

```

#---
#   Package management
#---
list.of.pkgs <-  c("IRISseismic", "ggplot2", "data.table",
                  "dbhydroR", "dplyr", "tidyr", "future",
                  "listenv", "scales", "lubridate")

new.pkgs <- list.of.pkgs[!(list.of.pkgs %in% installed.packages()[, "Package"])]

if (length(new.pkgs)){ install.packages(new.pkgs) }
for (pkg in list.of.pkgs){ library(pkg,character.only = TRUE) }

## Evaluate futures in parallel
plan(multiprocess)

stageDBK <- c('S3103','S0629','S3105','S0631','05025','IY133')
gwDBK <- c('S3009','S3011','S3013','S3015','S5133',
          'S5131','S5129','S5127','S3017','S3019',
          'S3021','S3023','S5141','S5139','S5137','S5135')
airDBK <- c('UP245')
dates <- c('2009-12-19','2010-02-17')

#Looking up unknown dbkeys on the fly
dbhydroR::get_hydro(stationid = "L31NS", category = "WEATHER",
                    date_min = dates[1],
                    date_max = dates[2])

GW_data %<-% dbhydroR::get_hydro(dbkey = gwDBK,
                                date_min = dates[1],
                                date_max = dates[2])
SW_data %<-% dbhydroR::get_hydro(dbkey = stageDBK,
                                date_min = dates[1],
                                date_max = dates[2])
Air_data %<-% dbhydroR::get_hydro(stationid = "L31NS",
                                category = "WEATHER",
                                date_min = dates[1],
                                date_max = dates[2])

meltData <- vector(mode = "list", length = 3)

meltData[[1]] <- melt(GW_data,id='date',na.rm=T,value.name='dataVal')
meltData[[1]]$variable <- gsub('_WELL_', ' ',WELL,', meltData[[1]]$variable)
meltData[[1]]$variable <- gsub('_H2OT_', ' ',H2OT,', meltData[[1]]$variable)

meltData[[2]] <- melt(SW_data,id='date',na.rm=T,value.name='dataVal')
meltData[[2]]$variable <- gsub('_STG_', ' ',STG,', meltData[[2]]$variable)

#---
#   Run plotSeismic.R prior to  additional statements
#---
seismic <- dtTSBy10Sec
names(seismic) <- c('date','dataVal')
seismic$date <- as.POSIXct(seismic$date,tz="UTC")
seismic$variable =paste(st@traces[[1]]@id,'Shakes','10Sec',sep=',')
seismic$dataVal <- (seismic$dataVal/10000000)+5.75
meltData[[3]]<- seismic

# meltData[[4]] <- melt(Air_data,id='date',na.rm=T,value.name='dataVal')
# meltData[[4]]$variable <- gsub('_AIRT_', ' ',AIRT,', meltData[[3]]$variable)

df <- do.call("rbind", meltData)

allData<-df %>%
  separate(variable, c("station", "type", "units"), ",")
allData$ea15<-lubridate::as_datetime(
  lubridate::round_date(allData$date, "15 minutes") )
allData$ea15 <- as.POSIXct(allData$ea15)
lims <- as.POSIXct(strptime(c("2010-01-12 16:21:45",
                             "2010-01-12 17:22:45"),
                          format = "%Y-%m-%d %H:%M:%S"), tz="America/New_York")
# dateRange = as.POSIXct(c('2009-12-19','2010-02-17'))
#shocks/250000
p<-ggplot2::ggplot(data=allData[allData$type %in% c("WELL","Shakes"),],
                  aes(x=ea15,y=dataVal,color=factor(station))) +
  labs(x = "Time") +
  scale_x_datetime(limits =lims, breaks = '5 min',
                  labels = date_format("%m-%d %H:%M")) +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  #geom_line(aes(x=ea15,y=dataVal,color=factor(station)))+
  # ggplot2::ggplot(subset(allData,units %in% c('10Sec')))+
  # ggplot2::geom_line(aes(x=ea15,y=dataVal/250000,
  #                       color=factor(station)))

```

```
geom_line(data=function(x){x[x$type %in% c("WELL"), ]}) +  
geom_line(data=function(x){x[x$type %in% c("Shakes"), ]},color='black')
```