

# Annual Report - Time Series

Jess S. Glanz | SMS-BEL Program Coordinator | jess.glanz@gmail.com

8 Mar 2023

## Load the packages

```
library(readxl)
library(dplyr)
library(tidyverse)
library(tidyr)
library(ggplot2)
library(ggpubr)
```

## Load the data

For this example, the sediment organic and water content data will be loaded.

First, load the sheet titled “MSites\_AllData”. This contains all the organic and water content data as it was entered for each replicate.

```
ow_cur<-read_excel("~/Desktop/SMS-BEL/SMS-EnvlAnalysis/Data/2022.12.09_SLEOS.xlsx", sheet = 4)
```

Then, load the sheet titled “MeanSE21”. This contains the mean and standard error across replicates for each site, depth, and sampling event.

```
ow_past<-read_excel("~/Desktop/SMS-BEL/SMS-EnvlAnalysis/Data/2022.12.09_SLEOS.xlsx", sheet=3)
```

## Tidy the data

Convert the percent water and percent organic variables to numeric, in case they’re not already.

```
ow_cur$pwater<-as.numeric(ow_cur$pwater)
ow_cur$pLOI<-as.numeric(ow_cur$pLOI)
ow_past$pwmean<-as.numeric(ow_past$pwmean)
ow_past$pwse<-as.numeric(ow_past$pwse)
ow_past$pomean<-as.numeric(ow_past$pomean)
ow_past$pose<-as.numeric(ow_past$pose)
```

Filter the first loaded sheet for the most recent values and the second loaded sheet for all previous collection dates.

```
ow_cur<-ow_cur%>%filter(rsd>=2021.25)

pastJa21<-ow_past%>%filter(rsd<2021.25)
```

Calculate mean percent water and mean percent organics at each site per recent collection date and depth.

```
sowcr<- ow_cur %>%
  group_by(Site,rsd,Depth) %>%
  summarize(pwmean=mean(pwater, na.rm=TRUE), pwse=sd(pwater, na.rm=TRUE)/sqrt(length(na.omit(pwater))),
            pomean=mean(pLOI, na.rm=TRUE), pose=sd(pLOI, na.rm=TRUE)/sqrt(length(na.omit(pLOI))))
```

Select and reorder the matching columns between pastJa21 and sowcr before binding the two. Then, remove the last two rows that are unnecessary.

```
pastJa21<-pastJa21%>%select(1:2,4:8) #select all columns except "Q", the only column sowcr doesn't have

sowcr<-sowcr %>% relocate(Depth, .after = Site) #reorders columns in dataframe

allmse<-rbind(pastJa21,sowcr) #binds these two datasets by stacking pastJa21's rows over sowcr's rows

allmse<-allmse[-c(2133:2134),] #removes last two rows of dataframe
```

Convert Depth into a factor variable for plotting.

```
allmse$Depth<-as.factor(allmse$Depth)
```

## Create percent water content plots for the River sites, one at a time and then arrange into one panel.

Running the code in the chunk below once prior to creating all of the time series plots will create one tick mark per quarter on the x axis, with only the January tick marks labeled with the year.

```
every_nth <- function(x, nth, empty = TRUE, inverse = FALSE)
{
  if (!inverse) {
    if(empty) {
      x[1:nth == 1] <- ""
      x
    } else {
      x[1:nth != 1]
    }
  } else {
    if(empty) {
      x[1:nth != 1] <- ""
      x
    } else {
      x[1:nth == 1]
    }
  }
}

custom_breaks <- seq(2005, 2022.75, 0.25)
```

Step 1: Filter the data by site.

```
m15<-allmse %>%  
  filter(Site == "M15")
```

Step 2: Create R label for the 0-2cm cores.

```
m15s<-m15%>%filter(Depth=="2") #filter data for 0-2cm depth  
  
cor15s<-cor.test(m15s$rsd,m15s$pwmean, method="spearman",na.rm=TRUE) #calculate R value  
  
sr15 <- sprintf(  
  "italic(R) ~ '=' ~ '%#.2f'",  
  cor15s$estimate) #create label to call when plotting
```

Step 3: Create R label for the 2-5cm cores.

```
m15d<-m15%>%filter(Depth=="5") #filter data for 2-5cm depth  
  
cor15d<-cor.test(m15d$rsd,m15d$pwmean, method="spearman",na.rm=TRUE) #calculate R value  
  
br15 <- sprintf(  
  "italic(R) ~ '=' ~ '%#.2f'",  
  cor15d$estimate) #create label to call when plotting
```

**Plot the data. This only plots the percent water mean +/- SE over time for site M15.**

```
p15<- m15%>% ggplot( aes(x=rsd, y=pwmean, linetype=Depth,group=Depth, shape=Depth)) +  
  geom_line() +  
  geom_point(size=2)+  
  geom_errorbar(aes(ymin=pwmean-pwse, ymax=pwmean+pwse), width=.2,  
    position=position_dodge(0.05))+  
  scale_shape_manual(values=c(1, 16), breaks=c("2", "5"),  
    labels=c( "0-2cm", "2-5cm"))+  
  scale_linetype_manual(values=c("dashed", "solid"), breaks=c("2", "5"),  
    labels=c( "0-2cm", "2-5cm"))+  
  
  ggtitle("M15") +  
  ylim(0,125)+  
  scale_x_continuous(breaks = custom_breaks,  
    labels = every_nth(custom_breaks, 4, inverse = TRUE))+  
  geom_smooth(method="lm",color="black",se=FALSE,size=0.5)+  
  theme_bw() +  
  theme(panel.grid.major = element_blank(),  
    panel.grid.minor = element_blank(),  
    panel.background = element_rect(colour = "#471164", size=5))+  
  annotate("text", x=2020.5, y=118, label= "0-2cm") +  
  annotate("text", x=2022, y=118, label= sr15, parse=TRUE) +  
  annotate("text", x = 2020.5, y=103, label = "2-5cm")+  
  annotate("text", x = 2022, y=103, label = br15, parse=TRUE)+  
  theme(legend.position = "none")+  
  theme(plot.title = element_text(vjust = - 9, hjust= 0.025))+
```

```
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank())+
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

## Create percent water content plots for the remaining River sites.

Repeat the above for the other four River sites: M1, M2, M4, and M14.

```
#Site M01
m1<-allmse %>%
  filter(Site == "M01")

m1s<-m1%>%filter(Depth=="2")
cor1s<-cor.test(m1s$rsd,m1s$pwmean, method="spearman",na.rm=TRUE)
sr1 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor1s$estimate)

m1d<-m1%>%filter(Depth=="5")
cor1d<-cor.test(m1d$rsd,m1d$pwmean, method="spearman",na.rm=TRUE)
br1 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor1d$estimate)

p1<- m1%>% ggplot( aes(x=rsd, y=pwmean, linetype=Depth,group=Depth, shape=Depth)) +
  geom_line() +
  geom_point(size=2)+
  geom_errorbar(aes(ymin=pwmean-pwse, ymax=pwmean+pwse), width=.2,
    position=position_dodge(0.05))+
  scale_shape_manual(values=c(1, 16), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  scale_linetype_manual(values=c("dashed", "solid"), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  ggtitle("M01") +
  ylim(0,125)+
  scale_x_continuous(breaks = custom_breaks,
    labels = every_nth(custom_breaks, 4, inverse = TRUE))+
  geom_smooth(method="lm",color="black",se=FALSE,size=0.5)+
  theme_bw() +
  theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(colour = "#471164", size=5))+
  annotate("text", x=2020.5, y=118, label= "0-2cm") +
  annotate("text", x=2022, y=118, label= sr1, parse=TRUE) +
  annotate("text", x = 2020.5, y=103, label = "2-5cm")+
  annotate("text", x = 2022, y=103, label = br1, parse=TRUE)+
  theme(legend.position = "none")+
  theme(plot.title = element_text(vjust = - 9, hjust= 0.025))+
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank())+
```

```

theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Site M02
m2<-allmse %>%
  filter(Site == "M02")

m2s<-m2%>%filter(Depth=="2")
cor2s<-cor.test(m2s$rsd,m2s$pwmean, method="spearman",na.rm=TRUE)
sr2 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor2s$estimate)

m2d<-m2%>%filter(Depth=="5")
cor2d<-cor.test(m2d$rsd,m2d$pwmean, method="spearman",na.rm=TRUE)
br2 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor2d$estimate)

p2<- m2%>% ggplot( aes(x=rsd, y=pwmean, linetype=Depth,group=Depth, shape=Depth)) +
  geom_line() +
  geom_point(size=2)+
  geom_errorbar(aes(ymin=pwmean-pwse, ymax=pwmean+pwse), width=.2,
    position=position_dodge(0.05))+
  scale_shape_manual(values=c(1, 16), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  scale_linetype_manual(values=c("dashed", "solid"), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  ggtitle("M02") +
  ylim(0,125)+
  scale_x_continuous(breaks = custom_breaks,
    labels = every_nth(custom_breaks, 4, inverse = TRUE))+
  geom_smooth(method="lm",color="black",se=FALSE,size=0.5)+
  theme_bw() +
  theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(colour = "#471164", size=5))+
  annotate("text", x=2020.5, y=118, label= "0-2cm") +
  annotate("text", x=2022, y=118, label= sr2, parse=TRUE) +
  annotate("text", x = 2020.5, y=103, label = "2-5cm")+
  annotate("text", x = 2022, y=103, label = br2, parse=TRUE)+
  theme(legend.position = "none")+
  theme(plot.title = element_text(vjust = - 9, hjust= 0.025))+
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank())+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Site M04
m4<-allmse %>%
  filter(Site == "M04")

m4s<-m4%>%filter(Depth=="2")

```

```

cor4s<-cor.test(m4s$rsd,m4s$pwmean, method="spearman",na.rm=TRUE)
sr4 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor4s$estimate)

m4d<-m4%>%filter(Depth=="5")
cor4d<-cor.test(m4d$rsd,m4d$pwmean, method="spearman",na.rm=TRUE)
br4 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor4d$estimate)

p4<- m4%>% ggplot( aes(x=rsd, y=pwmean, linetype=Depth,group=Depth, shape=Depth)) +
  geom_line() +
  geom_point(size=2)+
  geom_errorbar(aes(ymin=pwmean-pwse, ymax=pwmean+pwse), width=.2,
    position=position_dodge(0.05))+
  scale_shape_manual(values=c(1, 16), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  scale_linetype_manual(values=c("dashed", "solid"), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  ggtitle("M04") +
  ylim(0,125)+
  scale_x_continuous(breaks = custom_breaks,
    labels = every_nth(custom_breaks, 4, inverse = TRUE))+
  geom_smooth(method="lm",color="black",se=FALSE,size=0.5)+
  theme_bw() +
  theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(colour = "#471164", size=5))+
  annotate("text", x=2020.5, y=118, label= "0-2cm") +
  annotate("text", x=2022, y=118, label= sr4, parse=TRUE) +
  annotate("text", x = 2020.5, y=103, label = "2-5cm")+
  annotate("text", x = 2022, y=103, label = br4, parse=TRUE)+
  theme(legend.position = "none")+
  theme(plot.title = element_text(vjust = - 9, hjust= 0.025))+
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank())+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Site M14
m14<-allmse %>%
  filter(Site == "M14")

m14s<-m14%>%filter(Depth=="2")
cor14s<-cor.test(m14s$rsd,m14s$pwmean, method="spearman",na.rm=TRUE)
sr14 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor14s$estimate)

m14d<-m14%>%filter(Depth=="5")
cor14d<-cor.test(m14d$rsd,m14d$pwmean, method="spearman",na.rm=TRUE)

```

```
br14 <- sprintf(
  "italic(R) ~ '=' ~ '%#.2f'",
  cor14d$estimate)

p14<- m14%>% ggplot( aes(x=rsd, y=pwmean, linetype=Depth,group=Depth, shape=Depth)) +
  geom_line() +
  geom_point(size=2)+
  geom_errorbar(aes(ymin=pwmean-pwse, ymax=pwmean+pwse), width=.2,
    position=position_dodge(0.05))+
  scale_shape_manual(values=c(1, 16), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+
  scale_linetype_manual(values=c("dashed", "solid"), breaks=c("2", "5"),
    labels=c( "0-2cm", "2-5cm"))+

  ggtitle("M14") +
  ylim(0,125)+
  scale_x_continuous(breaks = custom_breaks,
    labels = every_nth(custom_breaks, 4, inverse = TRUE))+
  geom_smooth(method="lm",color="black",se=FALSE,size=0.5)+
  theme_bw() +
  theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(colour = "#471164", size=5))+
  annotate("text", x=2020.5, y=118, label= "0-2cm") +
  annotate("text", x=2022, y=118, label= sr14, parse=TRUE) +
  annotate("text", x = 2020.5, y=103, label = "2-5cm")+
  annotate("text", x = 2022, y=103, label = br14, parse=TRUE)+
  theme(legend.position = "none")+
  theme(plot.title = element_text(vjust = - 9, hjust= 0.025))+
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank())+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

## Save multiple plots within one panel.

First, arrange the five River plots in one panel.

```
figure <- ggarrange(p15, p1,p2,p4,p14,
  ncol = 1)
```

Annotate the figure by adding a common y-axis label.

```
wc_riv<-annotate_figure(figure,
  left = text_grob("Sediment Water Content (%)", size=18, rot = 90))
```

Save the 5-plot panel as a jpg.

```
ggsave(
  "wc_Riv.jpg",#file name
  plot = wc_riv,#plot to save, defaults to last plot displayed
  device = "jpg",#file format (jpg, pdf, png, eps, etc)
```

```
path = NULL, #defaults to working directory
scale = 1,
width = 8.5,
height = 11,
units = "in", #c("in", "cm", "mm", "px")
dpi = 300,
limitsize = TRUE)
```



