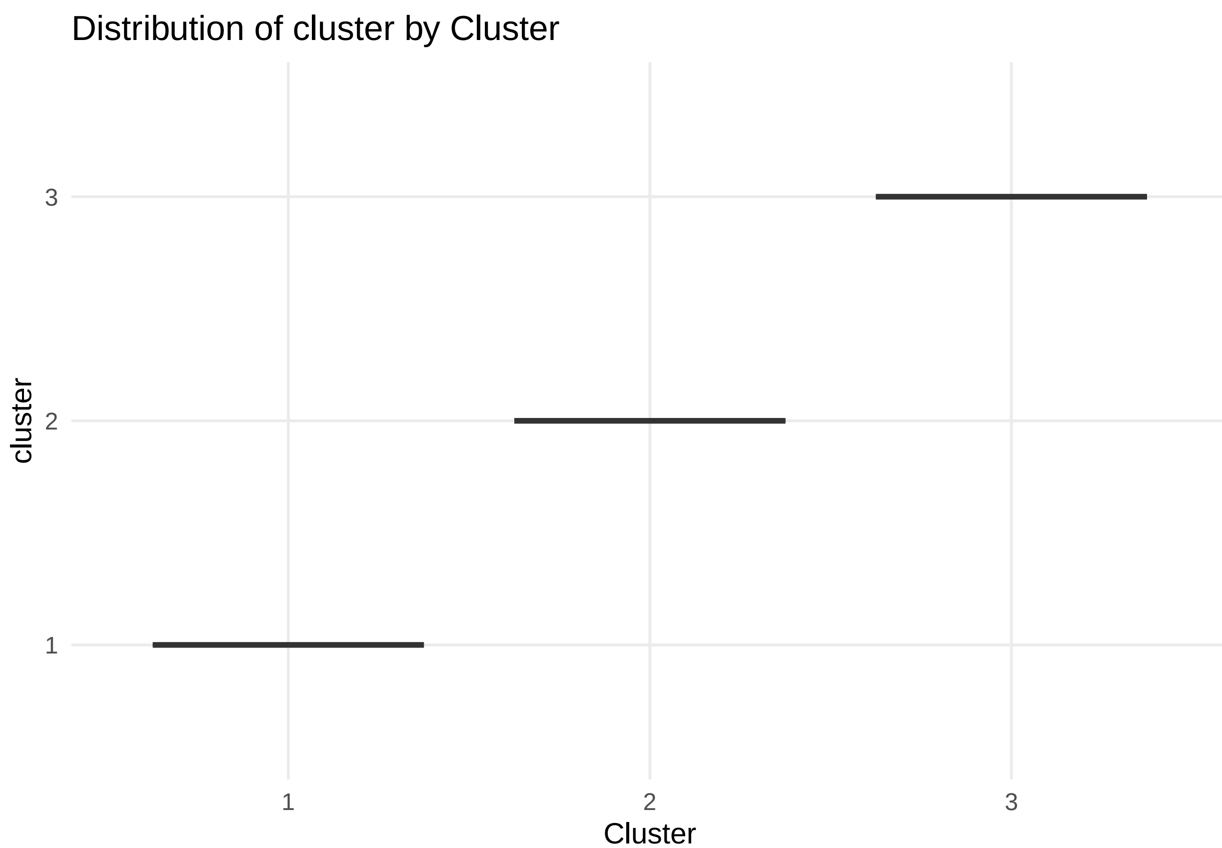

Distribution of Cr by Cluster
Cr
1.0
0.5
0.0
−0.5
−1.0
1
2
3
Cluster


Distribution of Li by Cluster
Li
0
−1
−2
−3
1
2
3
Cluster


Distribution of Mn by Cluster
Mn
1.0
0.5
0.0
−0.5
−1.0
−1.5
1
2
3
Cluster

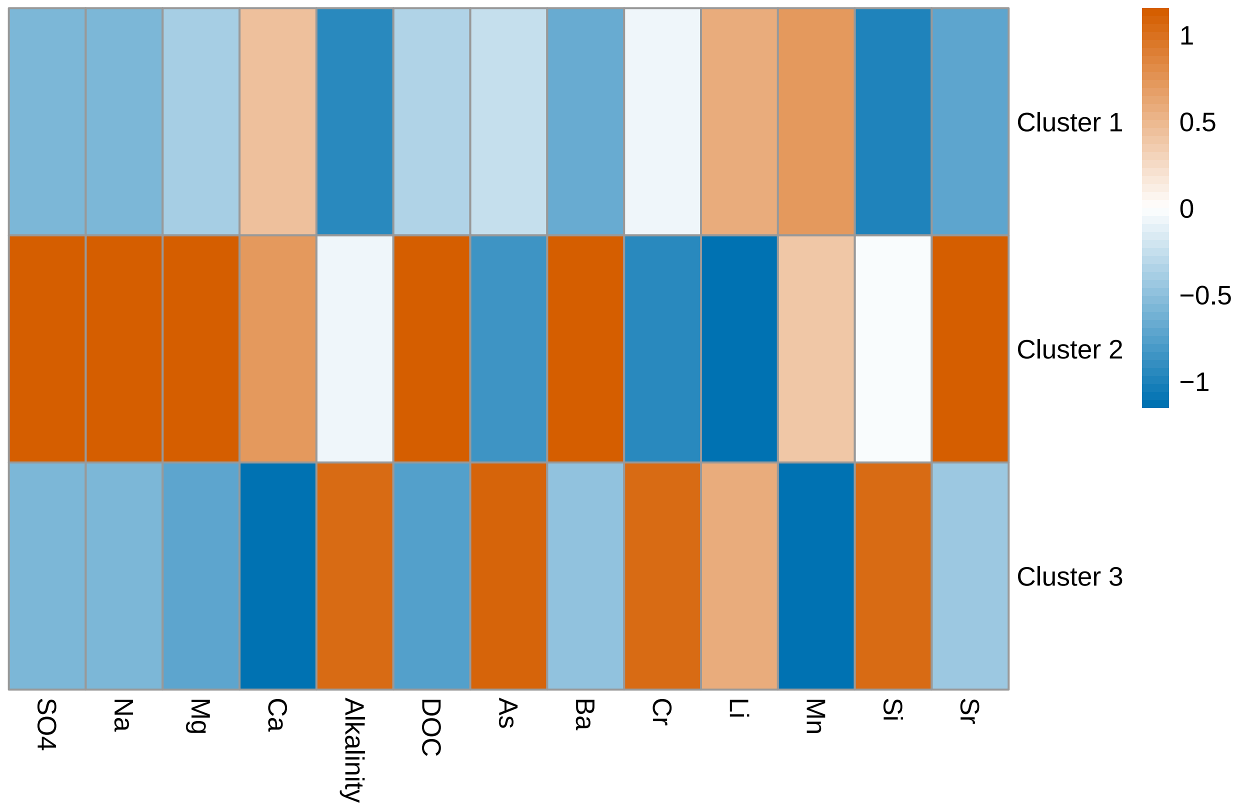

Distribution of Si by Cluster
2
1
0
−1
Si
1
2
3
Cluster


Distribution of Sr by Cluster
1
0
−1
−2
Sr
1
2
3
Cluster

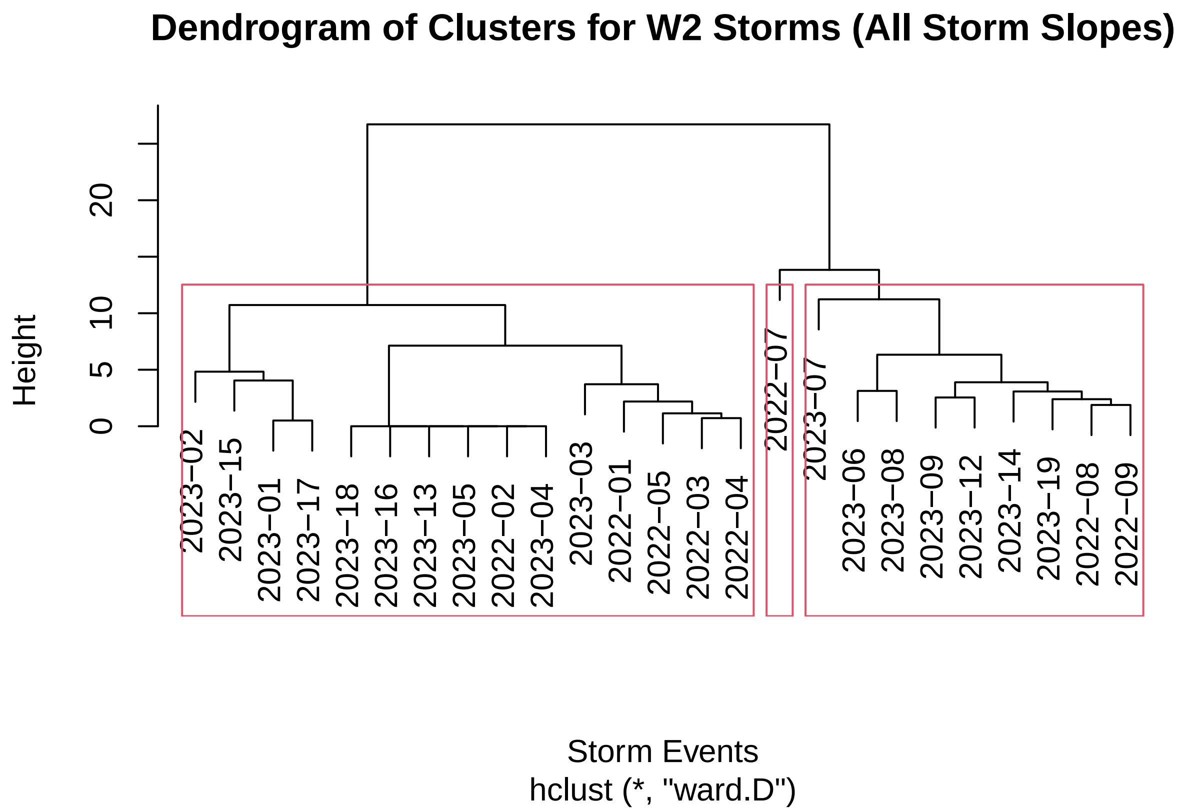



```r
# Let's make some heat maps!
# Cluster Profiling: Aggregate the original variables by cluster
cluster_summary_post_intercept <- aggregate(. ~ cluster, data = w2_std_post_intercept, FUN = mean)

# Remove the cluster column for visualization
heatmap_data_post_intercept <- cluster_summary_post_intercept[, -1]

# Set row names as cluster identifiers
rownames(heatmap_data_post_intercept) <- paste("Cluster", cluster_summary_post_intercept$cluster)

# Create a heatmap
pheatmap(
  mat = as.matrix(heatmap_data_post_intercept),
  main = "Heatmap of Cluster Solute Means for W2 Post Storm Intercepts",
  cluster_rows = FALSE,  # Don't re-cluster rows (clusters are predefined)
  cluster_cols = FALSE,   # Cluster variables
  scale = "column",      # Scale variables to make them comparable
  color = colorRampPalette(c("#0072B2", "white", "#D55E00"))(50) # Gradient color scheme
)
```

Heatmap of Cluster Solute Means for W2 Post Storm Intercepts




```r
# Hierarchical Clustering
dist_matrix_all_slope <- dist(w2_std_all_slope, method = "euclidean")
cluster_result_all_slope <- hclust(dist_matrix_all_slope, method = "ward.D")

# Plot dendrogram for visualization
plot(cluster_result_all_slope, labels = rownames(w2_std_all_slope), main = "Dendrogram of Clusters for W
     xlab = "Storm Events")
rect.hclust(cluster_result_all_slope, k = 3)
```

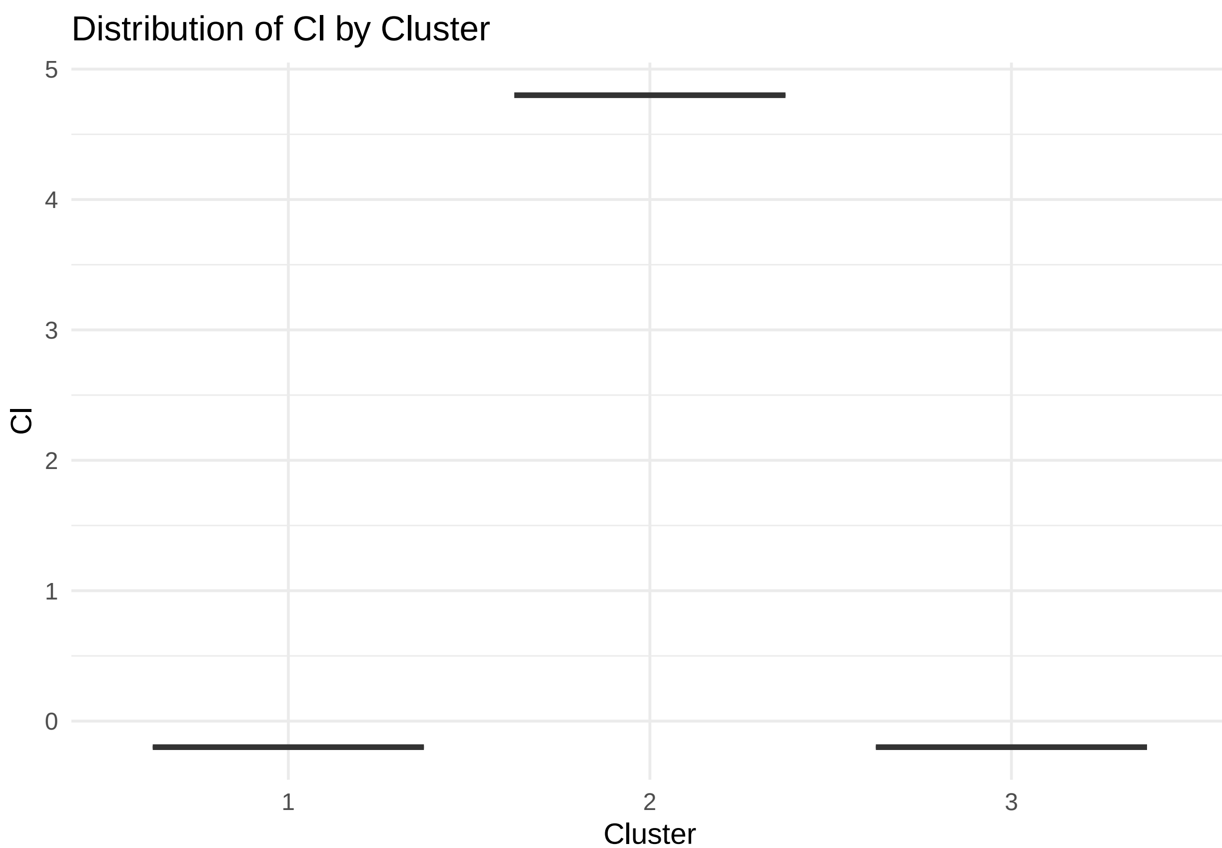


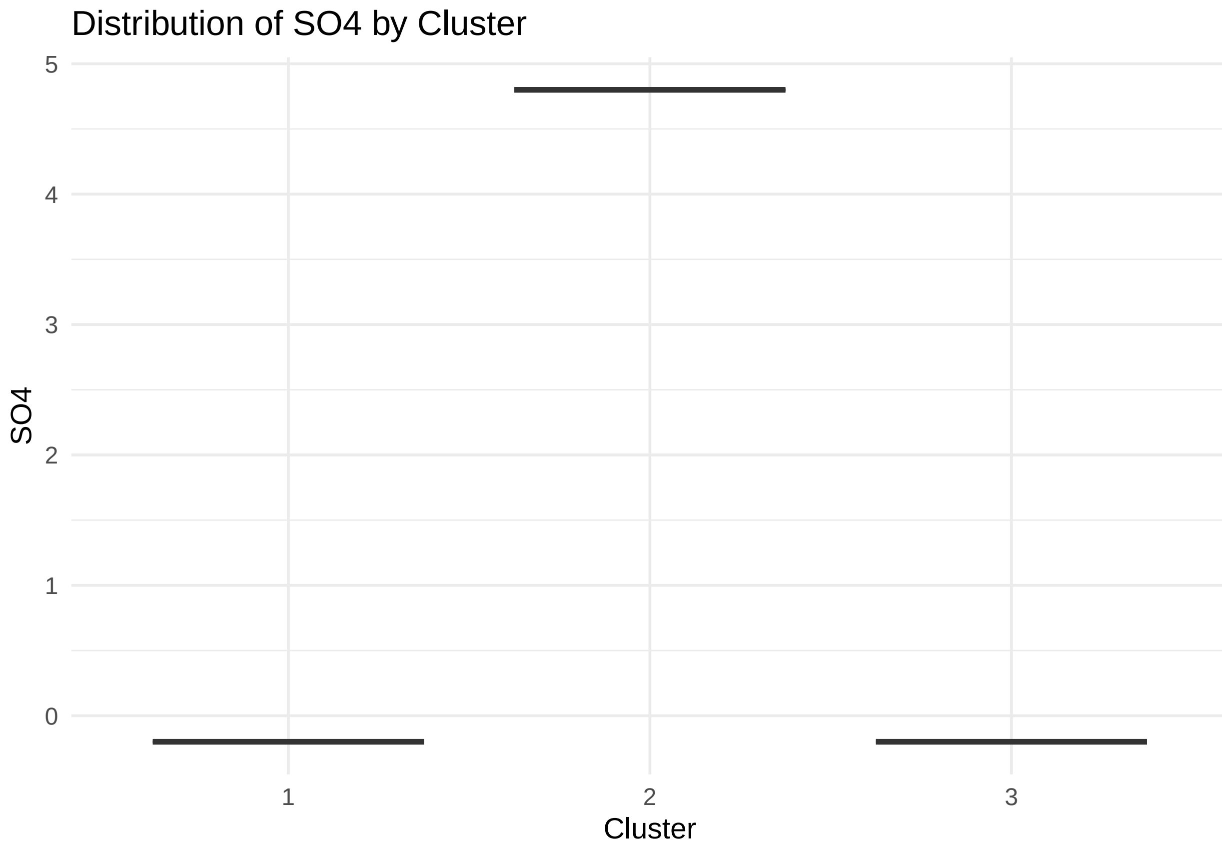

```r
# Cut the dendrogram to create 3 clusters
cuts_all_slope <- cutree(cluster_result_all_slope, k = 3)
w2_std_all_slope$cluster <- as.factor(cuts_all_slope)

# Cluster Profiling
# Aggregate the original variables by cluster to identify differences
cluster_summary_all_slope <- aggregate(w2_std_all_slope, by = list(cluster = cuts_all_slope), FUN = mear
print(cluster_summary_all_slope)
```

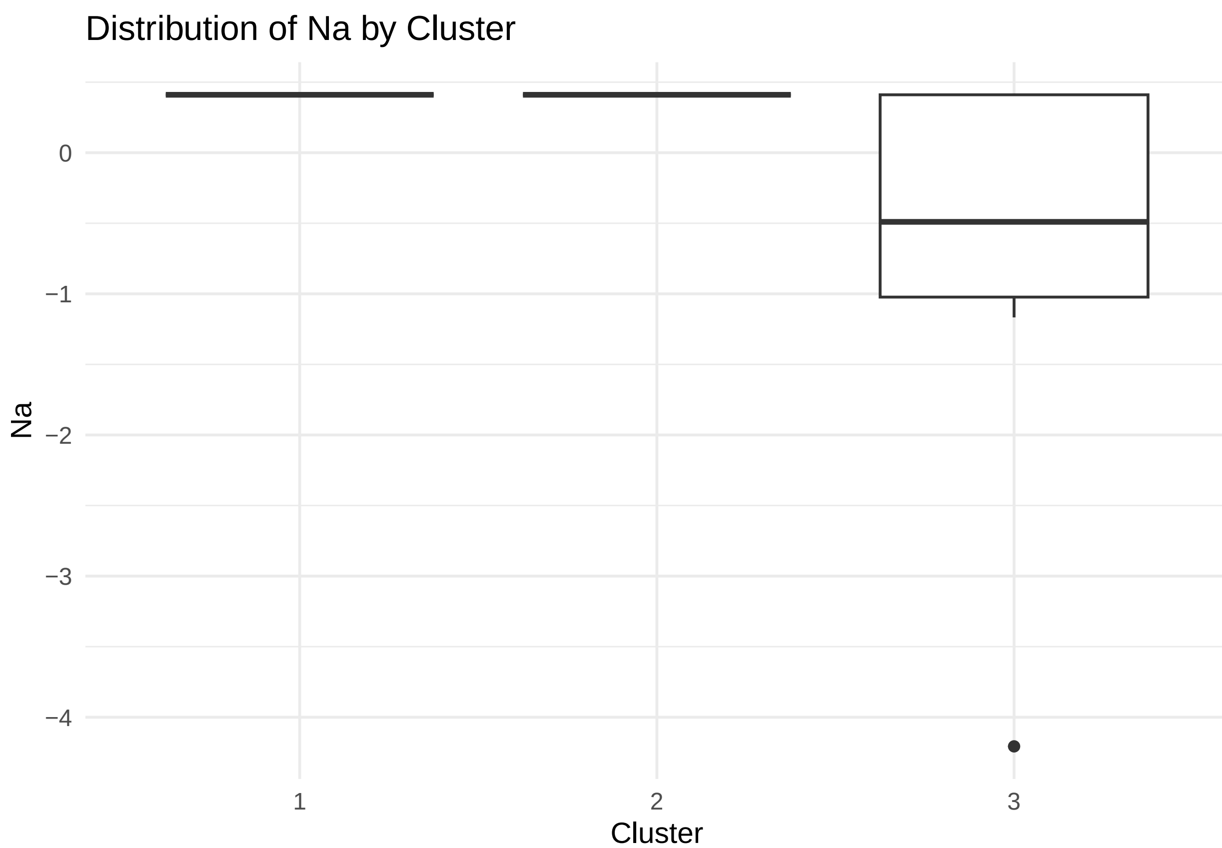
```
##   cluster   Cl  SO4         Na         Mg         Ca Alkalinity        DOC
## 1       1 -0.2 -0.2  0.4104945  0.4037185  0.2699294  0.7443602 -0.4383610
## 2       2  4.8  4.8  0.4104945 -0.3064559  0.2699294 -0.7940828  0.7928437
## 3       3 -0.2 -0.2 -0.7297680 -0.6388135 -0.4798745 -1.1523690  0.6425080
##           As         Ba         Cr         Li           Mn         Si
## 1 -0.3648739  0.4950002 -0.2454787  0.6765512  0.022972513  0.6559484
## 2 -1.7713406 -0.1477291  4.6499296 -1.7982100 -0.008118198 -0.3857707
## 3  0.8049388 -0.8085859 -0.1075276 -0.9277843 -0.037385499 -1.0503840
##           Sr cluster
## 1  0.6601846      NA
## 2 -0.5040130      NA
## 3 -1.0443063      NA
```

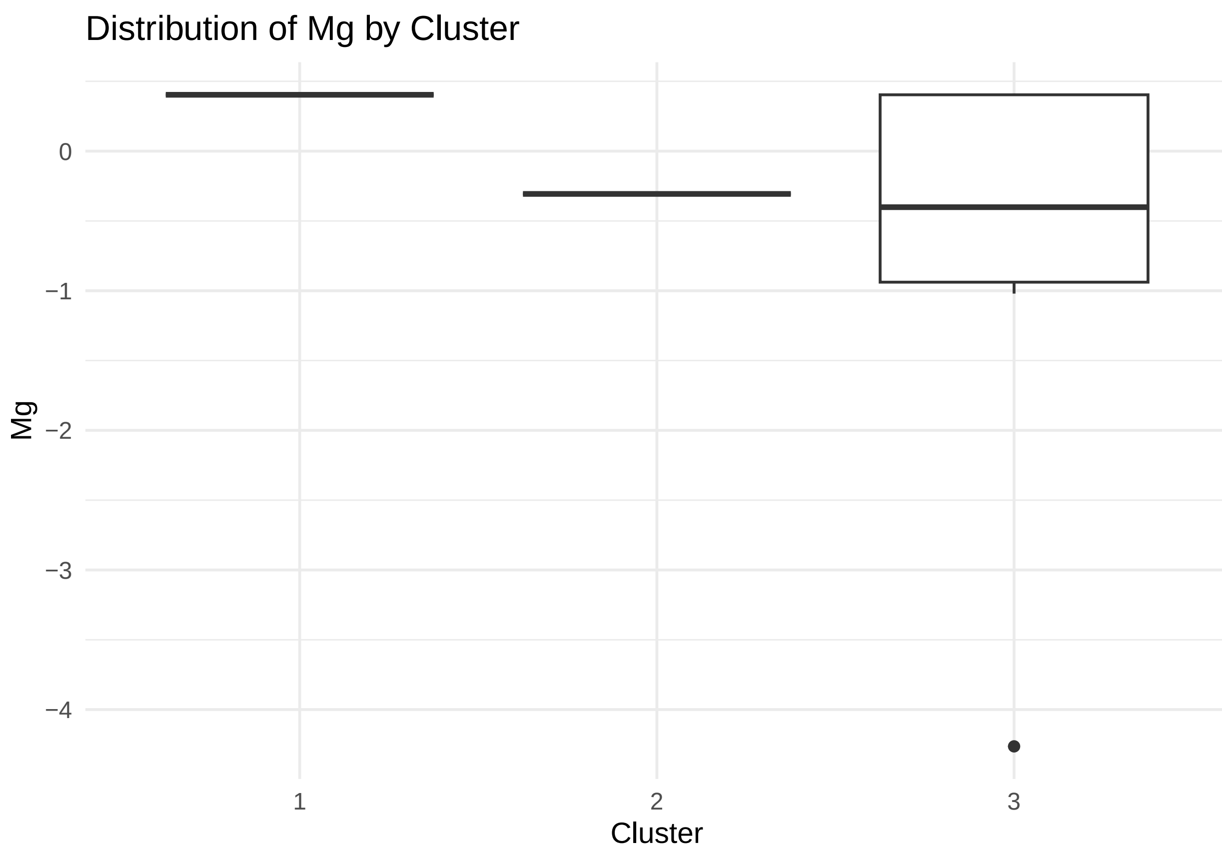
```r
# Boxplots to visualize differences in key variables across clusters
for (variable in colnames(w2_std_all_slope)) {
  ggplot(w2_std_all_slope, aes(x = as.factor(cuts_all_slope), y = .data[[variable]])) +
```

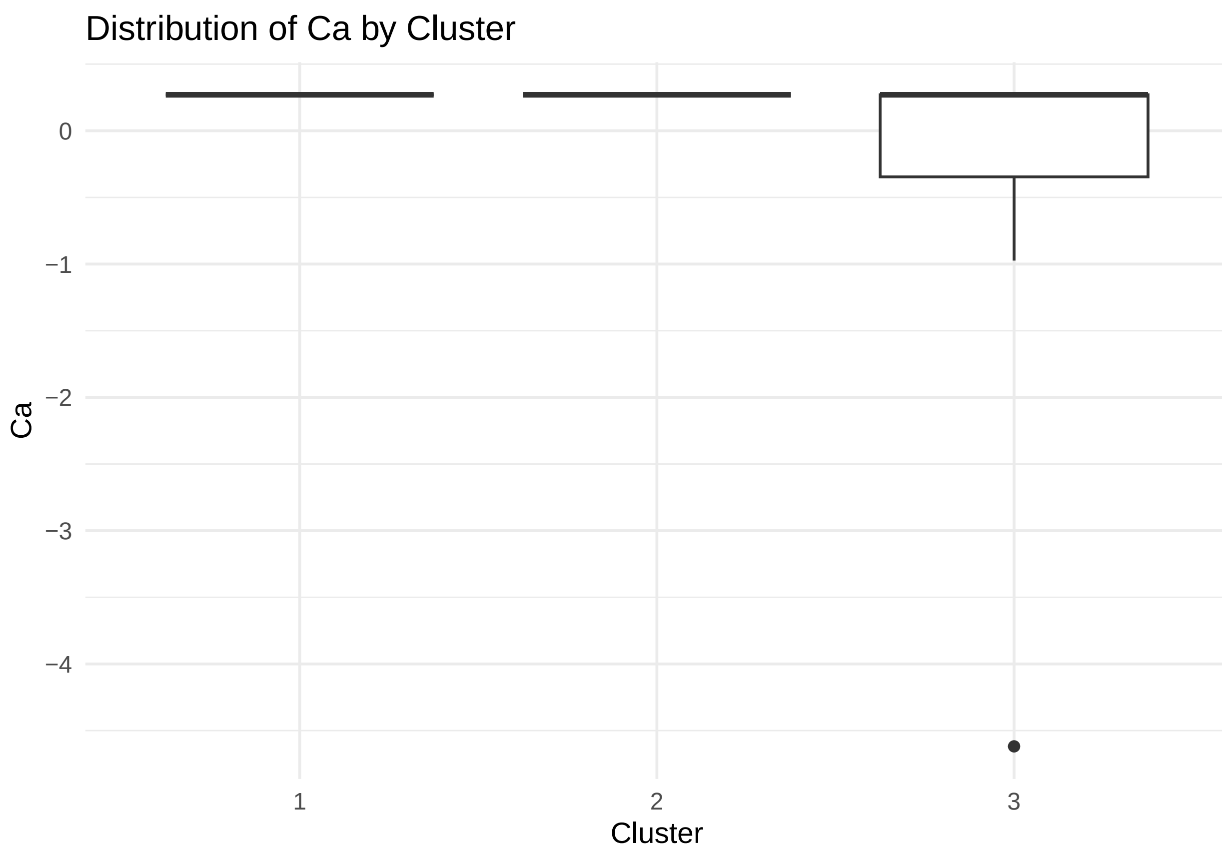
```
    geom_boxplot() +
    labs(title = paste("Distribution of", variable, "by Cluster"),
         x = "Cluster",
         y = variable) +
    theme_minimal() -> plot
  print(plot)
}
```

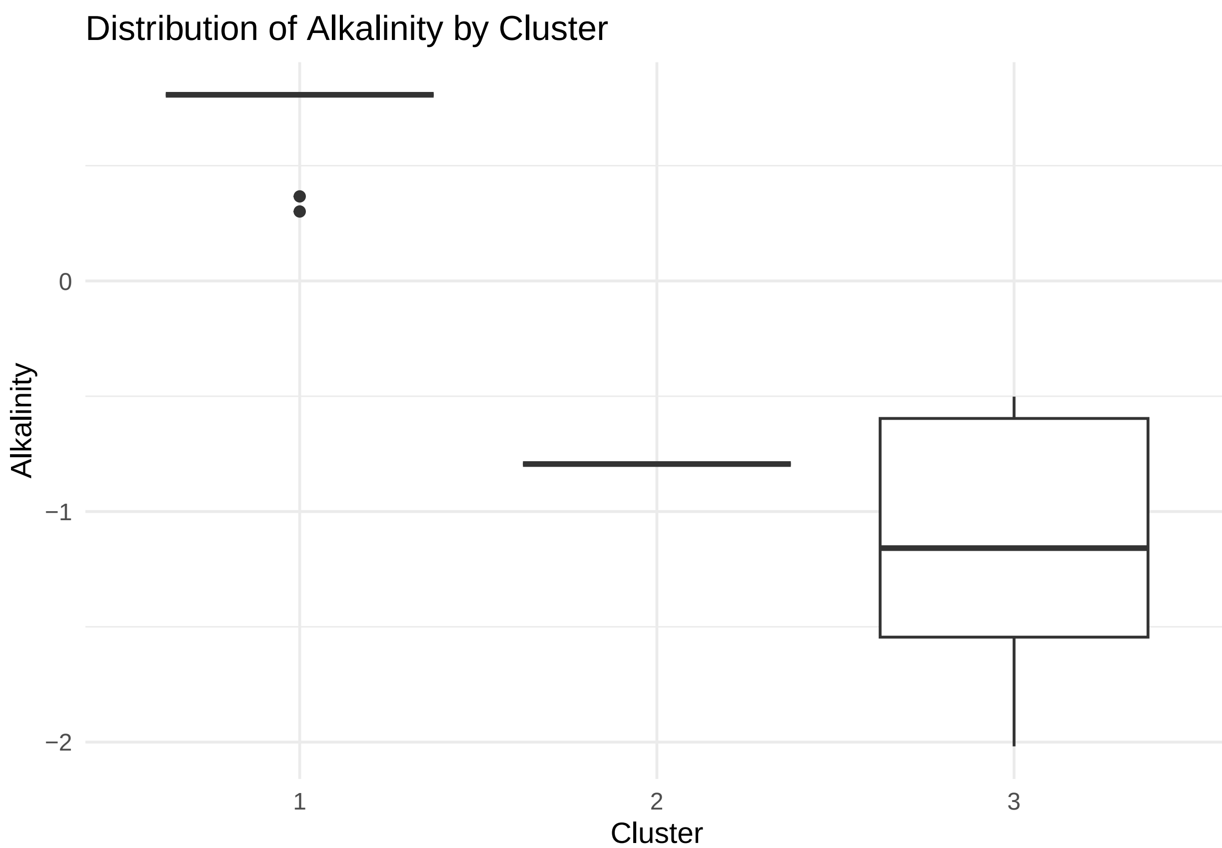


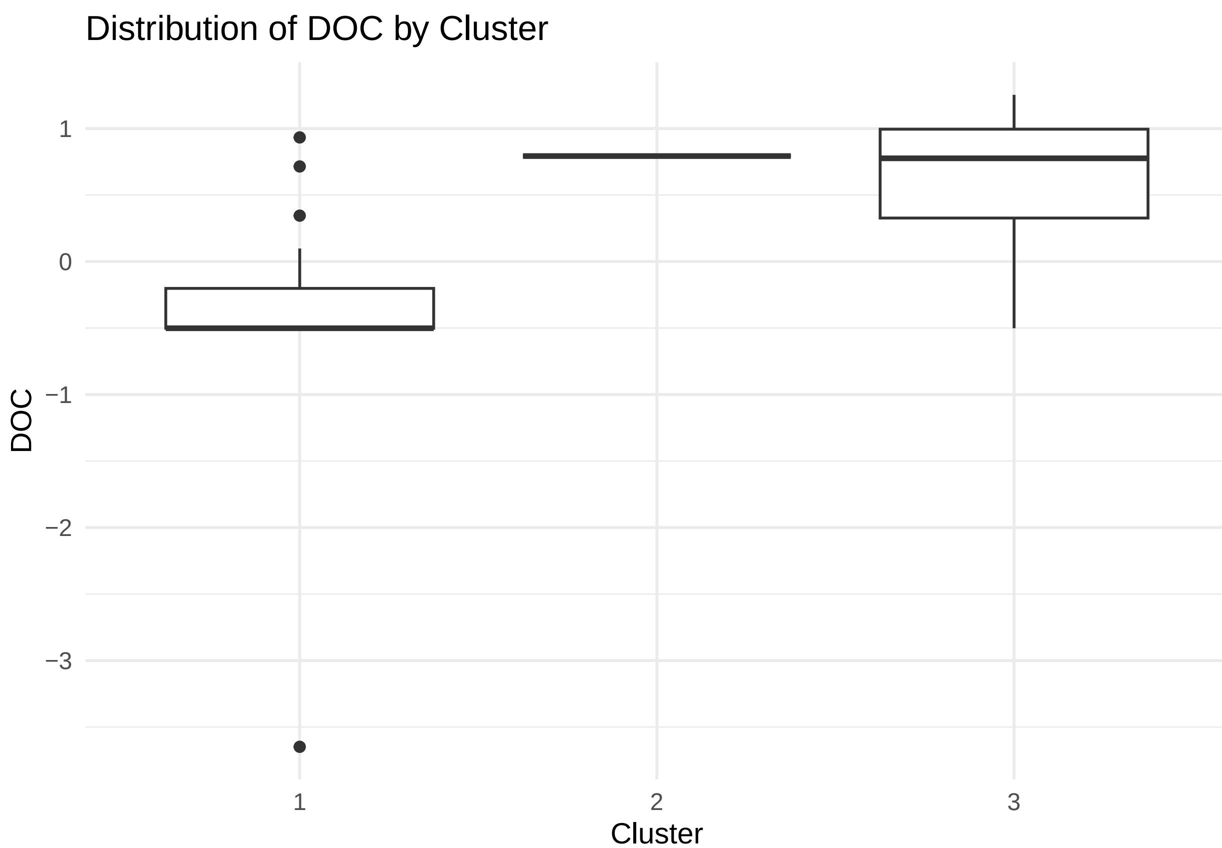

Distribution of SO4 by Cluster
SO4
5
4
3
2
1
0
1
2
3
Cluster

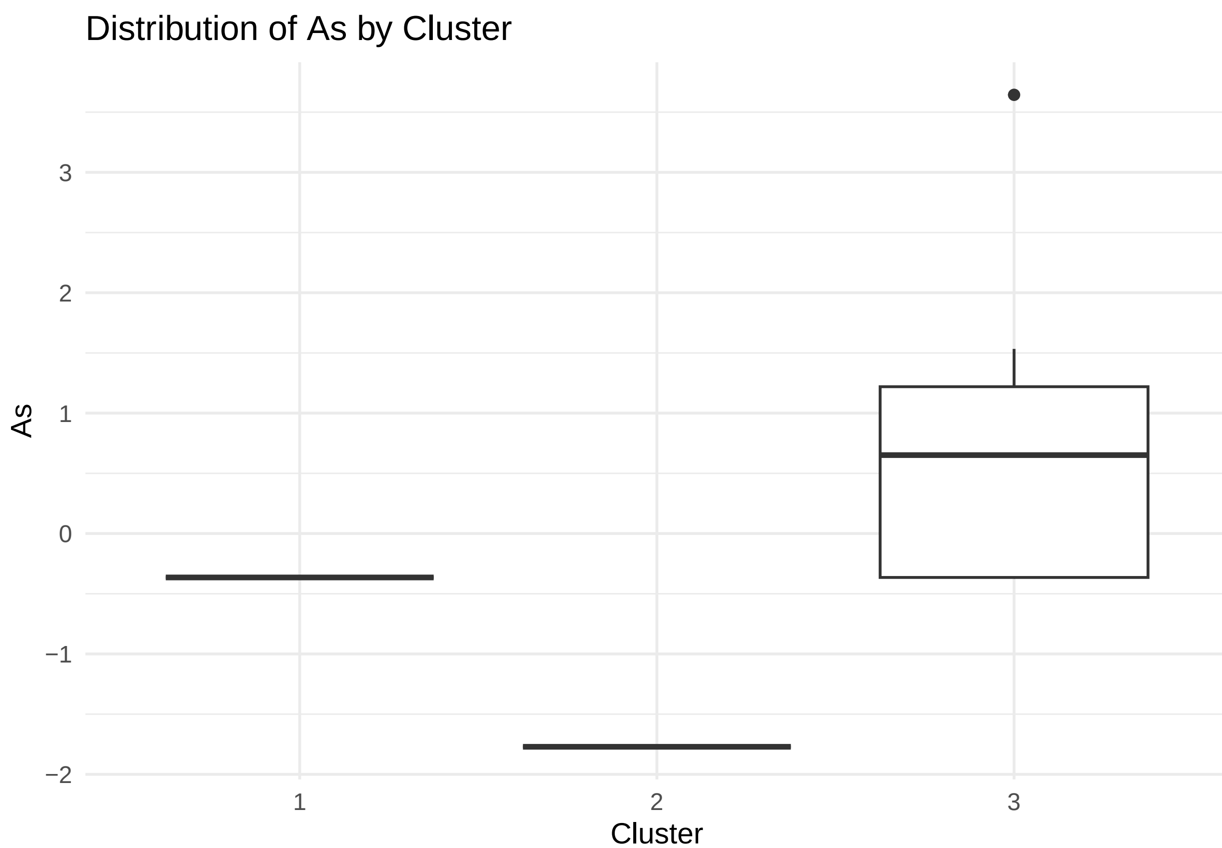

Distribution of Na by Cluster
0
−1
−2
−3
−4
Na
1
2
3
Cluster

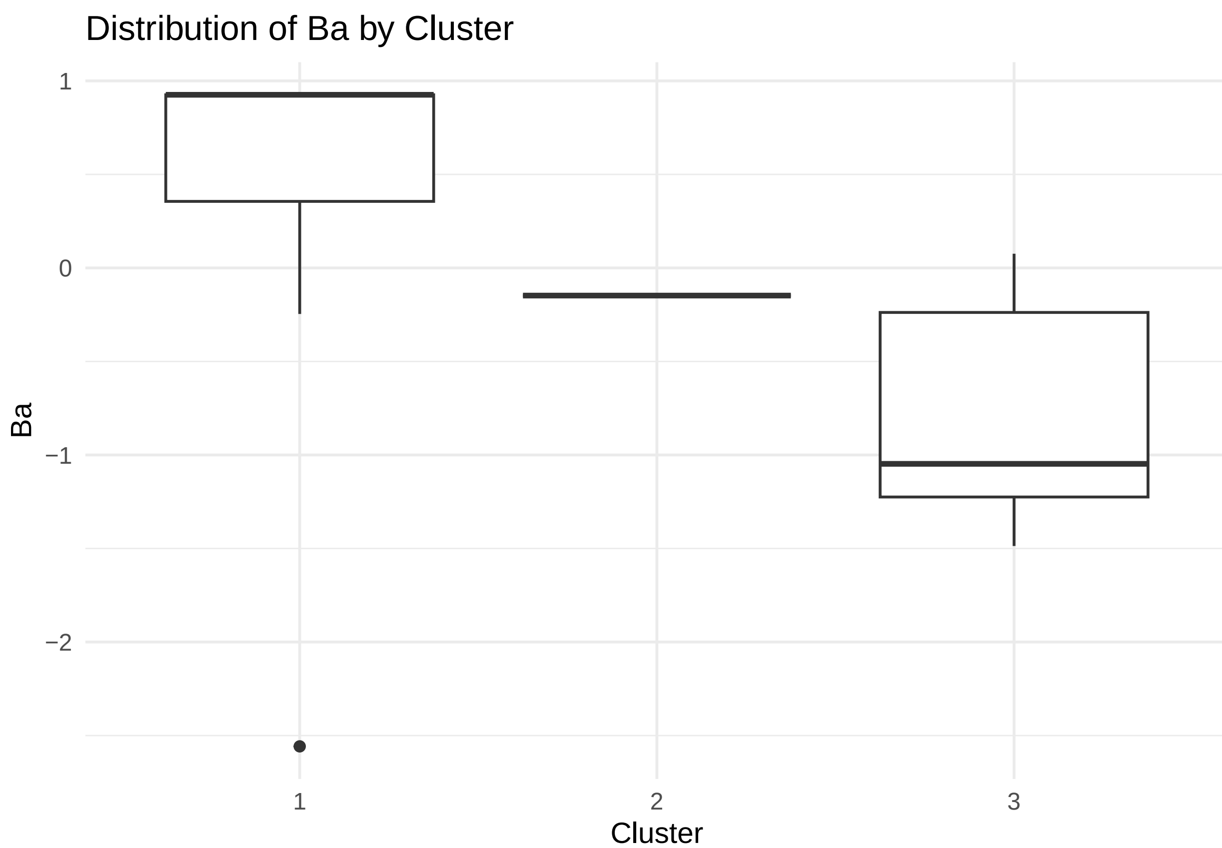

Distribution of Mg by Cluster
0
−1
−2
−3
−4
Mg
1
2
3
Cluster

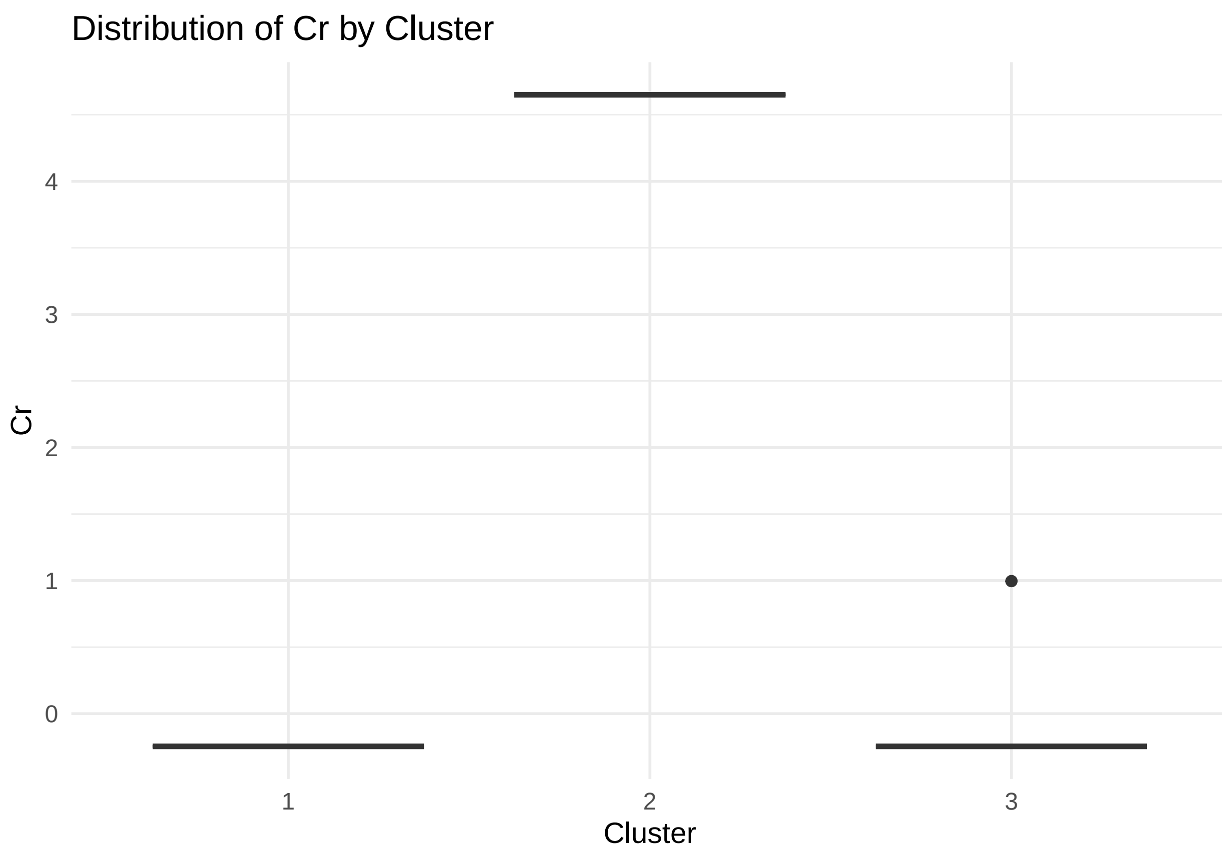

Distribution of Ca by Cluster
Ca
0
−1
−2
−3
−4
1
2
3
Cluster

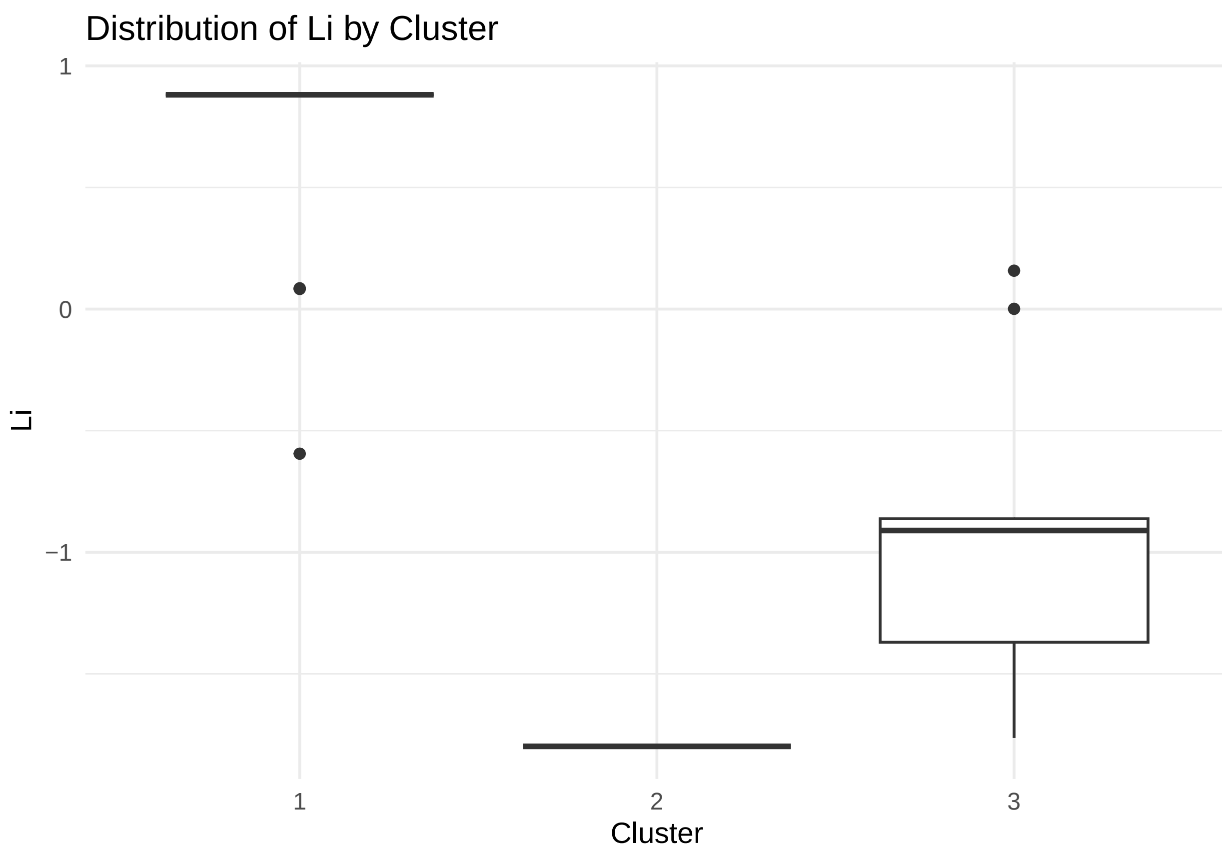

Distribution of Alkalinity by Cluster
Alkalinity
0
−1
−2
1
2
3
Cluster

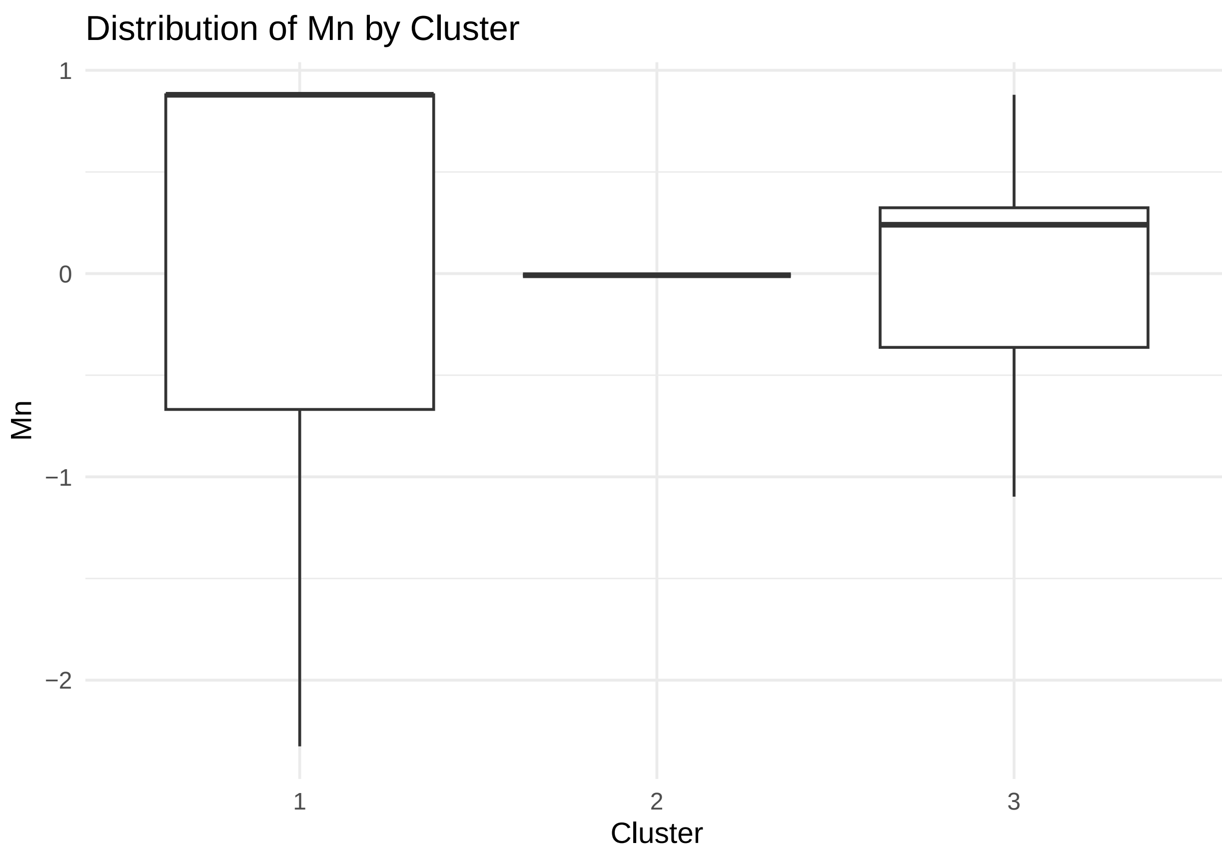

Distribution of DOC by Cluster
1
0
−1
−2
−3
DOC
1
2
3
Cluster

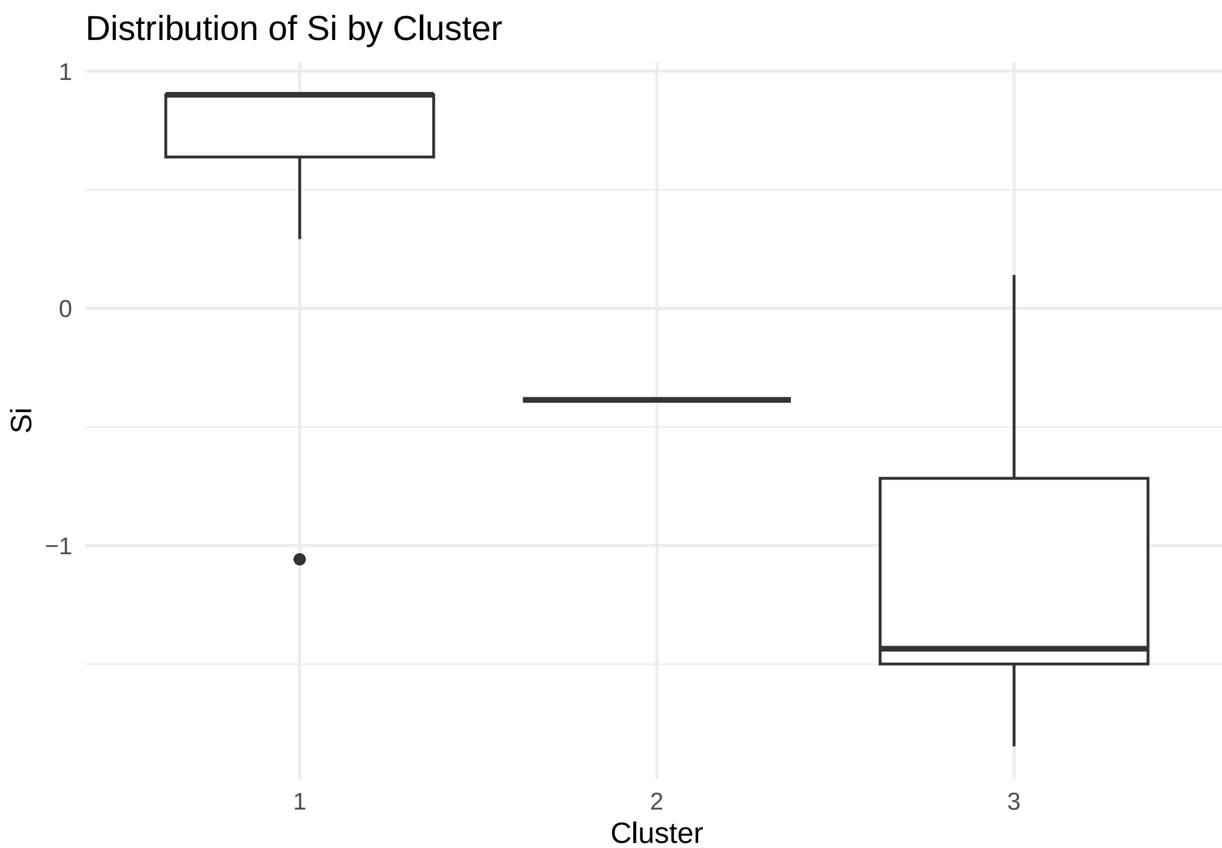

Distribution of As by Cluster
As
3
2
1
0
−1
−2
1
2
3
Cluster

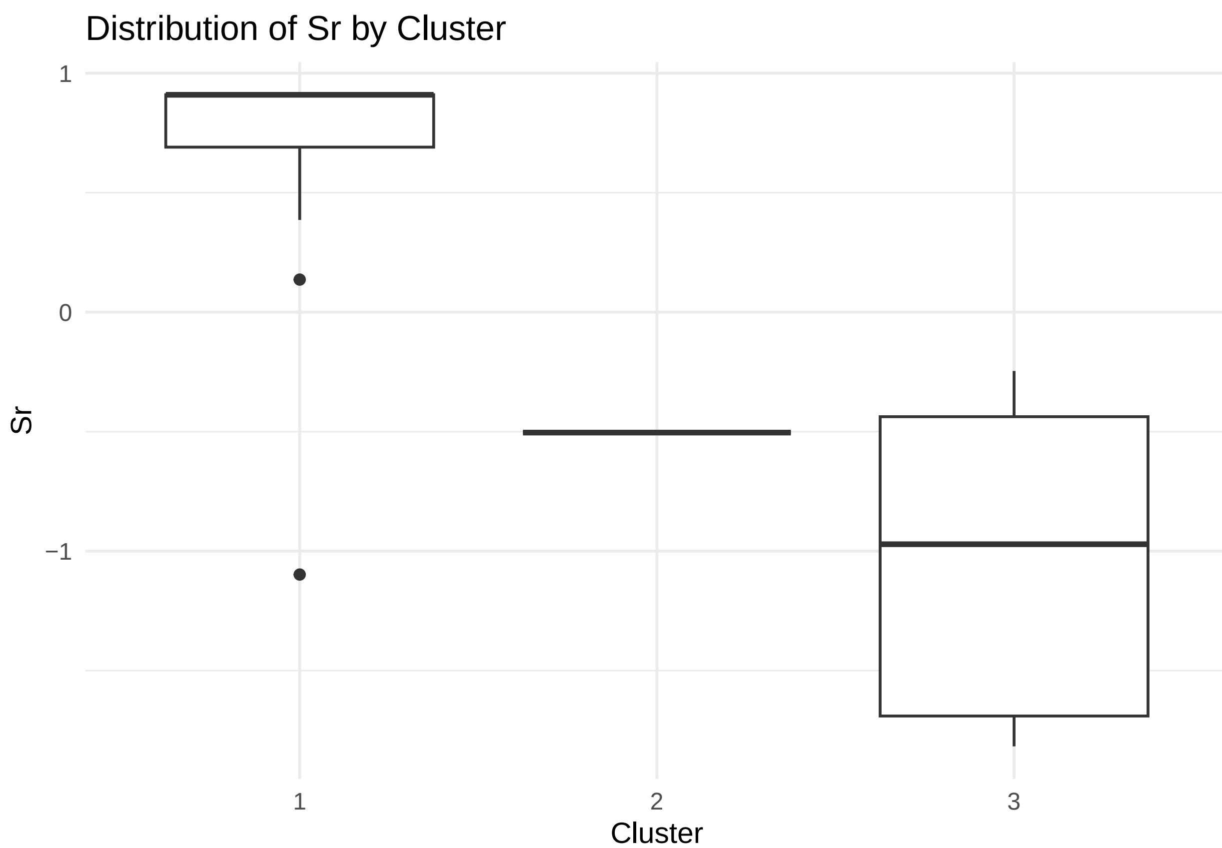

Distribution of Ba by Cluster
1
0
−1
−2
Ba
1
2
3
Cluster

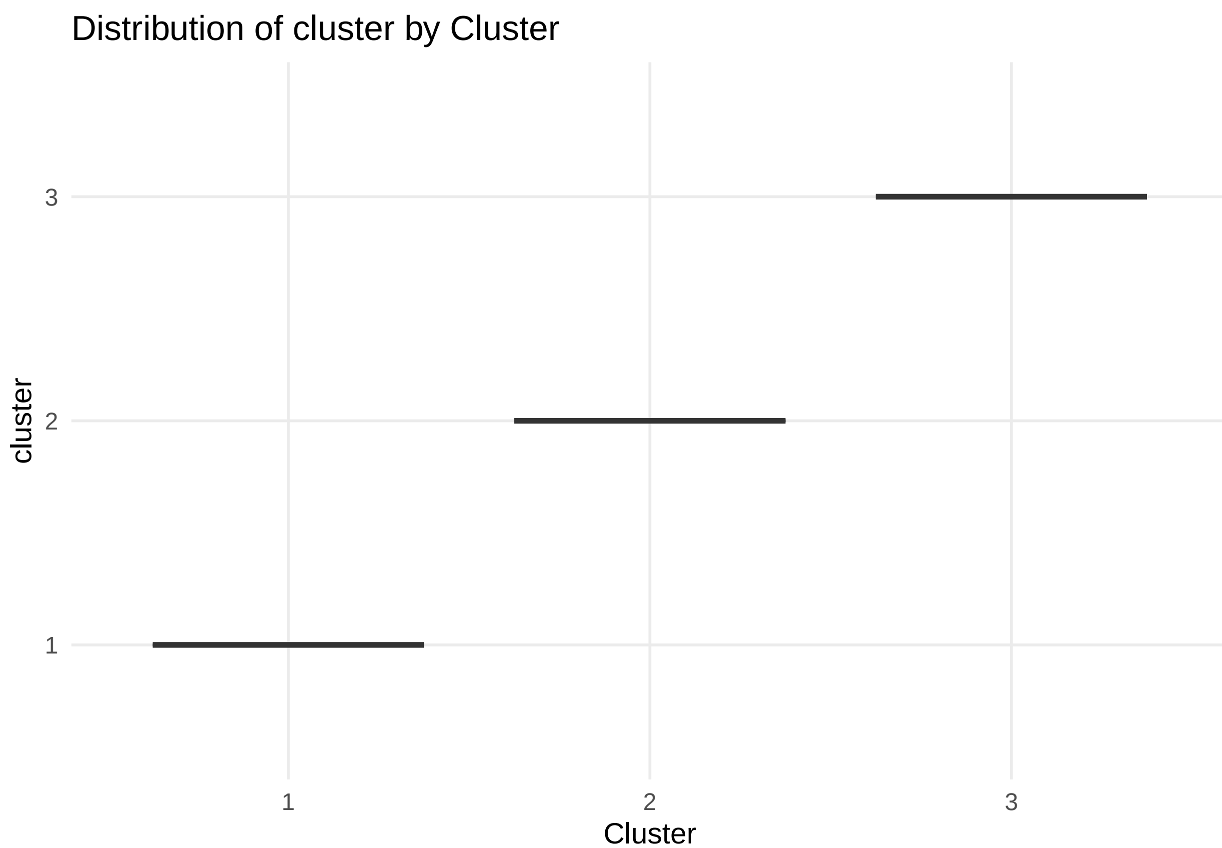

Distribution of Cr by Cluster
Cr
4
3
2
1
0
1
2
3
Cluster


Distribution of Li by Cluster
1
0
−1
Li
1
2
3
Cluster

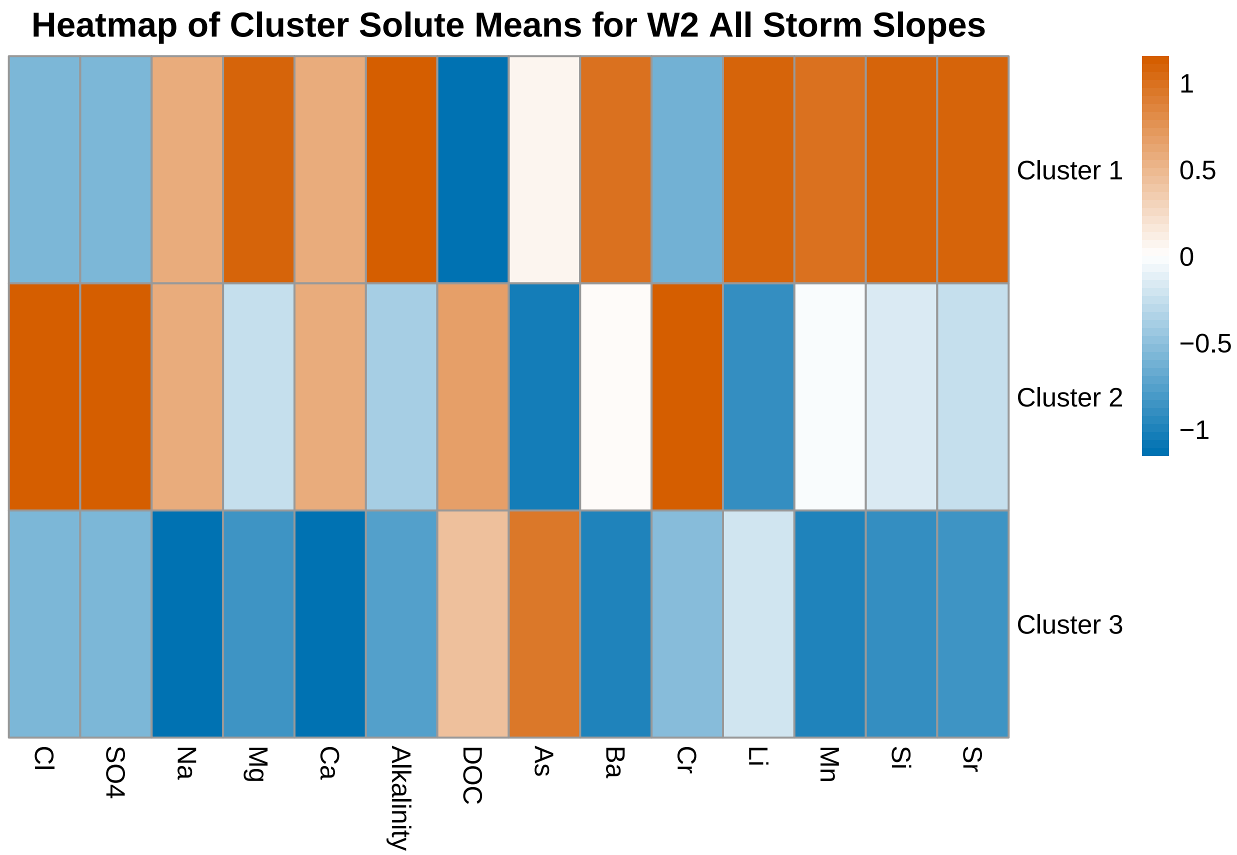

Distribution of Mn by Cluster
1
0
−1
−2
Mn
1
2
3
Cluster


Distribution of Si by Cluster
1
0
−1
Si
1
2
3
Cluster

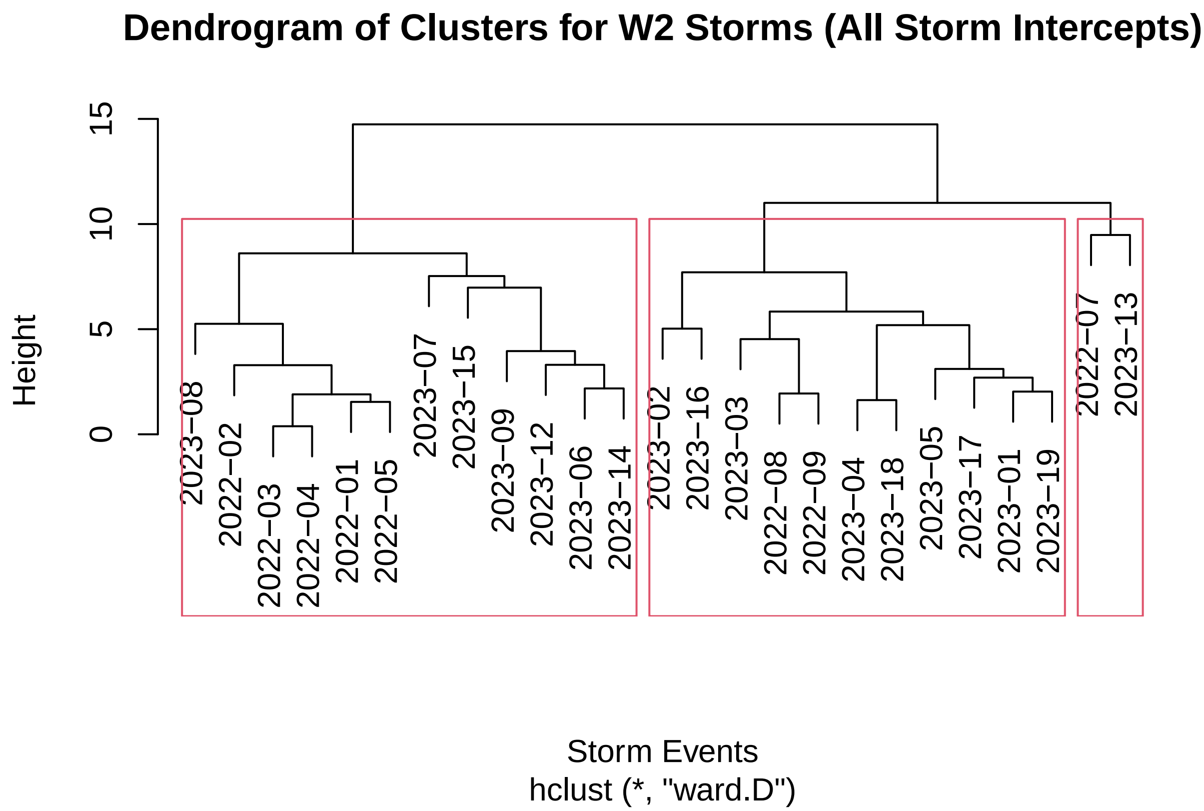

Distribution of Sr by Cluster
1
0
−1
Sr
1
2
3
Cluster




```r
# Let's make some heat maps!
# Cluster Profiling: Aggregate the original variables by cluster
cluster_summary_all_slope <- aggregate(. ~ cluster, data = w2_std_all_slope, FUN = mean)

# Remove the cluster column for visualization
heatmap_data_all_slope <- cluster_summary_all_slope[, -1]

# Set row names as cluster identifiers
rownames(heatmap_data_all_slope) <- paste("Cluster", cluster_summary_all_slope$cluster)

# Create a heatmap
pheatmap(
  mat = as.matrix(heatmap_data_all_slope),
  main = "Heatmap of Cluster Solute Means for W2 All Storm Slopes",
  cluster_rows = FALSE,  # Don't re-cluster rows (clusters are predefined)
  cluster_cols = FALSE,   # Cluster variables
  scale = "column",       # Scale variables to make them comparable
  color = colorRampPalette(c("#0072B2", "white", "#D55E00"))(50) # Gradient color scheme
)
```




```r
# Hierarchical Clustering
dist_matrix_all_intercept <- dist(w2_std_all_intercept, method = "euclidean")
cluster_result_all_intercept <- hclust(dist_matrix_all_intercept, method = "ward.D")

# Plot dendrogram for visualization
plot(cluster_result_all_intercept, labels = rownames(w2_std_all_intercept), main = "Dendrogram of Clust
     xlab = "Storm Events")
rect.hclust(cluster_result_all_intercept, k = 3)
```

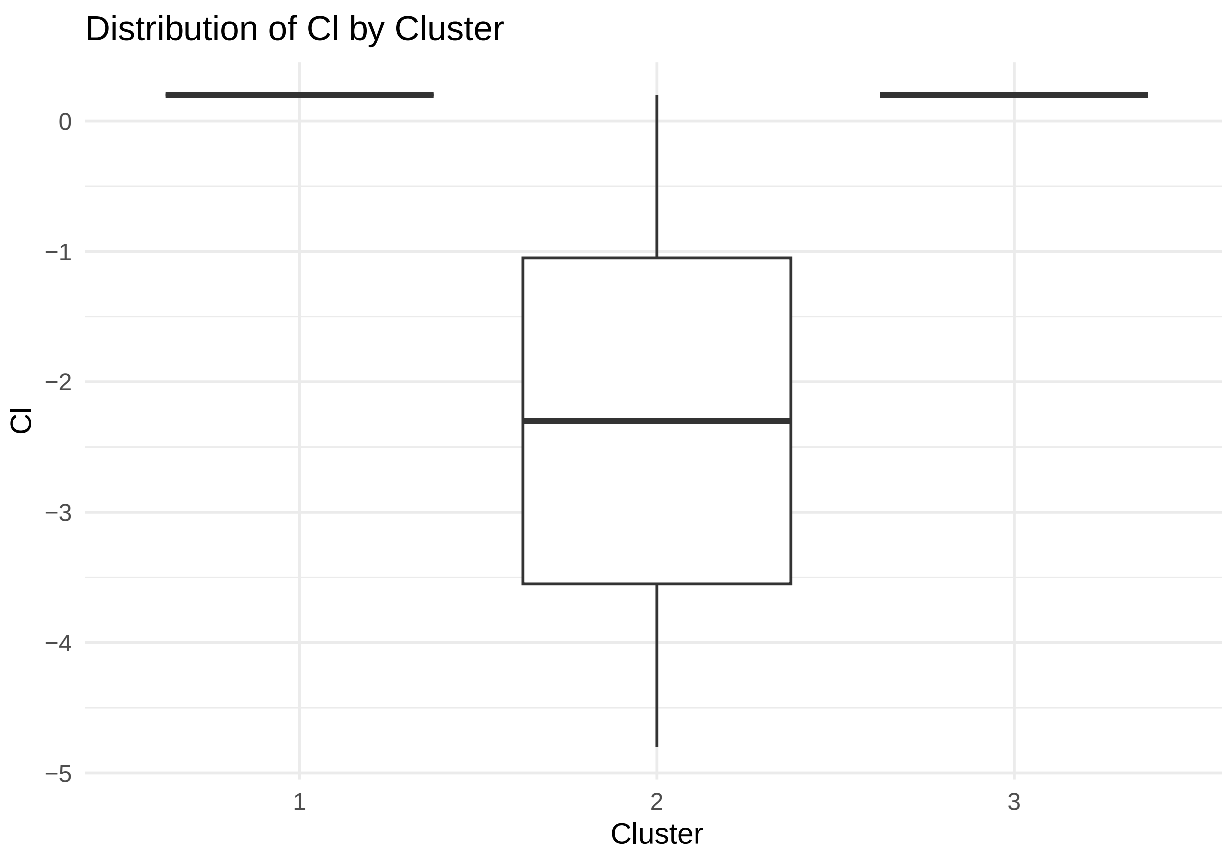



```r
# Cut the dendrogram to create 3 clusters
cuts_all_intercept <- cutree(cluster_result_all_intercept, k = 3)
w2_std_all_intercept$cluster <- as.factor(cuts_all_intercept)

# Cluster Profiling
# Aggregate the original variables by cluster to identify differences
cluster_summary_all_intercept <- aggregate(w2_std_all_intercept, by = list(cluster = cuts_all_intercept)
print(cluster_summary_all_intercept)
```

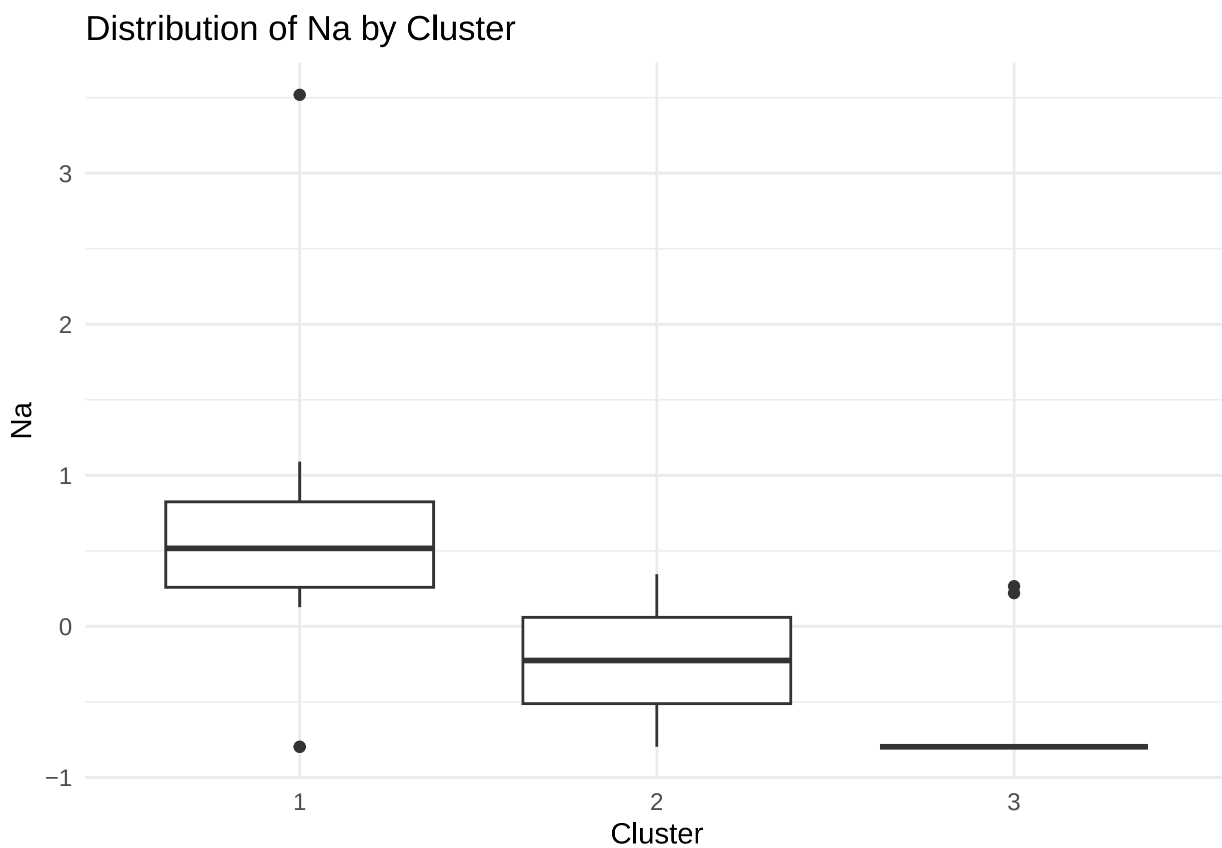
```
##   cluster   Cl        SO4         Na         Mg          Ca Alkalinity
## 1       1  0.2  0.4466180  0.5949961  0.6077298  0.35112636  0.3966732
## 2       2 -2.3 -1.8327178 -0.2258612 -0.2749348 -1.89034898  1.4752994
## 3       3  0.2 -0.1539983 -0.6080210 -0.6129898 -0.03934712 -0.7009706
##           DOC          As         Ba         Cr         Li          Mn
## 1  0.35000823 -0.25945370  0.6026770 -0.1388070 -0.4717552 -0.03376542
## 2 -0.04609468  1.73220387 -0.3145297 -0.2430307  0.1991241 -0.70136636
## 3 -0.37344631 -0.03190576 -0.6002785  0.1956132  0.4784376  0.16435616
##           Si         Sr cluster
## 1  0.4171746  0.6154437      NA
## 2  1.0133874  0.1925182      NA
## 3 -0.6393518 -0.7063964      NA
```

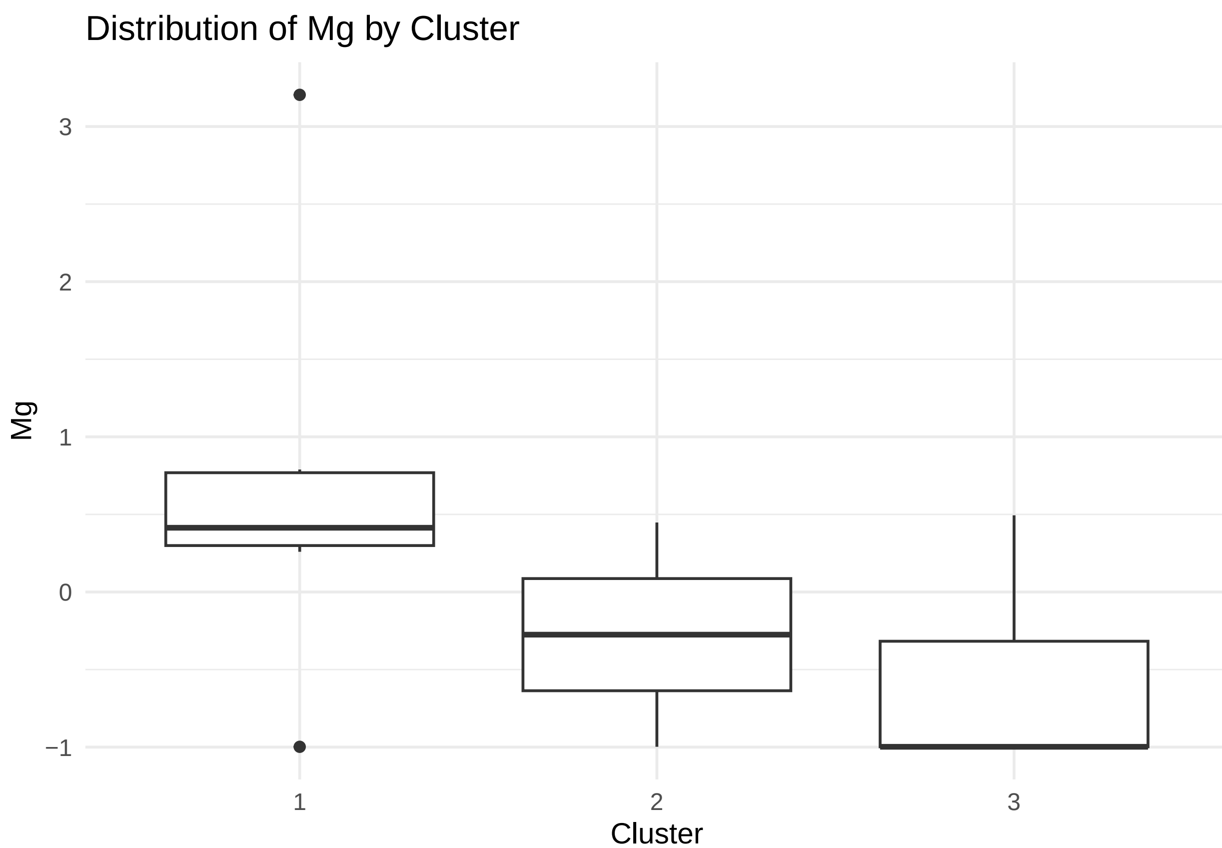
```r
# Boxplots to visualize differences in key variables across clusters
for (variable in colnames(w2_std_all_intercept)) {
  ggplot(w2_std_all_intercept, aes(x = as.factor(cuts_all_intercept), y = .data[[variable]])) +
```

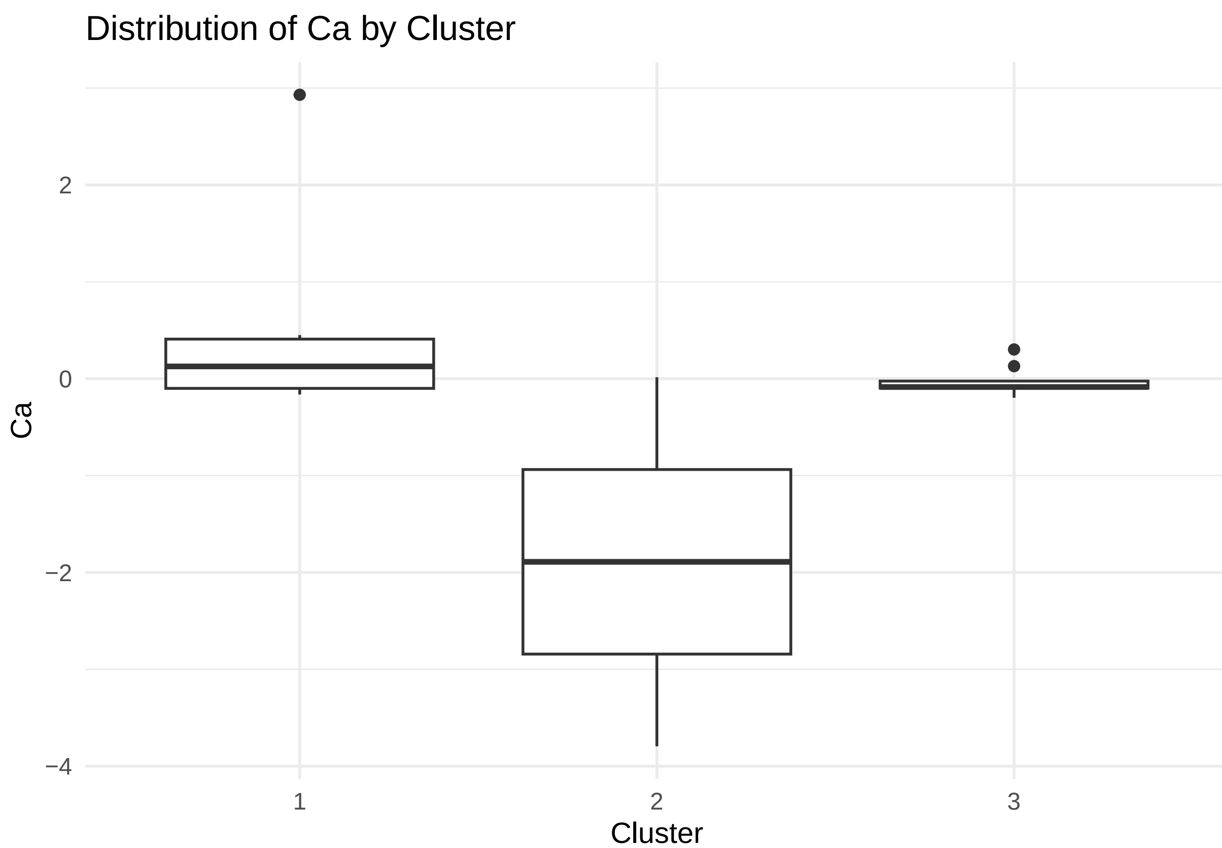
```
    geom_boxplot() +
    labs(title = paste("Distribution of", variable, "by Cluster"),
         x = "Cluster",
         y = variable) +
    theme_minimal() -> plot
  print(plot)
}
```

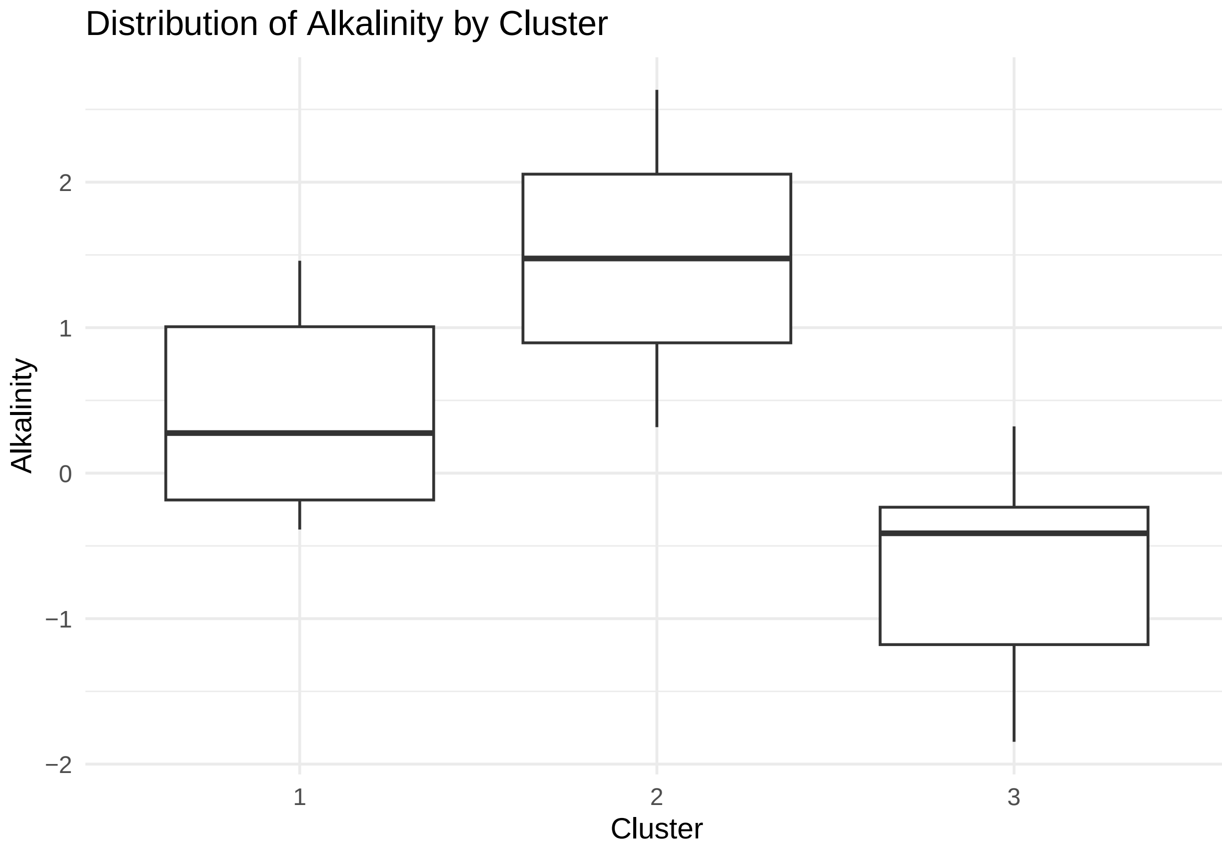


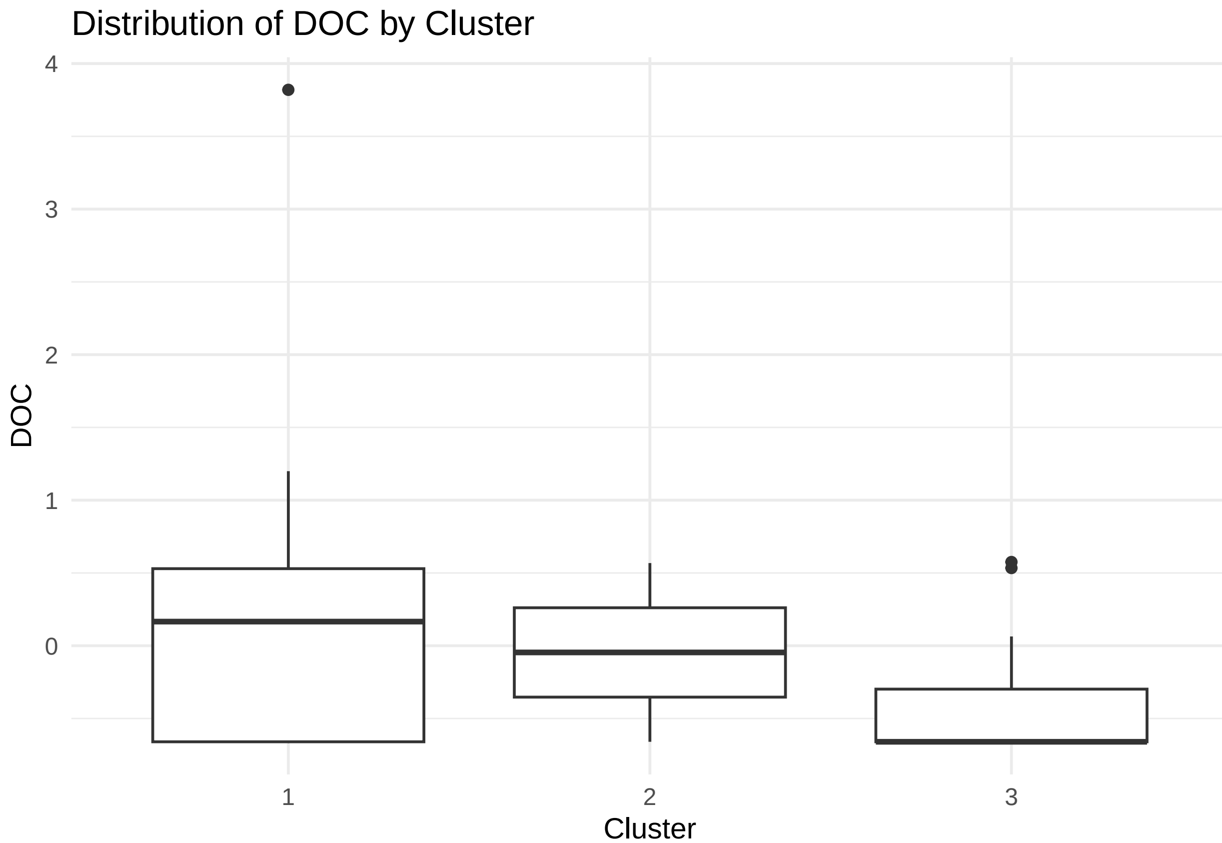

Distribution of SO4 by Cluster
SO4
1
0
−1
−2
−3
1
2
3
Cluster

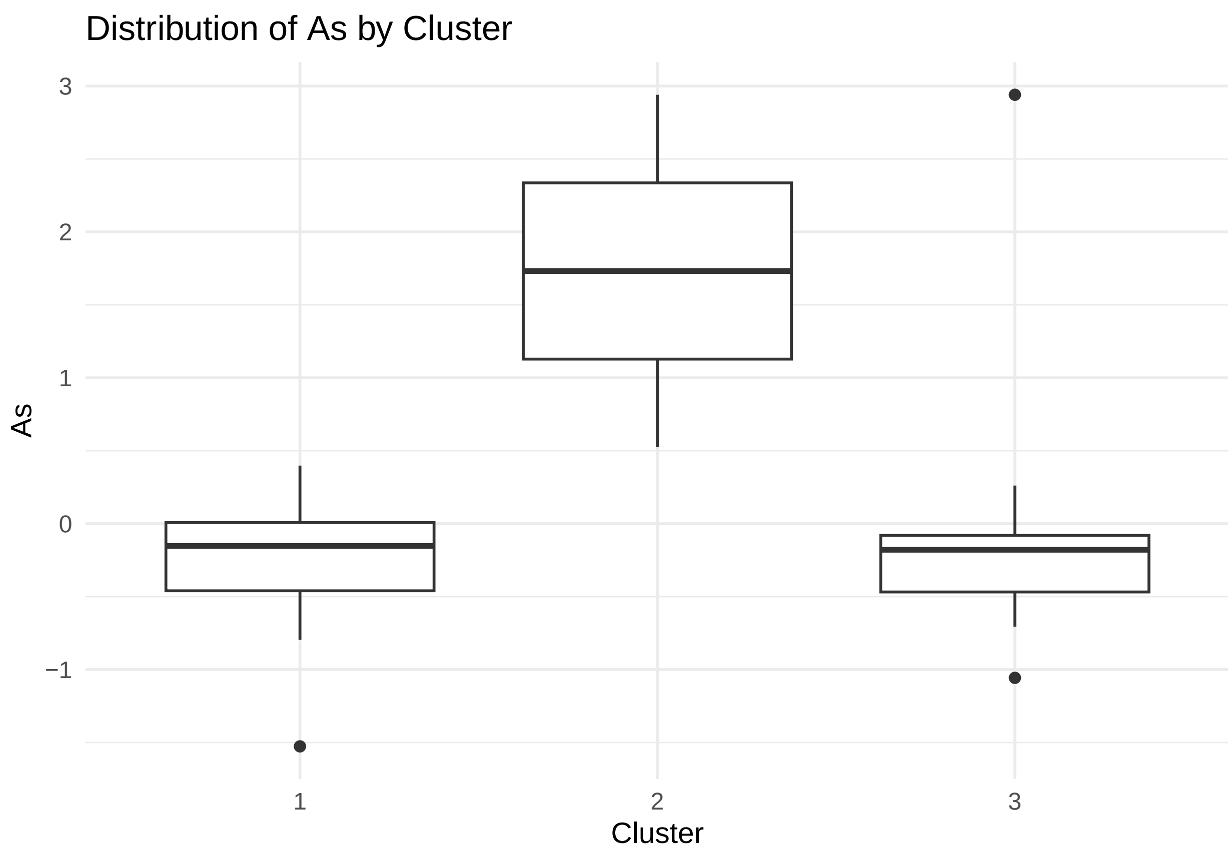

Distribution of Na by Cluster
Na
3
2
1
0
−1
1
2
3
Cluster

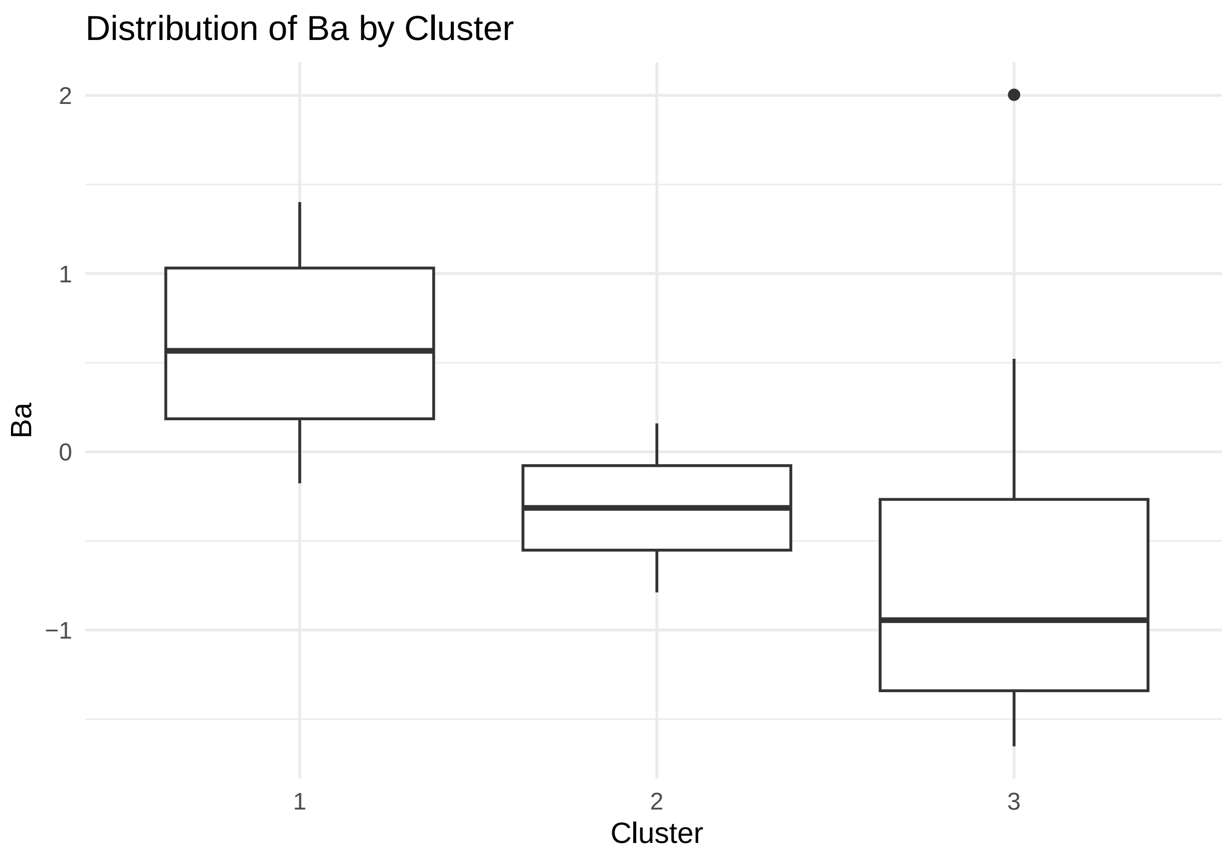

Distribution of Mg by Cluster
Mg
3
2
1
0
−1
1
2
3
Cluster

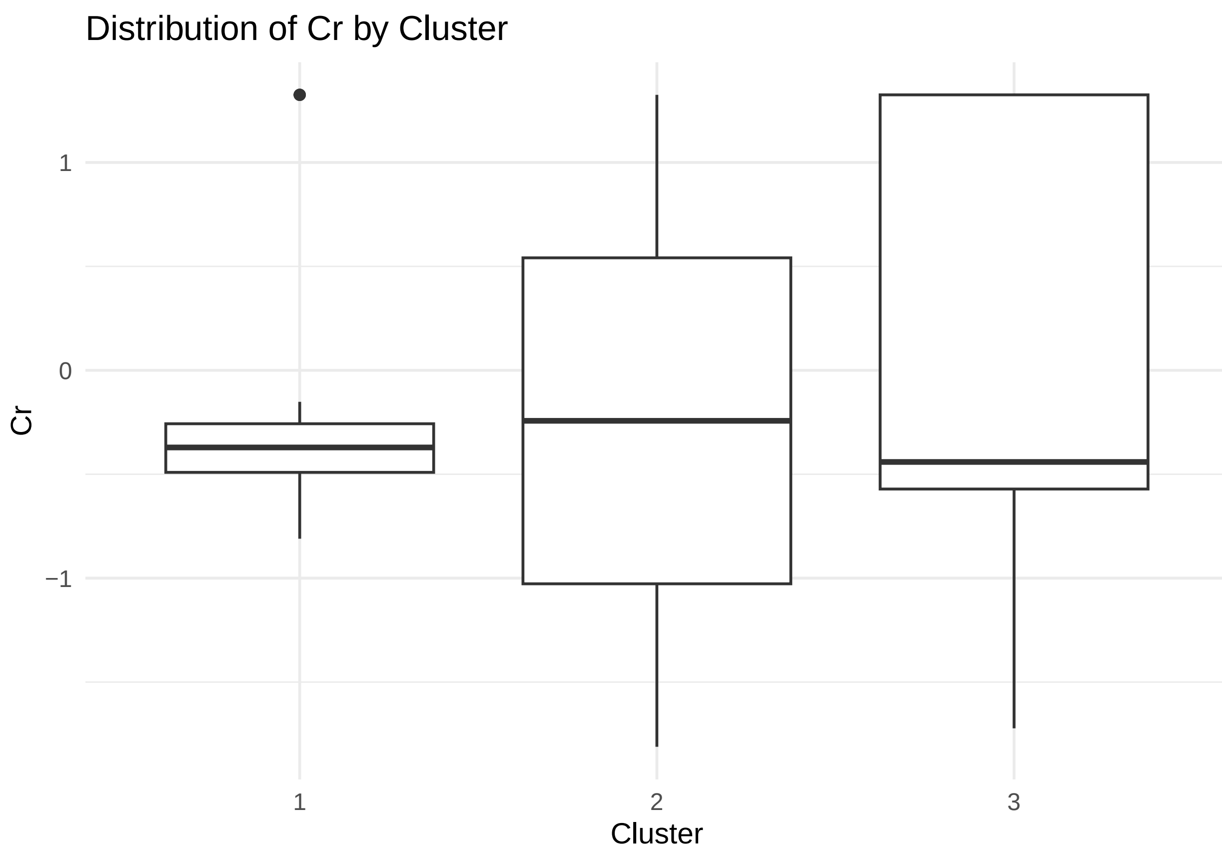

Distribution of Ca by Cluster
2
0
−2
−4
Ca
1
2
3
Cluster

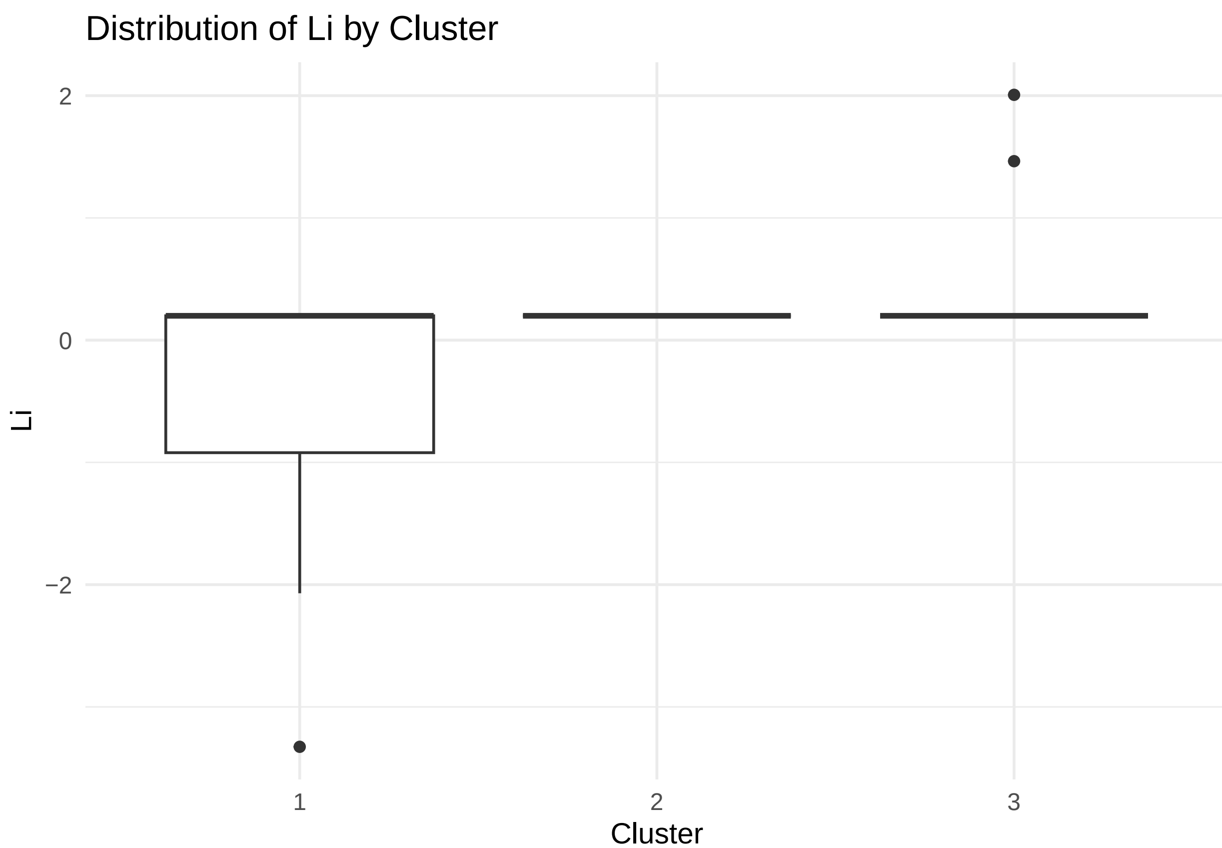

Distribution of Alkalinity by Cluster
Alkalinity
2
1
0
−1
−2
1
2
3
Cluster

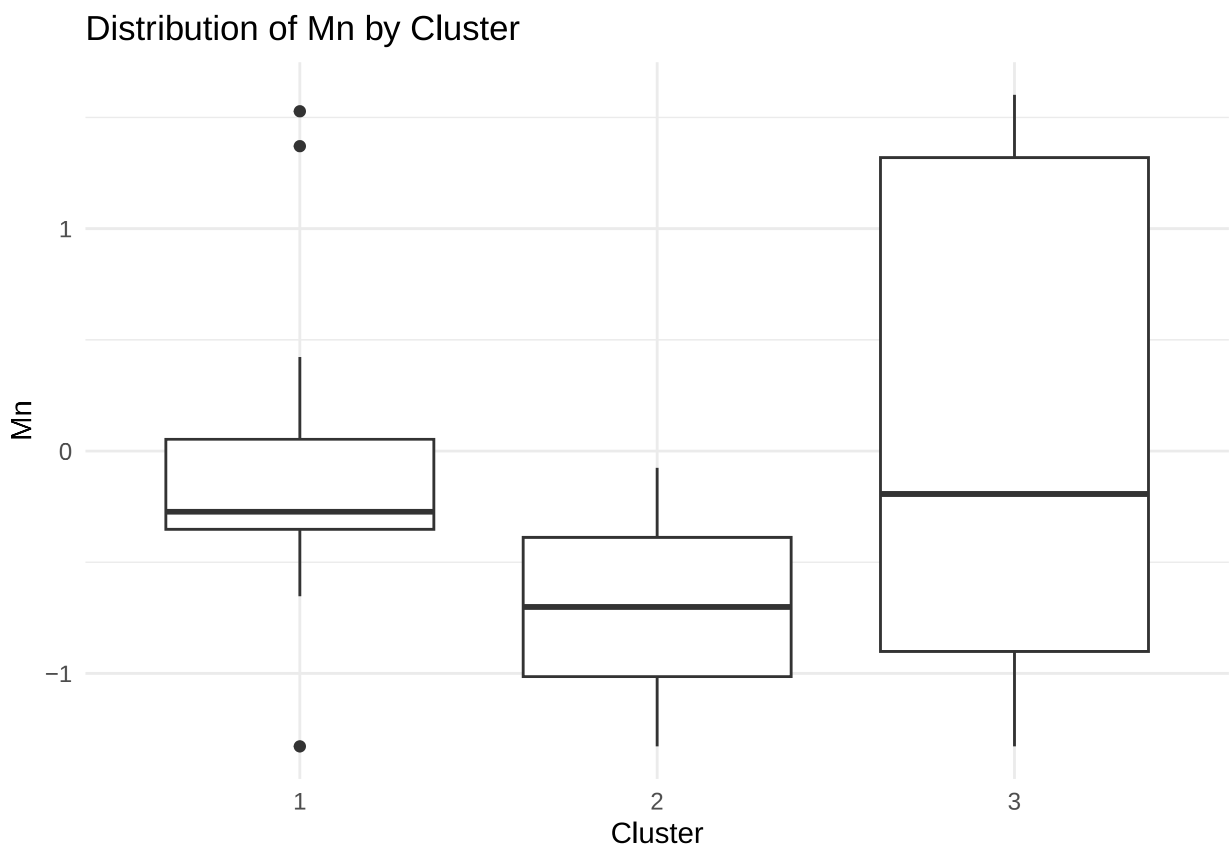

Distribution of DOC by Cluster
DOC
4
3
2
1
0
1
2
3
Cluster


Distribution of As by Cluster
As
3
2
1
0
−1
1
2
3
Cluster

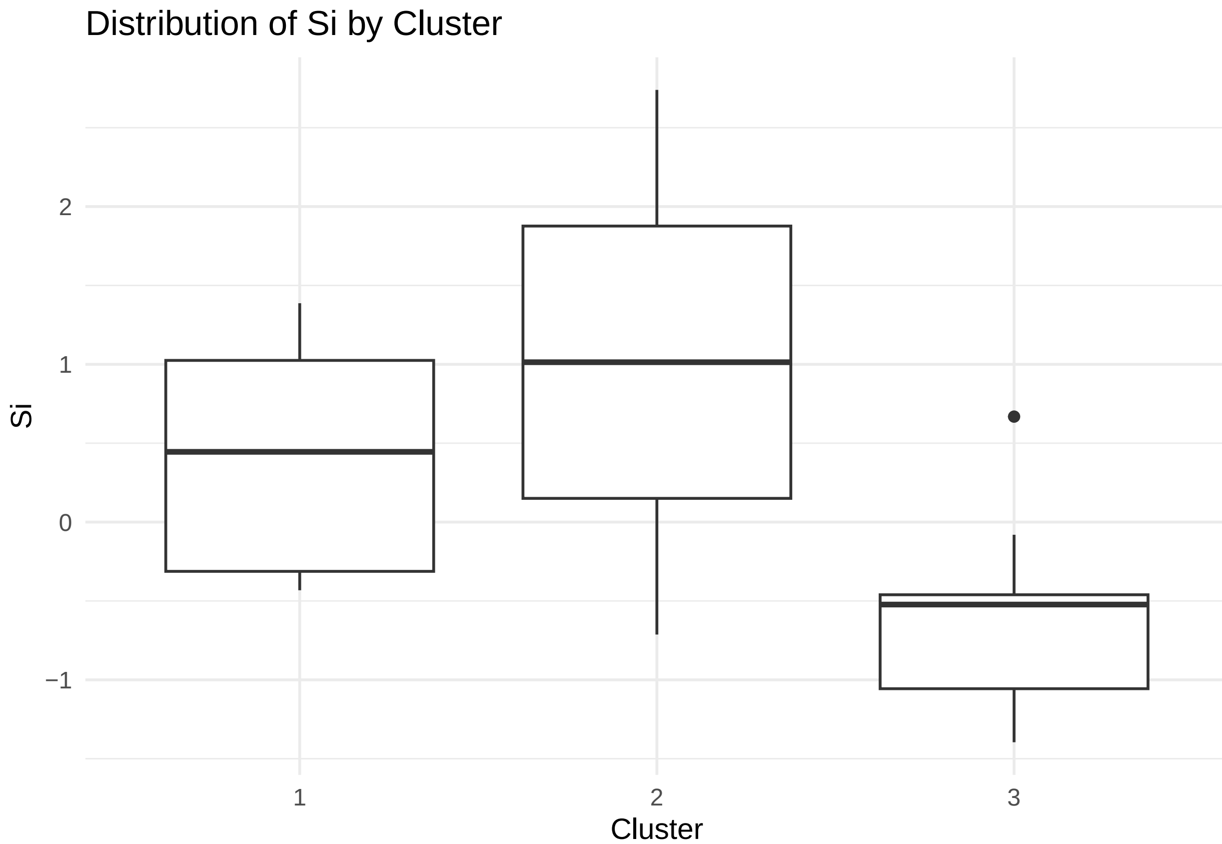

Distribution of Ba by Cluster
Ba
2
1
0
−1
1
2
3
Cluster

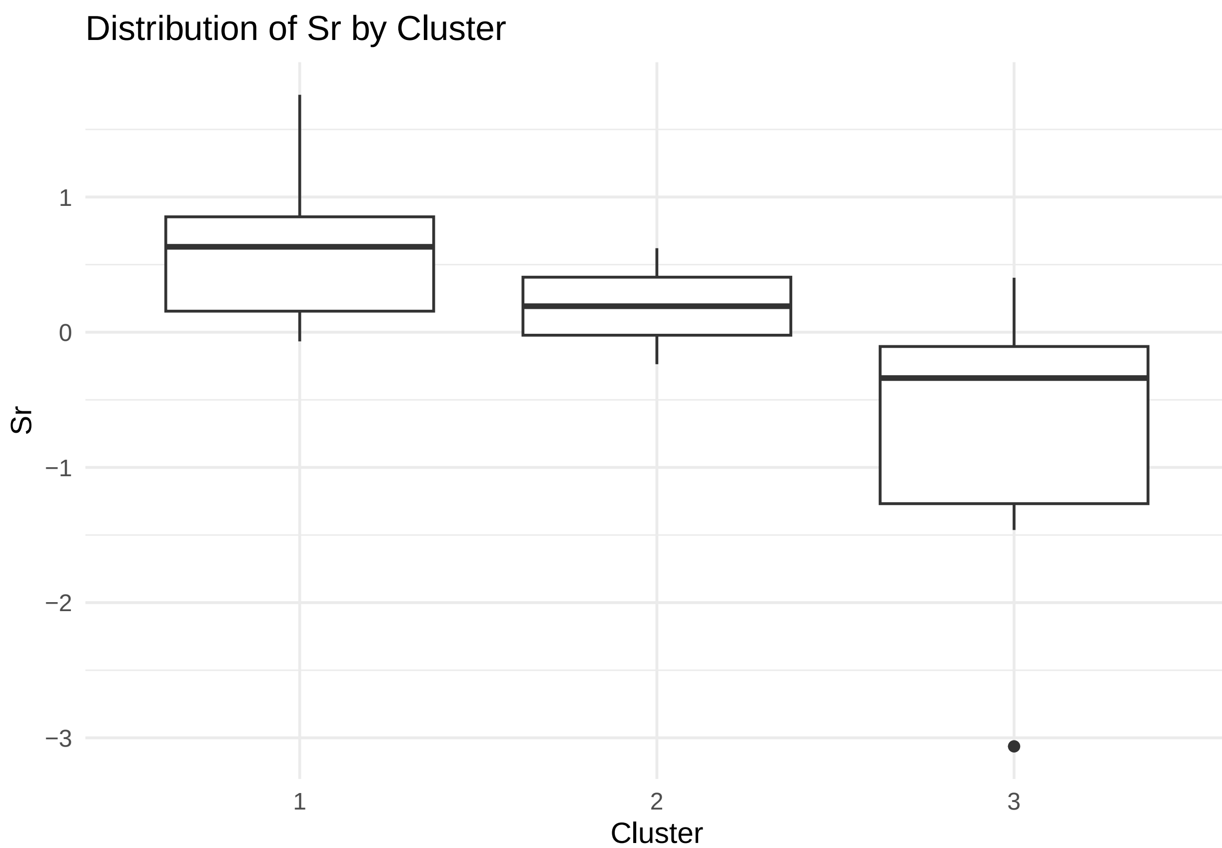

Distribution of Cr by Cluster
Cr
1
0
−1
1
2
3
Cluster


Distribution of Li by Cluster
2
0
−2
Li
1
2
3
Cluster

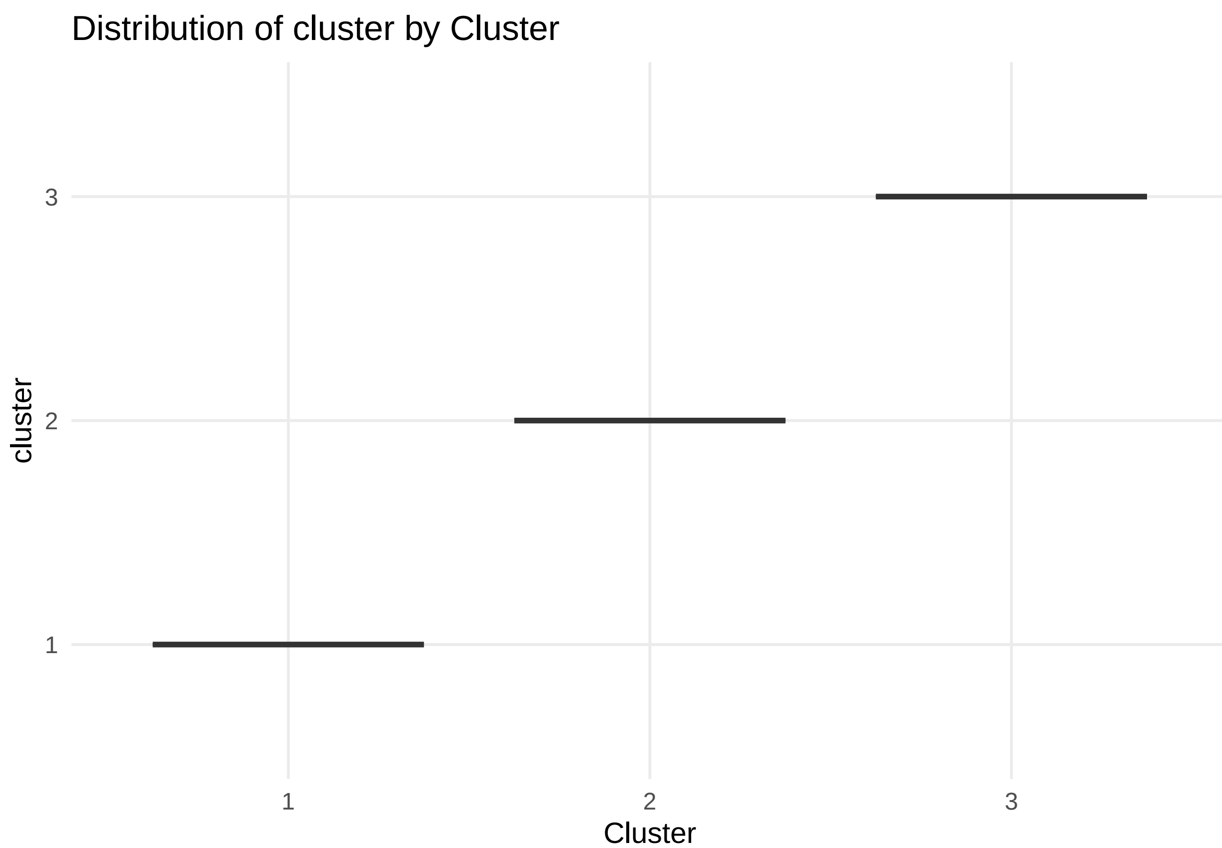


Distribution of Mn by Cluster
Mn
1
0
−1
1
2
3
Cluster

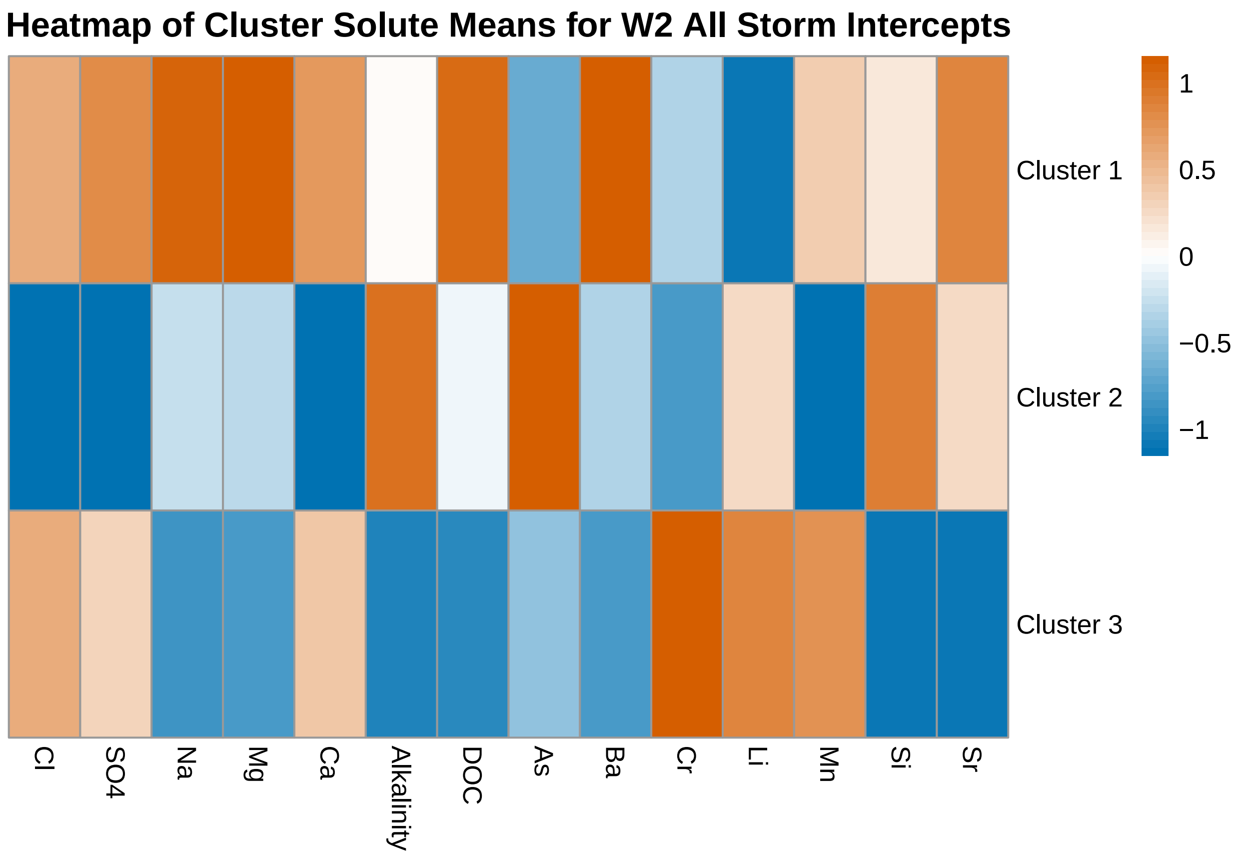

Distribution of Si by Cluster
Si
2
1
0
−1
1
2
3
Cluster



Distribution of Sr by Cluster
1
0
−1
−2
−3
Sr
1
2
3
Cluster




```r
# Cluster Profiling: Aggregate the original variables by cluster
cluster_summary_all_intercept <- aggregate(. ~ cluster, data = w2_std_all_intercept, FUN = mean)

# Remove the cluster column for visualization
heatmap_data_all_intercept <- cluster_summary_all_intercept[, -1]

# Set row names as cluster identifiers
rownames(heatmap_data_all_intercept) <- paste("Cluster", cluster_summary_all_intercept$cluster)

# Create a heatmap
pheatmap(
  mat = as.matrix(heatmap_data_all_intercept),
  main = "Heatmap of Cluster Solute Means for W2 All Storm Intercepts",
  cluster_rows = FALSE,  # Don't re-cluster rows (clusters are predefined)
  cluster_cols = FALSE,   # Cluster variables
  scale = "column",       # Scale variables to make them comparable
  color = colorRampPalette(c("#0072B2", "white", "#D55E00"))(50) # Gradient color scheme
)
```



Let's rank and pull out the variables with the highest and lowest means for each cluster and the variables that vary the most between clusters. We will use this in a post-hoc way to describe cluster membership.

```
# Function to identify top variables per cluster
identify_top_variables <- function(cluster_summary, n = 5) {
  cluster_summary %>%
    pivot_longer(cols = -cluster, names_to = "Variable", values_to = "Mean") %>%
    group_by(cluster) %>%
    arrange(desc(abs(Mean))) %>% # Sort by absolute mean
    slice_head(n = n) %>%        # Select top `n` variables
    ungroup()
}

# Apply to each dataset
top_variables_all_slope <- identify_top_variables(cluster_summary_all_slope)
top_variables_post_slope <- identify_top_variables(cluster_summary_post_slope)
top_variables_pre_slope <- identify_top_variables(cluster_summary_pre_slope)
top_variables_all_intercept <- identify_top_variables(cluster_summary_all_intercept)
top_variables_post_intercept <- identify_top_variables(cluster_summary_post_intercept)
top_variables_preintercept <- identify_top_variables(cluster_summary_pre_intercept)

# View results
top_variables_all_slope
```

```
## # A tibble: 15 x 3
##    cluster Variable     Mean
##    <fct>   <chr>       <dbl>
```

```
##  1 1       Alkalinity  0.744
##  2 1       Li          0.677
##  3 1       Sr          0.660
##  4 1       Si          0.656
##  5 1       Ba          0.495
##  6 2       Cl          4.8
##  7 2       SO4         4.80
##  8 2       Cr          4.65
##  9 2       Li         -1.80
## 10 2       As         -1.77
## 11 3       Alkalinity -1.15
## 12 3       Si         -1.05
## 13 3       Sr         -1.04
## 14 3       Li         -0.928
## 15 3       Ba         -0.809
```

```
top_variables_post_slope
```

```
## # A tibble: 15 x 3
##    cluster Variable     Mean
##    <fct>   <chr>       <dbl>
##  1 1       Si          1.01
##  2 1       Alkalinity  0.980
##  3 1       Ba          0.974
##  4 1       Sr          0.970
##  5 1       Li          0.925
##  6 2       Ba         -0.797
##  7 2       Si         -0.790
##  8 2       Alkalinity -0.775
##  9 2       Li         -0.772
## 10 2       Sr         -0.753
## 11 3       Ca         -3.22
## 12 3       Mg         -3.01
## 13 3       Na         -2.96
## 14 3       Sr         -1.30
## 15 3       Si         -1.30
```

```
top_variables_pre_slope
```

```
## # A tibble: 15 x 3
##    cluster Variable     Mean
##    <fct>   <chr>       <dbl>
##  1 1       Mg          0.427
##  2 1       Cr         -0.359
##  3 1       Alkalinity  0.337
##  4 1       Li          0.323
##  5 1       Na          0.289
##  6 2       SO4         3.18
##  7 2       Cl          3.18
##  8 2       Cr          3.07
##  9 2       As         -2.51
## 10 2       Mg         -1.98
## 11 3       Ca         -3.18
```

```
## 12 3       Na         -3.18
## 13 3       Mg         -2.30
## 14 3       Li         -1.60
## 15 3       Alkalinity -1.53
```

```
top_variables_all_intercept
```

```
## # A tibble: 15 x 3
##    cluster Variable     Mean
##    <fct>   <chr>       <dbl>
##  1 1       Sr          0.615
##  2 1       Mg          0.608
##  3 1       Ba          0.603
##  4 1       Na          0.595
##  5 1       Li         -0.472
##  6 2       Cl         -2.3
##  7 2       Ca         -1.89
##  8 2       SO4        -1.83
##  9 2       As          1.73
## 10 2       Alkalinity  1.48
## 11 3       Sr         -0.706
## 12 3       Alkalinity -0.701
## 13 3       Si         -0.639
## 14 3       Mg         -0.613
## 15 3       Na         -0.608
```

```
top_variables_post_intercept
```

```
## # A tibble: 15 x 3
##    cluster Variable     Mean
##    <fct>   <chr>       <dbl>
##  1 1       Si         -1.01
##  2 1       Alkalinity -0.920
##  3 1       Ba         -0.898
##  4 1       Sr         -0.894
##  5 1       Na         -0.642
##  6 2       Ba          0.749
##  7 2       Sr          0.733
##  8 2       Mg          0.568
##  9 2       Na          0.550
## 10 2       As         -0.487
## 11 3       Ca         -2.74
## 12 3       As          2.09
## 13 3       Si          2.04
## 14 3       Alkalinity  1.97
## 15 3       Mn         -1.37
```

```
top_variables_preintercept
```

```
## # A tibble: 15 x 3
##    cluster Variable     Mean
##    <fct>   <chr>       <dbl>
```

```
##   1 1       Na          0.661
##   2 1       Mg          0.658
##   3 1       SO4         0.526
##   4 1       DOC         0.402
##   5 1       Sr          0.400
##   6 2       Cl         -3.18
##   7 2       SO4        -2.64
##   8 2       As          1.64
##   9 2       Cr         -1.54
##  10 2       Alkalinity  1.34
##  11 3       Mg         -1.35
##  12 3       Na         -1.33
##  13 3       Alkalinity -1.02
##  14 3       DOC        -0.978
##  15 3       Sr         -0.943
```

```
# Function to calculate variability across clusters
compute_variable_variability <- function(cluster_summary) {
  cluster_summary %>%
    pivot_longer(cols = -cluster, names_to = "Variable", values_to = "Mean") %>%
    group_by(Variable) %>%
    summarise(
      Range = max(Mean, na.rm = TRUE) - min(Mean, na.rm = TRUE), # Compute range
      SD = sd(Mean, na.rm = TRUE),                               # Compute standard deviation
      .groups = "drop"                                           # Ungroup after summarizing
    ) %>%
    arrange(desc(Range)) # Sort by variability (Range)
}

# Apply to each dataset
variability_all_slope <- compute_variable_variability(cluster_summary_all_slope)
variability_post_slope <- compute_variable_variability(cluster_summary_post_slope)
variability_pre_slope <- compute_variable_variability(cluster_summary_pre_slope)
variability_all_intercept <- compute_variable_variability(cluster_summary_all_intercept)
variability_post_intercept <- compute_variable_variability(cluster_summary_post_intercept)
variability_pre_intercept <- compute_variable_variability(cluster_summary_pre_intercept)

# View results
variability_all_slope
```

```
## # A tibble: 14 x 3
##    Variable    Range    SD
##    <chr>       <dbl> <dbl>
##  1 Cl          5     2.89
##  2 SO4         5.00  2.89
##  3 Cr          4.90  2.79
##  4 As          2.58  1.29
##  5 Li          2.47  1.26
##  6 Alkalinity 1.90  1.01
##  7 Si          1.71  0.860
##  8 Sr          1.70  0.871
##  9 Ba          1.30  0.652
## 10 DOC         1.23  0.672
## 11 Na          1.14  0.658
```

```
## 12 Mg         1.04   0.533
## 13 Ca         0.750  0.433
## 14 Mn         0.0604 0.0302
```

```
variability_post_slope
```

```
## # A tibble: 11 x 3
##    Variable   Range    SD
##    <chr>      <dbl> <dbl>
##  1 Ca         3.57  2.02
##  2 Na         3.52  1.88
##  3 Mg         3.51  1.90
##  4 Si         2.30  1.21
##  5 Sr         2.27  1.18
##  6 Alkalinity 2.21  1.17
##  7 Ba         2.04  1.11
##  8 Li         1.85  1.03
##  9 DOC        1.33  0.702
## 10 As         1.02  0.586
## 11 Mn         0.379 0.194
```

```
variability_pre_slope
```

```
## # A tibble: 14 x 3
##    Variable   Range     SD
##    <chr>      <dbl>  <dbl>
##  1 As         3.83  1.96
##  2 Ca         3.46  2
##  3 Na         3.46  2
##  4 SO4        3.46  2
##  5 Cl         3.46  2
##  6 Cr         3.43  1.78
##  7 Mg         2.72  1.49
##  8 Alkalinity 2.18  1.18
##  9 Li         1.96  1.12
## 10 Si         1.32  0.726
## 11 Sr         1.29  0.722
## 12 DOC        1.19  0.682
## 13 Ba         0.424 0.223
## 14 Mn         0.138 0.0704
```

```
variability_all_intercept
```

```
## # A tibble: 14 x 3
##    Variable   Range    SD
##    <chr>      <dbl> <dbl>
##  1 Cl         2.5   1.44
##  2 SO4        2.28  1.18
##  3 Ca         2.24  1.20
##  4 Alkalinity 2.18  1.09
##  5 As         1.99  1.09
##  6 Si         1.65  0.837
```

```
##  7 Sr         1.32  0.675
##  8 Mg         1.22  0.630
##  9 Na         1.20  0.615
## 10 Ba         1.20  0.628
## 11 Li         0.950 0.488
## 12 Mn         0.866 0.454
## 13 DOC        0.723 0.362
## 14 Cr         0.439 0.229
```

```
variability_post_intercept
```

```
## # A tibble: 13 x 3
##    Variable   Range    SD
##    <chr>      <dbl> <dbl>
##  1 Ca         3.19  1.71
##  2 Si         3.05  1.52
##  3 Alkalinity 2.89  1.45
##  4 As         2.57  1.32
##  5 Mn         1.66  0.884
##  6 Ba         1.65  0.912
##  7 Sr         1.63  0.879
##  8 Mg         1.43  0.766
##  9 Cr         1.37  0.686
## 10 Na         1.19  0.688
## 11 DOC        0.922 0.486
## 12 SO4        0.760 0.439
## 13 Li         0.515 0.297
```

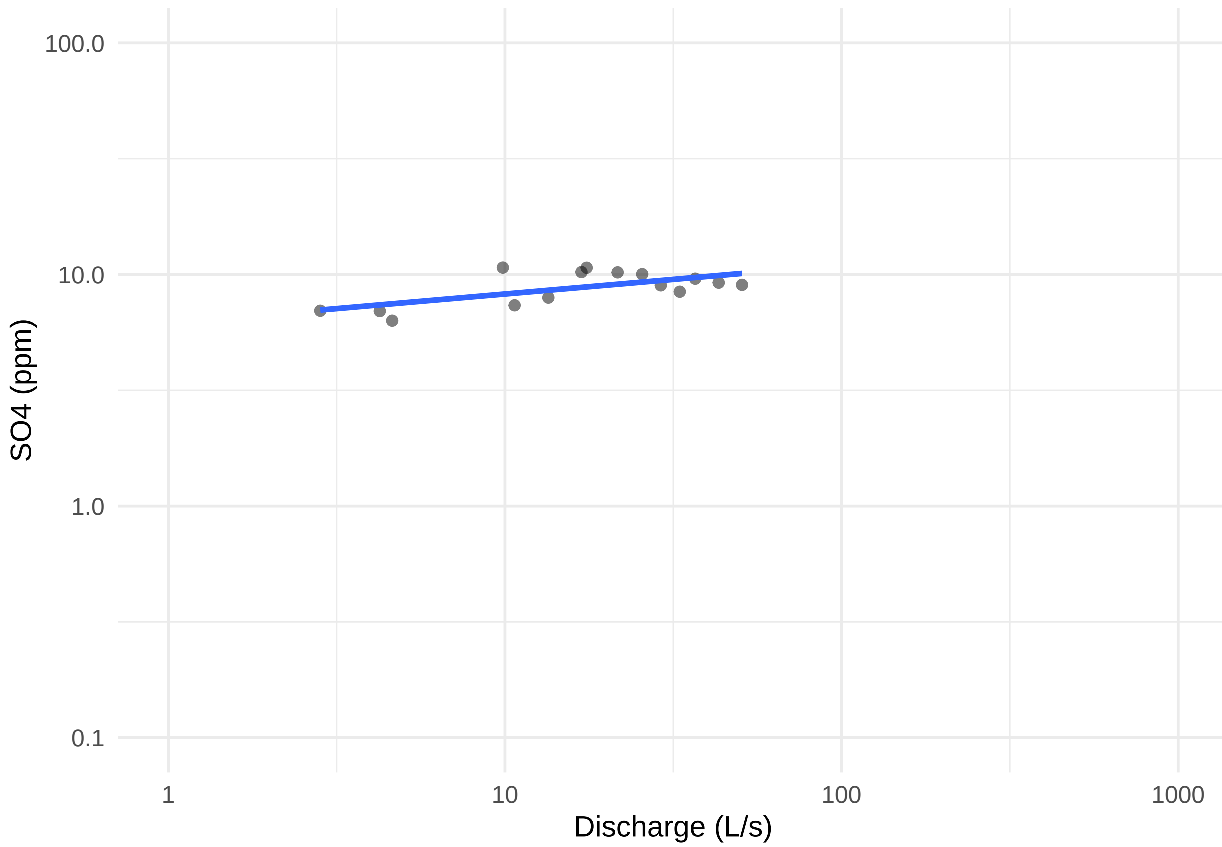
```
variability_pre_intercept
```

```
## # A tibble: 14 x 3
##    Variable   Range    SD
##    <chr>      <dbl> <dbl>
##  1 Cl         3.46  2
##  2 SO4        3.16  1.65
##  3 Alkalinity 2.37  1.19
##  4 As         2.29  1.16
##  5 Mg         2.14  1.20
##  6 DOC        2.08  1.06
##  7 Na         2.01  1.16
##  8 Cr         1.99  1.04
##  9 Sr         1.91  0.982
## 10 Ca         1.00  0.551
## 11 Si         0.567 0.285
## 12 Ba         0.431 0.216
## 13 Mn         0.417 0.230
## 14 Li         0.196 0.113
```

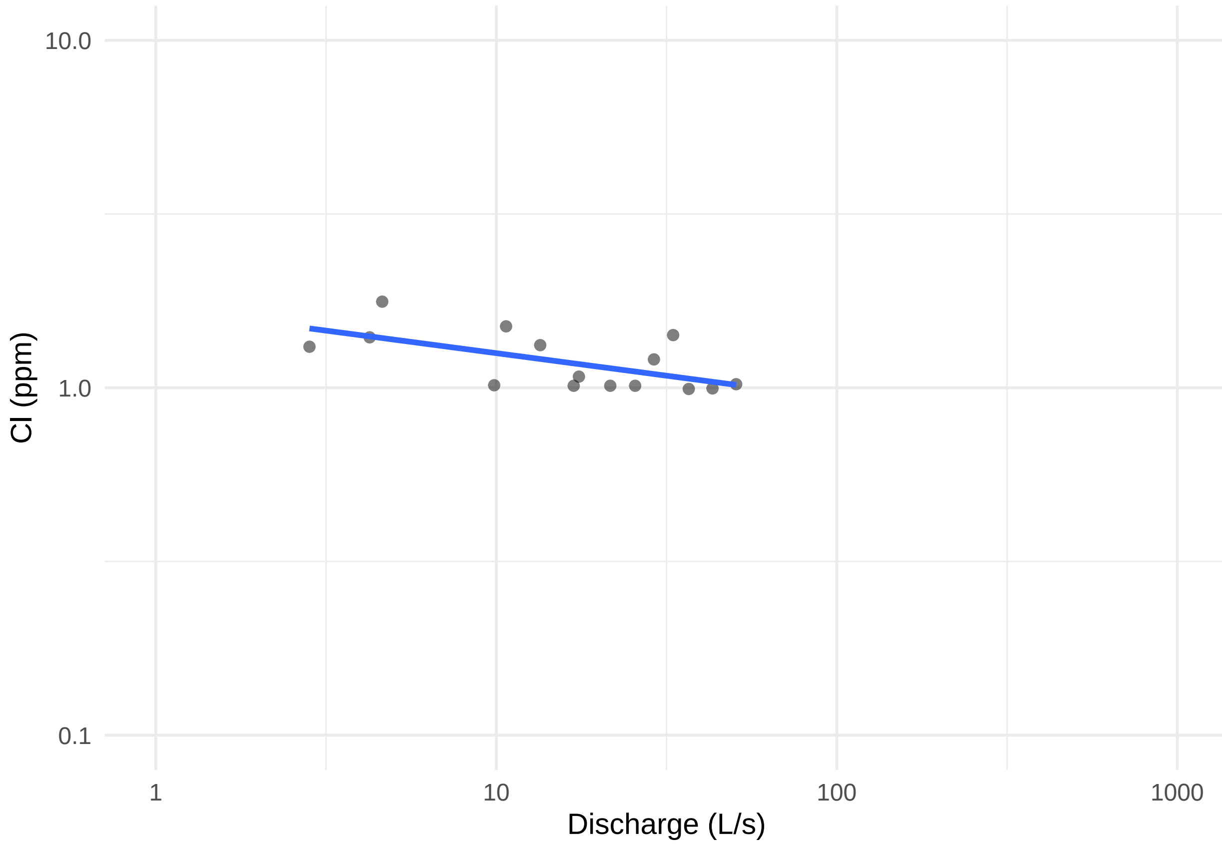
Part VII: Investing Storm 2022-07

This storm clusters quite independently, why?

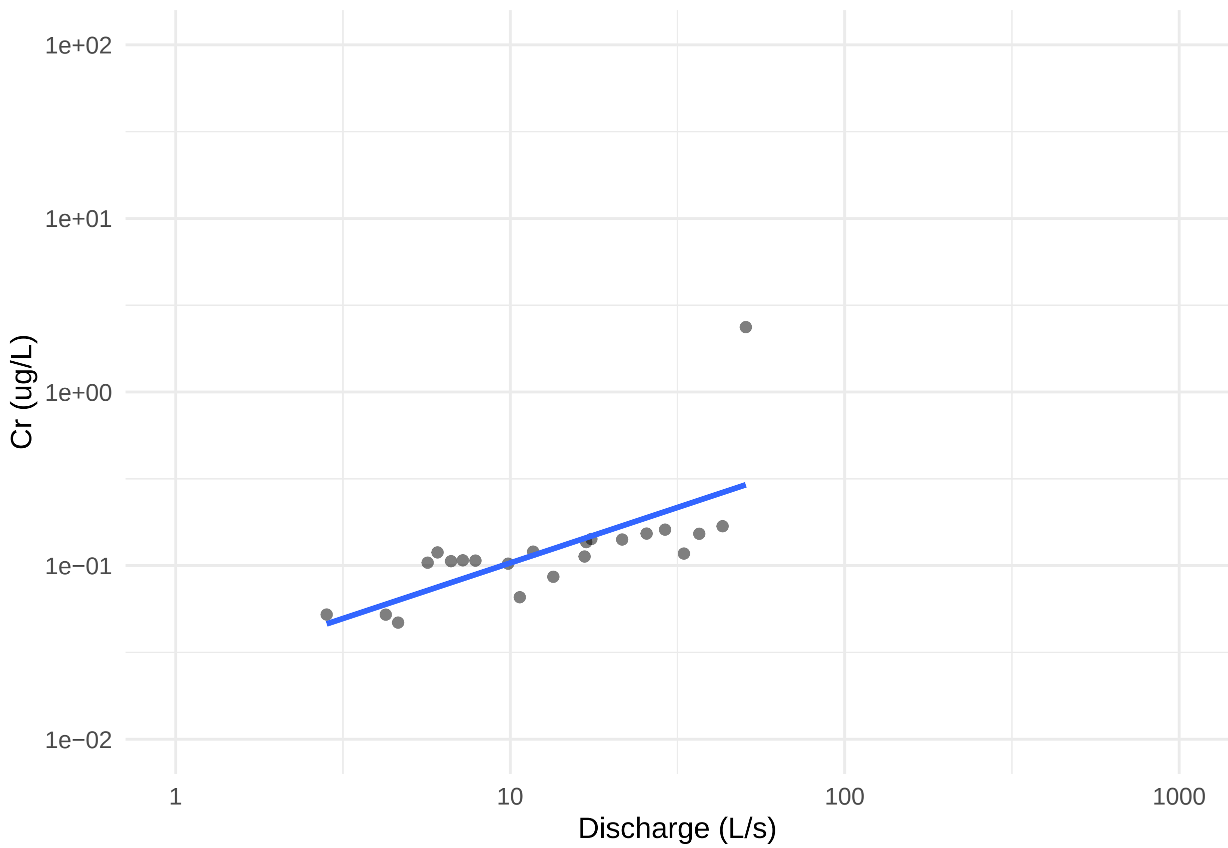
```r
# Filter the data for the specific storm and date
storm_filtered <- W2_Stormwater_2 %>%
    filter(Rain_Event_Number == "2022-07")

# Plot the filtered data
ggplot(storm_filtered, aes(x = Q, y = SO4_ppm)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(0.1, 100)) + ylab("SO4 (ppm)") +
    xlab("Discharge (L/s)") + theme_minimal()
```



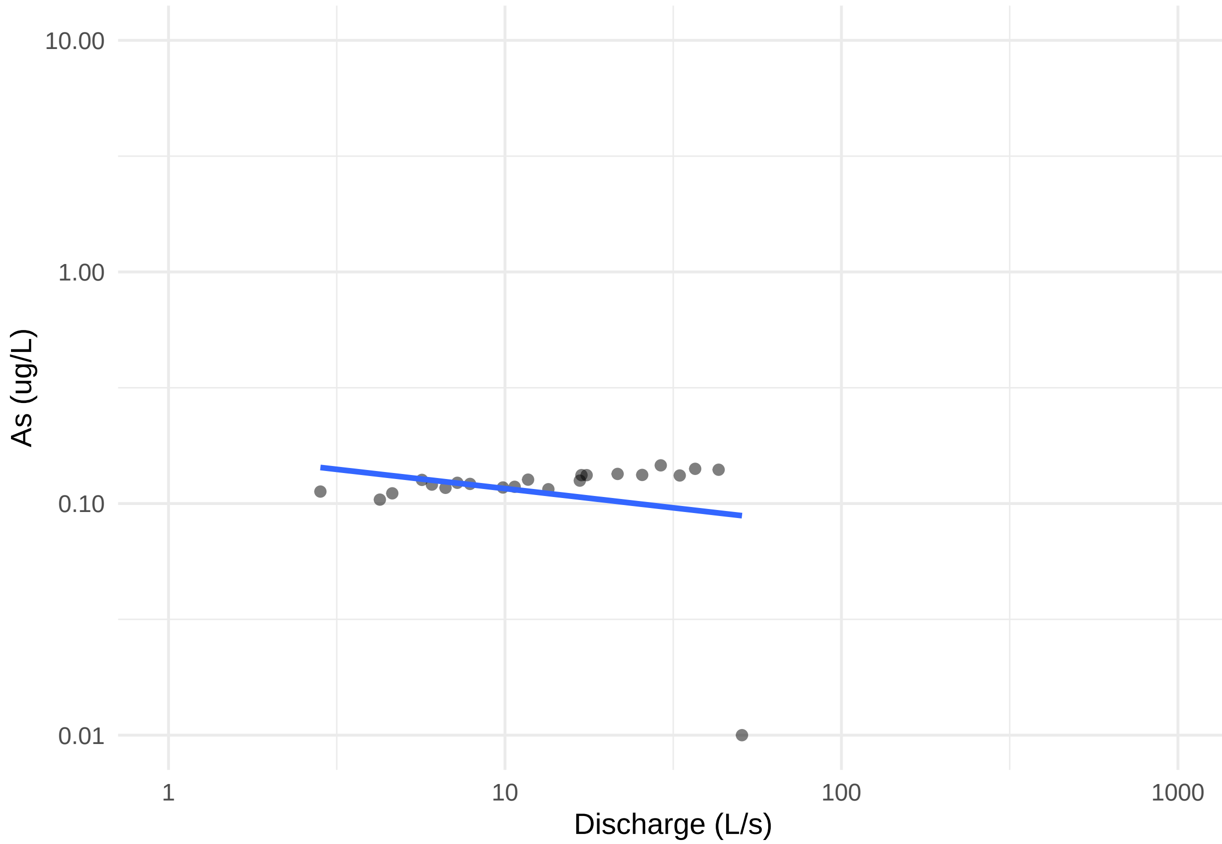

```r
ggplot(storm_filtered, aes(x = Q, y = Cl_ppm)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(0.1, 10)) + ylab("Cl (ppm)") +
    xlab("Discharge (L/s)") + theme_minimal()
```



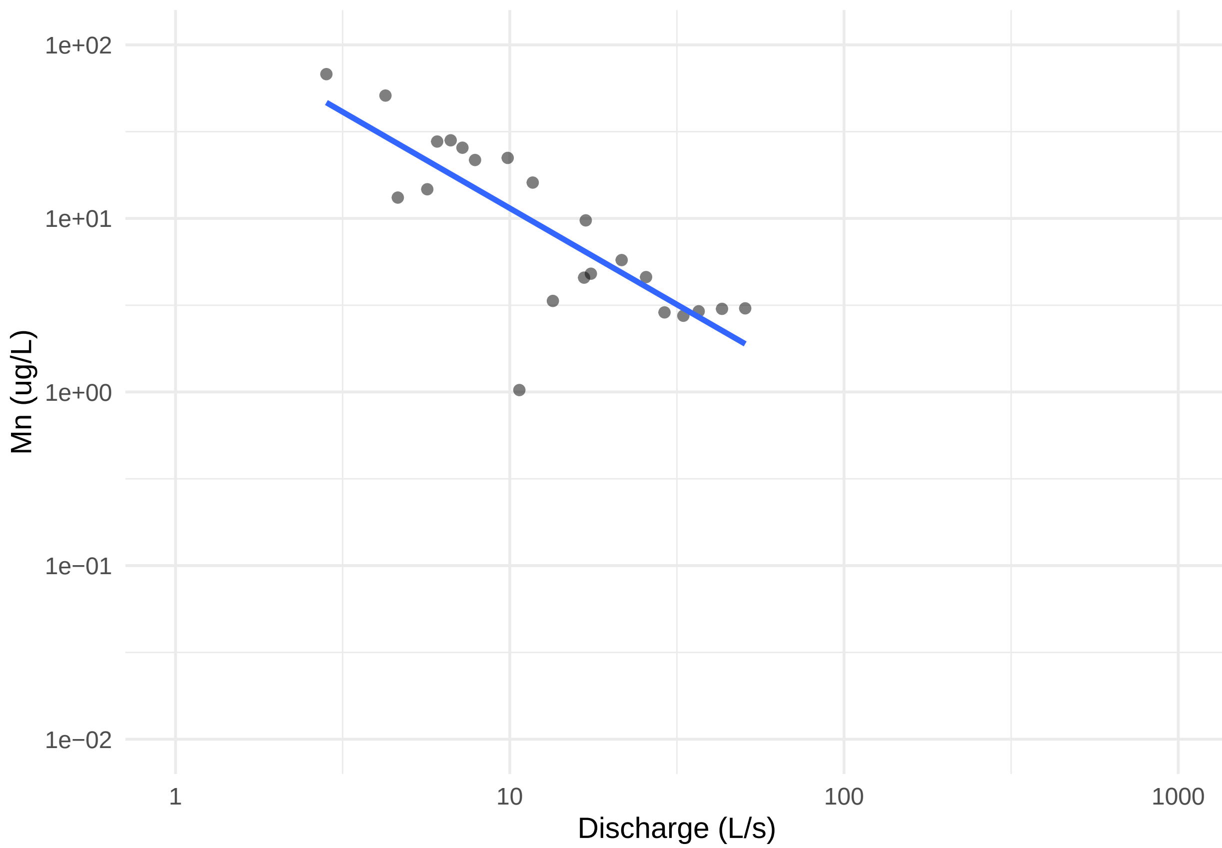

```
ggplot(storm_filtered, aes(x = Q, y = Cr_ugL)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(0.01, 100)) + ylab("Cr (ug/L)") +
    xlab("Discharge (L/s)") + theme_minimal()
```



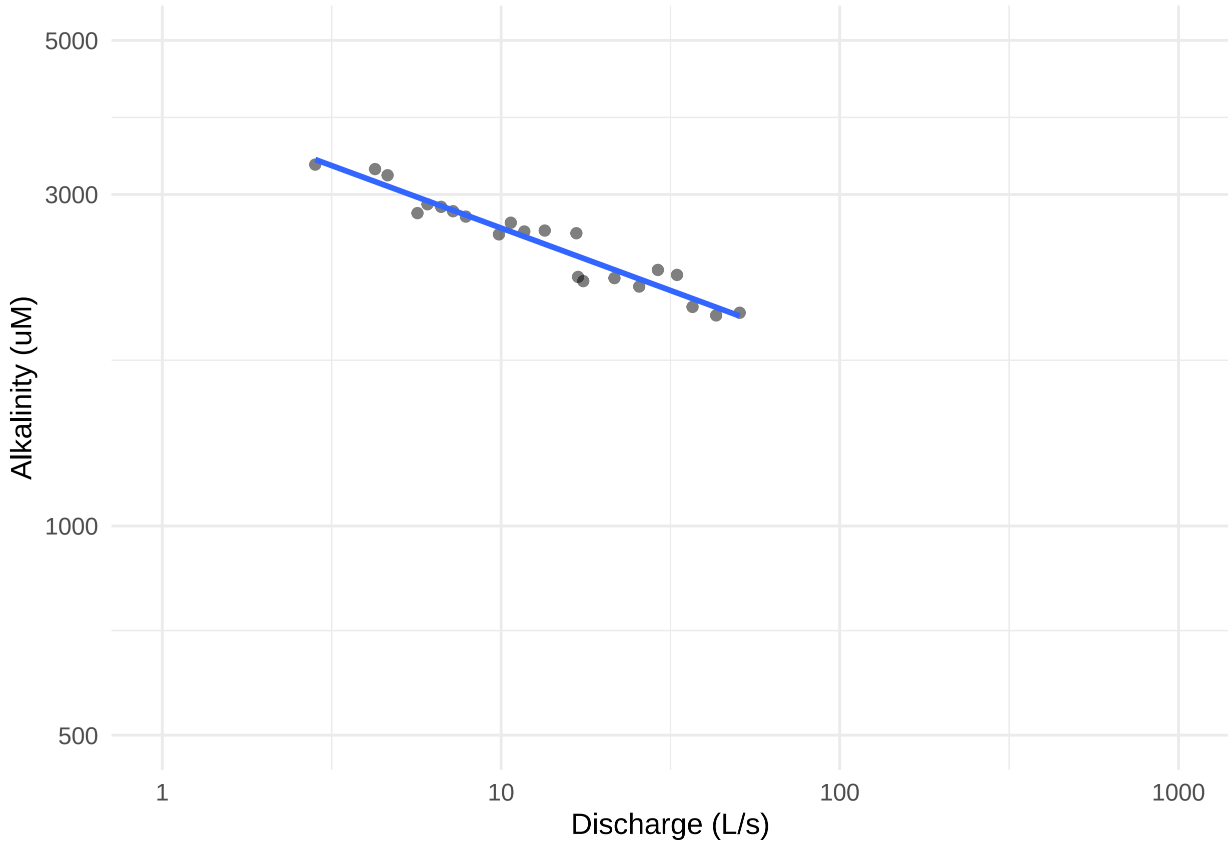

```
ggplot(storm_filtered, aes(x = Q, y = As_ugL)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(0.01, 10)) + ylab("As (ug/L)") +
    xlab("Discharge (L/s)") + theme_minimal()
```



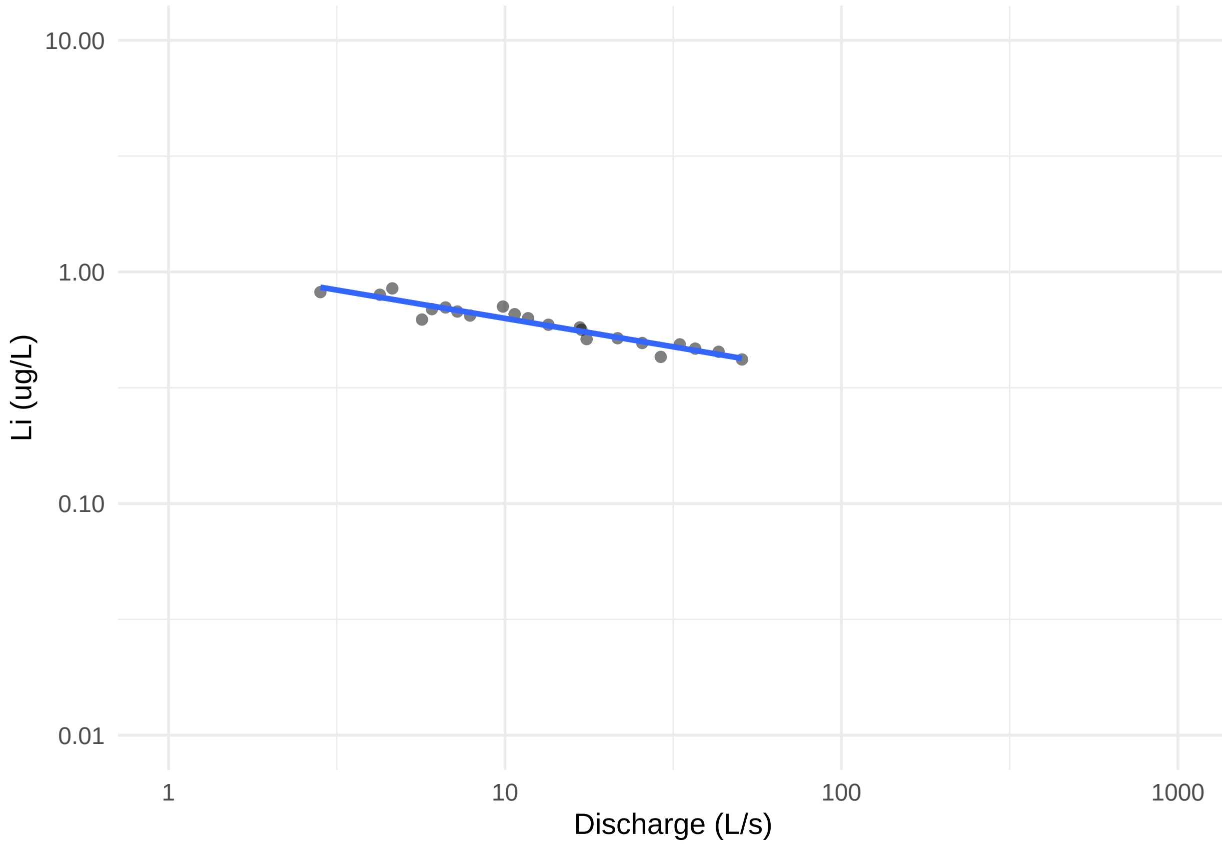

```r
ggplot(storm_filtered, aes(x = Q, y = Mn_ugL)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(0.01, 100)) + ylab("Mn (ug/L)") +
    xlab("Discharge (L/s)") + theme_minimal()
```



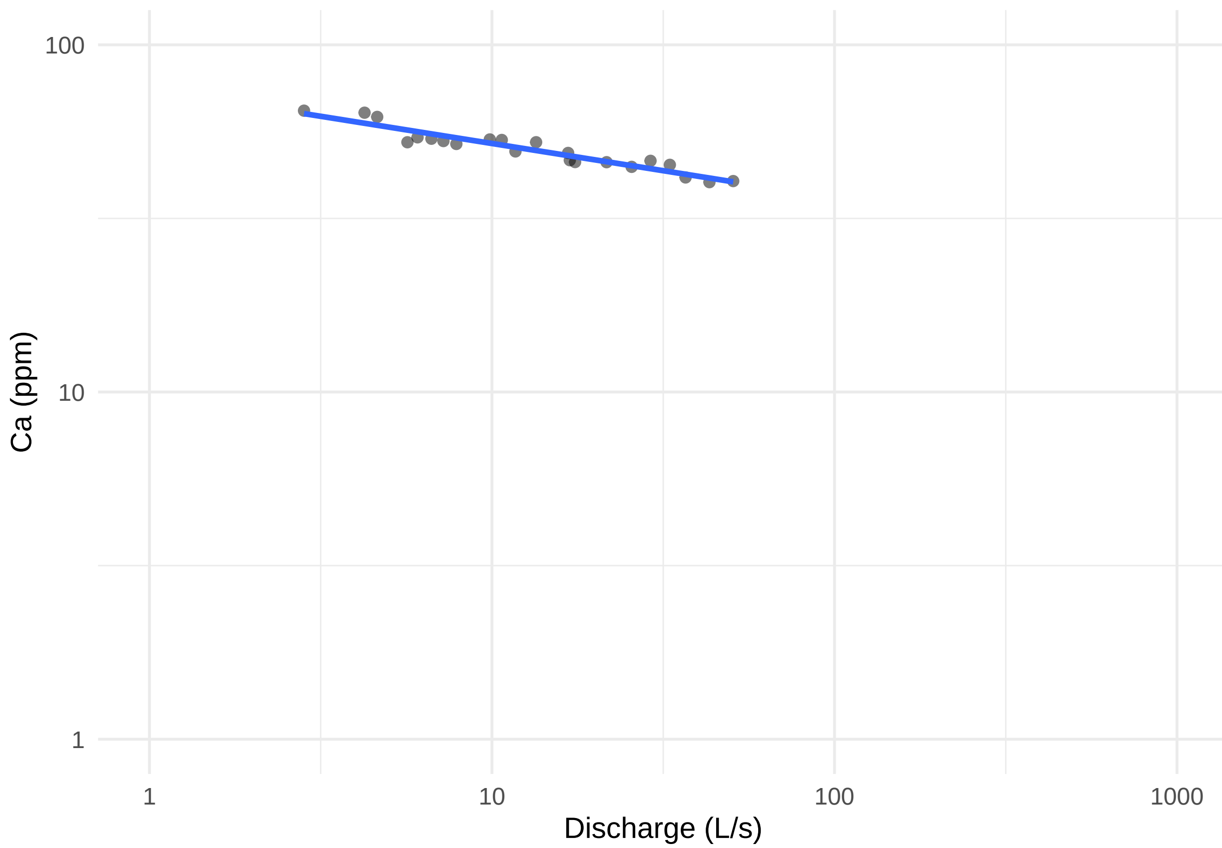

```r
ggplot(storm_filtered, aes(x = Q, y = Alkalinity_uM)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(500, 5000)) + ylab("Alkalinity (uM)") +
    xlab("Discharge (L/s)") + theme_minimal()
```




```
ggplot(storm_filtered, aes(x = Q, y = Li_ugL)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(0.01, 10)) + ylab("Li (ug/L)") +
    xlab("Discharge (L/s)") + theme_minimal()
```




```
ggplot(storm_filtered, aes(x = Q, y = Ca_ppm)) + geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", se = FALSE) + scale_x_log10(limits = c(1,
    1000)) + scale_y_log10(limits = c(1, 100)) + ylab("Ca (ppm)") +
    xlab("Discharge (L/s)") + theme_minimal()
```



Part VIII: Power Analysis

This is still in development - currently trial and erroring the code

```
# Step 1: Extract significant variables and compute statistics
# Based on `final_t_test_summary` which has already been computed
significant_stats <- final_t_test_summary %>%
  filter(P_Value < 0.05) %>%
  select(Variable, Mean_Difference, P_Value)

# Compute standard deviation of differences for each significant variable
significant_stats <- significant_stats %>%
  mutate(Std_Dev_Diff = sapply(Variable, function(var) {
    pre <- final_data[[var]][final_data$Treatment == "Pre"]
    post <- final_data[[var]][final_data$Treatment == "Post"]
    sd(pre - post, na.rm = TRUE)
  }))
print(significant_stats)
```

```
##                   Variable Mean_Difference    P_Value Std_Dev_Diff
## 1 intercept_Alkalinity_uM     -0.15018267 0.01314822    0.1762002
## 2          intercept_Si_ugL     -0.10226444 0.03240005    0.1447526
## 3          intercept_Sr_ugL     -0.09571092 0.03779729    0.1404735
```

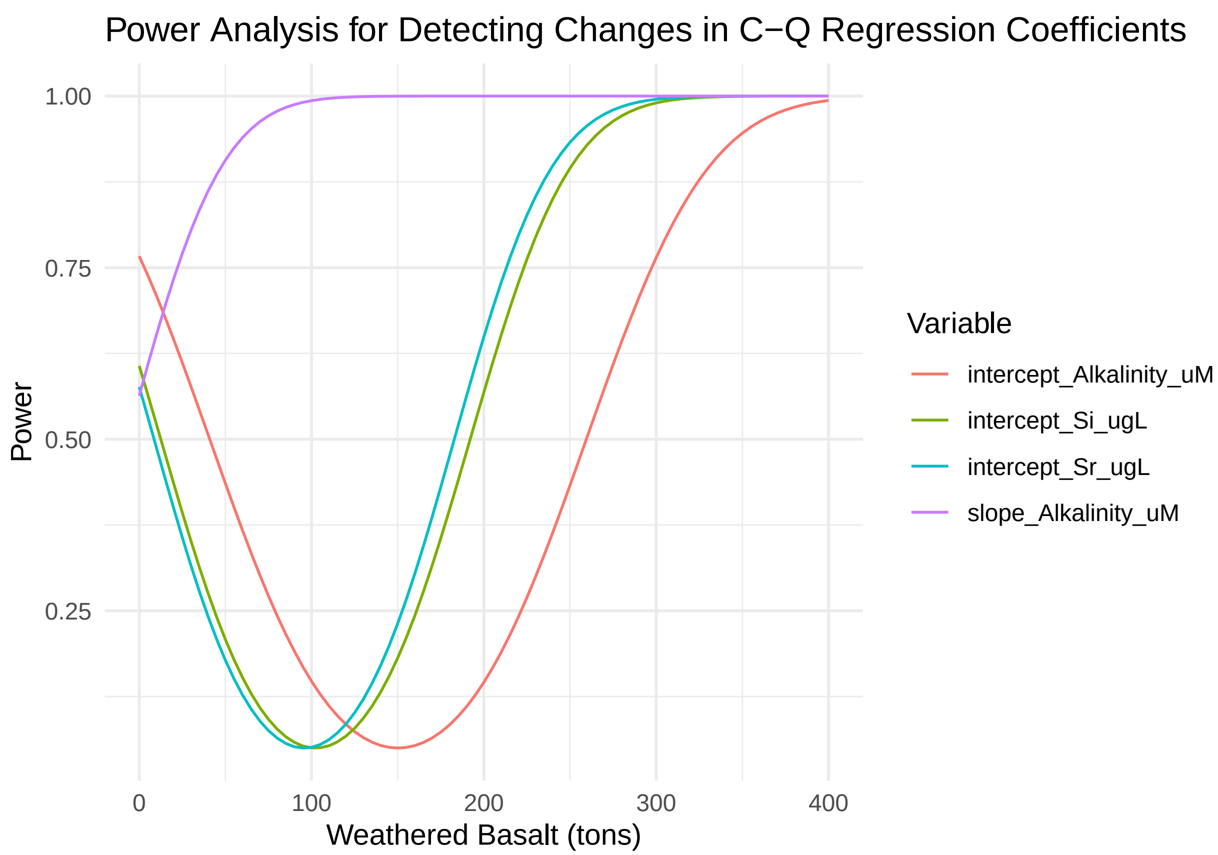
```
## 4     slope_Alkalinity_uM      0.09035598 0.04030749    0.1346945
```

```r
# Step 2: Perform power analysis for each significant variable
power_results <- significant_stats %>%
  rowwise() %>%
  mutate(Power = pwr.t.test(
    n = nrow(final_data) / 2,  # Average size of Pre and Post groups
    d = Mean_Difference / Std_Dev_Diff,  # Cohen's d
    sig.level = 0.05,  # Significance level
    power = NULL,  # Calculate power
    type = "paired"
  )$power)
print(power_results)
```

```
## # A tibble: 4 x 5
## # Rowwise:
##   Variable                Mean_Difference P_Value Std_Dev_Diff Power
##   <chr>                             <dbl>   <dbl>        <dbl> <dbl>
## 1 intercept_Alkalinity_uM         -0.150   0.0131        0.176 0.767
## 2 intercept_Si_ugL                -0.102   0.0324        0.145 0.607
## 3 intercept_Sr_ugL                -0.0957  0.0378        0.140 0.576
## 4 slope_Alkalinity_uM              0.0904  0.0403        0.135 0.563
```

```r
# Step 3: Simulate power for increasing basalt weathering
# Define the range of weathering amounts
weathered_amounts <- seq(0, 400, by = 5)  # Adjust as needed - based on our total amount applied

# Function to simulate power for each weathering amount
simulate_power <- function(amount, stats) {
  stats %>%
    rowwise() %>%
    mutate(
      Simulated_Effect = Mean_Difference + 0.001 * amount,  # Adjust scaling factor
      Power = pwr.t.test(
        n = nrow(final_data) / 2,
        d = Simulated_Effect / Std_Dev_Diff,  # Adjust effect size
        sig.level = 0.05,
        power = NULL,
        type = "paired"
      )$power
    )
}

# Simulate power for each weathering amount
simulated_power <- lapply(weathered_amounts, function(amount) {
  simulate_power(amount, significant_stats) %>% mutate(Weathered_Amount = amount)
})

# Combine results into a single dataframe
simulated_power_df <- do.call(rbind, simulated_power)

# Step 4: Visualize the results
ggplot(simulated_power_df, aes(x = Weathered_Amount, y = Power, color = Variable)) +
```

```r
  geom_line() +
  labs(
    title = "Power Analysis for Detecting Changes in C-Q Regression Coefficients",
    x = "Weathered Basalt (tons)",
    y = "Power",
    color = "Variable"
  ) +
  theme_minimal()
```




```r
# Step 5: Identify the basalt weathering required for desired power
desired_power <- 0.9
# Ensure the data frame is not rowwise
simulated_power_df <- as_tibble(simulated_power_df)

# Filter and summarize
required_weathering <- simulated_power_df %>%
  filter(!is.na(Power), Power >= desired_power) %>%  # Ensure no NA values and filter by desired power
  group_by(Variable) %>%
  summarise(Min_Weathering = min(Weathered_Amount, na.rm = TRUE), .groups = "drop")  # Drop grouping st

# View the results
print(required_weathering)
```

```
## # A tibble: 4 x 2
##   Variable                Min_Weathering
```

```
##   <chr>                              <dbl>
## 1 intercept_Alkalinity_uM              335
## 2 intercept_Si_ugL                     255
## 3 intercept_Sr_ugL                     245
## 4 slope_Alkalinity_uM                   50
```

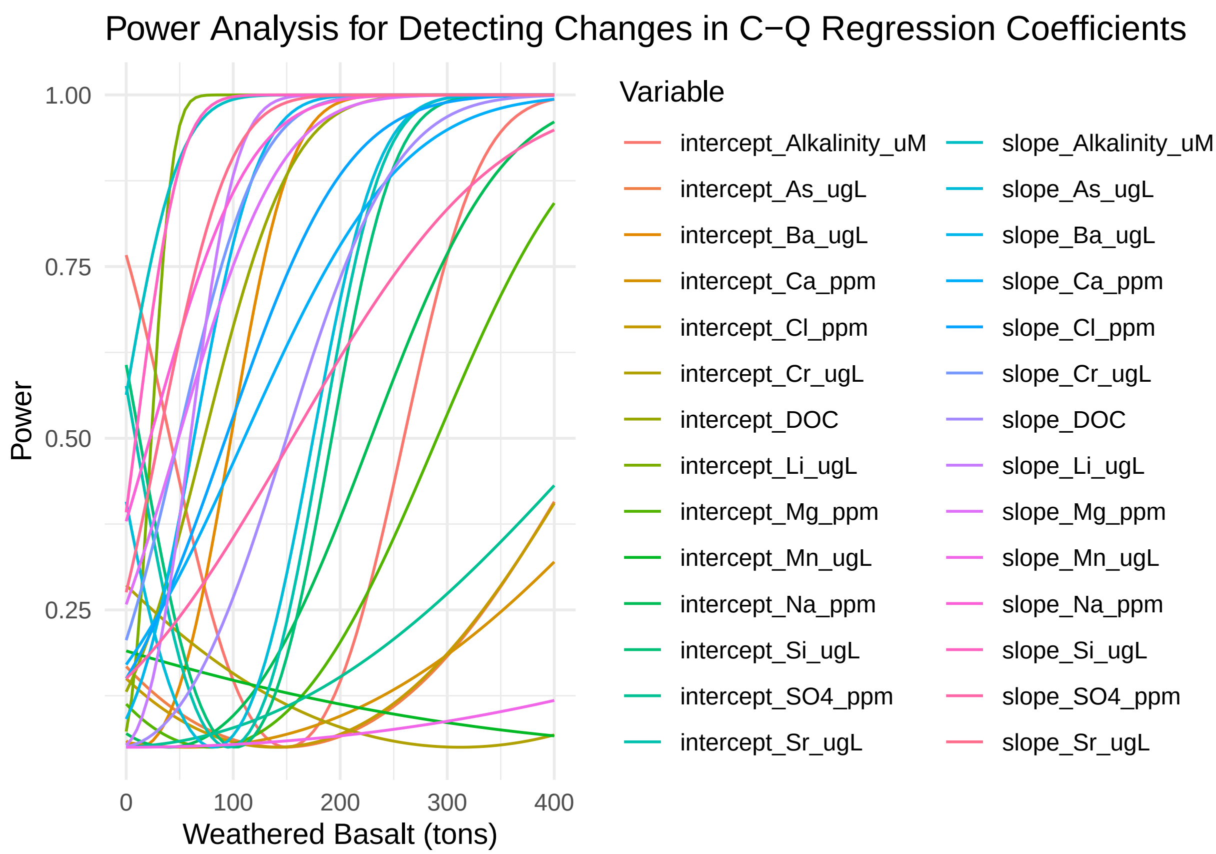
Now, we might want to see if there is effect size growth if we want to know how much of our treatment is needed to produce a detectable effect, which helps us explore new patterns and potentailly assess how sensitive our system is. Be clear - this is not a geochemical model. We may well require more basalt than is possible.

```r
# Include all variables, not just significant ones
all_stats <- final_t_test_summary %>%
    select(Variable, Mean_Difference, P_Value)

# Now calculate std dev of all variables
all_stats <- all_stats %>%
    mutate(Std_Dev_Diff = sapply(Variable, function(var) {
        pre <- final_data[[var]][final_data$Treatment == "Pre"]
        post <- final_data[[var]][final_data$Treatment == "Post"]
        sd(pre - post, na.rm = TRUE)
    }))

# Step 2: Perform power analysis for all variables
power_results <- all_stats %>%
    rowwise() %>%
    mutate(Power = pwr.t.test(
        n = nrow(final_data) / 2,  # Average size of Pre and Post groups
        d = Mean_Difference / Std_Dev_Diff,  # Cohen's d
        sig.level = 0.05,  # Significance level
        power = NULL,  # Calculate power
        type = "paired"
    )$power)
print(power_results)
```

```
## # A tibble: 28 x 5
## # Rowwise:
##    Variable                Mean_Difference P_Value Std_Dev_Diff  Power
##    <chr>                             <dbl>   <dbl>        <dbl>  <dbl>
##  1 intercept_Cl_ppm                -0.139   0.339         0.480 0.150
##  2 intercept_SO4_ppm                0.0140  0.948         0.733 0.0504
##  3 intercept_Na_ppm                -0.0398  0.659         0.305 0.0700
##  4 intercept_Mg_ppm                -0.0786  0.442         0.342 0.113
##  5 intercept_Ca_ppm                -0.0575  0.789         0.727 0.0572
##  6 intercept_Alkalinity_uM         -0.150   0.0131        0.176 0.767
##  7 intercept_DOC                    0.0527  0.387         0.202 0.131
##  8 intercept_As_ugL                -0.146   0.302         0.466 0.168
##  9 intercept_Ba_ugL                -0.0123  0.763         0.138 0.0592
## 10 intercept_Cr_ugL                -0.312   0.156         0.709 0.286
## # i 18 more rows
```

```r
# Step 3: Simulate power for increasing basalt weathering
# Define the range of weathering amounts
```

```r
weathered_amounts <- seq(0, 400, by = 5)  # Adjust as needed - based on our total amount applied

# Function to simulate power for each weathering amount
simulate_power <- function(amount, stats) {
    stats %>%
        rowwise() %>%
        mutate(
            Simulated_Effect = Mean_Difference + 0.001 * amount,  # Adjust scaling factor
            Power = pwr.t.test(
                n = nrow(final_data) / 2,
                d = Simulated_Effect / Std_Dev_Diff,  # Adjust effect size
                sig.level = 0.05,
                power = NULL,
                type = "paired"
            )$power
        )
}

# Simulate power for each weathering amount
simulated_power_nonsig <- lapply(weathered_amounts, function(amount) {
    simulate_power(amount, all_stats) %>% mutate(Weathered_Amount = amount)
})

# Combine results into a single dataframe
simulated_power_df_nonsig <- do.call(rbind, simulated_power_nonsig)

# Step 4: Visualize the results
ggplot(simulated_power_df_nonsig, aes(x = Weathered_Amount, y = Power, color = Variable)) +
    geom_line() +
    labs(
        title = "Power Analysis for Detecting Changes in C-Q Regression Coefficients",
        x = "Weathered Basalt (tons)",
        y = "Power",
        color = "Variable"
    ) +
    theme_minimal()
```




```r
# Step 5: Identify the basalt weathering required for desired power
desired_power <- 0.9

# Ensure the data frame is not rowwise
simulated_power_df_nonsig <- as_tibble(simulated_power_df_nonsig)

# Filter and summarize
required_weathering <- simulated_power_df_nonsig %>%
    filter(!is.na(Power), Power >= desired_power) %>%
    group_by(Variable) %>%
    summarise(Min_Weathering = min(Weathered_Amount, na.rm = TRUE), .groups = "drop")

# View the results
print(required_weathering)
```

```
## # A tibble: 20 x 2
##    Variable                Min_Weathering
##    <chr>                            <dbl>
##  1 intercept_Alkalinity_uM            335
##  2 intercept_Ba_ugL                   155
##  3 intercept_DOC                      160
##  4 intercept_Li_ugL                    45
##  5 intercept_Na_ppm                   355
##  6 intercept_Si_ugL                   255
##  7 intercept_Sr_ugL                   245
##  8 slope_Alkalinity_uM                 50
```

```
##  9 slope_As_ugL                    240
## 10 slope_Ba_ugL                    125
## 11 slope_Ca_ppm                    260
## 12 slope_Cl_ppm                    210
## 13 slope_Cr_ugL                    125
## 14 slope_DOC                       255
## 15 slope_Li_ugL                    105
## 16 slope_Mg_ppm                    150
## 17 slope_Na_ppm                    120
## 18 slope_SO4_ppm                   350
## 19 slope_Si_ugL                     55
## 20 slope_Sr_ugL                    100
```