Appendix A - ADS506-01-FA22 - Final Project

Team 1

12/05/2022

Page 11 is where I cut in to put parameters into columns using the owt_df02_gb07 dataframe

RMarkdown global setup

```
knitr::opts_chunk$set(fig.align = 'center')
```

```
library(AppliedPredictiveModeling)
library(BioStatR)
library(car)
library(caret)
library(class)
library(corrplot)
library(datasets)
library(e1071)
library(Hmisc)
library(mlbench)
library(gridExtra)
library(psych)
library(randomForest)
library(RANN)
library(reshape2)
library(rpart)
library(rpart.plot)
library(scales)
library(tidyverse)
library(tseries)
library(zoo)

set.seed(1699)
```

Create function to generate boxplots for continuous variables

```
# Define function to produce formatted boxplots
box_comp <- function(xcol = c(), df = NA, rtn_met = TRUE) {
  sig <- 3
  metrics_df01 <- data.frame(metric = c("",
                                        "Total N:",
                                        "Count",
                                        "NA Count",
                                        "Mean",
                                        "Median",
                                        "Standard Deviation",
                                        "Variance",
                                        "Range",
                                        "Min",
                                        "Max",
                                        "25th Percentile",
                                        "75th Percentile",
                                        "Subset w/o Outliers:",
                                        "Count",
                                        "%",
                                        "Outlier %",
                                        "NA Count",
                                        "Mean",
                                        "Median",
                                        "Standard Deviation",
                                        "Variance",
                                        "Range",
                                        "Min",
                                        "Max"
                                        ))
  for (var in xcol) {
    df_s1 <- df[, var]
    df_s1s1 <-data.frame(df_s1)
    df_s1_fit <- preProcess(df_s1s1,
                            method = c("center", "scale"))
    df_s1_trans <- predict(df_s1_fit, df_s1s1)


    # Calculate quartiles
    var_iqr_lim <- IQR(df_s1) * 1.5
    var_q1 <- quantile(df_s1, probs = c(.25))
    var_otlow <- var_q1 - var_iqr_lim
    var_q3 <- quantile(df_s1, probs = c(.75))
    var_othigh <- var_q3 + var_iqr_lim

    # Subset non-outlier data
    var_non_otlr_df01 <- subset(df, (abs(df_s1_trans) <= 3))
    #var_non_otlr_df01 <- subset(df, (df_s1 > var_otlow & df_s1 < var_othigh))
    df_s2 <- var_non_otlr_df01[, var]

    # Begin calculating measures of centrality & dispersion
    var_mean <- mean(df_s1)
    var_non_otlr_df01_trunc_mean <- mean(df_s2)
```

```
    var_med <- median(df_s1)
    var_non_otlr_df01_trunc_med <- median(df_s2)
    var_mode <- mode(df_s1)
    var_non_otlr_df01_trunc_mode <- mode(df_s2)
    var_stde <- sd(df_s1)
    var_non_otlr_df01_trunc_stde <- sd(df_s2)
    var_vari <- var(df_s1)
    var_non_otlr_df01_trunc_vari <- var(df_s2)
    var01_min <- min(df[, var])
    var01_max <- max(df[, var])
    var01_range <- var01_max - var01_min
    var02_min <- min(var_non_otlr_df01[, var])
    var02_max <- max(var_non_otlr_df01[, var])
    var02_range <- var02_max - var02_min

    # Configure y-axis min & max to sync graphs
    plot_min <- min(var01_min, var02_min)
    plot_max <- max(var01_max, var02_max)
    nonoutlier_perc <- round((as.numeric(dim(var_non_otlr_df01)[1] / as.numeric(dim(df)[1]))) *
100, 1)
    # Fill in metrics table
    measure_val01 <- c(paste0("Variable: ", var),
                       "",
                       as.character(dim(df)[1]),
                       sum(is.na(df_s1)),
                       round(var_mean, sig),
                       round(var_med, sig),
                       round(var_stde, sig),
                       round(var_vari, sig),
                       round(var01_range, sig),
                       round(var01_min, sig),
                       round(var01_max, sig),
                       round(var_q1, sig),
                       round(var_q3, sig),
                       "",
                       as.character(dim(var_non_otlr_df01)[1]),
                       paste0(nonoutlier_perc, "%"),
                       paste0(round(100 - nonoutlier_perc, 1), "%"),
                       sum(is.na(df_s2)),
                       round(var_non_otlr_df01_trunc_mean, sig),
                       round(var_non_otlr_df01_trunc_med, sig),
                       round(var_non_otlr_df01_trunc_stde, sig),
                       round(var_non_otlr_df01_trunc_vari, sig),
                       round(var02_range, sig),
                       round(var02_min, sig),
                       round(var02_max, sig)
                       )

    var_name <- paste0("Variable: ", var)
    metrics_df01[, ncol(metrics_df01) + 1] <- measure_val01
  }
  # Format boxplot titles based on number of plots
```

```
  if(length(xcol == 1)) {
    boxplot(df,
            ylab = "Parameter Values",
            main = paste0("Boxplot for ", xcol, " (", grph_title, ")")) }
  else {
    boxplot(df,
            ylab="ParameterValues",
            main=paste0("BoxplotforMultipleParameters","(",grph_title,")"))
  }
  #Returnandprintmetricstable(s)
  if(rtn_met==TRUE){
    print(metrics_df01)
    return(metrics_df01)
    }
  }
```

Importing Train/Test Datasets

```
# Import 4 separate CSV files
owt_df01a <- read.csv("water_quality_1990_1999_datasd.csv",
                      header = TRUE, sep = ",")
owt_df01b <- read.csv("water_quality_2000_2010_datasd.csv",
                      header = TRUE, sep = ",")
owt_df01c <- read.csv("water_quality_2011_2019_datasd.csv",
                      header = TRUE, sep = ",")
owt_df01d <- read.csv("water_quality_2020_2021_datasd.csv",
                      header = TRUE, sep = ",")

# Merge 4 seperate dataframes into 1
owt_df01 <- rbind(owt_df01a, owt_df01b, owt_df01c, owt_df01d)

print(head(owt_df01))
```

```
##       sample station depth_m date_sample time project   parameter qualifier
## 1 9011158743      C5       9  1990-11-15         PLOO CHLOROPHYLL
## 2 9011158743      C5       9  1990-11-15         PLOO     DENSITY
## 3 9011158743      C5       9  1990-11-15         PLOO          DO
## 4 9011158743      C5       9  1990-11-15         PLOO          PH
## 5 9011158743      C5       9  1990-11-15         PLOO    SALINITY
## 6 9011158743      C5       9  1990-11-15         PLOO        TEMP
##    value   units
## 1  0.870    ug/L
## 2 23.855 sigma-t
## 3  6.550    mg/L
## 4  8.080      pH
## 5 33.617     ppt
## 6 19.430       C
```

```
#describe(owt_df01)

#write.csv(owt_df01, "../data/Ocean Water/ocean_df01.csv")
```

FActorize and format column types; print NA counts

```
#Listofparametervalues
param_lst01<-c("CHLOROPHYLL", "DENSITY", "DO", "ENTERO", "FECAL",
               "OG", "PH", "SALINITY", "SUSO", "TEMP", "TOTAL", "XMS")

#Listofcolnames
col_lst01<-c("sample", "station", "date_sample", "time", "project",
             "parameter", "qualifier", "units")

#Citation:https://www.geeksforgeeks.org/find-columns-and-rows-with-na-in-r-dataframe/
owt_df02<-owt_df01

for(c in col_lst01){
  owt_df02[,c]<-as.factor(owt_df02[,c])
}

#Generate NA summary tables
#Citation:https://www.geeksforgeeks.org/replace-character-value-with-na-in-r/
owt_df02[owt_df02==""]<-NA
print(head(owt_df02))
```

```
##       sample station depth_m date_sample time project   parameter qualifier
## 1 9011158743      C5       9  1990-11-15 <NA>    PLOO CHLOROPHYLL      <NA>
## 2 9011158743      C5       9  1990-11-15 <NA>    PLOO     DENSITY      <NA>
## 3 9011158743      C5       9  1990-11-15 <NA>    PLOO          DO      <NA>
## 4 9011158743      C5       9  1990-11-15 <NA>    PLOO          PH      <NA>
## 5 9011158743      C5       9  1990-11-15 <NA>    PLOO    SALINITY      <NA>
## 6 9011158743      C5       9  1990-11-15 <NA>    PLOO        TEMP      <NA>
##    value   units
## 1  0.870    ug/L
## 2 23.855 sigma-t
## 3  6.550    mg/L
## 4  8.080      pH
## 5 33.617     ppt
## 6 19.430       C
```

```
owt_df02_na<-sapply(owt_df02,function(x)sum(is.na(x)))

owt_df02_notna<-sapply(owt_df02,function(x)sum(!is.na(x)))

owt_df02_tbl01<-rbind(owt_df02_notna, owt_df02_na)

owt_df02_tbl02<-rbind(owt_df02_tbl01,
                      round(prop.table(owt_df02_tbl01, margin=2), 4))

print("Allparameters")
```

```
## [1] "Allparameters"
```

```
print(owt_df02_tbl02)
```

```
##                  sample station     depth_m date_sample       time project
## owt_df02_notna 1236769 1236769 1152608.000     1236769 1075929.00 1236769
## owt_df02_na          0       0   84161.000           0  160840.00       0
## owt_df02_notna       1       1       0.932           1       0.87       1
## owt_df02_na          0       0       0.068           0       0.13       0
##                parameter   qualifier        value   units
## owt_df02_notna   1236769 394867.0000 1231466.0000 1236769
## owt_df02_na            0 841902.0000    5303.0000       0
## owt_df02_notna         1      0.3193       0.9957       1
## owt_df02_na            0      0.6807       0.0043       0
```

```
owt_df02a<-owt_df02[which(is.na(owt_df02),arr.ind=TRUE),]
#print(head(owt_df02a))

for(p in param_lst01){
  df=owt_df02[owt_df02$parameter==p,]
  #print(head(df[which(is.na(df$value),arr.ind=TRUE),]))
  df_na<-sapply(df,function(x)sum(is.na(x)))
  df_notna<-sapply(df,function(x)sum(!is.na(x)))
  df_tbl01<-rbind(df_notna,df_na)
  df_tbl02<-rbind(df_tbl01,round(prop.table(df_tbl01,margin=2),4))
  rownames(df_tbl02)<-c("NotNAn","NAn","NotNA%","NA%")
  #print(p)
  #print(df_tbl02)
  #Thistablew/nullcountsandproportionsforeachfeaturenot displayedforspacepurposes→
}
```

Bin depth_m variable

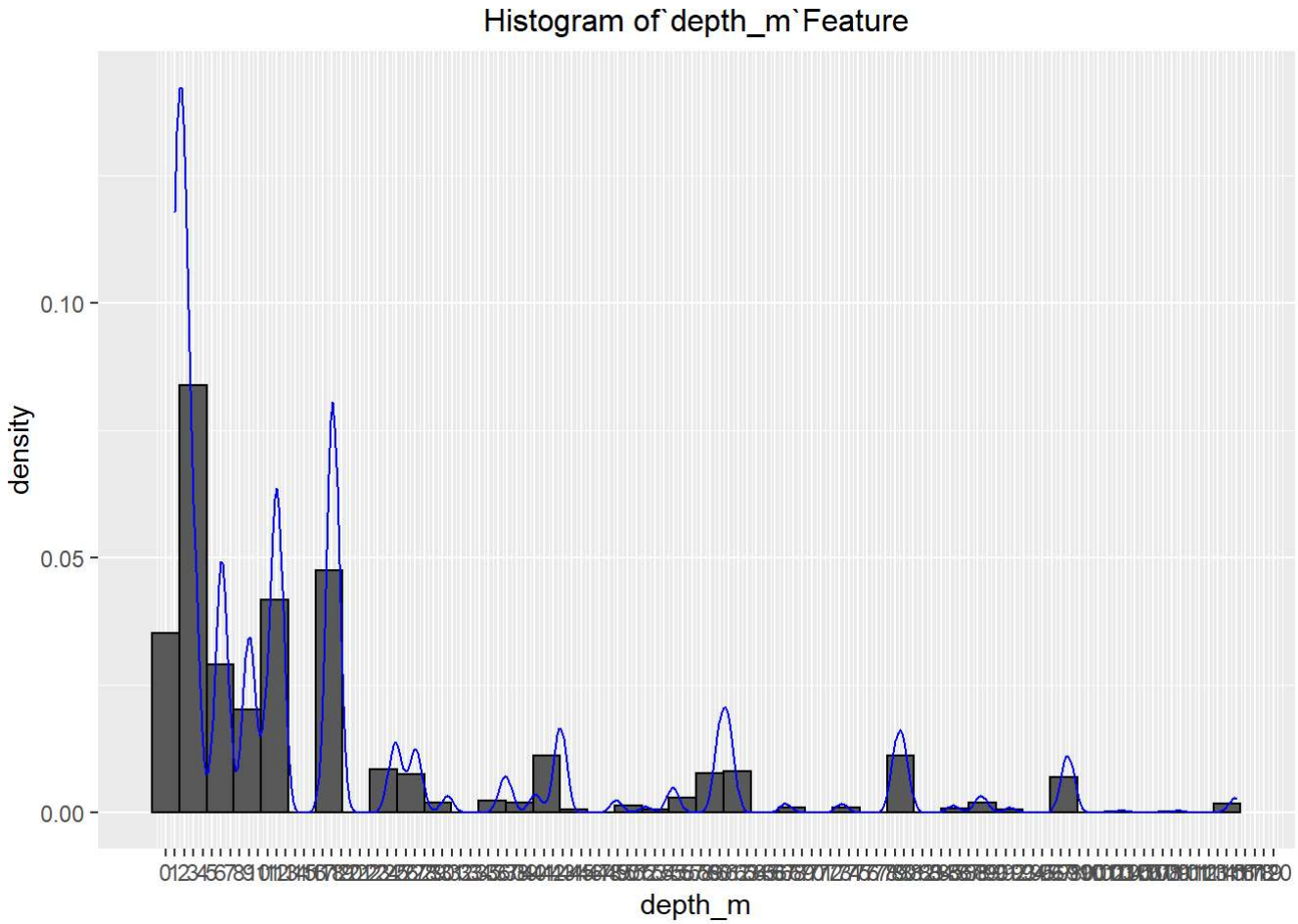
```
owt_df02$date_sample<-as.Date(owt_df02$date_sample,"%Y-%m-%d")
#Createbinsfordepth_mvalues
depth_lvls01<-c("[0,8)", "[8,33)", "[33,47)", "[47,70)", "[70,90)",
                "[90,112)", "[112,120]", "Unknown")

#Plotdistributionofdepth_mvalues #Citation: https://community.rstudio.com/t/ggplot-x-axis-y-axis
-ticks-labels-breaks-and-limits/119123/2→

ggplot(owt_df02,aes(x=depth_m))+
  geom_histogram(color="black", bins=40,aes(y=stat(density)))+
  geom_density(col="blue")+ labs(title="Histogram of`depth_m`Feature")+
  scale_x_continuous(breaks=seq(0,120,1))+
  theme(plot.title=element_text(hjust=0.5,size=12))
```

```
## Warning: Removed 84161 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 84161 rows containing non-finite values (stat_density).
```

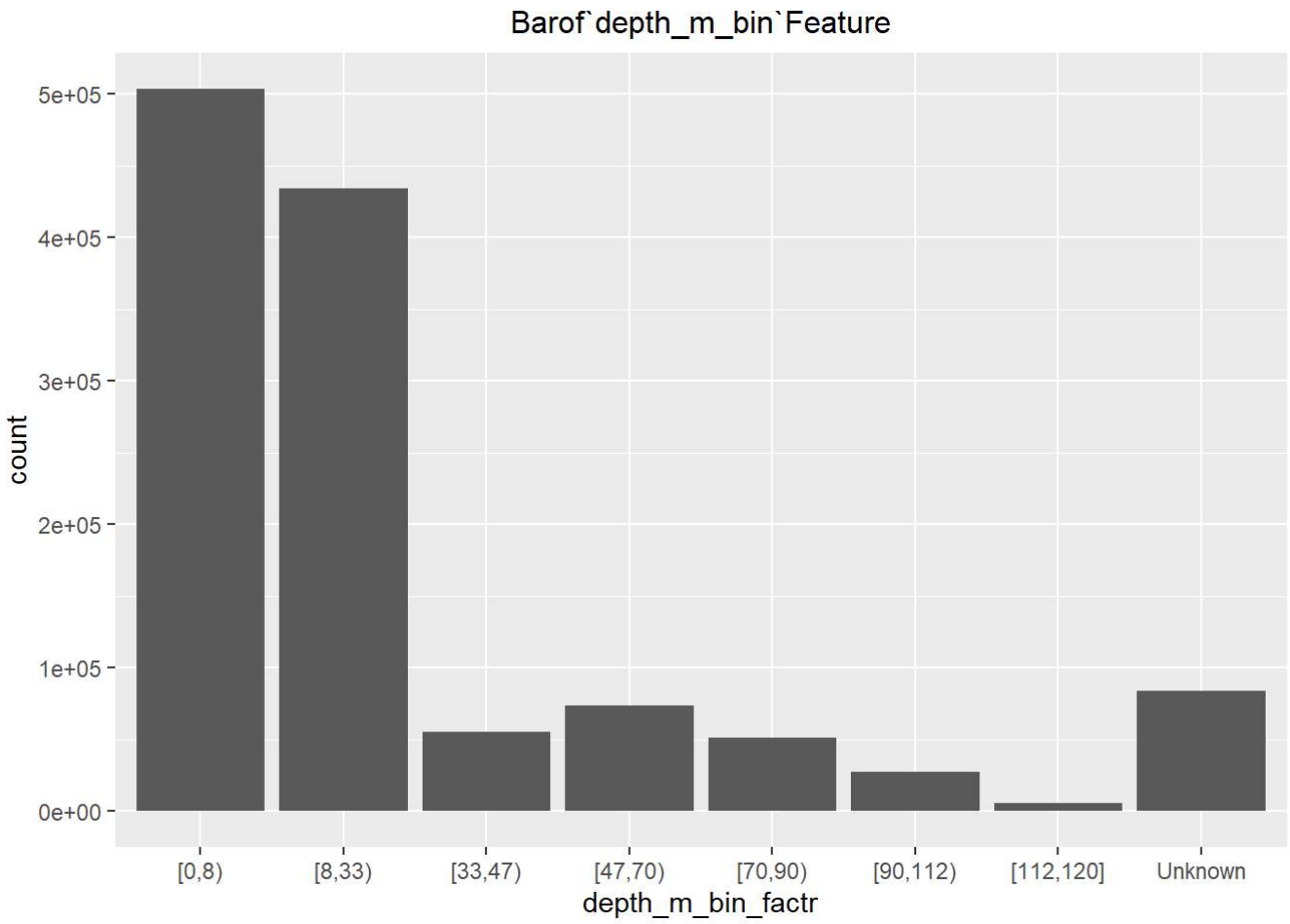


```
#Createnewcolumnwithbins
#Citation: https://www.marsja.se/r-add-column-to-dataframe-based-on-other-columns-conditions-dplyr/→
owt_df02<-mutate(owt_df02,depth_m_bin=case_when(depth_m<8~"[0,8)",
                                                depth_m<33~"[8,33)",
                                                depth_m<47~"[33,47)",
                                                depth_m<70~"[47,70)",
                                                depth_m<90~"[70,90)",
                                                depth_m<112~"[90,112)",
                                                depth_m>=112~"[112,120]"))
#ReplaceNAswith"Unknown"
#Citation:https://statisticsglobe.com/r-replace-na-with-0/
owt_df02$depth_m_bin<-replace_na(owt_df02$depth_m_bin,"Unknown")

#Citation:https://www.statology.org/order-bars-ggplot2-bar-chart/
owt_df02$depth_m_bin_factr=factor(owt_df02$depth_m_bin,levels=depth_lvls01)
print(head(owt_df02))
```

```
##       sample station depth_m date_sample time project   parameter qualifier
## 1 9011158743      C5       9  1990-11-15 <NA>    PLOO CHLOROPHYLL      <NA>
## 2 9011158743      C5       9  1990-11-15 <NA>    PLOO     DENSITY      <NA>
## 3 9011158743      C5       9  1990-11-15 <NA>    PLOO          DO      <NA>
## 4 9011158743      C5       9  1990-11-15 <NA>    PLOO          PH      <NA>
## 5 9011158743      C5       9  1990-11-15 <NA>    PLOO    SALINITY      <NA>
## 6 9011158743      C5       9  1990-11-15 <NA>    PLOO        TEMP      <NA>
##    value   units depth_m_bin depth_m_bin_factr
## 1  0.870    ug/L      [8,33)            [8,33)
## 2 23.855 sigma-t      [8,33)            [8,33)
## 3  6.550    mg/L      [8,33)            [8,33)
## 4  8.080      pH      [8,33)            [8,33)
## 5 33.617     ppt      [8,33)            [8,33)
## 6 19.430       C      [8,33)            [8,33)
```

```
#Display transformed barchart
ggplot(owt_df02,aes(x=depth_m_bin_factr))+
  geom_bar()+ labs(title="Barof`depth_m_bin`Feature")+
  theme(plot.title=element_text(hjust=0.5,size=12))
```



Perform several aggregations on the data for performing EDA at multiple levels

```
#Display aggregations by different features:
  # Station grouping = owt_df02_gb01
owt_df02_gb01<-owt_df02%>%
  group_by(station)%>%
  summarise(Count=n())
print(owt_df02_gb01[owt_df02_gb01$Count==min(owt_df02_gb01$Count),])
```

```
## # A tibble: 2 × 2
##   station Count
##   <fct>   <int>
## 1 A15        30
## 2 A16        30
```

```
print(owt_df02_gb01[owt_df02_gb01$Count==max(owt_df02_gb01$Count),])
```

```
## # A tibble: 1 × 2
##   station Count
##   <fct>   <int>
## 1 A1      55217
```

```
# Project grouping: owt_df02_gb02
owt_df02_gb02<-owt_df02 %>%
  group_by(project) %>%
  summarise(Count=n())

# Date_sample grouping: owt_df02_gb03
owt_df02_gb03<-owt_df02%>%
  group_by(date_sample)%>%
  summarise(Count=n())

#MainDF1 (owt_df02_gb04 Date and Paramter grouping)
owt_df02_gb04<-owt_df02%>%
  group_by(date_sample,parameter)%>%
  summarise(Avg=mean(value))
```

```
## `summarise()` has grouped output by 'date_sample'. You can override using the
## `.groups` argument.
```

```
# Paramter count group = owt_df02_gb05
owt_df02_gb05<-owt_df02 %>%
  group_by(parameter)%>%
  summarise(Count=n())

# depth_m grouping owt_df02_gb06
owt_df02_gb06<-owt_df02%>%
  group_by(depth_m)%>%
  summarise(Count=n())

#MainDF3:
  # owt_df02_gb07 = date_sample, project, depth bin, parameter groupings by avg
owt_df02_gb07<-owt_df02%>%
  group_by(date_sample, project, depth_m_bin, parameter)%>%
  summarise(Avg=mean(value))
```

```
## `summarise()` has grouped output by 'date_sample', 'project', 'depth_m_bin'.
## You can override using the `.groups` argument.
```

```
# owt_df02_gb08 depth bin groupings vy count
owt_df02_gb08<-owt_df02%>%
  group_by(depth_m_bin)%>%
  summarise(Count=n())

#MainDF2
  # owt_df02_gb09= date, project, paramter groupings vy average
owt_df02_gb09<-owt_df02%>%
  group_by(date_sample, project, parameter)%>%
  summarise(Avg=mean(value))
```

```
## `summarise()` has grouped output by 'date_sample', 'project'. You can override
## using the `.groups` argument.
```

putting parameters into own columns via mutate:

```
# putting paramters into columns via mutate:
  # citation: https://stackoverflow.com/questions/55516010/make-one-column-into-multiple-columns
-in-r
split_try <- owt_df02_gb07 %>%
  mutate(rn=row_number()) %>%
  spread(parameter, Avg) %>%
  select(-rn)

head(split_try, 5)
```

date_sample <date>	project <fct>	depth_m_bin <chr>	CHLOROPH… <dbl>	DENSI… <dbl>	DO <dbl>	ENT… <dbl>	FE… <dbl>	… <dbl>	PH <dbl>
1990-11-15	PLOO	[8,33)	0.87	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
1990-11-15	PLOO	[8,33)	*NA*	23.855	*NA*	*NA*	*NA*	*NA*	*NA*
1990-11-15	PLOO	[8,33)	*NA*	*NA*	6.55	*NA*	*NA*	*NA*	*NA*
1990-11-15	PLOO	[8,33)	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	8.08
1990-11-15	PLOO	[8,33)	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*

5 rows | 1-10 of 15 columns

```
# dropping the TOTAL column as that was vague parameter
split_try <- data.frame(split_try)
splitted <- subset(split_try, select = -c(TOTAL))
#head(splitted, 3)
```

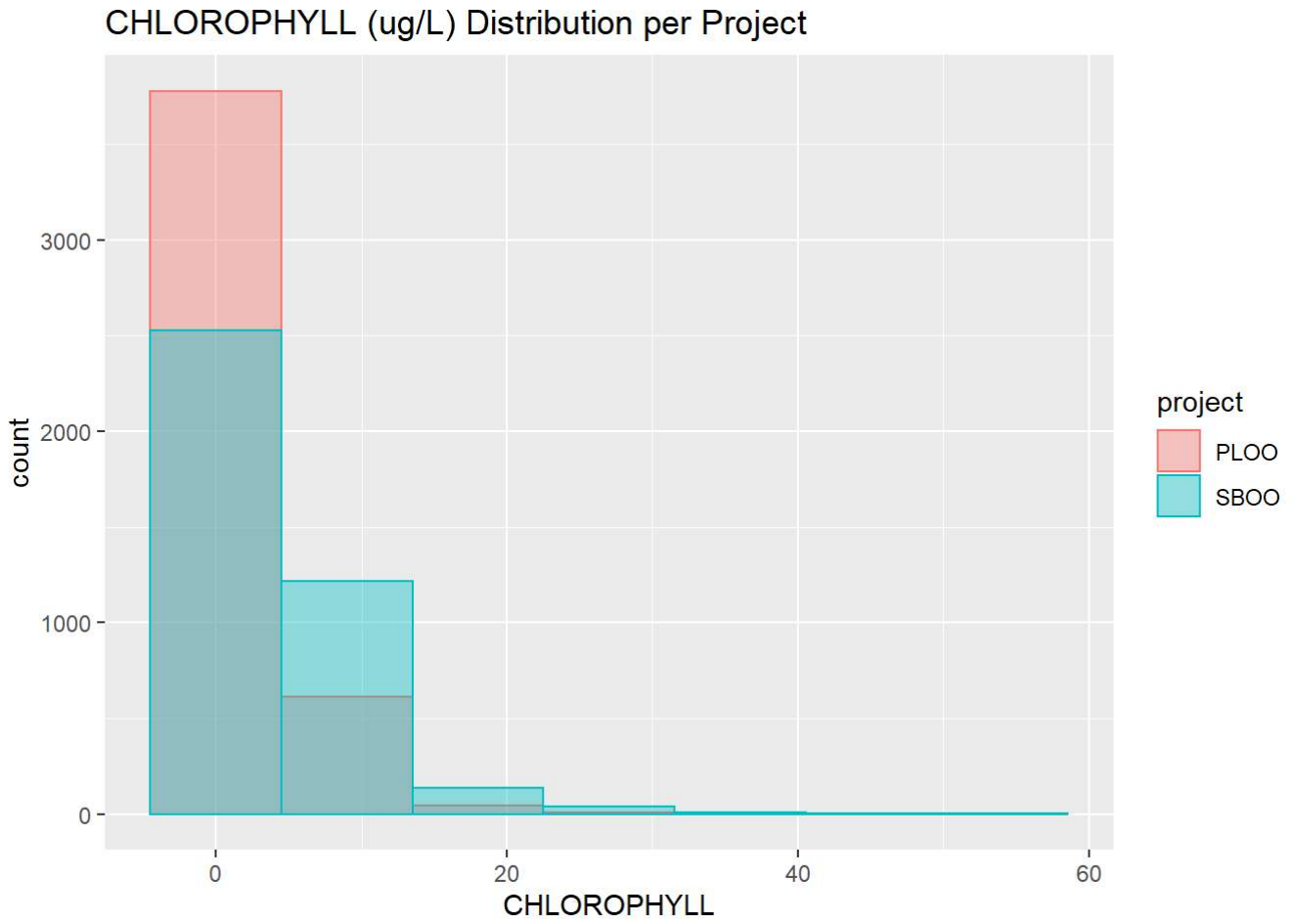
```
sapply(splitted, function(x) sum(is.na(x)))
```

```
## date_sample     project depth_m_bin CHLOROPHYLL     DENSITY          DO
##           0           0           0      112900      112659      111444
##      ENTERO       FECAL          OG          PH    SALINITY        SUSO
##      105753      106412      120171      111591      111442      118263
##        TEMP         XMS
##      109276      109317
```

PLotting histograms to see skewness/distribution

```
# chlorophyll
ggplot(splitted, aes(x= CHLOROPHYLL)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "CHLOROPHYLL (ug/L) Distribution per Project")
```

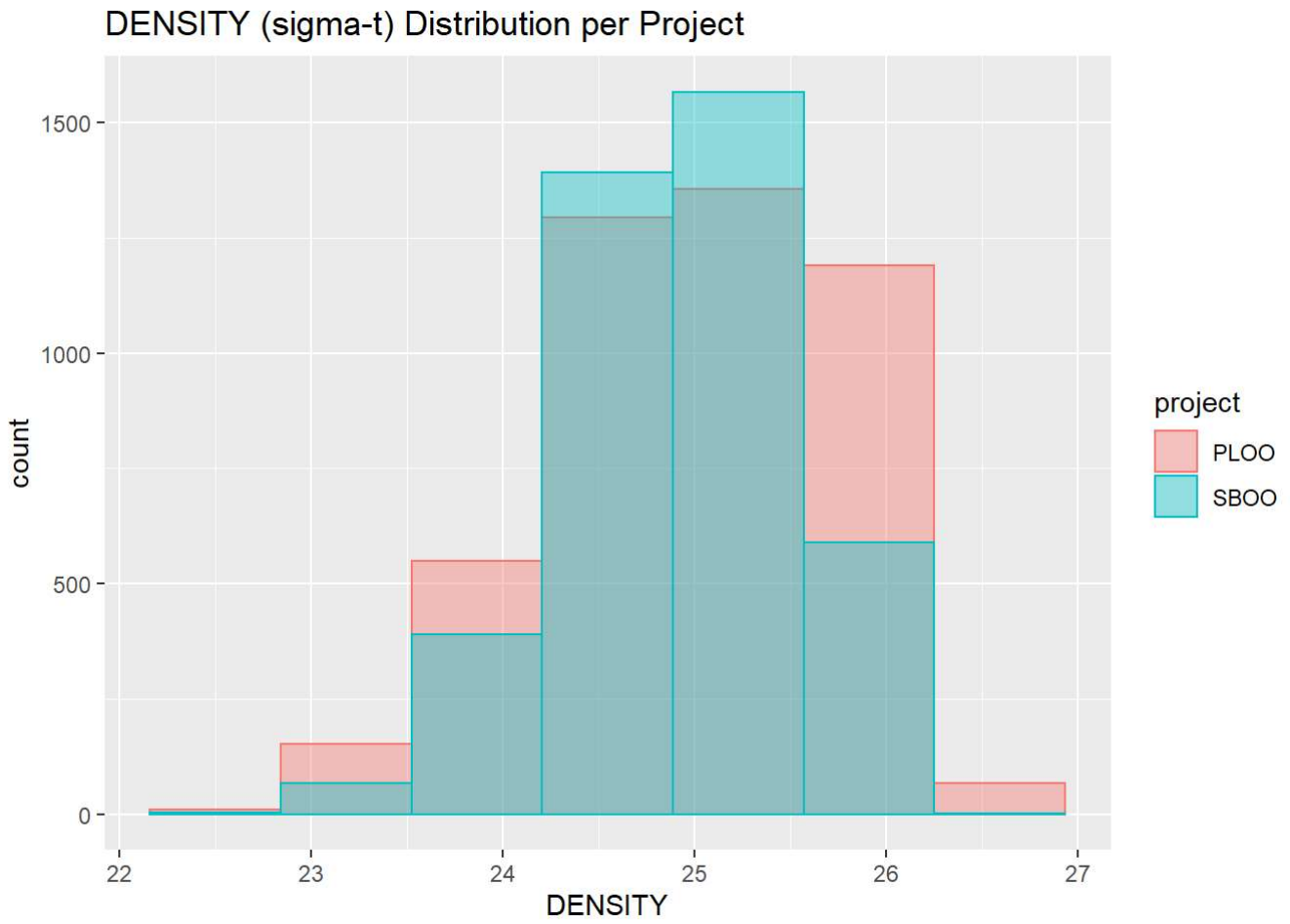
```
## Warning: Removed 112900 rows containing non-finite values (stat_bin).
```



```
  # skewed with most obs towards left (low values)

# density
ggplot(splitted, aes(x= DENSITY)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "DENSITY (sigma-t) Distribution per Project")
```

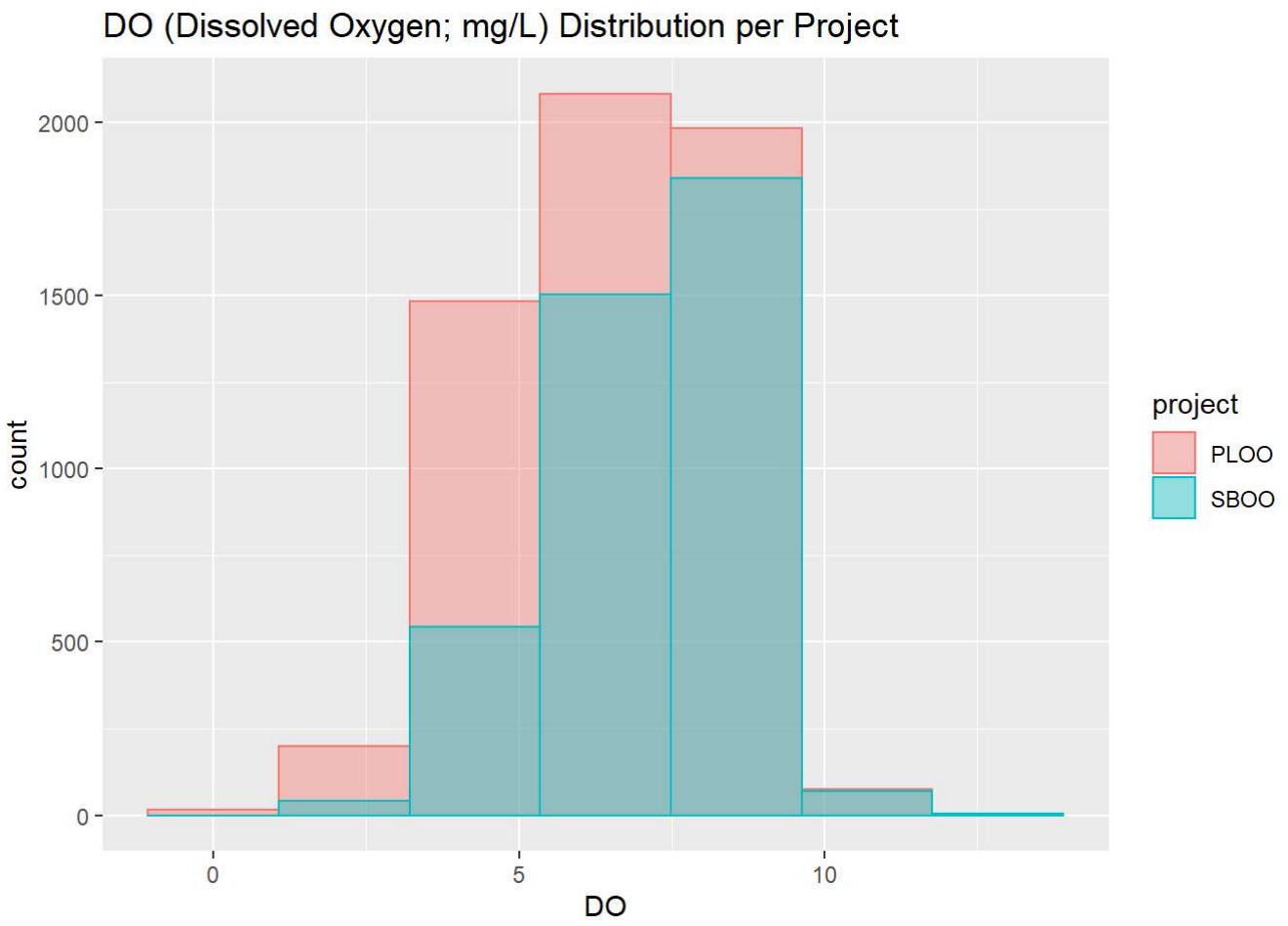
```
## Warning: Removed 112659 rows containing non-finite values (stat_bin).
```




```
  # relatively normal distributed

# DO
ggplot(splitted, aes(x= DO)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "DO (Dissolved Oxygen; mg/L) Distribution per Project")
```

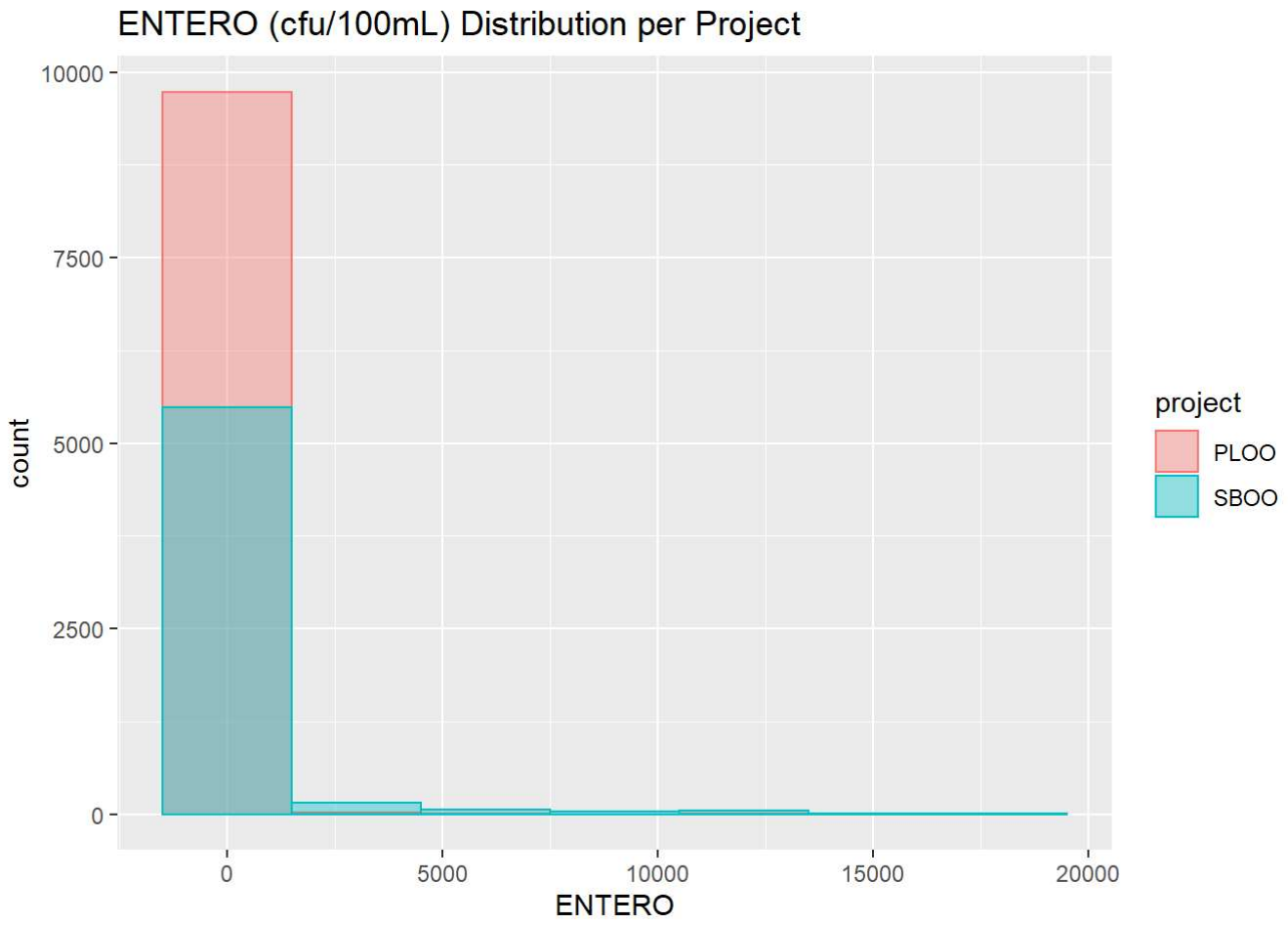
```
## Warning: Removed 111444 rows containing non-finite values (stat_bin).
```




```
  # relatively normal distributed

# ENTERO
ggplot(splitted, aes(x= ENTERO)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "ENTERO (cfu/100mL) Distribution per Project")
```

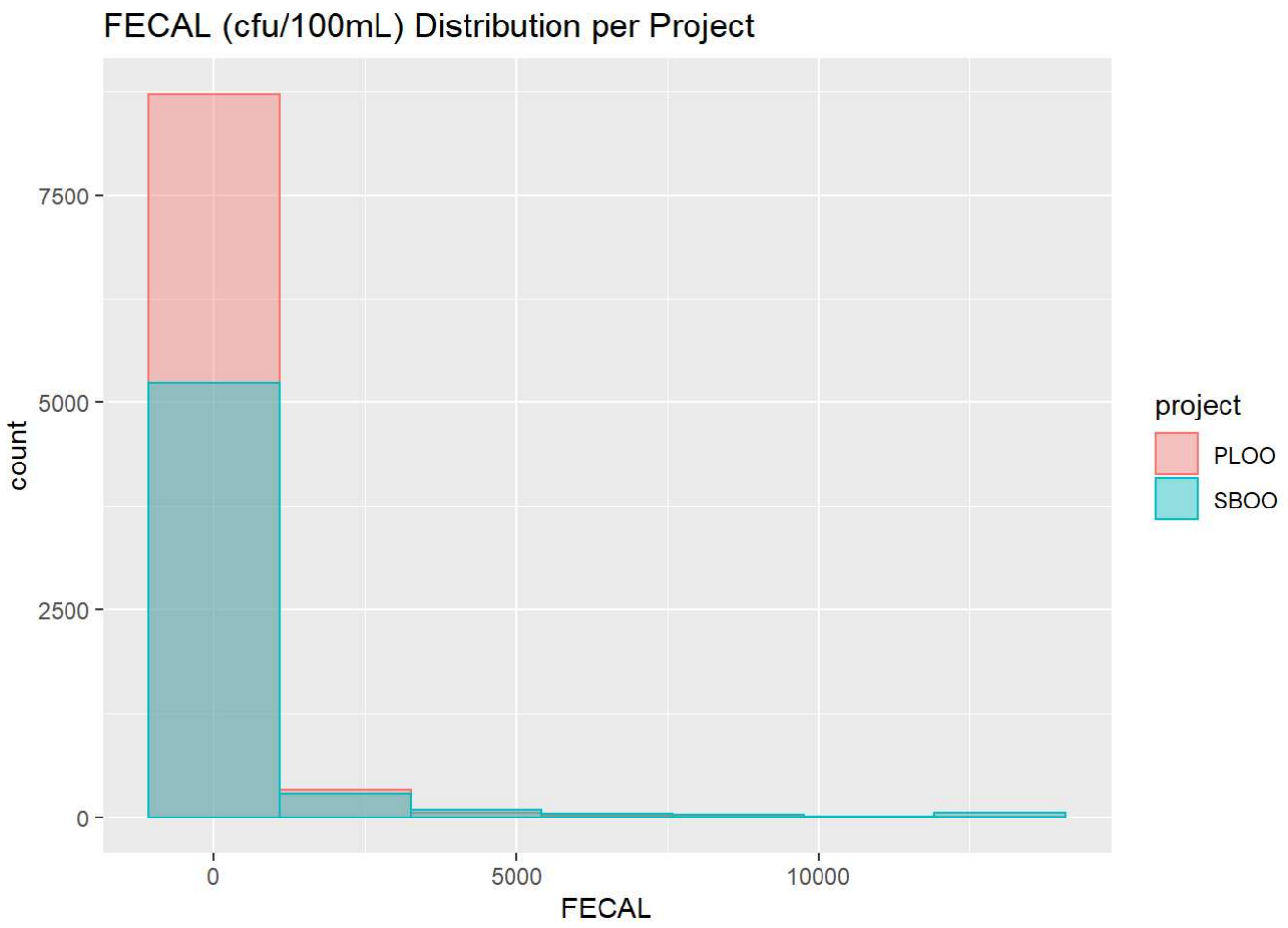
```
## Warning: Removed 105753 rows containing non-finite values (stat_bin).
```




```
  # skewed with most obs towards left (low values)

# FECAL
ggplot(splitted, aes(x= FECAL)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "FECAL (cfu/100mL) Distribution per Project")
```

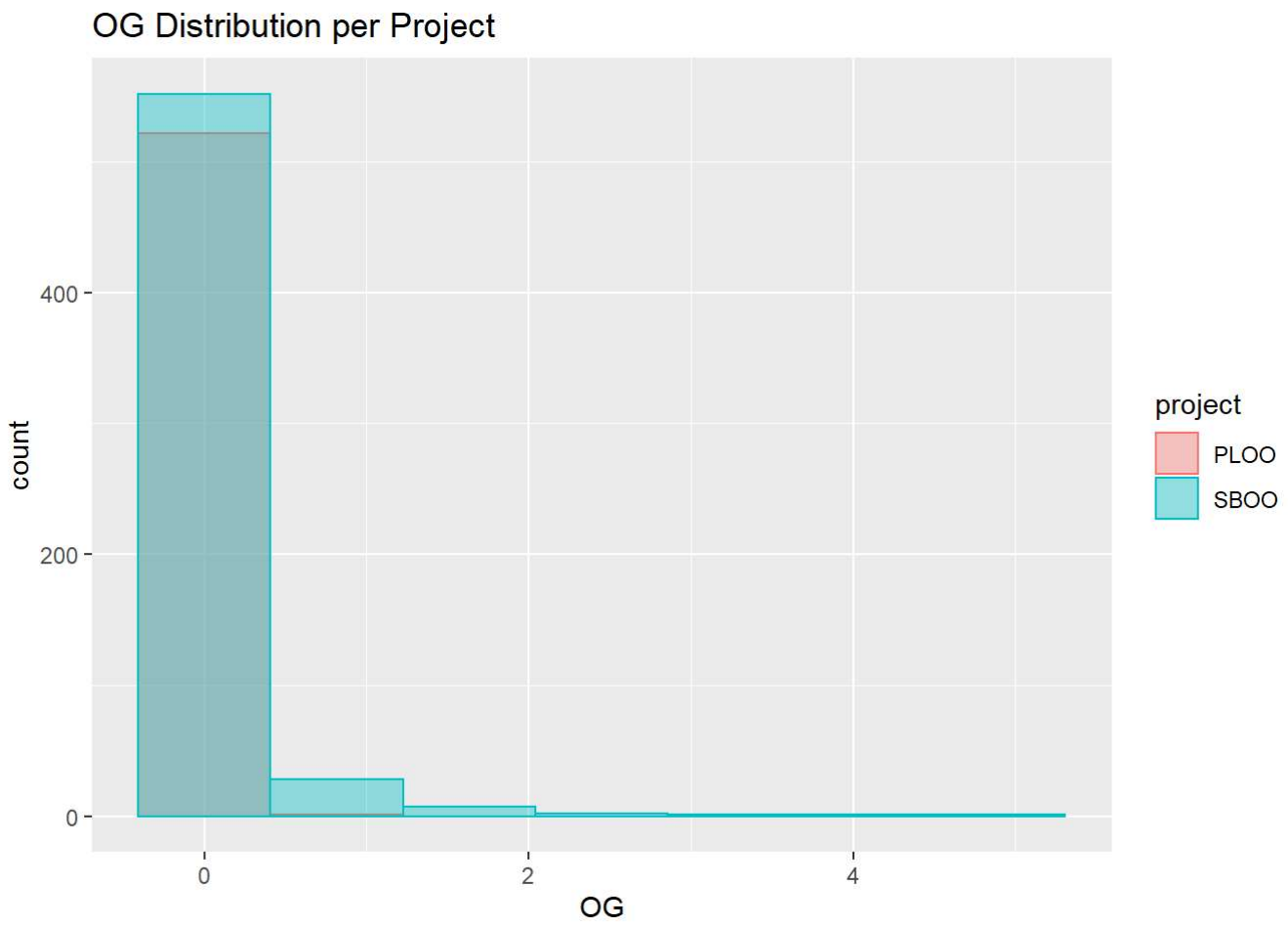
```
## Warning: Removed 106412 rows containing non-finite values (stat_bin).
```




```
  # skewed with most obs towards left (low values)

# OG
ggplot(splitted, aes(x= OG)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "OG Distribution per Project")
```

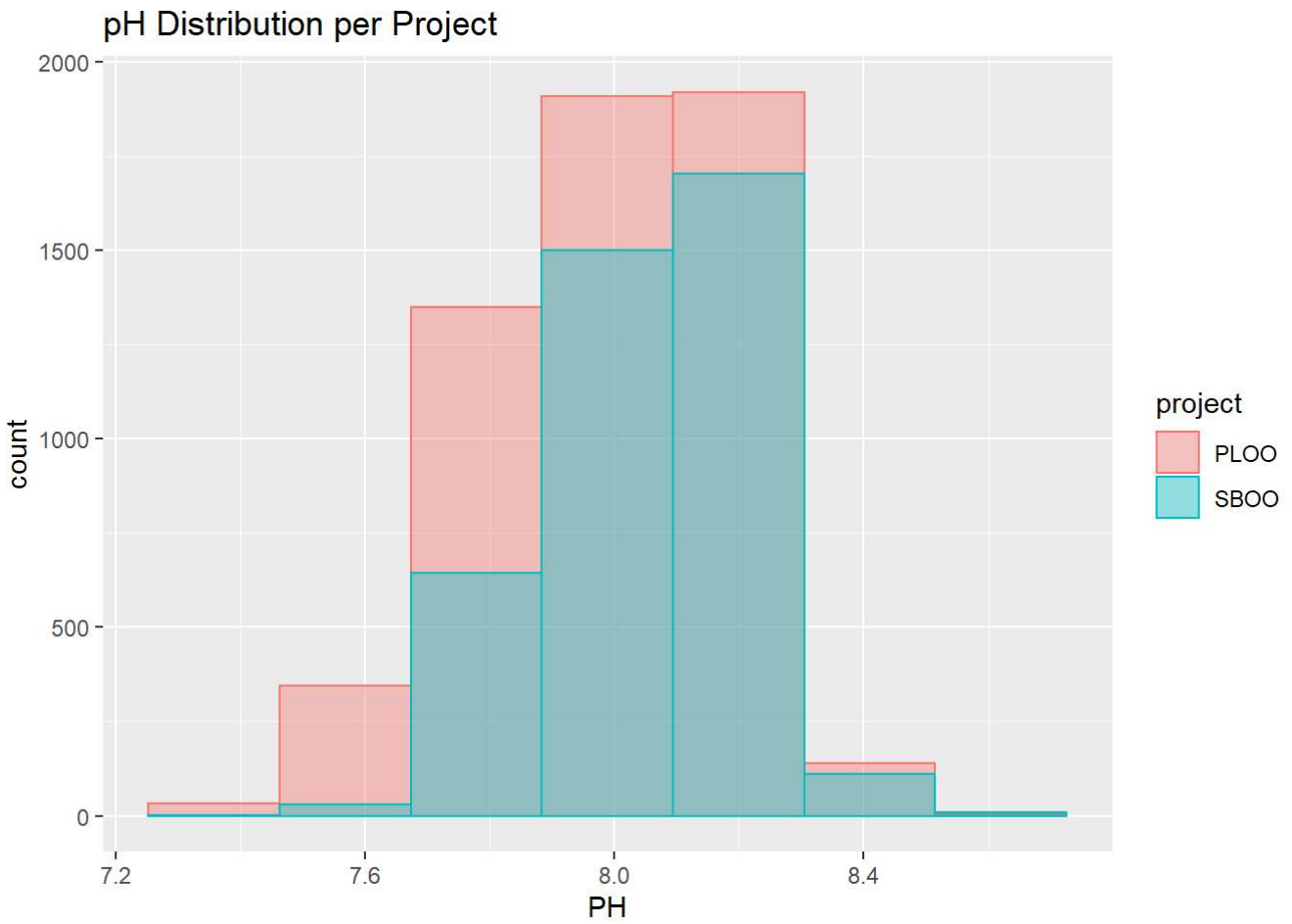
```
## Warning: Removed 120171 rows containing non-finite values (stat_bin).
```




```
  # skewed with most obs towards left (low values)

# PH
ggplot(splitted, aes(x= PH)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "pH Distribution per Project")
```

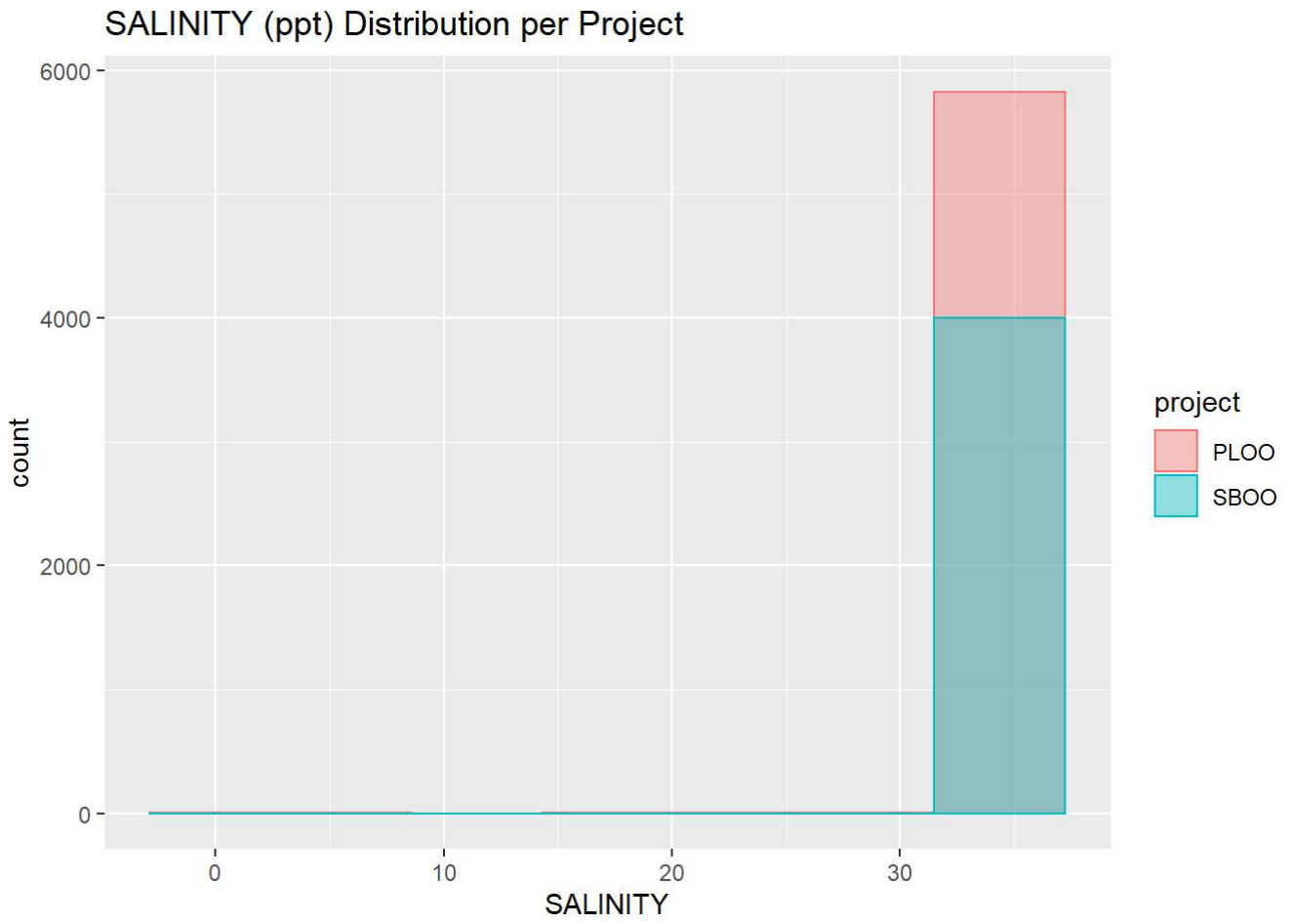
```
## Warning: Removed 111591 rows containing non-finite values (stat_bin).
```




```
  # near normal distribution, slightly stronger drop off on alkaline side

# SALINITY
ggplot(splitted, aes(x= SALINITY)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "SALINITY (ppt) Distribution per Project")
```

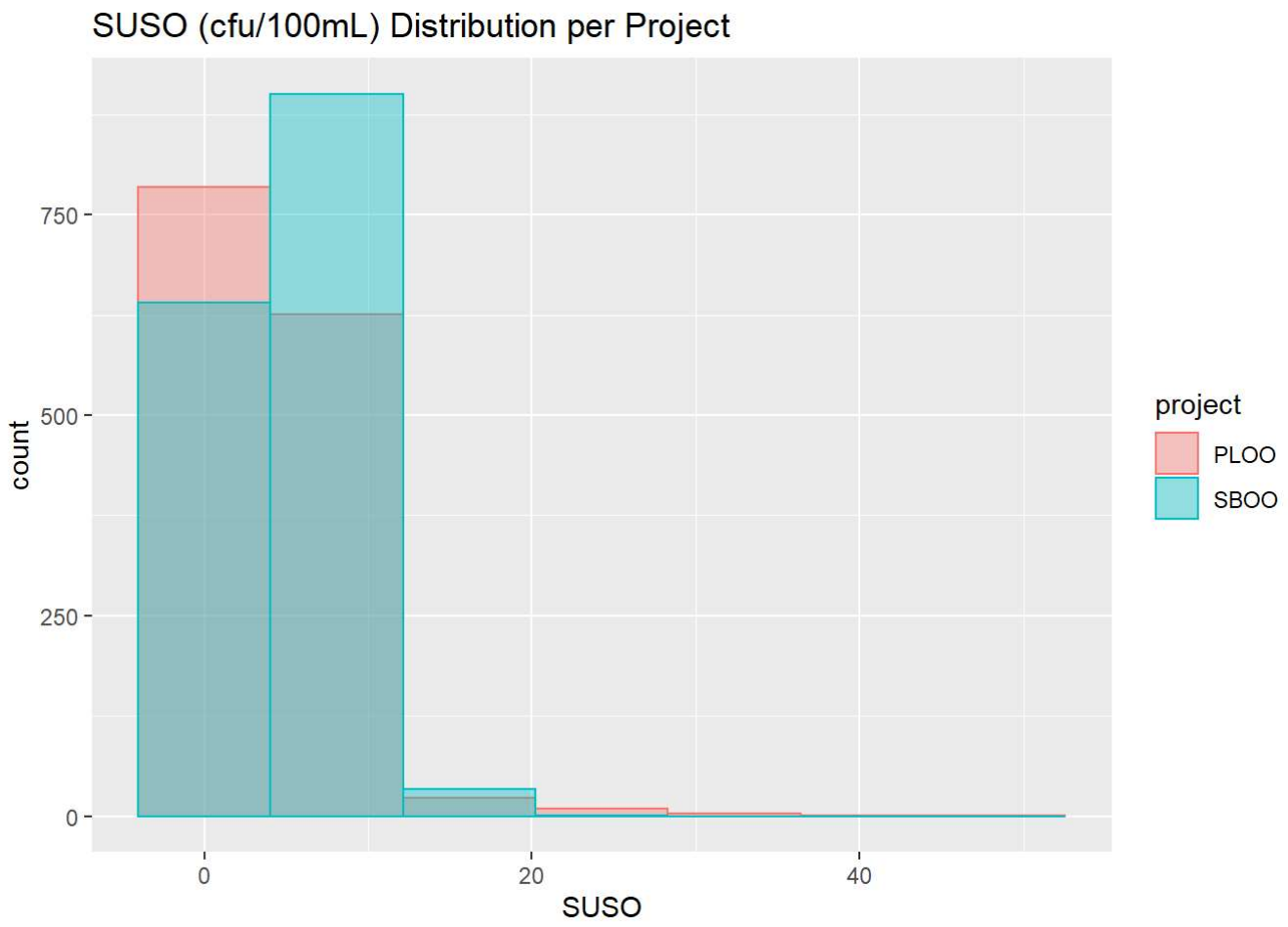
```
## Warning: Removed 111442 rows containing non-finite values (stat_bin).
```




```
    # essentially only values between 30 and 40, not interesting

# SUSO
ggplot(splitted, aes(x= SUSO)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "SUSO (cfu/100mL) Distribution per Project")
```

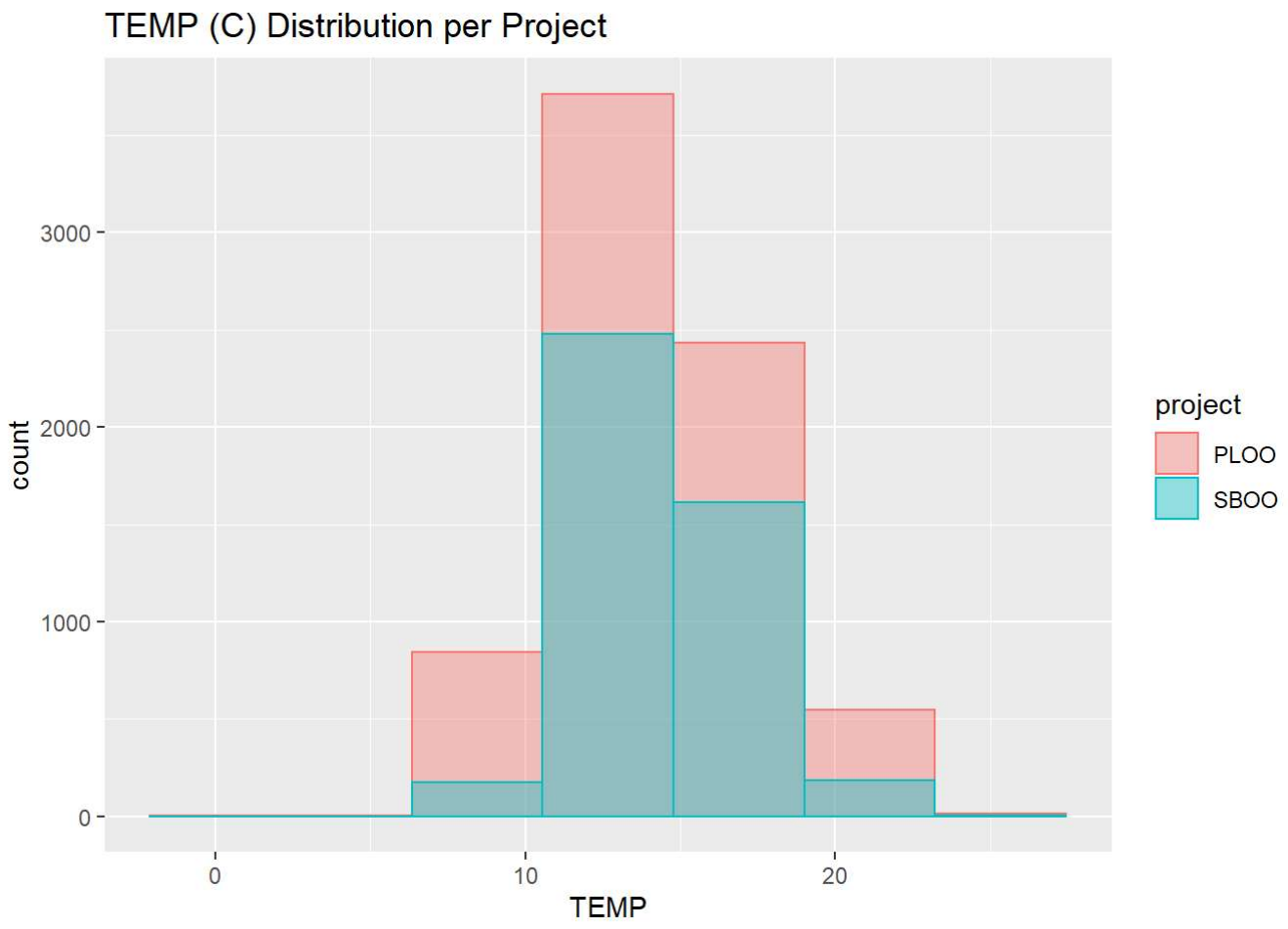
```
## Warning: Removed 118263 rows containing non-finite values (stat_bin).
```




```
  # skewed with most obs towards left (low values)

# TEMP
ggplot(splitted, aes(x= TEMP)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "TEMP (C) Distribution per Project")
```

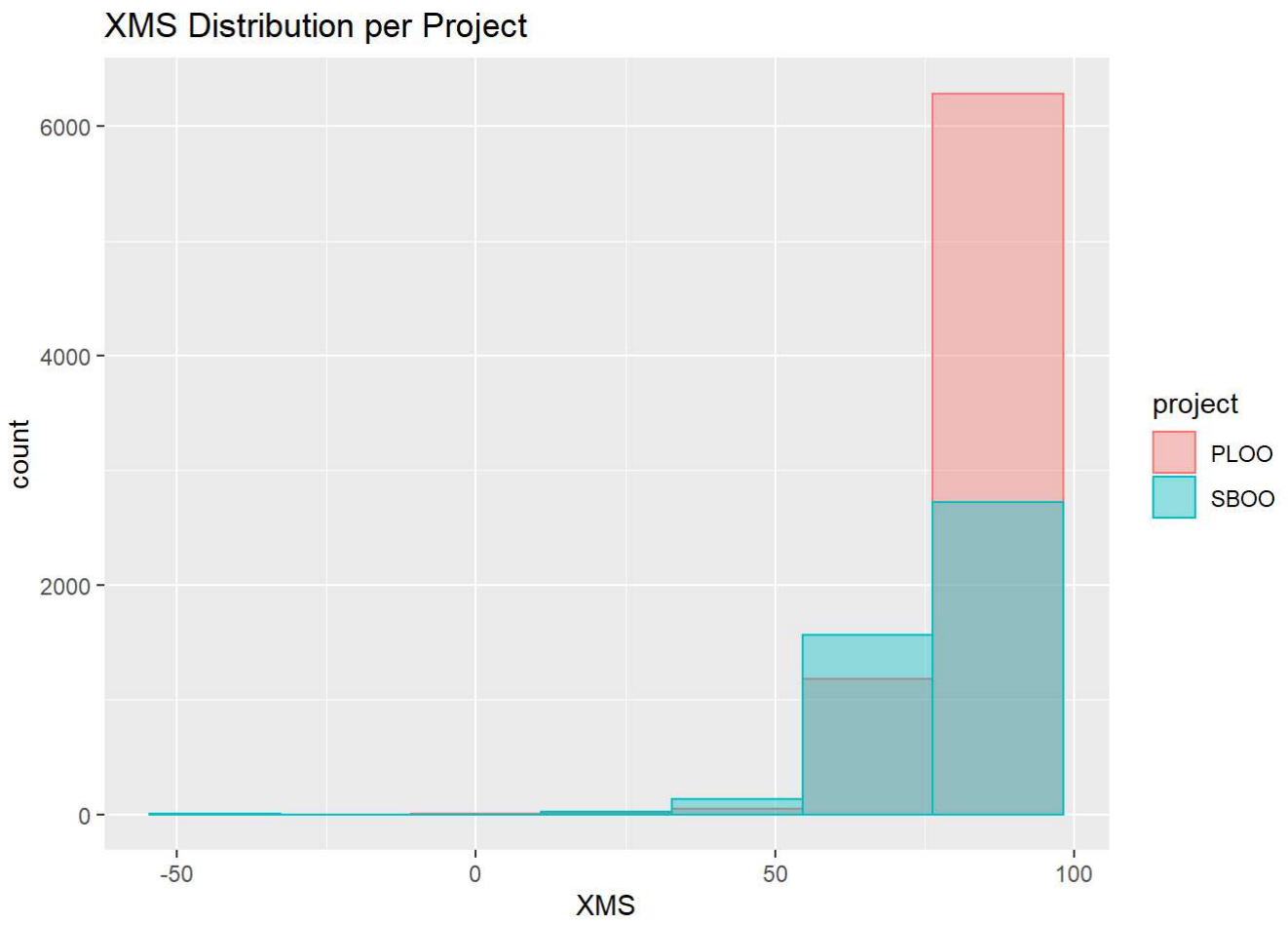
```
## Warning: Removed 109276 rows containing non-finite values (stat_bin).
```




```
  # near normal distribution

# XMS
ggplot(splitted, aes(x= XMS)) +
  geom_histogram(aes(color= project, fill= project),
                 position= "identity", bins=7, alpha=.4) +
  labs(title= "XMS Distribution per Project")
```

```
## Warning: Removed 109317 rows containing non-finite values (stat_bin).
```




```
  # skewed to right with most obs for 50-100%, especially >75%

# Entero, fecal, OG, salinity, and arguably, XMS do not have interesting distributions.
  # I think we can safely not do salinity and XMS since they are so uniform.
# Entero, fecal, and Suso are measures relating to bacteria that often come from fecal matter an
d fecal matter (fecal) so they are likely going to show the same thing (modeling-wise).
  # I think we should pick one (Suso has most diverse histogram) of the three.
```

Summary (Descriptive Stats) with NA's included:

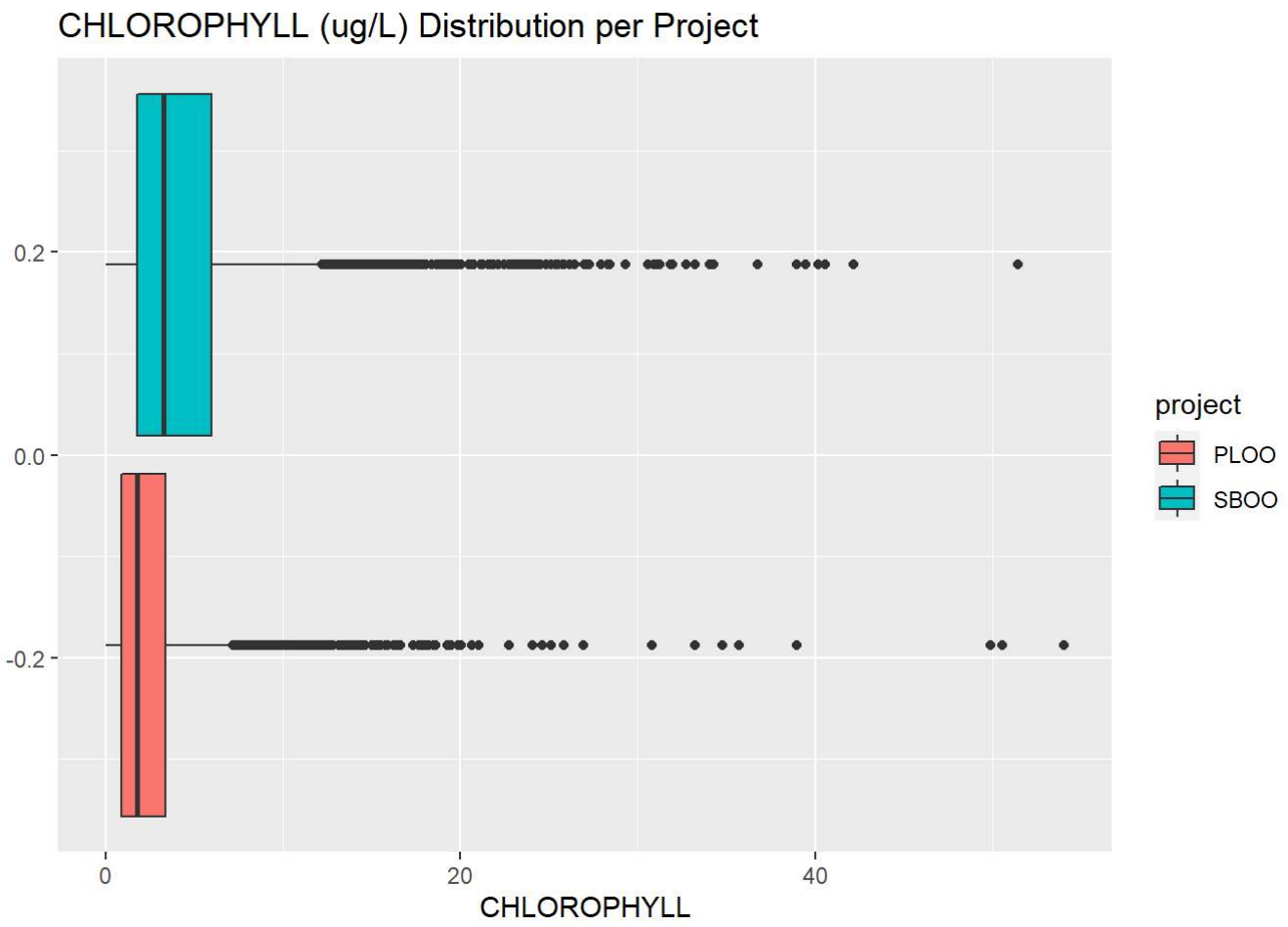
```
summary(splitted)
```

```
##   date_sample         project      depth_m_bin         CHLOROPHYLL
##  Min.   :1990-11-15   PLOO:72690   Length:121286      Min.   : 0.00
##  1st Qu.:2000-09-05   SBOO:48596   Class :character   1st Qu.: 1.19
##  Median :2006-09-13                Mode  :character   Median : 2.38
##  Mean   :2006-11-21                                   Mean   : 3.60
##  3rd Qu.:2013-05-10                                   3rd Qu.: 4.48
##  Max.   :2021-12-29                                   Max.   :54.04
##                                                       NA's   :112900
##     DENSITY             DO             ENTERO             FECAL
##  Min.   :22.39    Min.   : 0.00    Min.   :    0.00   Min.   :    0.00
##  1st Qu.:24.53    1st Qu.: 5.50    1st Qu.:    2.00   1st Qu.:    2.00
##  Median :24.98    Median : 7.10    Median :    2.80   Median :    3.85
##  Mean   :24.96    Mean   : 6.70    Mean   :  150.11   Mean   :  269.82
##  3rd Qu.:25.48    3rd Qu.: 7.95    3rd Qu.:   15.82   3rd Qu.:   30.24
##  Max.   :26.48    Max.   :12.83    Max.   :18000.00   Max.   :13000.00
##  NA's   :112659   NA's   :111444   NA's   :105753     NA's   :106412
##        OG                PH            SALINITY           SUSO
##  Min.   :0.20     Min.   :7.33     Min.   : 0.00    Min.   : 0.20
##  1st Qu.:0.20     1st Qu.:7.89     1st Qu.:33.42    1st Qu.: 3.08
##  Median :0.20     Median :8.05     Median :33.54    Median : 4.18
##  Mean   :0.24     Mean   :8.01     Mean   :33.52    Mean   : 4.77
##  3rd Qu.:0.20     3rd Qu.:8.15     3rd Qu.:33.65    3rd Qu.: 5.71
##  Max.   :5.10     Max.   :8.59     Max.   :34.36    Max.   :48.70
##  NA's   :120171   NA's   :111591   NA's   :111442   NA's   :118263
##       TEMP              XMS
##  Min.   : 0.00    Min.   :-37.00
##  1st Qu.:12.07    1st Qu.: 76.40
##  Median :14.14    Median : 82.04
##  Mean   :14.27    Mean   : 80.22
##  3rd Qu.:15.99    3rd Qu.: 86.46
##  Max.   :25.34    Max.   : 93.83
##  NA's   :109276   NA's   :109317
```

Plotting boxplots to see distribution

```
# chlorophyll
ggplot(splitted, aes(x= CHLOROPHYLL, fill=project)) +
  geom_boxplot() +
  labs(title= "CHLOROPHYLL (ug/L) Distribution per Project")
```

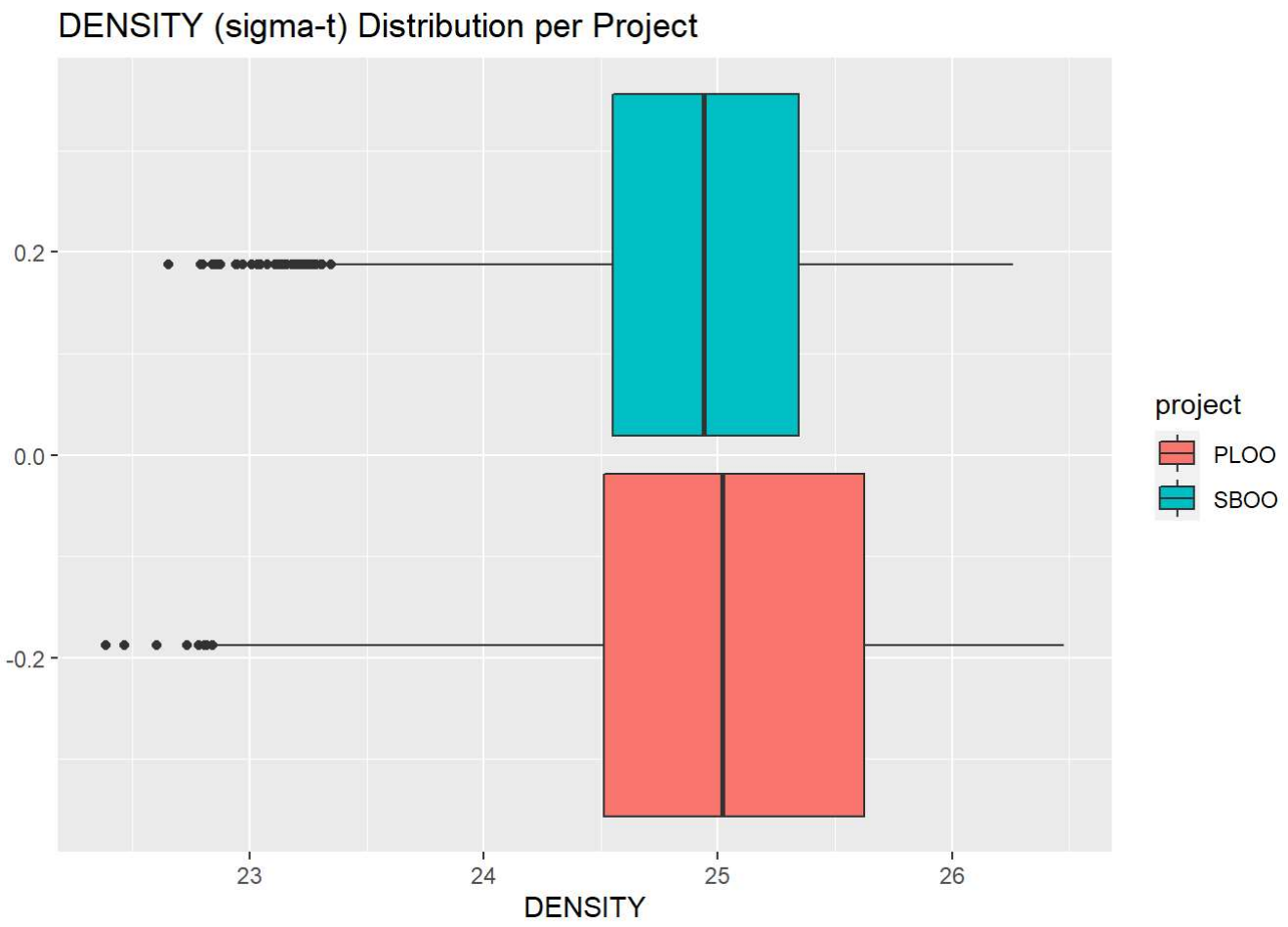
```
## Warning: Removed 112900 rows containing non-finite values (stat_boxplot).
```




```
  # primarily outlier filled

# density
ggplot(splitted, aes(x= DENSITY, fill=project)) +
  geom_boxplot() +
  labs(title= "DENSITY (sigma-t) Distribution per Project")
```

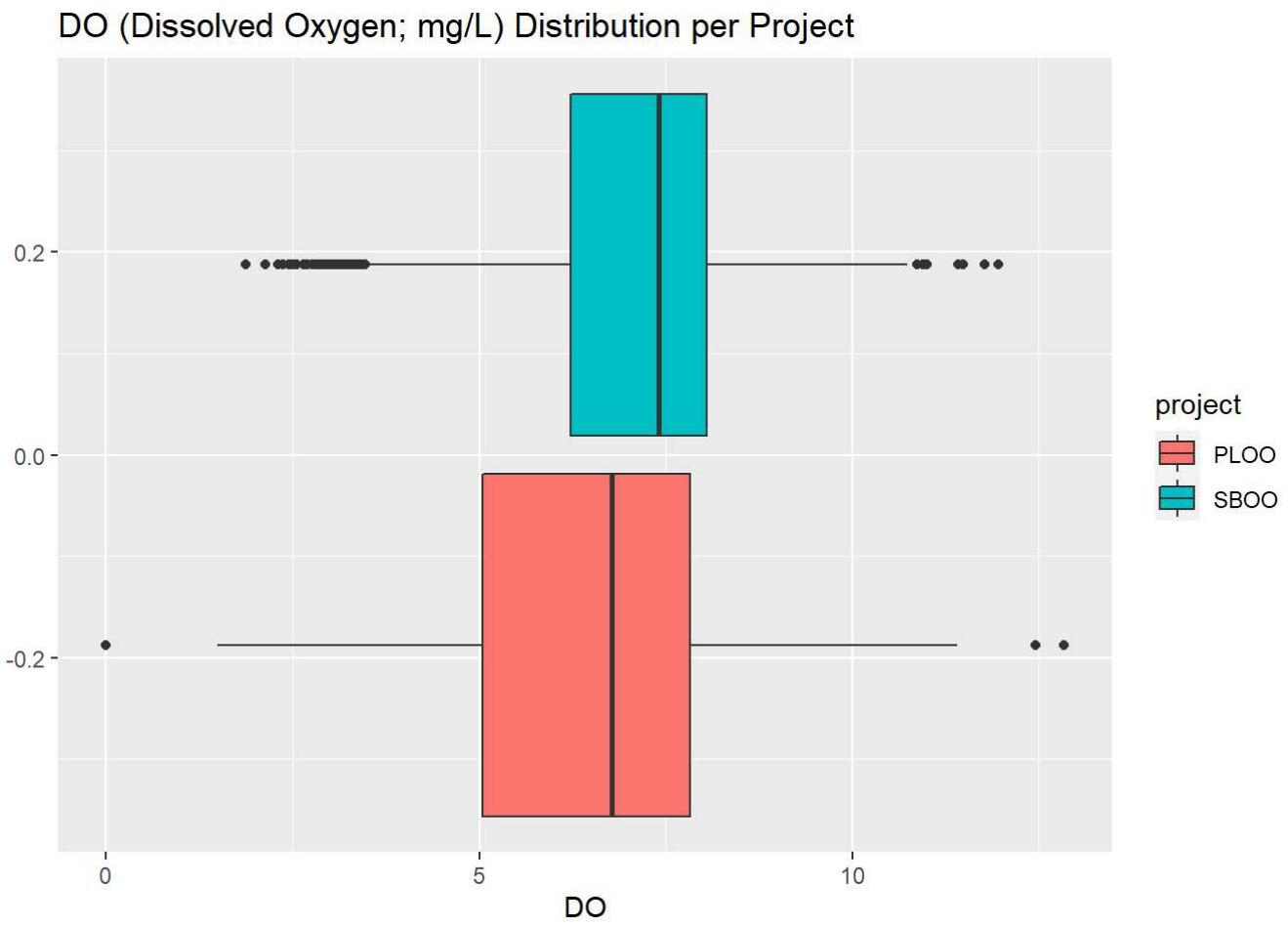
```
## Warning: Removed 112659 rows containing non-finite values (stat_boxplot).
```




```
  # most outliers on low density measures

# DO
ggplot(splitted, aes(x= DO, fill=project)) +
  geom_boxplot() +
  labs(title= "DO (Dissolved Oxygen; mg/L) Distribution per Project")
```

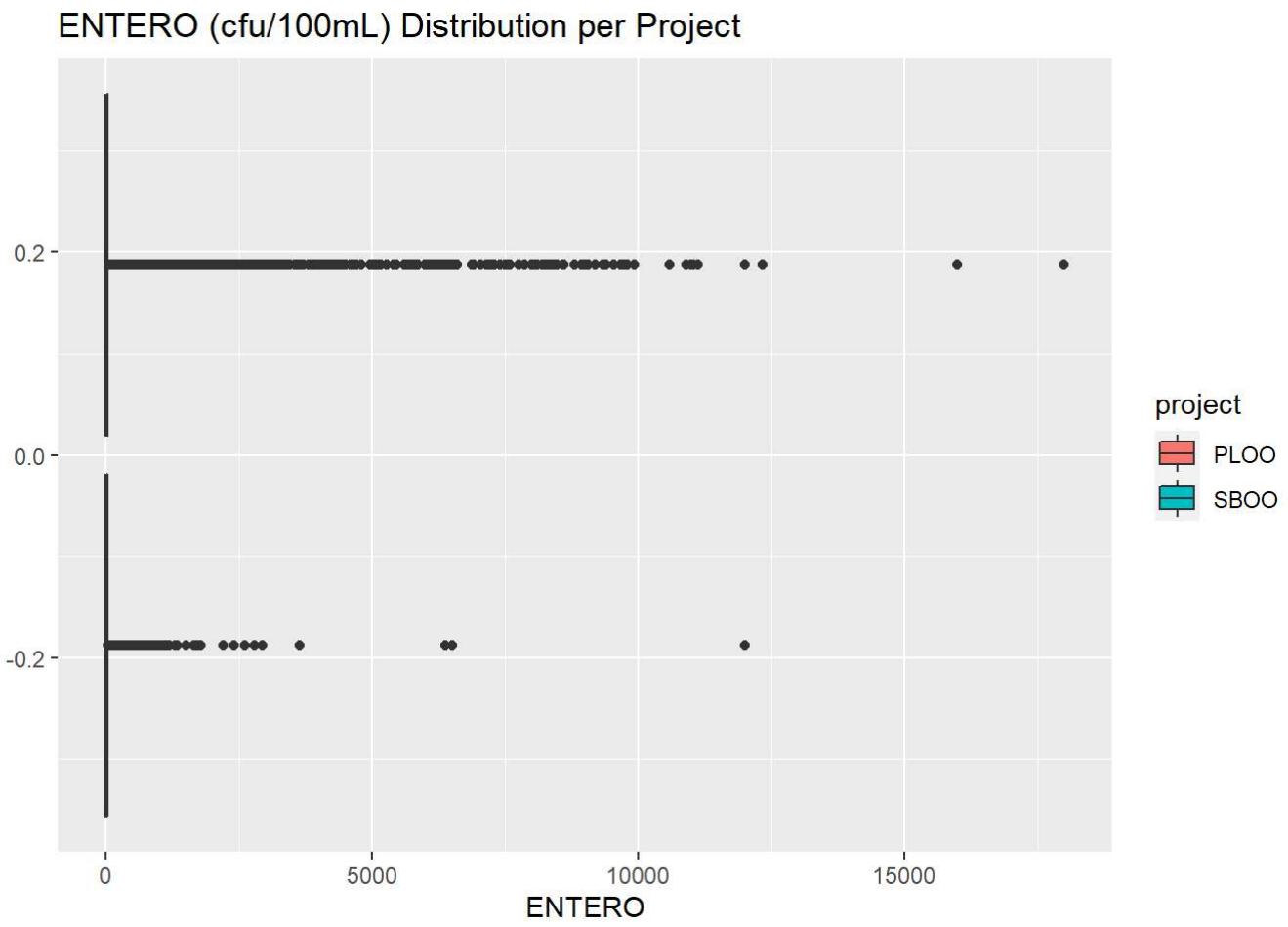
```
## Warning: Removed 111444 rows containing non-finite values (stat_boxplot).
```




```
  # very few outliers

# ENTERO
ggplot(splitted, aes(x= ENTERO, fill=project)) +
  geom_boxplot() +
  labs(title= "ENTERO (cfu/100mL) Distribution per Project")
```

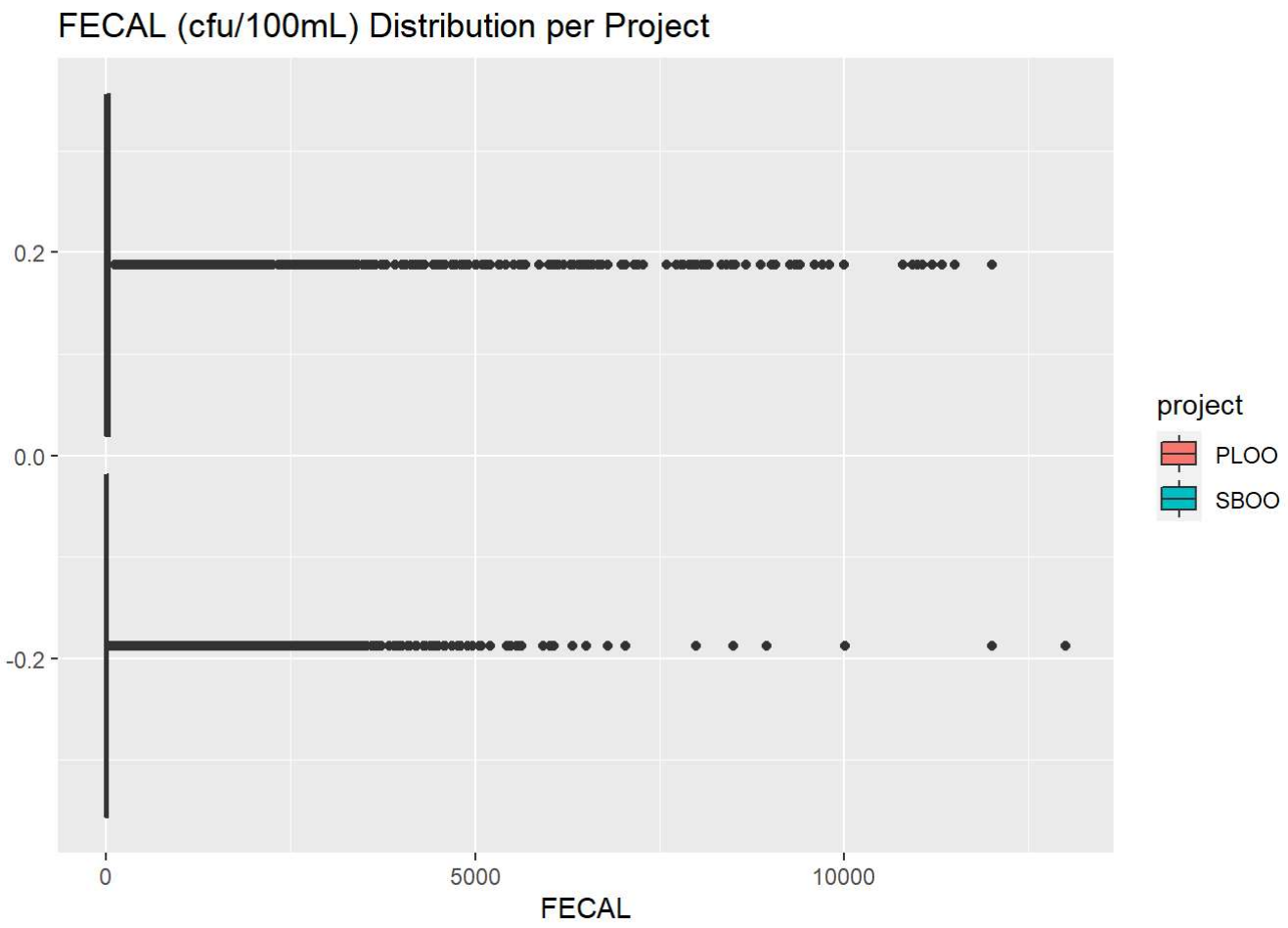
```
## Warning: Removed 105753 rows containing non-finite values (stat_boxplot).
```




```
  # the plot is essentially all outliers

# FECAL
ggplot(splitted, aes(x= FECAL, fill=project)) +
  geom_boxplot() +
  labs(title= "FECAL (cfu/100mL) Distribution per Project")
```

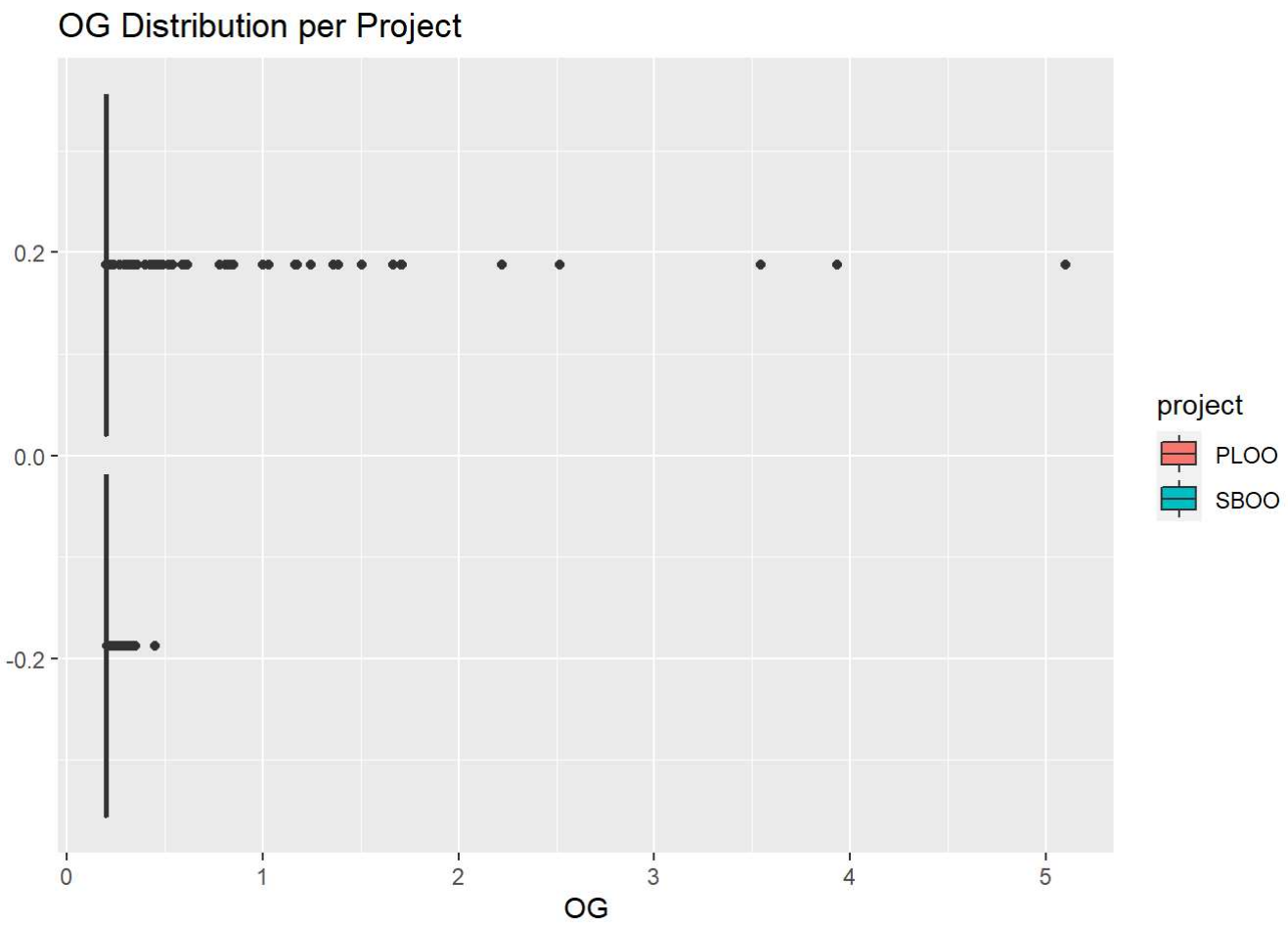
```
## Warning: Removed 106412 rows containing non-finite values (stat_boxplot).
```




```
  # the plot is essentially all outliers

# OG
ggplot(splitted, aes(x= OG, fill=project)) +
  geom_boxplot() +
  labs(title= "OG Distribution per Project")
```

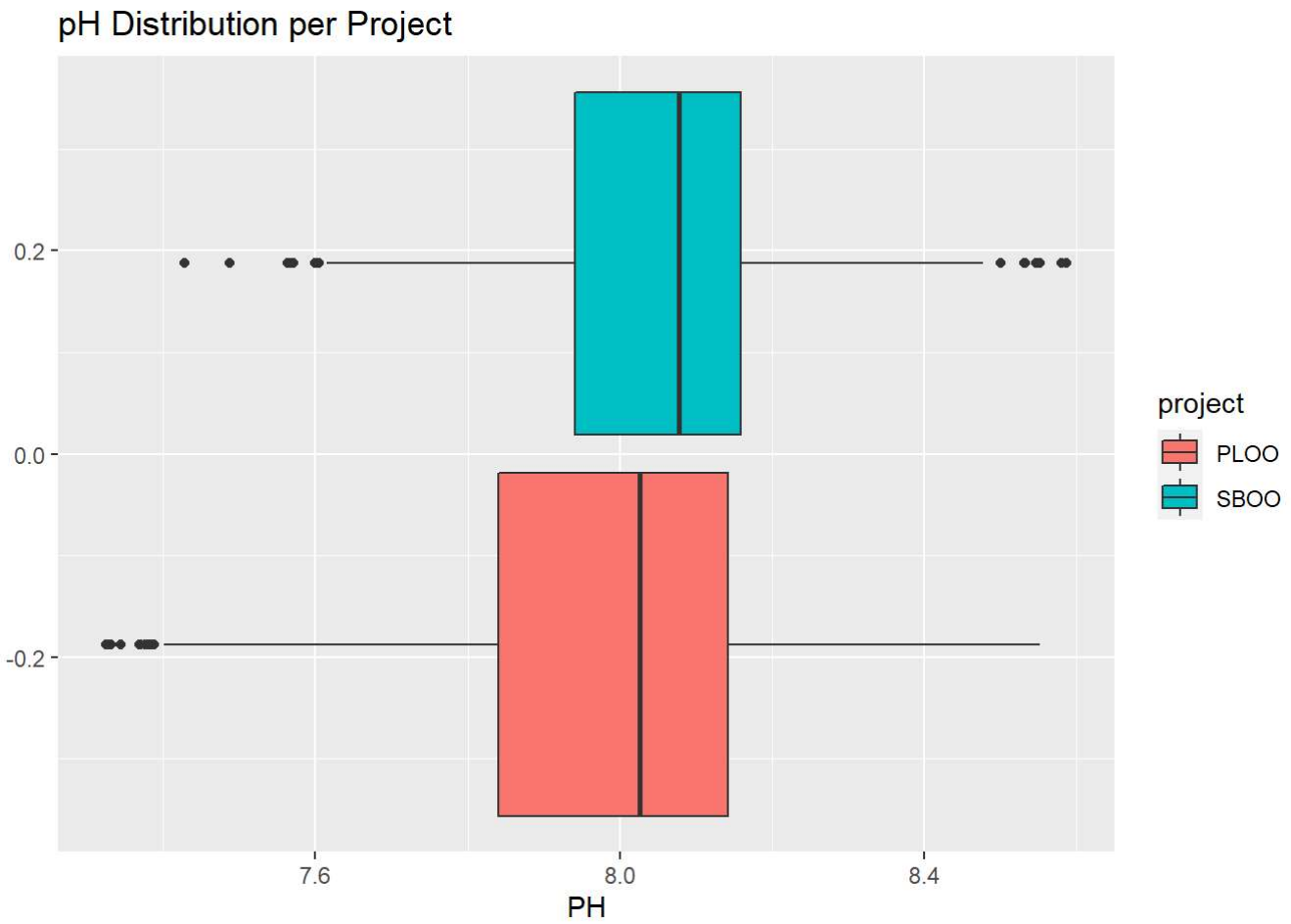
```
## Warning: Removed 120171 rows containing non-finite values (stat_boxplot).
```




```
  # the plot is essentially all outliers

# PH
ggplot(splitted, aes(x= PH, fill=project)) +
  geom_boxplot() +
  labs(title= "pH Distribution per Project")
```

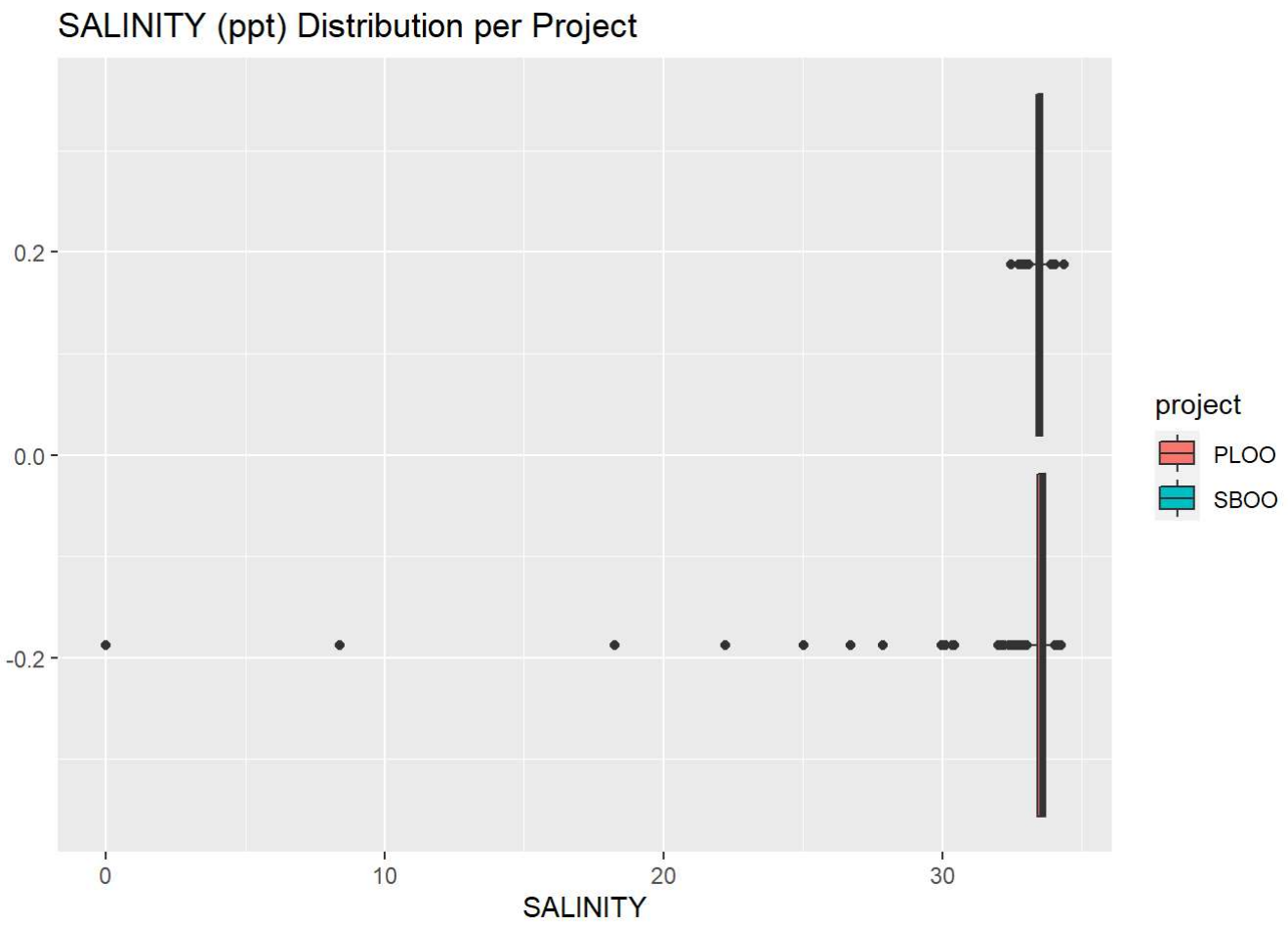
```
## Warning: Removed 111591 rows containing non-finite values (stat_boxplot).
```




```
  # more outliers on the acidic (low value) end more than alkaline end

# SALINITY
ggplot(splitted, aes(x= SALINITY, fill=project)) +
  geom_boxplot() +
  labs(title= "SALINITY (ppt) Distribution per Project")
```

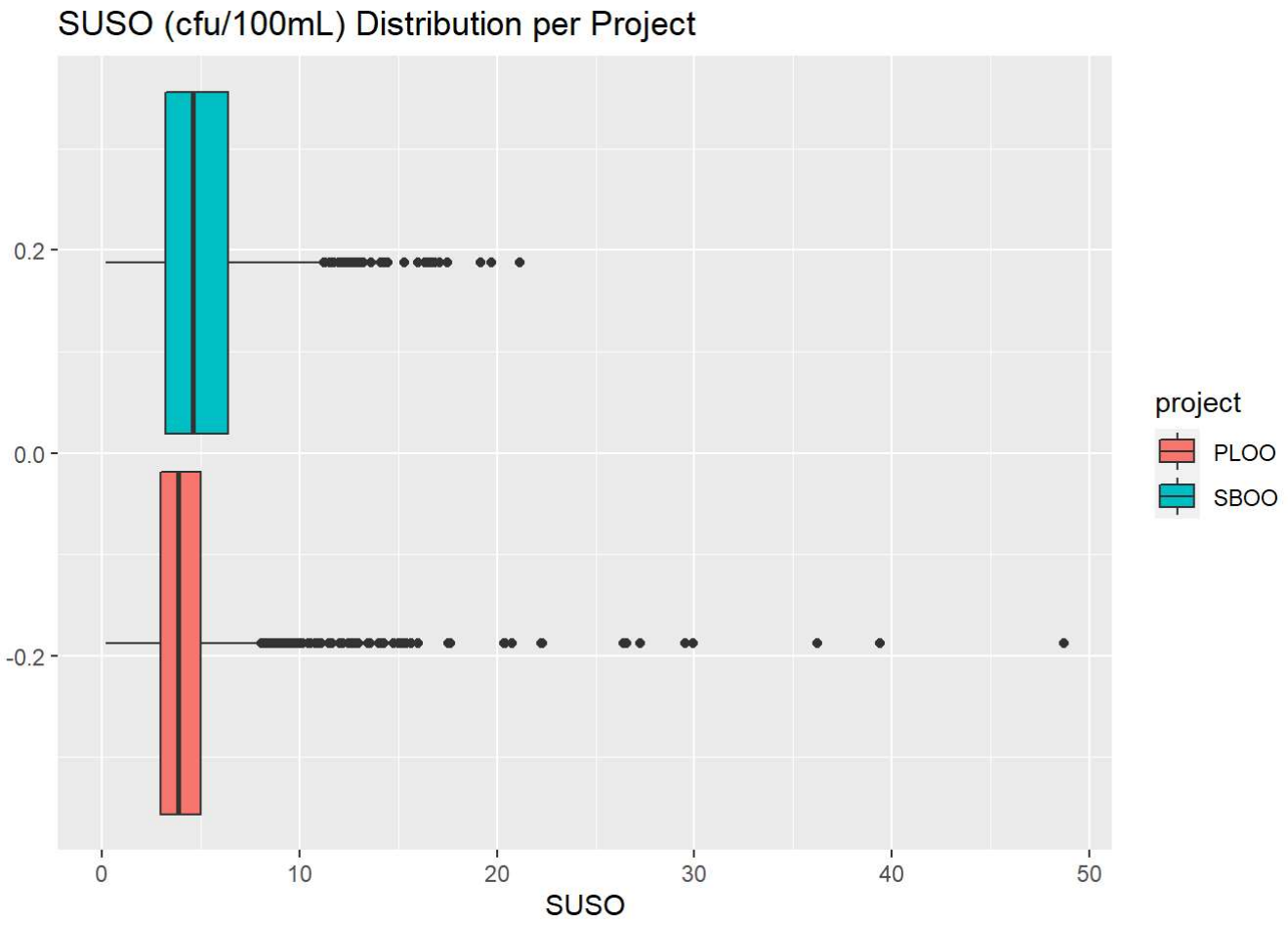
```
## Warning: Removed 111442 rows containing non-finite values (stat_boxplot).
```




```
    # the plot is essentially all outliers

# SUSO
ggplot(splitted, aes(x= SUSO, fill=project)) +
  geom_boxplot() +
  labs(title= "SUSO (cfu/100mL) Distribution per Project")
```

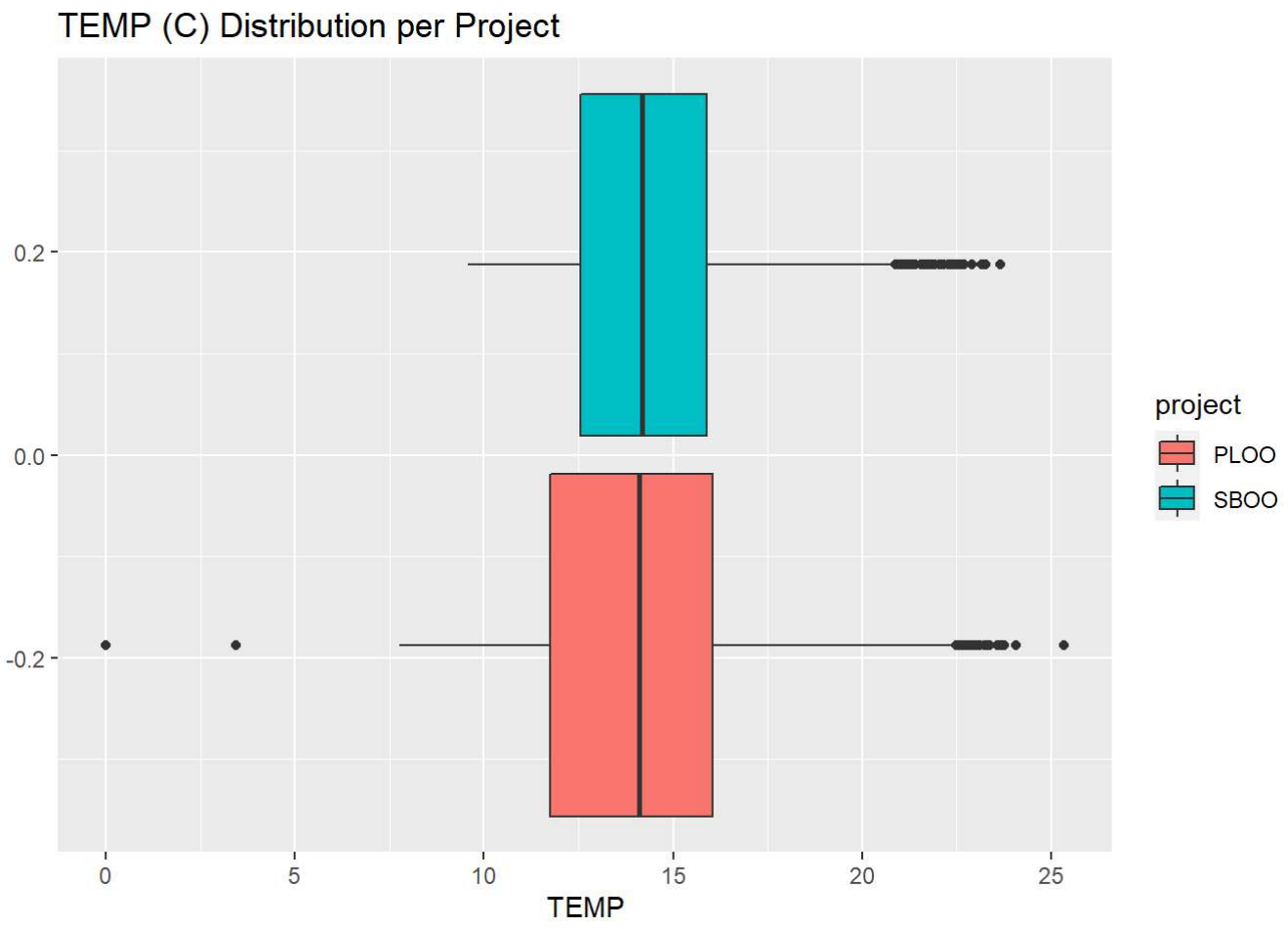
```
## Warning: Removed 118263 rows containing non-finite values (stat_boxplot).
```




```
  # the plot is essentially all outliers,
    # fewer than other bacterial measures (fecal, entero)

# TEMP
ggplot(splitted, aes(x= TEMP, fill=project)) +
  geom_boxplot() +
  labs(title= "TEMP (C) Distribution per Project")
```

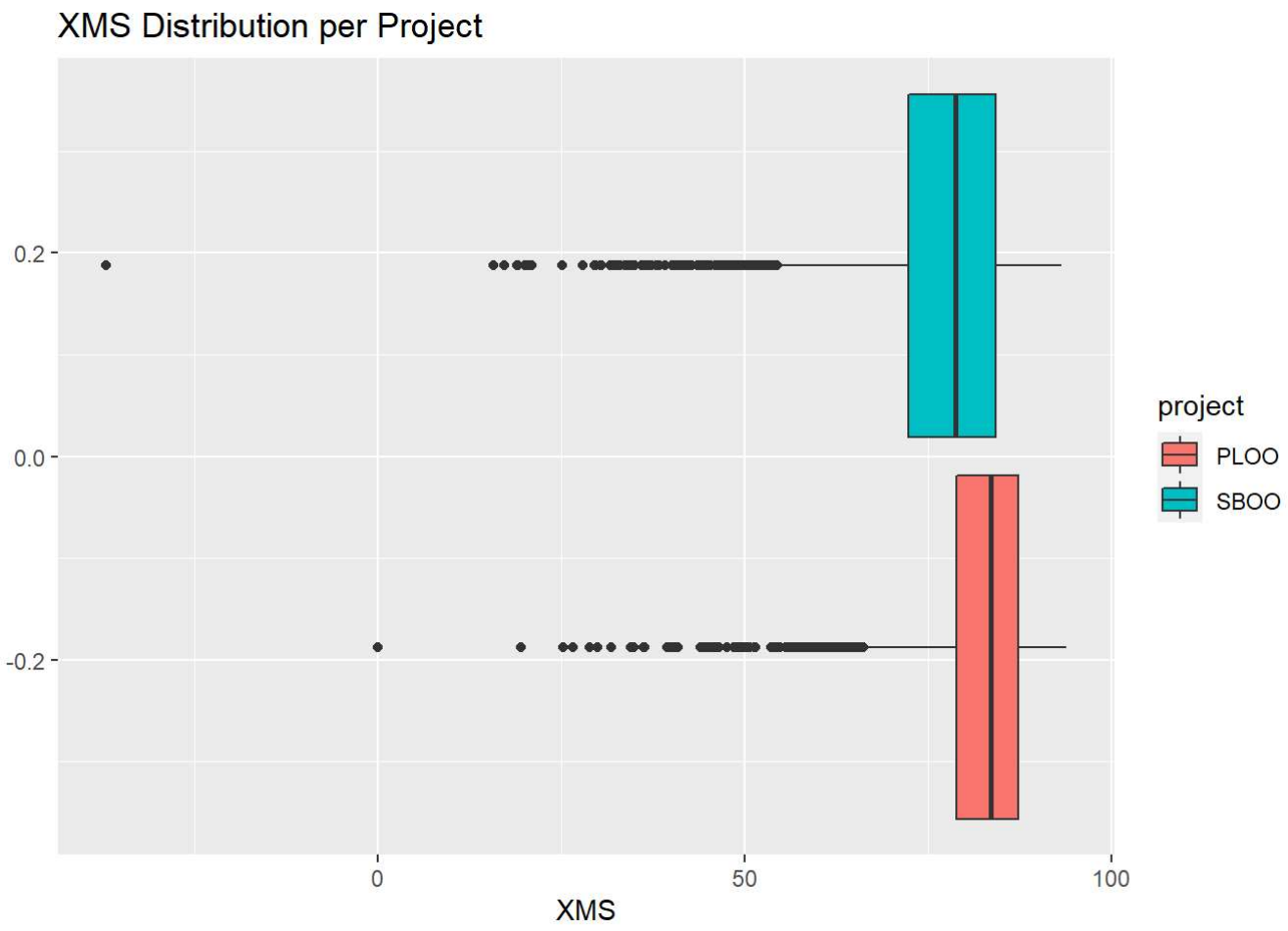
```
## Warning: Removed 109276 rows containing non-finite values (stat_boxplot).
```




```
  # some outliers, primarily at warmer temperatures

# XMS
ggplot(splitted, aes(x= XMS, fill=project)) +
  geom_boxplot() +
  labs(title= "XMS Distribution per Project")
```

```
## Warning: Removed 109317 rows containing non-finite values (stat_boxplot).
```




```
  # primarily outliers

# Again the fecal, entero, OG, and salinity look nearly identical and equally uninteresting
```