Reading and processing data for use in the LTER-LIFE digital twins: RWS physical data

Karline Soetaert

first version: 01-10-2023; current version: 30 January 2024

This document prepares the physical data that are to be used in the LTER life dtWad package. These data are made available from the Rijkswaterstaat (RWS). Data from the RWS that are close to or in the Wadden area are read, and units converted.

# RWS water temperature and water height data

The water temperature and water height data were requested from RWS, via the website: <https://waterinfo.rws.nl/>.

The data thus obtained are csv files, where the long variable names are in Dutch.

The files are read with the function *readRWS* from the package *dtLife*. This function

* Removes Values that are suspect (these are given a value = 999999999)
* The height is recorded in cm; it is converted to m.
* Several columns are removed.
* Remaining columns are given english names
* The variables are translated from Dutch into english.

## Water heights

* Some water heights are calculated and not measured; these are removed from the data
* The date and time column are merged and converted to POSIXct format

All water heights are versus Normaal Amsterdams Peil (NAP).

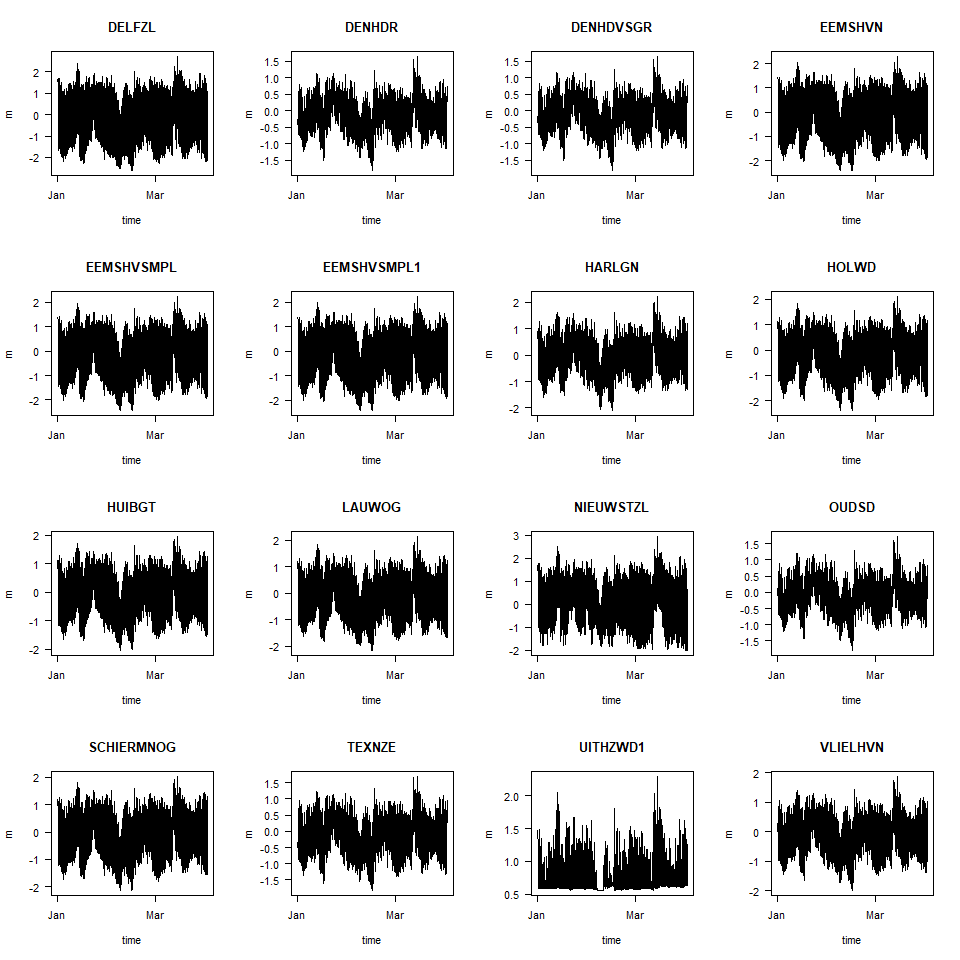
filename <- "20231022\_001.csv" # all stations, including in Northsea  
  
# Read the data file - takes a while  
  
rws.h <- readRWS(dir = "../raw\_data/rws/",   
 file = filename,   
 format = "long")  
  
stats <- attributes(rws.h)$stations  
  
# select only the water height data  
  
Heightwad <- subset(rws.h,   
 subset = variable == "Height" &   
 value < 1e5 &  
 ! is.na(datetime) )  
  
Heightwad <- Heightwad[order(Heightwad$station, Heightwad$datetime),  
 c("station", "datetime", "value")]

The data set comprises 1022874 data points, at 10 minute resolution. The size of these data is too large, so 0.5 hour averages are taken, and the data are converted into wide format.

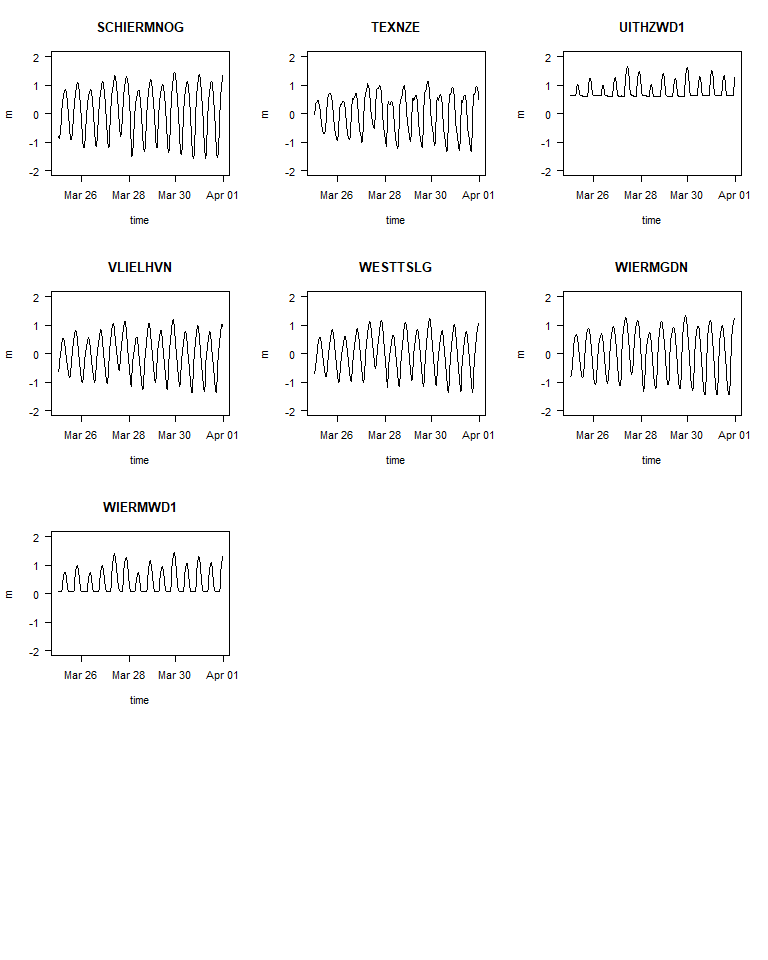
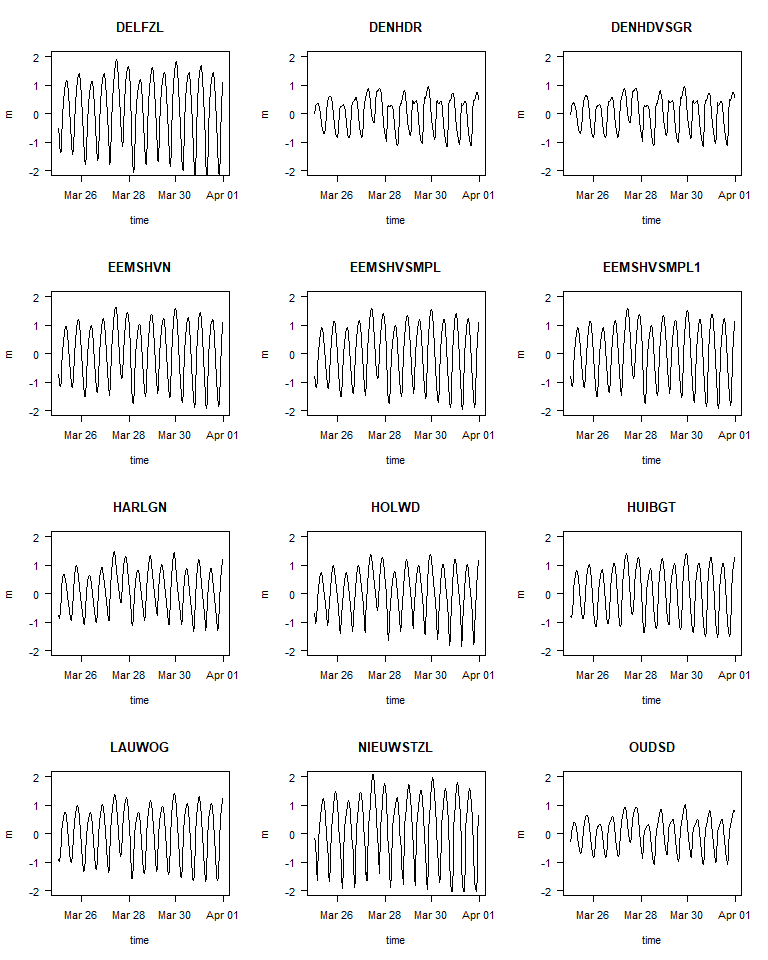
# take 0.5 hour averages  
HH <- average\_timeseries(Heightwad,   
 avgOver = "min",   
 avgTime = 30,  
 by = "station")  
colnames(HH)[3] <- "height"  
  
# convert to wide format  
WadHeightHR <- reshape(data = HH,   
 direction = "wide",   
 timevar = "station",   
 idvar = "datetime")  
colnames(WadHeightHR) <- sub("height.", "", colnames(WadHeightHR))  
  
# Add attributes  
attributes(WadHeightHR) <- c(attributes(WadHeightHR),   
 attributes(Heightwad)[c("variables", "stations", "datasource", "EPSG",   
 "file", "processing", "fun" )])  
attributes(WadHeightHR)$format <- "wide"  
  
WadHeightHR <- subset(WadHeightHR,   
 subset = datetime < "2021-04-01")  
toremove <- "EEMSHVSMPL2" # Lacks data for the selected interval  
WadHeightHR <- WadHeightHR[,-which(colnames(WadHeightHR) %in% toremove)]

The data are shown, first for the entire year, then for the first few months

par(las=1)  
plot(WadHeightHR[,1:17], ylab="m",   
 mfrow=c(4,4), mar=c(3,3,3,2),   
 type="l")



# plot results for small time frame  
sub <- subset(WadHeightHR, subset= datetime > "2021-03-25")  
plot(sub, mfrow=c(4,3), ylab="m", type="l", ylim=c(-2,2), las=1)

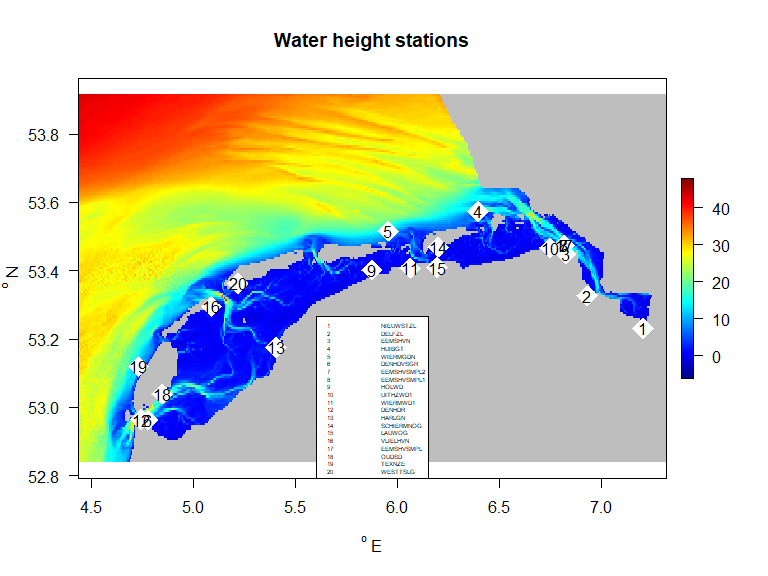


Based on the data, the water heights for stations WIERMD1, UITHZWD1 and EEMSHVWMPL2 are removed, as these are data from intertidal flats. The resulting data.frame is saved as binary R file.

toremove <- c("WIERMWD1", "UITHZWD1", "EEMSHVSMPL2")  
WadHeightHR <- WadHeightHR[,-which(colnames(WadHeightHR) %in% toremove)]  
  
atts <- attributes(WadHeightHR)$stations  
ii <- which(atts$station %in% toremove)  
if (length(ii)) atts <- atts[-ii,]  
attributes(WadHeightHR)$stations <- atts  
  
save(file="../processed\_data/WadHeightHR.rda", WadHeightHR)

## Positions of the stations

plotBathymetry(WadDepth,   
 pts=stats[,c("longitude", "latitude")],   
 ptlist=list(cex=3, col= "white"), NAcol="grey", type="image",  
 main="Water height stations")  
text(stats$longitude, stats$latitude, labels=1:nrow(stats))  
legend("bottom", legend = c(1:nrow(stats), stats$station), ncol=2, cex=0.4)



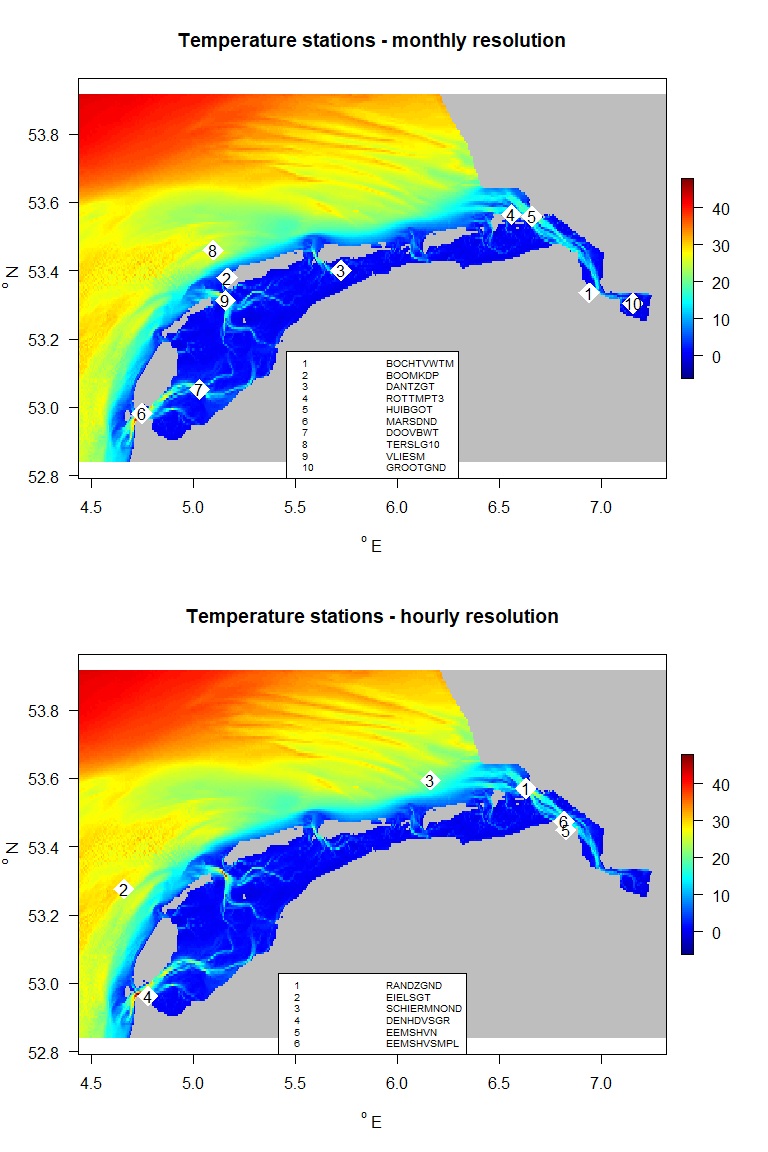
## Water temperature

Temperature was measured at varying temporal resolution, either during monitoring cruises, on quasi-monthly resolution, or at 10 minute resolution. For the latter, hourly averages are calculated.

filename <- "20231022\_002.csv" # all stations, including in Northsea  
  
rws.t <- readRWS(dir = "../raw\_data/rws/",   
 file = filename,   
 format = "long")  
  
stats <- attributes(rws.t)$stations  
  
# Check validity of the data  
  
Tempwad <- subset(rws.t,   
 subset = value < 1e5 )  
Tempwad <- Tempwad[,c("station", "datetime", "value")]  
  
# split into stations with a lot of data and few data  
Tnum <- table(Tempwad$station) # number of data per station  
  
# High resolution data  
# ======================  
  
HRstation <- names(Tnum)[Tnum > 10000]  
  
TempwadHR <- subset(Tempwad,   
 subset = station %in% HRstation)  
statHR <- subset(stats,   
 subset = station %in% HRstation)  
  
# take hourly averages  
THR <- average\_timeseries(TempwadHR,   
 avgOver = "hour",   
 avgTime = 1,  
 by = "station")  
colnames(THR)[3] <- "T"  
  
  
# convert to wide format   
WadTempHR <- reshape(data = THR,   
 direction = "wide",   
 timevar = "station",   
 idvar = "datetime")  
  
# add station attributes  
attributes(WadTempHR)$stations <- statHR  
  
# create good column names  
cn <- colnames(WadTempHR)[-1]  
cn <- sub("T.", "", cn)  
colnames(WadTempHR)[-1] <- cn  
  
attributes(WadTempHR)$format <- "wide"  
  
# make dataset smaller by restricting time  
WadTempHR <- subset(WadTempHR,   
 subset = datetime < "2021-04-01")  
  
# Low resolution data  
# ======================  
  
TempwadLR <- subset(Tempwad,   
 subset = ! station %in% HRstation)  
statLR <- subset(stats,   
 subset = ! station %in% HRstation)  
  
# average over 15 days !  
TLR <- average\_timeseries(TempwadLR,   
 avgOver = "day",   
 avgTime = 15,  
 by = "station")  
colnames(TLR)[3] <- "T"  
  
WadTempLR <- reshape(data = TLR,   
 direction = "wide",   
 timevar = "station",   
 idvar = "datetime")  
WadTempLR <- WadTempLR[order(WadTempLR$datetime),]  
attributes(WadTempLR)$stations <- statLR  
  
cn <- colnames(WadTempLR)[-1]  
cn <- sub("T.", "", cn)  
colnames(WadTempLR)[-1] <- cn  
  
attributes(WadTempLR)$format <- "wide"  
  
  
# save results  
save(file="../processed\_data/WadTempHR.rda", WadTempHR)  
save(file="../processed\_data/WadTempLR.rda", WadTempLR)

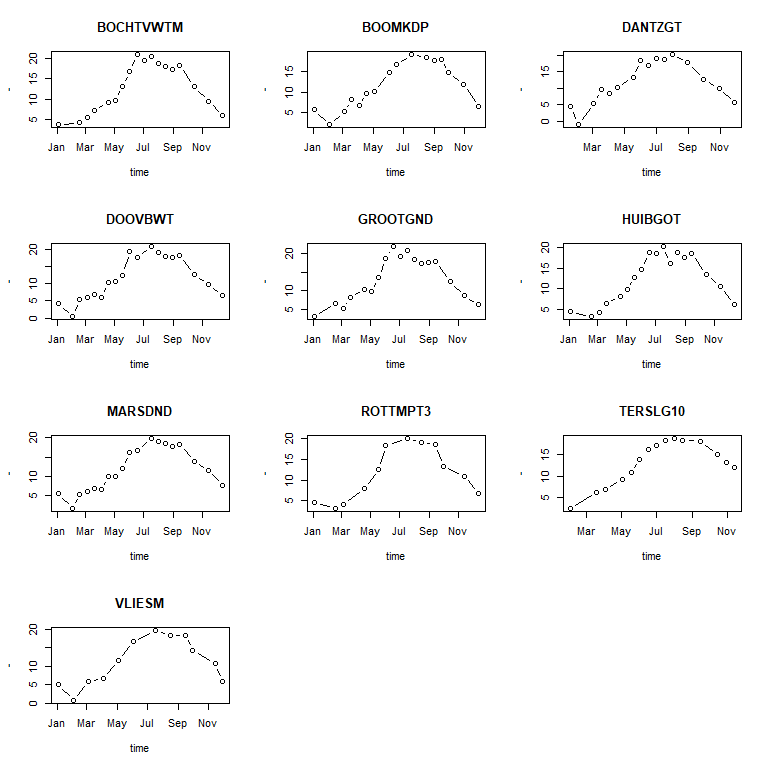
## Positions of the stations

par(mfrow=c(2,1))  
stats <- attributes(WadTempLR)$stations  
plotBathymetry(WadDepth,   
 pts=stats[,c("longitude", "latitude")],   
 ptlist=list(cex=3, col= "white"), NAcol="grey", type="image",  
 main="Temperature stations - monthly resolution")  
text(stats$longitude, stats$latitude, labels=1:nrow(stats))  
legend("bottom", legend = c(1:nrow(stats), stats$station), ncol=2, cex=0.6)  
  
stats <- attributes(WadTempHR)$stations  
plotBathymetry(WadDepth,   
 pts=stats[,c("longitude", "latitude")],   
 ptlist=list(cex=3, col= "white"), NAcol="grey", type="image",  
 main="Temperature stations - hourly resolution")  
text(stats$longitude, stats$latitude, labels=1:nrow(stats))  
legend("bottom", legend = c(1:nrow(stats), stats$station), ncol=2, cex=0.6)



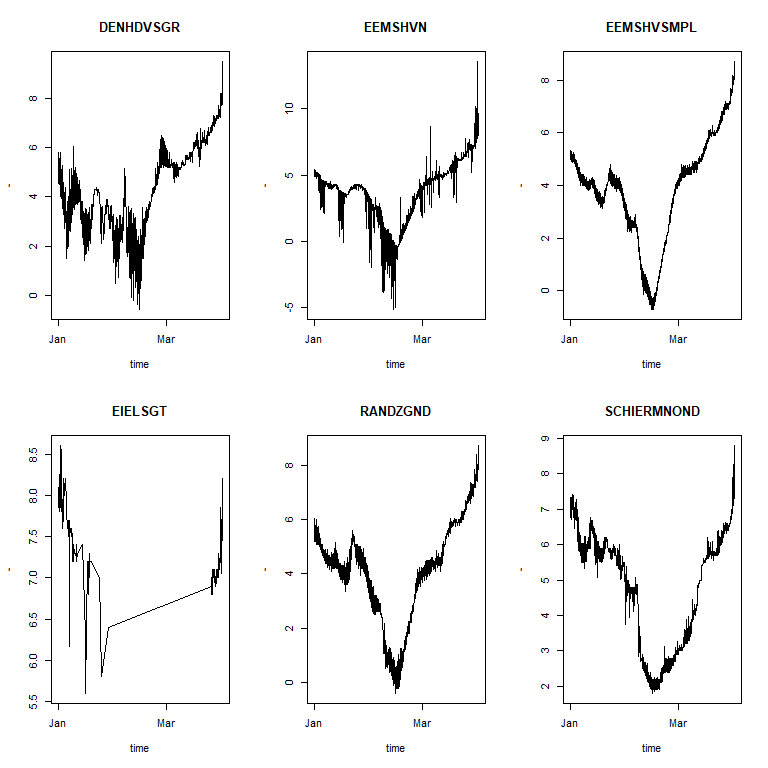
## The low-resolution temperature data

plot(WadTempLR, mfrow=c(4,3), type="b")



## The high-resolution temperature data

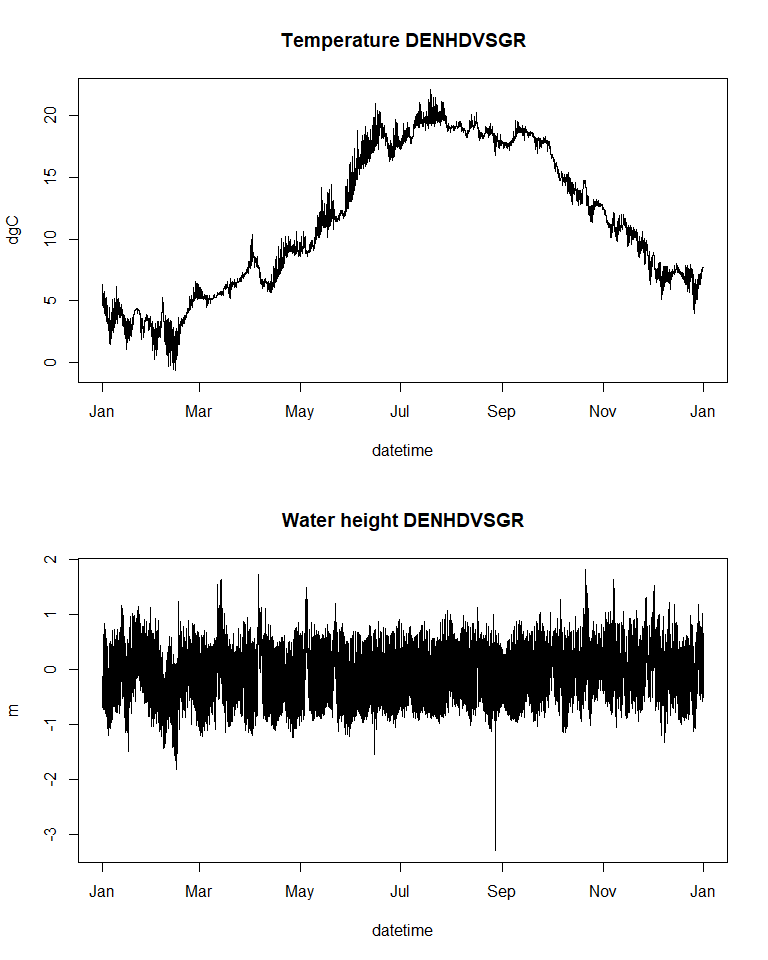
par(mfrow=c(3,2))  
plot(WadTempHR, type="l")



## Single datasets

These are data from one station only; 20 minute averages are taken here.

RWS <- readRWS(dir = "../raw\_data/rws/",   
 file = "20230915\_019.csv",   
 format = "long")  
  
Temp2021 <- subset(RWS,   
 subset = variable == "T" &   
 station == "DENHDVSGR" &  
 value < 1e10 &  
 datetime < "2022-01-01")  
  
Temp2021 <- na.omit(Temp2021[,c("station", "datetime", "value")])  
  
Temp2021 <- Temp2021[order(Temp2021$datetime),]  
  
Temp2021 <- average\_timeseries(Temp2021,   
 avgOver = "min",   
 avgTime = 20,   
 value = "value",  
 by = "station")  
colnames(Temp2021)[3] <- "T"  
row.names(Temp2021) <- NULL  
  
Height2021 <- subset(RWS,   
 subset = variable == "Height" &   
 station == "DENHDVSGR" &   
 value < 1e5 &  
 datetime < "2022-01-01")  
  
Height2021 <- na.omit(Height2021[,c("station", "datetime", "value")])  
  
Height2021 <- Height2021[order(Height2021$datetime),]  
  
Height2021 <- average\_timeseries(Height2021,   
 avgOver = "min",   
 avgTime = 20,   
 value = "value",  
 by = "station")  
  
colnames(Height2021)[3] <- "Height"  
row.names(Height2021) <- NULL  
  
par(mfrow=c(2,1))  
with(Temp2021, plot(datetime, T, type="l", ylab="dgC",   
 main="Temperature DENHDVSGR"))  
with(Height2021, plot(datetime, Height, type="l", ylab= "m",  
 main= "Water height DENHDVSGR"))



# merge these two datasets  
WadHT2021<- merge(Height2021, Temp2021)  
  
# merge the metadata of these two datasets  
mH <- meta(Height2021)  
mT <- meta(Temp2021)  
nn <- names(mH)  
  
META <- list()  
for (i in 1:length(nn)){  
 name <- nn[i]  
 if (is.vector(mH[[name]]))  
 META[[i]] <- c (mH[[name]], mT[[name]])  
 else   
 META[[i]] <- rbind(mH[[name]], mT[[name]])  
}  
names(META) <- nn  
  
# update processing info and add metadata to merged data  
META$processing <- c(META$processing, paste("Two files merged at", Sys.time()))  
  
attributes(WadHT2021) <- c(attributes(WadHT2021), META)  
attributes(WadHT2021)$format <- "wide"  
  
save(file="../processed\_data/WadHT2021.rda", WadHT2021)  
save(file="../processed\_data/Temp2021.rda", Temp2021)  
save(file="../processed\_data/Height2021.rda", Height2021)

# References

The following R-sources were used for this work:

R-core:

* R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

R-package dtWad,

* Soetaert K (2024). dtWad: Waddensea Digital Twin: general utilities. R package version 0.0.1.