MSc Project

SOBOYEJO LUKMAN ADEBOYE

01/06/2020

## Packages Installed —-

# pkgs <-  
 (c("forecast", "patchwork", "ggplot2", "plotly", "tidyverse", "pander", "ggthemes", "reshape", "cowplot"))

## [1] "forecast" "patchwork" "ggplot2" "plotly" "tidyverse" "pander"   
## [7] "ggthemes" "reshape" "cowplot"

# install.packages(pkgs)  
  
library(forecast) # For the moving avearage timeseries

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(patchwork) # For combining plots  
library(ggplot2) # For Visualization  
library(plotly) # For interactive plot

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(lubridate) #

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(knitr)  
library(tidyverse)

## -- Attaching packages --------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v tibble 3.0.0 v dplyr 0.8.5  
## v tidyr 1.1.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0  
## v purrr 0.3.4

## -- Conflicts ------------------------------------------------------------------------------------------------ tidyverse\_conflicts() --  
## x lubridate::as.difftime() masks base::as.difftime()  
## x lubridate::date() masks base::date()  
## x dplyr::filter() masks plotly::filter(), stats::filter()  
## x lubridate::intersect() masks base::intersect()  
## x dplyr::lag() masks stats::lag()  
## x lubridate::setdiff() masks base::setdiff()  
## x lubridate::union() masks base::union()

library(pander)  
library(ggthemes)  
library(reshape2)

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

library(cowplot)

##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Note: As of version 1.0.0, cowplot does not change the

## default ggplot2 theme anymore. To recover the previous

## behavior, execute:  
## theme\_set(theme\_cowplot())

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggthemes':  
##   
## theme\_map

## The following object is masked from 'package:lubridate':  
##   
## stamp

## The following object is masked from 'package:patchwork':  
##   
## align\_plots

## Theme —-

ann.theme <- function(){  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "plain",  
 size = 8),  
 axis.title = element\_text(face = "plain",  
 size = 8))  
}  
  
mon.theme <- function(){  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "bold",  
 size = 8),  
 axis.title = element\_text(face = "bold",  
 size = 8))  
}  
  
season.theme <- function(){  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "plain",  
 size = 8),  
 axis.title = element\_text(face = "plain",  
 size = 8))  
}  
  
season.summary <- function(){  
 # Add a red line to the mean  
 stat\_summary(aes(ymax = ..y.., ymin = ..y..),  
 fun = "mean",  
 geom = "pointrange",   
 # Use geom\_errorbar to add line as mean  
 color = "red",  
 position = position\_dodge(width = 0.75),   
 # Add the line to each group  
 show.legend = FALSE)  
}  
  
avg.theme <- function() {  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "plain",  
 size = 8),  
 axis.title = element\_text(face = "plain",  
 size = 8),  
 legend.position = "right")  
}  
  
budget.theme <- function() {  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "plain",  
 size = 8),  
 axis.title = element\_text(face = "plain",  
 size = 8),  
 legend.position = "bottom")  
}  
  
drought.theme <- function() {  
# ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "plain",  
 size = 8),  
 axis.title = element\_text(face = "plain",  
 size = 8),  
 legend.position = "bottom")  
}

## Read file into R —-

# Monthly Time series  
  
san.mon.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Sanvitale.R/SanVitale Basin/san.mon.csv")  
ras.mon.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Rasponi.R/Rasponi Basin/ras.mon.csv")  
quin.mon.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Fosso\_Ghiaia.R/Quinto Basin/quin.mon.csv")  
  
  
# Yearly Time series  
  
quin.ann.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Fosso\_Ghiaia.R/Quinto Basin/quin.ann.csv")  
ras.ann.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Rasponi.R/Rasponi Basin/ras.ann.csv")  
san.ann.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Sanvitale.R/SanVitale Basin/san.ann.csv")  
  
## Seasonal Average  
wat\_bal\_san <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Sanvitale.R/SanVitale Basin/wat\_bal\_san.csv")  
wat\_bal\_ras <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Rasponi.R/Rasponi Basin/wat\_bal\_ras.csv")  
wat\_bal\_quin <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Fosso\_Ghiaia.R/Quinto Basin/wat\_bal\_quin.csv")  
  
## Water Budget  
water\_budget <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/dat.csv")  
  
  
## Drought Values  
spi.data <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Code/Drought\_Index.csv")  
Drought\_Index <- read.csv("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Code/Drought\_Index.csv")  
  
## Subsidence  
subsidence <- read.delim("D:/Bologna Project/Land\_Reclamation\_Excel\_Data/subsidence.txt")  
  
# kable()  
# pander()

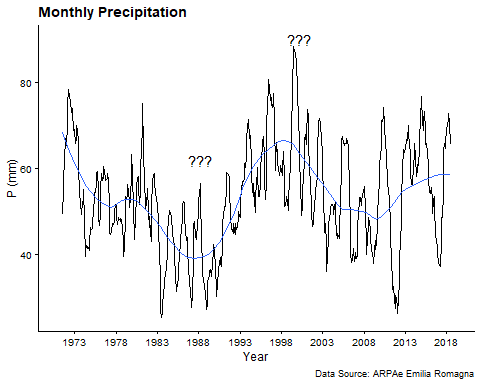
## Time series —-

link to learn more about ggthemes [link](https://jrnold.github.io/ggthemes/reference/index.html)

## Monthly Prec

## Date Conversion to POSXCIT  
newyear = parse\_date\_time(san.mon.data[,1], "by")  
date = as.Date.POSIXct(newyear)  
# # change to date format  
# mydate.ch <- format(date, "%b %Y")  
# # change to factor  
# mydate.fac <- factor(mydate.ch, unique(mydate.ch))  
  
# Data Manipulation for P  
# Time Series of precipitation  
rain.ts <- ts(san.mon.data$p, frequency = 12, start = c(1971,1))  
# moving average  
rain.ma = ma(rain.ts, order = 12, centre = T)  
# unclass rain moving average  
unclass.rain <- unclass(rain.ma)  
# dput  
  
  
  
# combined different type of object  
df.rain <- tibble(date, unclass.rain)  
  
  
  
# Data Visualization - ggplot2  
(ppt = ggplot(df.rain,  
 aes(x = date, y = unclass.rain)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE,  
 size = 0.5) +  
 labs(caption = "Data Source: ARPAe Emilia Romagna",  
 title = "Monthly Precipitation",  
 y = "P (mm)",  
 x = "Year",  
 tag = "") +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_labels = "%Y") +  
 theme\_cowplot(9) +  
 annotate(geom = "text",  
 x = as.Date("1988-04-01"),  
 y = 62,  
 label = "???",  
 hjust = 0.5,  
 size = 4) +  
 annotate(geom = "text",  
 x = as.Date("2000-04-01"),  
 y = 90,  
 label = "???",  
 hjust = 0.5,  
 size = 4)  
  
 )

## Warning: Removed 12 row(s) containing missing values (geom\_path).

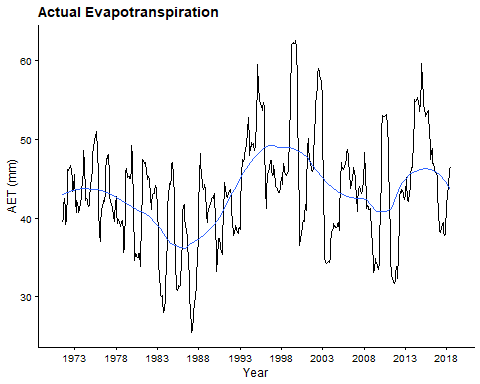


# annotate(geom = "segment", x = as.Date("1995-04-01"),  
 # y = 10, xend = as.Date("1990-04-01"),  
 # yend = 10,  
 # arrow = arrow(length = unit(2, "mm"))) +  
  
# ggsave(filename = "ppt.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Monthly AET

## Data Manipulation for PET  
# Time Series of AET  
aet.ts <- ts(san.mon.data$aet, frequency = 12, start = c(1971,1))  
# moving average  
aet.ma = ma(aet.ts, order = 12, centre = T)  
# unclass moving average  
unclass.aet <- unclass(aet.ma)  
# combined different type of object using tibble  
df.aet <- tibble(date, unclass.aet)  
  
## Data Visualization for AET  
(aet = ggplot(df.aet,  
 aes(x = date, y = unclass.aet)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE,  
 size = 0.5) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(title = "Actual Evapotranspiration",  
 y = "AET (mm)",  
 x = "Year",  
 tag = "") +  
 theme\_cowplot(9)   
)

## Warning: Removed 12 row(s) containing missing values (geom\_path).

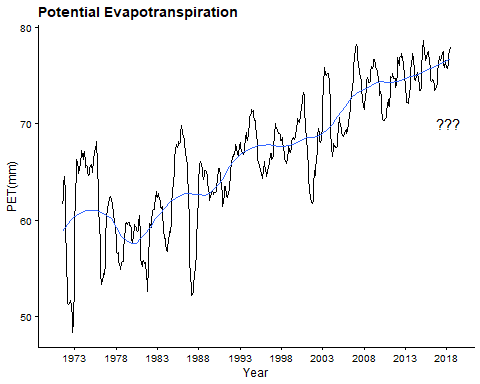


# ggsave(filename = "aet.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Monthly PET

## Data Manipulation for PET  
# Time Series of AET  
pet.ts <- ts(san.mon.data$pet, frequency = 12, start = c(1971,1))  
# moving average  
pet.ma = ma(pet.ts, order = 12, centre = T)  
# unclass moving average  
unclass.pet <- unclass(pet.ma)  
# combined different type of object using tibble  
df.pet <- tibble(date, unclass.rain)  
  
  
  
## Data Visualization for PET  
(pet = ggplot(df.pet,  
 aes(x = date, y = unclass.pet)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE,  
 size = 0.5) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(title = "Potential Evapotranspiration",  
 y = "PET(mm)",  
 x = "Year",  
 tag = "") +  
 theme\_cowplot(9) +  
 annotate(geom = "text",  
 x = as.Date("2018-04-01"),  
 y = 70,  
 label = "???",  
 hjust = 0.5,  
 size = 4)  
)

## Warning: Removed 12 row(s) containing missing values (geom\_path).



# ggsave(filename = "pet.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Combine Plt1 —-

plt1 <- plot\_grid(ppt, aet, pet, labels = c('A', 'B', 'C'), label\_size = 12,  
 ncol = 1, align = 'v', axis = 1)

## Warning: Removed 12 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 12 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 12 row(s) containing missing values (geom\_path).

ggsave(filename = "pplt.png",  
 dpi = 1200,  
 path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")

## Saving 5 x 4 in image

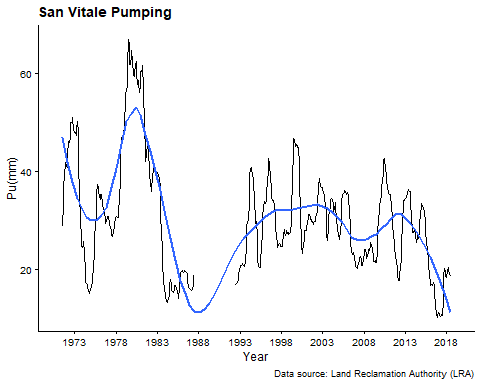
dev.off()

## null device   
## 1

## San Vitale Pu

## Data Manipulation for Pumping in San Vitale  
# Time series of Pumping  
san.pump.ts <- ts(san.mon.data$pu, frequency = 12, start = c(1971,1))  
# forecast using moving average  
san.pump.ma = ma(san.pump.ts, order = 12, centre = T)  
# combined different object to dataframe  
unclass.san.pump <- unclass(san.pump.ma)  
df.san.pump <- tibble(date, unclass.san.pump)  
  
  
## pumping in san vitale basin   
(san.pump = ggplot(df.san.pump,  
 aes(x = date, y = unclass.san.pump)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(caption = "Data source: Land Reclamation Authority (LRA)",  
 title = "San Vitale Pumping",  
 y = "Pu(mm)",  
 x = "Year",   
 tag = "") +  
 theme\_cowplot(9)   
 )

## Warning: Removed 12 row(s) containing missing values (geom\_path).

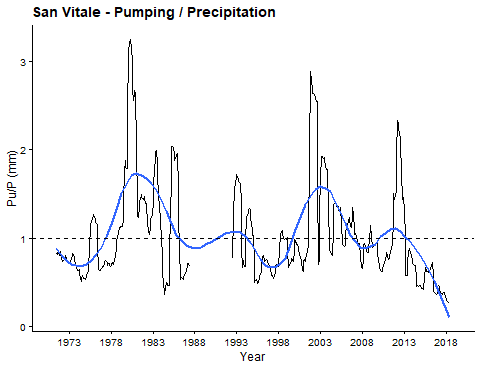


# ggsave(filename = "san.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## San Vitale Pu/P

## Data Manipulation for Pu/P  
# Time Series  
san.pup.ts <- ts(san.mon.data$pu.p, frequency = 12, start = c(1971,1))  
# forecast using moving average  
san.pup.ma = ma(san.pup.ts, order = 12, centre = T)  
unclass.san.pup <- unclass(san.pup.ma)  
# combined different object to data frame  
df.san.pup <- tibble(date, unclass.san.pup)  
  
  
  
## Visualization with ggplot  
(san.pup = ggplot(df.san.pup,  
 aes(x = date, y = unclass.san.pup)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(caption = "",  
 title = "San Vitale - Pumping / Precipitation",  
 y = "Pu/P (mm)",  
 x = "Year",   
 tag = "") +  
 theme\_cowplot(9) +  
 geom\_hline(yintercept = 1, linetype = "dashed")  
)

## Warning: Removed 12 row(s) containing missing values (geom\_path).

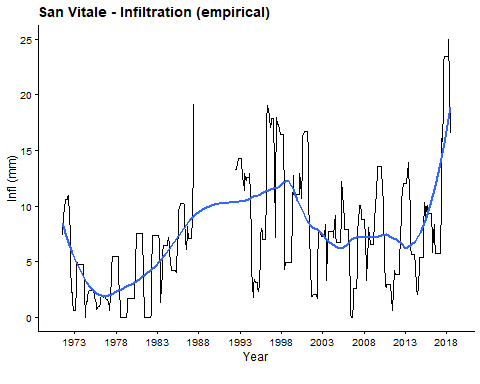


#   
# ggsave(filename = "san.pup.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## San Vitale Infl

# Data Manipulation for Infl  
# Time Series  
san.inf.ts <- ts(san.mon.data$infl, frequency = 12, start = c(1971,1))  
  
# forecast using moving average  
san.inf.ma = ma(san.inf.ts, order = 12, centre = T)  
unclass.san.inf <- unclass(san.inf.ma)  
# combined different object to data frame  
df.san.inf <- tibble(date, unclass.san.inf)  
  
## Infiltration in San Vitale  
(san.infl = ggplot(df.san.inf,  
 aes(x = date, y = unclass.san.inf)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(caption = "",  
 title = "San Vitale - Infiltration (empirical)",  
 y = "Infl (mm)",  
 x = "Year",   
 tag = "") +  
 theme\_cowplot(9)  
)

## Warning: Removed 12 row(s) containing missing values (geom\_path).

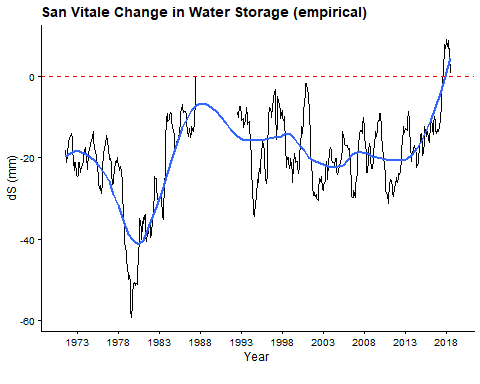


# ggsave(filename = "san.infl.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## San Vitale dS

# Data Manipulation for storage  
# Time Series  
san.ds.ts <- ts(san.mon.data$ds, frequency = 12, start = c(1971,1))  
# forecast using moving average  
san.ds.ma = ma(san.ds.ts, order = 12, centre = T)  
  
# combined different object to data frame  
unclass.san.ds <- unclass(san.ds.ma)  
df.san.ds <- tibble(date, unclass.san.ds)  
  
## Data Visualization with ggplot2  
(san.ds = ggplot(df.san.ds,  
 aes(x = date, y = unclass.san.ds)) +  
 geom\_line() +  
 geom\_smooth(aes(x = date, y = unclass.san.ds),  
 method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(caption = "",  
 title = "San Vitale Change in Water Storage (empirical)",  
 y = "dS (mm)",  
 x = "Year",   
 tag = "") +  
 theme\_cowplot(9) +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red")  
)

## Warning: Removed 12 row(s) containing missing values (geom\_path).

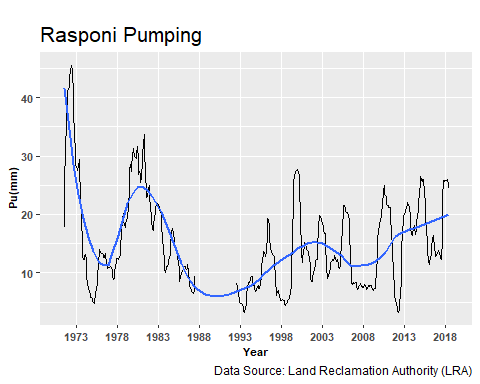


# ggsave(filename = "san.ds.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi Pu

## Data Manipulation  
# Time series  
ras.pump.ts <- ts(ras.mon.data$pu, frequency = 12, start = c(1971,1))  
# forecast using moving average  
ras.pump.ma = ma(ras.pump.ts, order = 12, centre = T)  
# combined different object to dataframe  
unclass.ras.pump <- unclass(ras.pump.ma)  
df.ras.pump <- tibble(date, unclass.ras.pump)  
  
  
## pumping in san vitale basin   
(san.pump = ggplot(df.ras.pump,  
 aes(x = date, y = unclass.ras.pump)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 labs(caption = "Data Source: Land Reclamation Authority (LRA)",  
 title = "Rasponi Pumping",  
 y = "Pu(mm)",  
 x = "Year",   
 tag = "") +  
 mon.theme()  
)

## Warning: Removed 12 row(s) containing missing values (geom\_path).

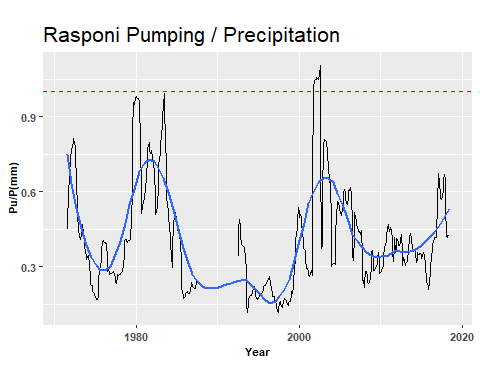


# ggsave(filename = "ras.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi Pu / P

## Data Manipulation  
# Times series  
ras.pup.ts <- ts(ras.mon.data$pu.p, frequency = 12, start = c(1971,1))  
# forecast using moving average  
ras.pup.ma = ma(ras.pup.ts, order = 12, centre = T)  
  
# combined different object to data frame  
unclass.ras.pup <- unclass(ras.pup.ma)  
df.ras.pup <- tibble(date, unclass.ras.pup)  
  
  
## Pumping over Precipitation  
(ras.pup = ggplot(df.ras.pup,  
 aes(x = date, y = unclass.ras.pup)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,  
 na.rm = TRUE) +  
 labs(caption = "",  
 title = "Rasponi Pumping / Precipitation",  
 y = "Pu/P(mm)",  
 x = "Year",   
 tag = "") +  
 mon.theme() +  
 geom\_hline(yintercept = 1, linetype = "dashed", color = "red"))

## Warning: Removed 12 row(s) containing missing values (geom\_path).

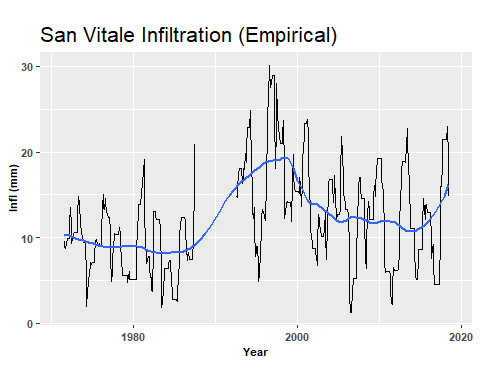


# ggsave(filename = "ras.pup.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi Infl

## Data Manipulation  
# Time Series  
ras.inf.ts <- ts(ras.mon.data$infl, frequency = 12, start = c(1971,1))  
# forecast using moving average  
ras.inf.ma = ma(ras.inf.ts, order = 12, centre = T)  
# combined different object to data frame  
unclass.ras.inf <- unclass(ras.inf.ma)  
df.ras.inf <- tibble(date, unclass.ras.inf)  
  
  
## Infiltration  
(ras.infl = ggplot(df.ras.inf,  
 aes(x = date, y = unclass.ras.inf)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 labs(caption = "",  
 title = "San Vitale Infiltration (Empirical)",  
 y = "Infl (mm)",  
 x = "Year",   
 tag = "") +  
 mon.theme())

## Warning: Removed 12 row(s) containing missing values (geom\_path).

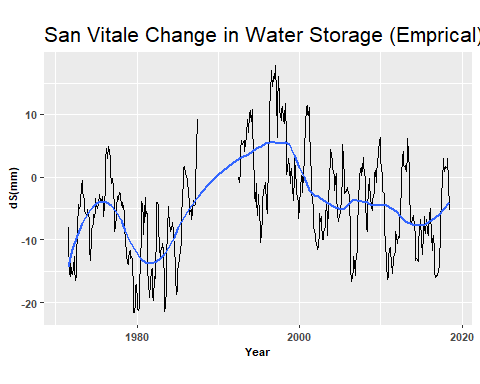


# ggsave(filename = "ras.infl.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi dS

## Data Manipulation  
# Time series  
ras.ds.ts <- ts(ras.mon.data$ds, frequency = 12, start = c(1971,1))  
# forecast using moving average  
ras.ds.ma = ma(ras.ds.ts, order = 12, centre = T)  
unclass.ras.ds <- unclass(ras.ds.ma)  
  
# combined different object to data frame  
df.ras.ds <- tibble(date, unclass.ras.ds)  
  
  
## Infiltration  
(ras.ds = ggplot(df.ras.ds,  
 aes(x = date, y = unclass.ras.ds)) +  
 geom\_line() +  
 geom\_smooth(aes(x = date, y = unclass.ras.ds),  
 method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 labs(caption = "",  
 title = "San Vitale Change in Water Storage (Emprical)",  
 y = "dS(mm)",  
 x = "Year",   
 tag = "") +   
 mon.theme())

## Warning: Removed 12 row(s) containing missing values (geom\_path).

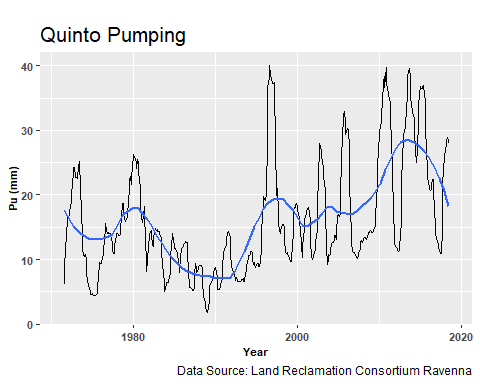


# ggsave(filename = "ras.ds.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto P

## Data Manipulation  
# Time Series  
quin.pump.ts <- ts(quin.mon.data$pu, frequency = 12, start = c(1971,1))  
# forecast using moving average  
quin.pump.ma = ma(quin.pump.ts, order = 12, centre = T)  
# combined different object to data frame  
unclass.quin.pump <- unclass(quin.pump.ma)  
df.quin.pump <- tibble(date, unclass.quin.pump)  
  
## pumping   
(quin.pump = ggplot(df.quin.pump,  
 aes(x = date, y = unclass.quin.pump)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 labs(caption = "Data Source: Land Reclamation Consortium Ravenna",  
 title = "Quinto Pumping",  
 y = "Pu (mm)",  
 x = "Year",   
 tag = "") +  
 mon.theme())

## Warning: Removed 12 row(s) containing missing values (geom\_path).

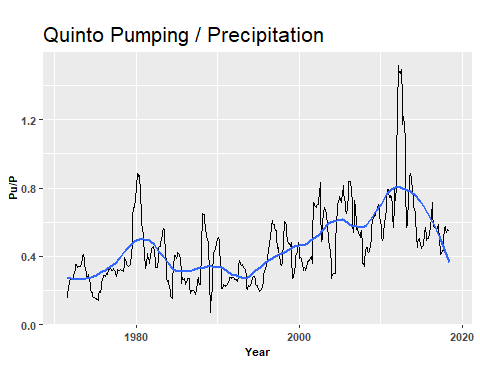


# ggsave(filename = "quin.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto Pu/P

## Data Manipulation  
# Time series  
quin.pup.ts <- ts(quin.mon.data$pu.p, frequency = 12, start = c(1971,1))  
# forecast using moving average  
quin.pup.ma = ma(quin.pup.ts, order = 12, centre = T)  
# combined different object to data frame  
unclass.quin.pup <- unclass(quin.pup.ma)  
df.quin.pup <- tibble(date, unclass.quin.pup)  
  
## Data Visualization with ggplot2  
(quin.pup = ggplot(df.quin.pup,  
 aes(x = date, y = unclass.quin.pup)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 labs(caption = "",  
 title = "Quinto Pumping / Precipitation",  
 y = "Pu/P",  
 x = "Year",   
 tag = "") +  
 mon.theme())

## Warning: Removed 12 row(s) containing missing values (geom\_path).

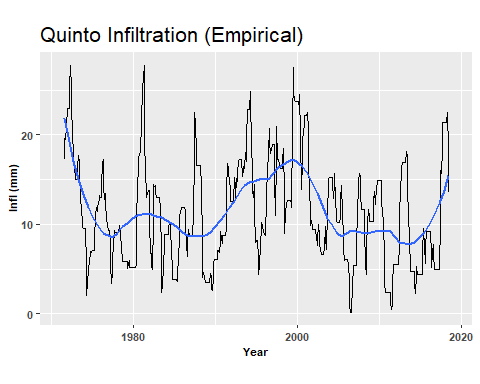


# ggsave(filename = "quin.pup.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto Infl

## Data Manipulation   
# Time Series  
quin.inf.ts <- ts(quin.mon.data$infl, frequency = 12, start = c(1971,1))  
# forecast using moving average  
quin.inf.ma = ma(quin.inf.ts, order = 12, centre = T)  
# unclass the time series  
unclass.quin.inf <- unclass(quin.inf.ma)  
# combined different object to data frame  
df.quin.inf <- tibble(date, unclass.quin.inf)  
  
  
## Data Visualization with ggplot2  
(quin.infl = ggplot(df.quin.inf,  
 aes(x = date, y = unclass.quin.inf)) +  
 geom\_line() +  
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 labs(caption = "",  
 title = "Quinto Infiltration (Empirical)",  
 y = "Infl (mm)",  
 x = "Year",   
 tag = "") +  
 mon.theme())

## Warning: Removed 12 row(s) containing missing values (geom\_path).

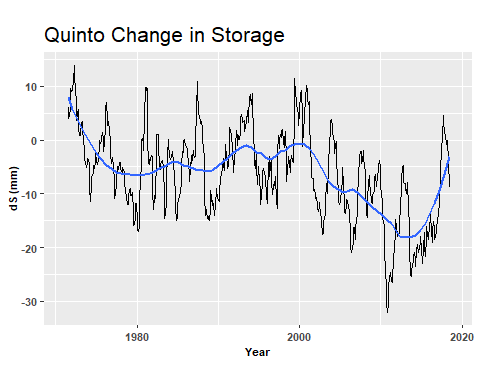


# ggsave(filename = "quin.infl.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto dS

## Data Manipulation   
# Time series  
quin.ds.ts <- ts(quin.mon.data$ds, frequency = 12, start = c(1971,1))  
# forecast using moving average  
quin.ds.ma = ma(quin.ds.ts, order = 12, centre = T)  
# combined different object to data frame  
unclass.quin.ds <- unclass(quin.ds.ma)  
df.quin.ds <- tibble(date, unclass.quin.ds)  
  
  
## Data Visualization with ggplot2  
(quin.ds = ggplot(df.quin.ds,  
 aes(x = date, y = unclass.quin.ds)) +  
 geom\_line() +  
 geom\_smooth(aes(x = date, y = unclass.quin.ds),  
 method = "loess",  
 formula = y ~ x,  
 span = 0.3,  
 level = 0.95,  
 se = FALSE,   
 na.rm = TRUE) +  
 labs(caption = "",  
 title = "Quinto Change in Storage",  
 y = "dS (mm)",  
 x = "Year",   
 tag = "") +  
 mon.theme())

## Warning: Removed 12 row(s) containing missing values (geom\_path).

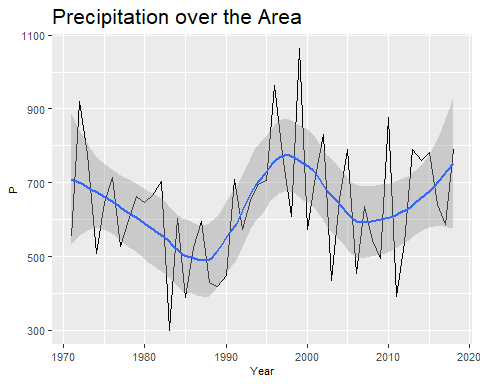


# ggsave(filename = "quin.ds.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

# Modeling —-

# Annual P

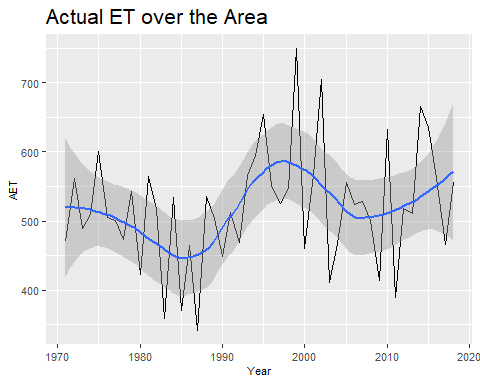
(rain.ann = ggplot(data = ras.ann.data,   
 mapping = aes(x = year, y = p)) +  
 geom\_line() +   
 geom\_smooth(method = "loess",   
 formula = y ~ x,   
 span = 0.5,   
 level = 0.95,   
 se = TRUE,   
 na.rm = FALSE)+  
 labs(x = "Year",   
 y = "P",   
 title = "Precipitation over the Area") +  
 ann.theme())



# ggsave(filename = "rain.ann.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Annual AET

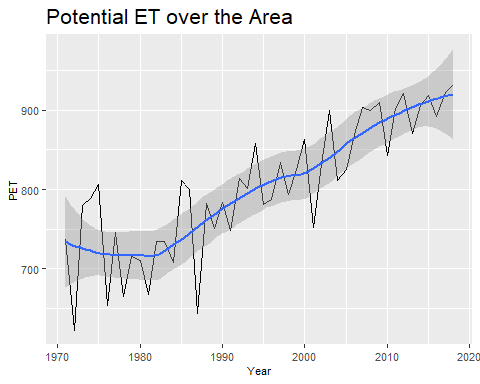
(aet.ann = ggplot(data = ras.ann.data,   
 mapping = aes(x = year, y = aet)) +  
 geom\_line() +   
 geom\_smooth(method = "loess",   
 formula = y ~ x,   
 span = 0.5,   
 level = 0.95, se = TRUE,   
 na.rm = FALSE) +  
 labs(x = "Year",   
 y = "AET",   
 title = "Actual ET over the Area") +  
 ann.theme())



# ggsave(filename = "aet.ann.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Annual PET

(pet = ggplot(data = ras.ann.data,   
 mapping = aes(x = year, y = pet)) +  
 geom\_line() +   
 geom\_smooth(method = "loess",   
 formula = y ~ x,   
 span = 0.5,   
 level = 0.95,   
 se = TRUE,   
 na.rm = FALSE) +  
 labs(x = "Year",   
 y = "PET",   
 title = "Potential ET over the Area") +  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 ann.theme())

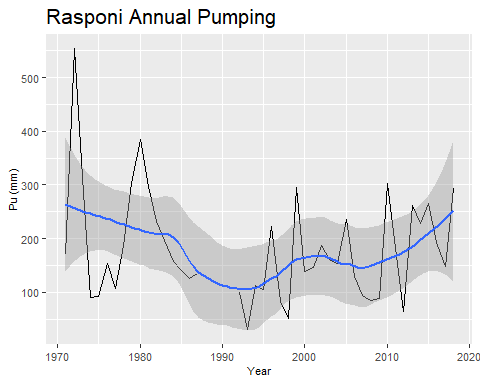


# ggsave(filename = "pet.ann.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

# Annual Rasponi

## Visualization with ggplot  
(ras.ann.pump = ggplot(data = ras.ann.data,   
 mapping = aes(x = year, y = pu)) +  
 geom\_line() +   
 geom\_smooth(method = "loess",   
 formula = y ~ x,   
 span = 0.5,   
 level = 0.96,   
 se = TRUE,   
 na.rm = FALSE)+  
 labs(x = "Year",   
 y = "Pu (mm)",   
 title = "Rasponi Annual Pumping") +  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 ann.theme())

## Warning: Removed 4 rows containing non-finite values (stat\_smooth).

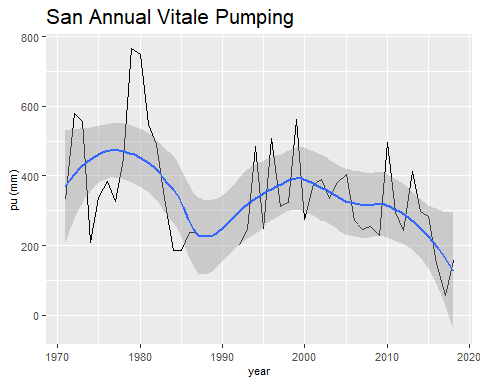


# ggsave(filename = "ras.ann.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Annual San Vitale

(san.ann.pump = ggplot(data = san.ann.data,  
 mapping = aes(x = year, y = pu)) +  
 geom\_line() +   
 geom\_smooth(method = "loess",   
 formula = y ~ x,   
 span = 0.5,   
 level = 0.95,   
 se = TRUE,   
 na.rm = FALSE) +  
 labs(x = "year",   
 y = "pu (mm)",   
 title = "San Annual Vitale Pumping") +  
 ann.theme())

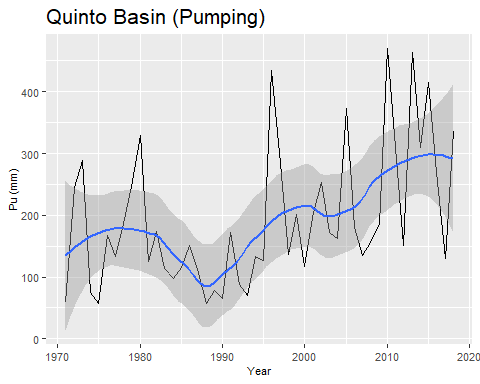
## Warning: Removed 4 rows containing non-finite values (stat\_smooth).



# ggsave(filename = "san.ann.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Annual Quinto

(quin.ann.pump = ggplot(data = quin.ann.data,  
 mapping = aes(x = year, y = pu)) +  
 geom\_line() +   
 geom\_smooth(method = "loess",  
 formula = y ~ x,  
 span = 0.5,   
 level = 0.95,  
 se = TRUE,   
 na.rm = FALSE) +  
 labs(x = "Year",   
 y = "Pu (mm)",   
 title = "Quinto Basin (Pumping)") +  
 # ## ggthemes  
 # theme\_solarized\_2(base\_size = 10, light = FALSE) +  
 theme(plot.title = element\_text(hjust=,  
 face = "plain",  
 size = 16),  
 axis.text = element\_text(face = "plain",  
 size = 8),  
 axis.title = element\_text(face = "plain",  
 size = 8)))

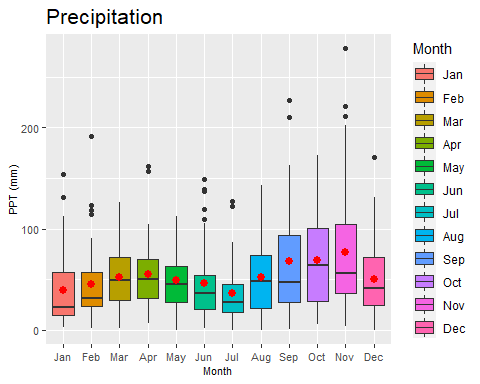


# ggsave(filename = "quin.ann.pump.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Seasonality —-

## Rainfall

# select the column  
rain <- san.mon.data[,2]  
# Combined Observations into data frame  
df.rain <- tibble(date, rain)  
  
season.rain <- df.rain %>%  
 # Convert data frame from lwide format to long format  
 gather(season.rain, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.rain = str\_replace(season.rain, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.rain$Month = factor(season.rain$Month, levels = month.abb)  
  
# Visualization with ggplot  
(ppt = ggplot(season.rain, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "PPT (mm)",  
 title = "Precipitation") +  
 season.theme() +  
 season.summary())

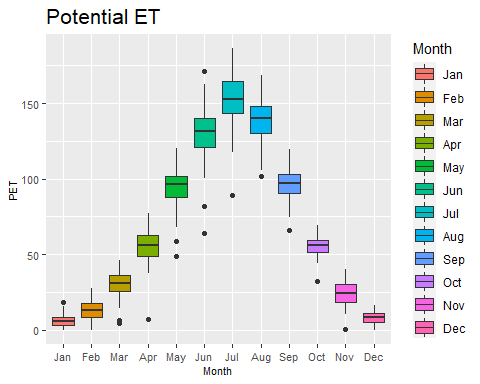


ggplotly(ppt)

# ggsave(filename = "mm.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## PET

# select the column  
pet <- san.mon.data[,3]  
# Combined Observations into data frame  
df.pet <- tibble(date, pet)  
  
season.pet <- df.pet %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pet, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pet = str\_replace(season.pet, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pet$Month = factor(season.pet$Month, levels = month.abb)  
  
# Visualization with ggplot  
(pet = ggplot(season.pet, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "PET",  
 title = "Potential ET") +  
 season.theme() +  
 season.theme())

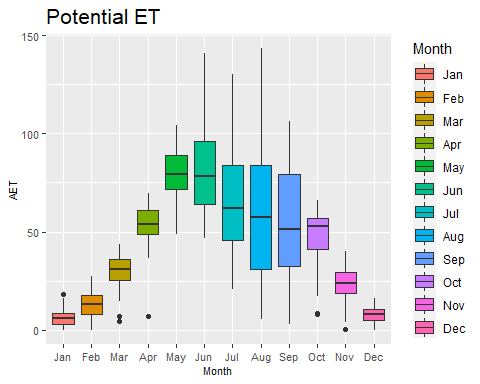


ggplotly(pet)

# ggsave(filename = "mm.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## AET

# select the column  
aet <- san.mon.data[,4]  
# Combined Observations into data frame  
df.aet <- tibble(date, aet)  
  
season.aet <- df.aet %>%  
 # Convert data frame from lwide format to long format  
 gather(season.aet, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.aet = str\_replace(season.aet, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.aet$Month = factor(season.aet$Month, levels = month.abb)  
  
# Visualization with ggplot  
(aet = ggplot(season.aet, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "AET",  
 title = "Potential ET") +  
 season.theme() +  
 season.theme())

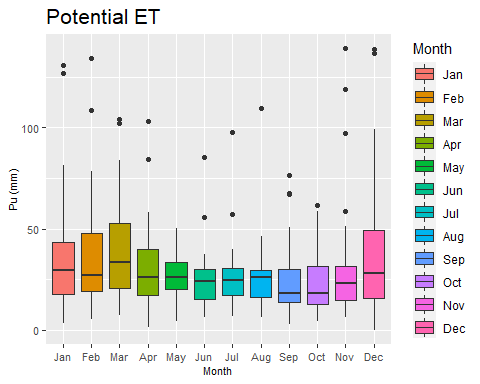


ggplotly(aet)

# ggsave(filename = "mm.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## San Vitale Pu

# select the column  
pu <- san.mon.data[,5]  
# Combined Observations into data frame  
df.pu <- tibble(date, pu)  
  
season.pu <- df.pu %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pu, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pu = str\_replace(season.pu, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pu$Month = factor(season.pu$Month, levels = month.abb)  
  
# Visualization with ggplot  
(san\_pu <- ggplot(season.pu, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Pu (mm)",  
 title = "Potential ET") +  
 season.theme() +  
 season.theme())

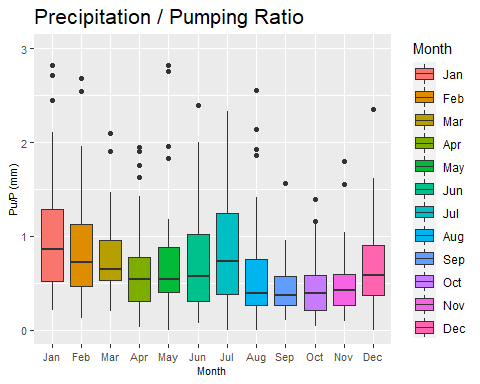


ggplotly(san\_pu)

# ggsave(filename = ".png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## SanVitale Pu/P

# select the column  
pup <- san.mon.data[,6]  
# Combined Observations into data frame  
df.pup <- tibble(date, pup)  
  
season.pup <- df.pup %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pup, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pup = str\_replace(season.pup, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pup$Month = factor(season.pup$Month, levels = month.abb)  
  
# Visualization with ggplot  
(san\_pup = ggplot(season.pup, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Pu/P (mm)",  
 title = "Precipitation / Pumping Ratio") +  
 season.theme() +  
 season.theme() +  
 scale\_y\_continuous(limits = c(0, 3)))

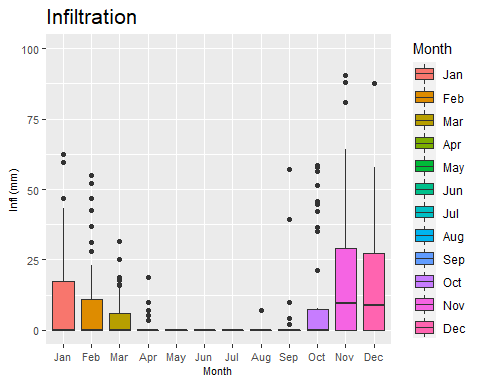


ggplotly(san\_pup)

# ggsave(filename = "pup.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## San Vitale Infl

# select the column  
infl <- san.mon.data[,7]  
# Combined Observations into data frame  
df.infl <- tibble(date, infl)  
  
season.infl <- df.infl %>%  
 # Convert data frame from lwide format to long format  
 gather(season.infl, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.infl = str\_replace(season.infl, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.infl$Month = factor(season.infl$Month, levels = month.abb)  
  
# Visualization with ggplot  
(san\_infl = ggplot(season.infl, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Infl (mm)",  
 title = "Infiltration") +  
 season.theme() +  
 season.theme() +   
 scale\_y\_continuous(limits = c(0, 100)))

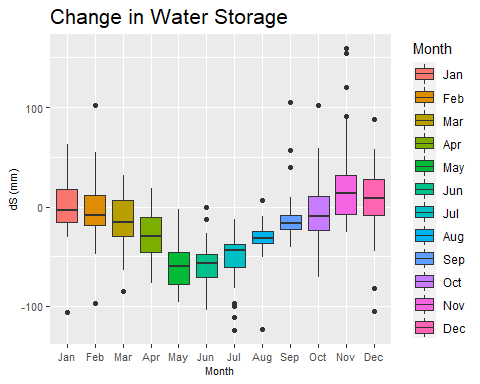


ggplotly(san\_infl)

# ggsave(filename = "pu.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## San Vitale dS

# select the column  
dS <- san.mon.data[,8]  
# Combined Observations into data frame  
df.dS <- tibble(date, dS)  
  
season.dS <- df.dS %>%  
 # Convert data frame from lwide format to long format  
 gather(season.dS, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.dS = str\_replace(season.dS, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.dS$Month = factor(season.dS$Month, levels = month.abb)  
  
# Visualization with ggplot  
(san\_dS = ggplot(season.dS, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "dS (mm)",  
 title = "Change in Water Storage") +  
 season.theme() +  
 season.theme())

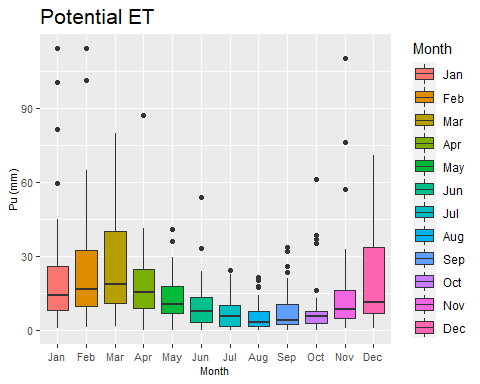


ggplotly(san\_dS)

# ggsave(filename = "pu.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi Pu

# select the column  
pu <- ras.mon.data[,5]  
# Combined Observations into data frame  
df.pu <- tibble(date, pu)  
  
season.pu <- df.pu %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pu, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pu = str\_replace(season.pu, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pu$Month = factor(season.pu$Month, levels = month.abb)  
  
# Visualization with ggplot  
(ras\_pu <- ggplot(season.pu, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Pu (mm)",  
 title = "Potential ET") +  
 season.theme() +  
 season.theme())

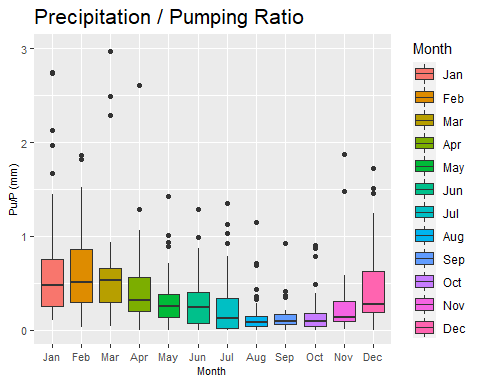


ggplotly(ras\_pu)

# ggsave(filename = ".png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi Pu/P

# select the column  
pup <- ras.mon.data[,6]  
# Combined Observations into data frame  
df.pup <- tibble(date, pup)  
  
season.pup <- df.pup %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pup, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pup = str\_replace(season.pup, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pup$Month = factor(season.pup$Month, levels = month.abb)  
  
# Visualization with ggplot  
(ras\_pup = ggplot(season.pup, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Pu/P (mm)",  
 title = "Precipitation / Pumping Ratio") +  
 season.theme() +  
 season.theme() +  
 scale\_y\_continuous(limits = c(0, 3)))

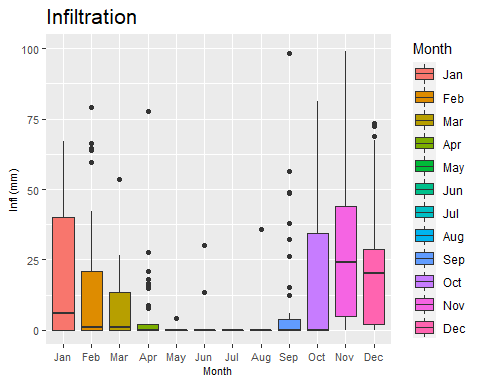


ggplotly(ras\_pup)

# ggsave(filename = "pup.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi Infl

# select the column  
infl <- ras.mon.data[,7]  
# Combined Observations into data frame  
df.infl <- tibble(date, infl)  
  
season.infl <- df.infl %>%  
 # Convert data frame from lwide format to long format  
 gather(season.infl, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.infl = str\_replace(season.infl, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.infl$Month = factor(season.infl$Month, levels = month.abb)  
  
# Visualization with ggplot  
(ras\_infl = ggplot(season.infl, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Infl (mm)",  
 title = "Infiltration") +  
 season.theme() +  
 season.theme() +   
 scale\_y\_continuous(limits = c(0, 100)))

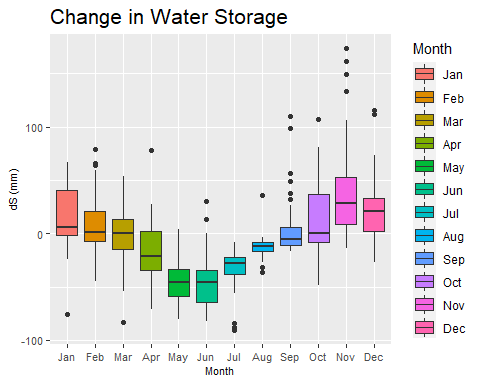


ggplotly(ras\_infl)

# ggsave(filename = "pu.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Rasponi dS

# select the column  
dS <- ras.mon.data[,8]  
# Combined Observations into data frame  
df.dS <- tibble(date, dS)  
  
season.dS <- df.dS %>%  
 # Convert data frame from lwide format to long format  
 gather(season.dS, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.dS = str\_replace(season.dS, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.dS$Month = factor(season.dS$Month, levels = month.abb)  
  
# Visualization with ggplot  
(ras\_dS = ggplot(season.dS, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "dS (mm)",  
 title = "Change in Water Storage") +  
 season.theme() +  
 season.theme())

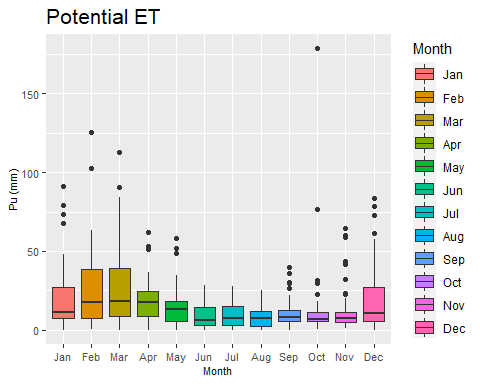


ggplotly(ras\_dS)

# ggsave(filename = "pu.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto Pu

# select the column  
pu <- quin.mon.data[,5]  
# Combined Observations into data frame  
df.pu <- tibble(date, pu)  
  
season.pu <- df.pu %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pu, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pu = str\_replace(season.pu, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pu$Month = factor(season.pu$Month, levels = month.abb)  
  
# Visualization with ggplot  
(quin\_pu <- ggplot(season.pu, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Pu (mm)",  
 title = "Potential ET") +  
 season.theme() +  
 season.theme())

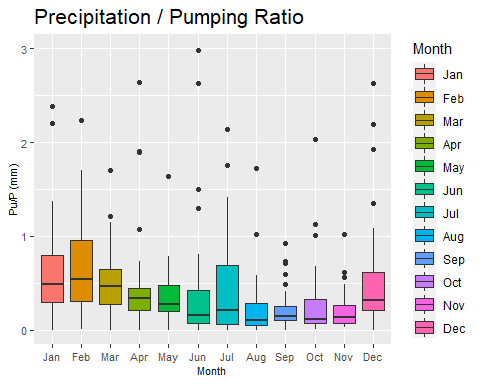


ggplotly(ras\_pu)

# ggsave(filename = ".png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto Pu/P

# select the column  
pup <- quin.mon.data[,6]  
# Combined Observations into data frame  
df.pup <- tibble(date, pup)  
  
season.pup <- df.pup %>%  
 # Convert data frame from lwide format to long format  
 gather(season.pup, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.pup = str\_replace(season.pup, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.pup$Month = factor(season.pup$Month, levels = month.abb)  
  
# Visualization with ggplot  
(quin\_pup = ggplot(season.pup, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Pu/P (mm)",  
 title = "Precipitation / Pumping Ratio") +  
 season.theme() +  
 season.theme() +  
 scale\_y\_continuous(limits = c(0, 3)))

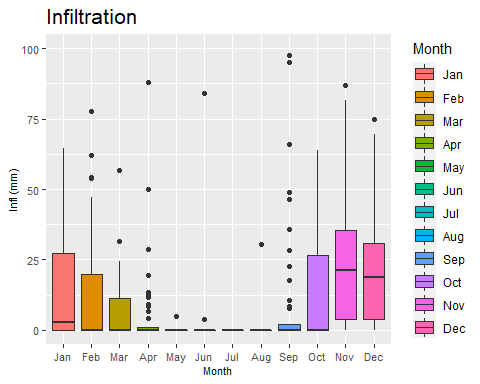


ggplotly(quin\_pup)

# ggsave(filename = "pup.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto Infl

# select the column  
infl <- quin.mon.data[,7]  
# Combined Observations into data frame  
df.infl <- tibble(date, infl)  
  
season.infl <- df.infl %>%  
 # Convert data frame from lwide format to long format  
 gather(season.infl, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.infl = str\_replace(season.infl, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.infl$Month = factor(season.infl$Month, levels = month.abb)  
  
# Visualization with ggplot  
(quin\_infl = ggplot(season.infl, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "Infl (mm)",  
 title = "Infiltration") +  
 season.theme() +  
 season.theme() +   
 scale\_y\_continuous(limits = c(0, 100)))

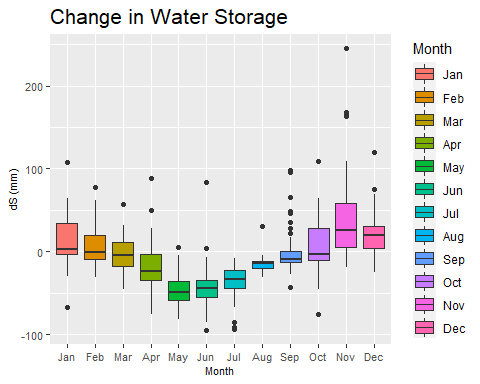


ggplotly(quin\_infl)

# ggsave(filename = "pu.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Quinto dS

# select the column  
dS <- quin.mon.data[,8]  
# Combined Observations into data frame  
df.dS <- tibble(date, dS)  
  
season.dS <- df.dS %>%  
 # Convert data frame from lwide format to long format  
 gather(season.dS, value, -date) %>%  
 # Remove "obs" in the Observation column  
 mutate(season.dS = str\_replace(season.dS, "date", "")) %>%  
 # Convert the DATE column to date class  
 mutate(DATE = ymd(date)) %>%  
 # Create Month column  
 mutate(Month = month(date)) %>%  
 # Create Season column  
 mutate(Month = case\_when(  
 Month %in% c(1) ~ "Jan",  
 Month %in% c(2) ~ "Feb",  
 Month %in% c(3) ~ "Mar",  
 Month %in% c(4) ~ "Apr",  
 Month %in% c(5) ~ "May",  
 Month %in% c(6) ~ "Jun",  
 Month %in% c(7) ~ "Jul",  
 Month %in% c(8) ~ "Aug",  
 Month %in% c(9) ~ "Sep",  
 Month %in% c(10) ~ "Oct",  
 Month %in% c(11) ~ "Nov",  
 Month %in% c(12) ~ "Dec",  
 TRUE ~ NA\_character\_  
 ))  
  
# Convert to factor  
season.dS$Month = factor(season.dS$Month, levels = month.abb)  
  
# Visualization with ggplot  
(quin\_dS = ggplot(season.dS, aes(x = Month, y = value,  
 fill = Month,   
 class = Month)) +  
 # Specify the geom to be boxplot  
 geom\_boxplot() +  
 labs(x = "Month",  
 y = "dS (mm)",  
 title = "Change in Water Storage") +  
 season.theme() +  
 season.theme())



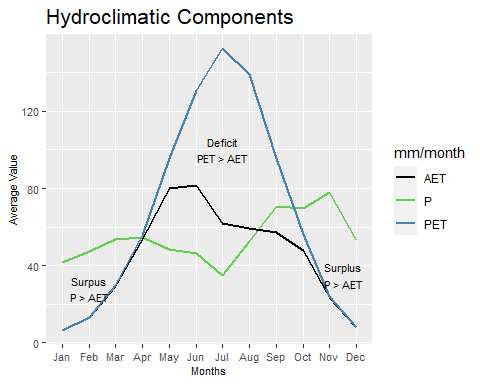
ggplotly(quin\_dS)

# ggsave(filename = "pu.png",  
# dpi = 1200, height = 4,  
# width = 8,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()

## Seasonal AVG —-

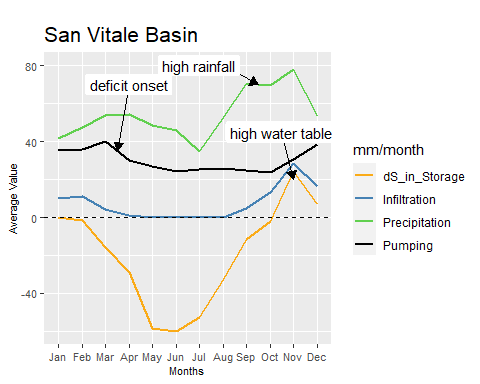
## Hydroclimate

wat\_bal\_san$mont = factor(wat\_bal\_san$Mon, levels = month.abb)  
  
(san\_avg <- ggplot(wat\_bal\_san)+  
 geom\_line(aes(mont, P, group = 1, color = "P"),  
 size = 1) +  
 geom\_line(aes(mont, AET, group = 1, color = "AET"),  
 size = 1) +  
 geom\_line(aes(mont, PET, group = 1, color = "PET"),  
 size = 1) +  
 scale\_color\_manual(name = "mm/month",  
 values=c("P" = "307FE2",  
 "AET" = "1E3F63",   
 "PET" ="steelblue")) +  
 labs(x = "Months",  
 y = "Average Value",  
 title = "Hydroclimatic Components") +  
 avg.theme() +   
 annotate("text", x = c(2,7, 11.5),  
 y = c(28,100, 35),  
 label = c("Surpus\nP > AET", "Deficit\nPET > AET", "Surplus\nP > AET"), size = 3))

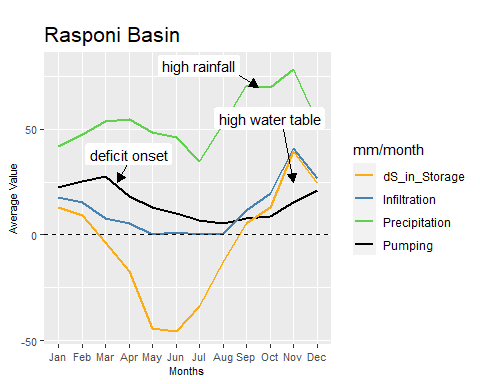


## San Vitale

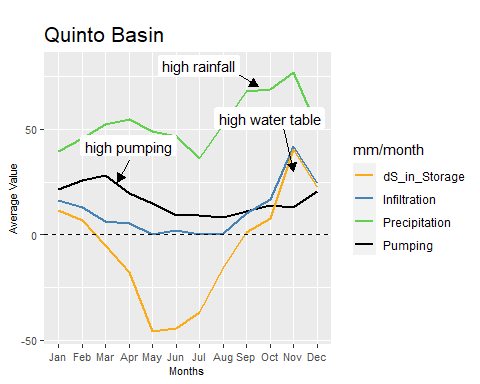
(ggplot(wat\_bal\_san)+  
 geom\_line(aes(mont, P, group = 1, color = "Precipitation"),  
 size = 1) +  
 geom\_line(aes(mont, Pu, group = 1, color = "Pumping"),  
 size = 1) +  
 geom\_line(aes(mont, Infl, group = 1, color = "Infiltration"),  
 size = 1)+  
 geom\_line(aes(mont, dS, group = 1, color = "dS\_in\_Storage"),  
 size = 1)+  
 scale\_color\_manual(name = "mm/month",  
 values=c("Precipitation" = "307FE2",  
 "Pumping" = "1E3F63",   
 "Infiltration" ="steelblue",  
 "dS\_in\_Storage" = "#FAAB18")) +  
 avg.theme() +  
 labs(x = "Months",   
 y = "Average Value",   
 title = "San Vitale Basin",  
 tag = "")+  
 annotate(geom = "segment",   
 x = c(10.5,4,8),   
 xend = c(11,3.5,9.5),   
 y = c(45,70, 80),   
 yend = c(20,35, 70),   
 arrow = arrow(type = "closed",  
 length = unit(0.10, "inches")))+  
 geom\_label(aes(x = 10.5, y = 45,   
 label = "high water table"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_label(aes(x = 4, y = 70,   
 label = "deficit onset"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_label(aes(x = 7, y = 80,   
 label = "high rainfall"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_hline(yintercept = 0,   
 linetype="dashed"))

 ## Rasponi

wat\_bal\_ras$mont = factor(wat\_bal\_ras$Mon, levels = month.abb)  
  
ggplot(wat\_bal\_ras)+  
 geom\_line(aes(mont, P, group = 1, color = "Precipitation"),  
 size = 1) +  
 geom\_line(aes(mont, Pu, group = 1, color = "Pumping"),  
 size = 1) +  
 geom\_line(aes(mont, Infl, group = 1, color = "Infiltration"),  
 size = 1)+  
 geom\_line(aes(mont, dS, group = 1, color = "dS\_in\_Storage"),  
 size = 1)+  
 scale\_color\_manual(name = "mm/month",  
 values=c("Precipitation" = "307FE2",  
 "Pumping" = "1E3F63",   
 "Infiltration" ="steelblue",  
 "dS\_in\_Storage" = "#FAAB18")) +  
 avg.theme() +  
 labs(x = "Months",   
 y = "Average Value",   
 title = "Rasponi Basin",  
 tag = "")+  
 annotate(geom = "segment",   
 x = c(10.5,4,8),   
 xend = c(11,3.5,9.5),   
 y = c(55,35, 80),   
 yend = c(25,25, 70),   
 arrow = arrow(type = "closed",  
 length = unit(0.10, "inches")))+  
 geom\_label(aes(x = 10, y = 55,   
 label = "high water table"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_label(aes(x = 4, y = 38,   
 label = "deficit onset"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_label(aes(x = 7, y = 80,   
 label = "high rainfall"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_hline(yintercept = 0,   
 linetype="dashed")

 ## Quinto

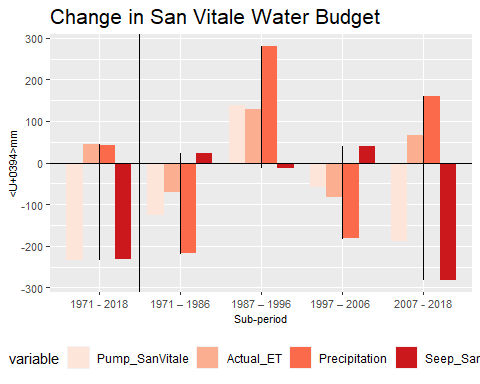
wat\_bal\_quin$mont = factor(wat\_bal\_quin$Mon, levels = month.abb)  
  
ggplot(wat\_bal\_quin)+  
 geom\_line(aes(mont, P, group = 1, color = "Precipitation"),  
 size = 1) +  
 geom\_line(aes(mont, Pu, group = 1, color = "Pumping"),  
 size = 1) +  
 geom\_line(aes(mont, Infl, group = 1, color = "Infiltration"),  
 size = 1)+  
 geom\_line(aes(mont, dS, group = 1, color = "dS\_in\_Storage"),  
 size = 1)+  
 scale\_color\_manual(name = "mm/month",  
 values=c("Precipitation" = "307FE2",  
 "Pumping" = "1E3F63",   
 "Infiltration" ="steelblue",  
 "dS\_in\_Storage" = "#FAAB18")) +  
 avg.theme() +   
 labs(x = "Months",   
 y = "Average Value",   
 title = "Quinto Basin",  
 tag = "")+  
 annotate(geom = "segment",   
 x = c(10.5,4,8),   
 xend = c(11,3.5,9.5),   
 y = c(55,35, 80),   
 yend = c(30,25, 70),   
 arrow = arrow(type = "closed",  
 length = unit(0.10, "inches")))+  
 geom\_label(aes(x = 10, y = 55,   
 label = "high water table"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_label(aes(x = 4, y = 42,   
 label = "high pumping"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_label(aes(x = 7, y = 80,   
 label = "high rainfall"),  
 hjust = 0.5,  
 vjust = 0.5,  
 fill = "white",   
 label.size = NA,  
 size = 4)+  
 geom\_hline(yintercept = 0,   
 linetype="dashed")



## Water Budget —-

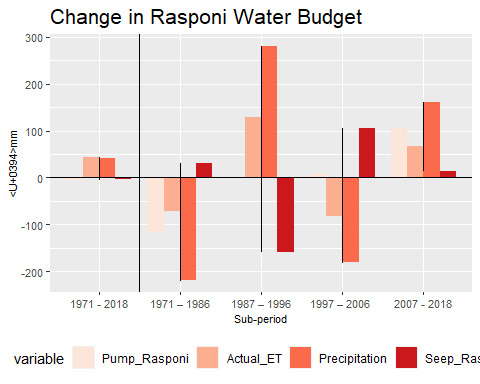
## San Vitale

df1 <- melt(water\_budget[,c("Sub\_period",  
 "Pump\_SanVitale","Actual\_ET",   
 "Precipitation", "Seep\_SanVitale")], id.vars = 1)  
  
  
(plt1 <- ggplot(df1,  
 aes(x = Sub\_period,  
 y =value)) +   
 geom\_bar(aes(fill = variable),   
 stat = "identity",   
 position = "dodge",  
 width = 0.8) +   
 geom\_line() +  
 labs(x = "Sub-period",  
 y = "Δmm",  
 title = "Change in San Vitale Water Budget") +  
 scale\_fill\_brewer(palette="Reds") +  
 budget.theme() +  
 annotate(geom = "text",   
 x = c(1,2,3, 4, 5),   
 y = c(200,200, -200, 200,200),   
 label = c("",  
 "",  
 "",  
 "",  
 ""),  
 size = 3) +  
 geom\_hline(yintercept = 0)+  
 geom\_vline(xintercept = 1.5))



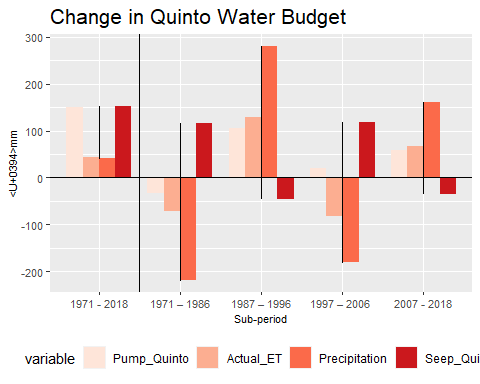
## Rasponi

df2 <- melt(water\_budget[,c("Sub\_period",  
 "Pump\_Rasponi","Actual\_ET",   
 "Precipitation", "Seep\_Rasponi")], id.vars = 1)  
  
  
(plt2 <- ggplot(df2,  
 aes(x = Sub\_period,  
 y =value)) +   
 geom\_bar(aes(fill = variable),   
 stat = "identity",   
 position = "dodge",  
 width = 0.8) +   
 geom\_line() +  
 labs(x = "Sub-period",  
 y = "Δmm",  
 title = "Change in Rasponi Water Budget") +  
 scale\_fill\_brewer(palette="Reds") +  
 budget.theme() +  
 annotate(geom = "text",   
 x = c(1,2,3, 4, 5),   
 y = c(200,200, -200, 200,200),   
 label = c("",  
 "",  
 "",  
 "",  
 ""),  
 size = 3) +  
 geom\_hline(yintercept = 0)+  
 geom\_vline(xintercept = 1.5))



## Quinto

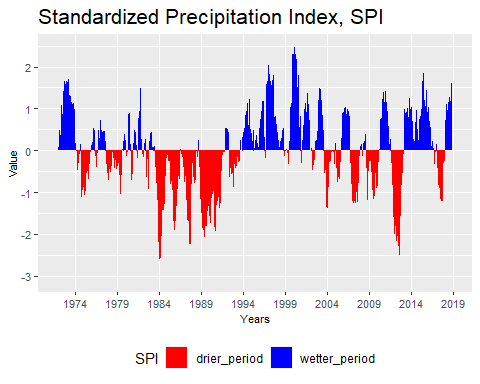
df3 <- melt(water\_budget[,c("Sub\_period",  
 "Pump\_Quinto","Actual\_ET",   
 "Precipitation", "Seep\_Quinto")], id.vars = 1)  
  
  
(plt3 <- ggplot(df3,  
 aes(x = Sub\_period,  
 y =value)) +   
 geom\_bar(aes(fill = variable),   
 stat = "identity",   
 position = "dodge",  
 width = 0.8) +   
 geom\_line() +  
 labs(x = "Sub-period",  
 y = "Δmm",  
 title = "Change in Quinto Water Budget") +  
 scale\_fill\_brewer(palette="Reds") +  
 budget.theme() +  
 annotate(geom = "text",   
 x = c(1,2,3, 4, 5),   
 y = c(200,200, -200, 200,200),   
 label = c("",  
 "",  
 "",  
 "",  
 ""),  
 size = 3) +  
 geom\_hline(yintercept = 0)+  
 geom\_vline(xintercept = 1.5))



## Drought Index —-

## SPI

# Concert date to POSXCIT  
newyear <- parse\_date\_time(spi.data[,1], "by")  
dates <- as.Date.POSIXct(newyear)  
  
Drought\_pos1 <- ifelse(spi.data$spi <= 0, 0, spi.data$spi)  
Drought\_neg1 <- ifelse(spi.data$spi >= 0, 0, spi.data$spi)  
  
df <- tibble(spi.data, Drought\_neg1, Drought\_pos1, dates)  
  
(drought.spi <- ggplot(df) +  
 geom\_area(aes(dates, Drought\_pos1, fill = "wetter\_period")) +  
 geom\_area(aes(dates, Drought\_neg1, fill = "drier\_period")) +  
 labs(x = "Years",   
 y = "Value",   
 title = "Standardized Precipitation Index, SPI") +  
 drought.theme() +  
 scale\_x\_date(date\_breaks = "5 year",  
 date\_minor\_breaks = "5 year",  
 date\_labels = "%Y") +  
 scale\_fill\_manual("SPI",   
 values=c(wetter\_period ="blue",   
 drier\_period="red")))

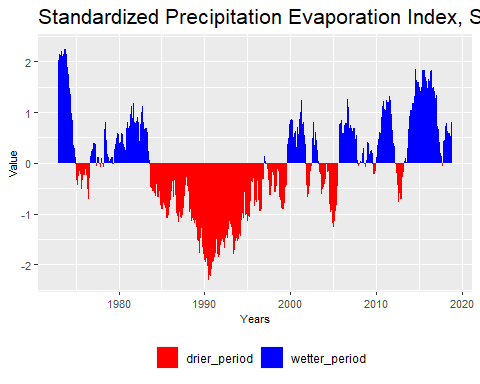


# ggsave(filename = "img.spi.png",  
# dpi = 1200, height = 4,  
# width = 10,  
# path = "D:/Bologna Project/Land\_Reclamation\_Excel\_Data/Plot")  
# dev.off()  
   
#+scale\_color\_manual(" ", values=c(wetter="blue", drier="red"))

## SPEI

newyear = parse\_date\_time(Drought\_Index[,1], "by")  
dates = as.Date.POSIXct(newyear)  
  
#spei index  
Drought1 = Drought\_Index  
Drought\_pos <- ifelse(Drought1$spei <= 0, 0, Drought1$spei)   
  
Drought2 = Drought\_Index  
Drought\_neg <- ifelse(Drought2$spei >= 0, 0, Drought2$spei)   
  
df <- cbind.data.frame(Drought\_Index, Drought\_neg, Drought\_pos)  
  
(drought.spei = ggplot(df) +   
 geom\_area(aes(dates, Drought\_pos,  
 fill = "wetter\_period"),  
 alpha = 1) +  
 geom\_area(aes(dates, Drought\_neg,  
 fill = "drier\_period"), alpha = 1) +  
 labs(x = "Years",   
 y = "Value",   
 title = "Standardized Precipitation Evaporation Index, SPEI") +  
 drought.theme() +  
 scale\_fill\_manual(" ",   
 values=c(wetter\_period ="blue",   
 drier\_period="red",   
 alpha = 0.5)))

## Warning: Removed 23 rows containing missing values (position\_stack).  
  
## Warning: Removed 23 rows containing missing values (position\_stack).



## Subsidence —-

## Columns of Subsidence  
 Q <- subsidence[,2]  
 R <- subsidence[,3]  
 S <- subsidence[,4]  
## Adding up in cumulative value  
 Q <- cumsum(subsidence$Q)   
 R <- cumsum(subsidence$R)  
 S <- cumsum(subsidence$S)  
year <- subsidence$Year  
## Combined into data frame  
 df <- cbind.data.frame(Q, R, S, year)  
  
   
(p1 <- ggplot(df) +   
 geom\_line(aes(year,   
 Q,   
 color = "Quinto\_basin"),   
 size = 1) +  
 geom\_line(aes(year,   
 S,   
 color = "S.Vitale\_basin"),   
 size = 1) +  
 geom\_line(aes(year,   
 R,   
 color = "Rasponi\_basin"),   
 size = 1) +  
 labs(x = "Year",   
 y = "subsidence",   
 title = "Cummulative Subsidence (1972 to 2016)",  
 values=c("Quinto\_basin" = "307FE2",   
 "S.Vitale\_basin" ="steelblue",  
 "Rasponi\_basin" = "#FAAB18")) +  
 theme(axis.text = element\_text(face = "plain",  
 size = 8),  
 plot.title = element\_text(face = "plain", size = 16),  
 legend.justification=c(0.8,0.2),  
 legend.position=c(0.95,0.1)) +  
 annotate(geom = "segment",  
 x = 1985,   
 xend = 1974,   
 y = 200,   
 yend = 200,  
 arrow = arrow(type = "closed",  
 length = unit(0.130, "inches"))) +  
 geom\_label(aes(x = 1987, y = 200,   
 label = "Anthropogenic"),hjust = 0.5,  
 vjust = 0.5,   
 fill = "white", label.size = NA,  
 size = 4))

