Auswertung\_Reaktordaten

# Auswertung der Reaktordaten

## Vorbereitung

Zuerst müssen die benötigten Packages installiert und geladen werden: install.packages(“tidyverse”) install.packages(“tidyr”) install.packages(“ggplot2”) install.packages(“ggpmisc”) install.packages(“readr”) install.packages(“lubridate”) install.packages(“ggpubr”)

library("tidyverse")  
library("tidyr")  
library("ggplot2")  
library("ggpmisc")  
library("readr")  
library("lubridate")  
library("ggpubr")

## Daten einlesen

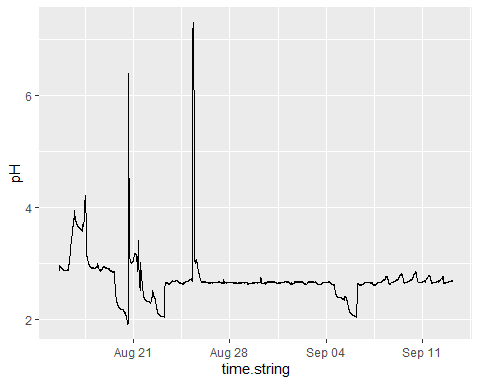
Der zweite Schritt ist das Einlesen der verschiendenen Daten, welche vom Reaktor stammen. Die Daten “aktuell” sind jene aus dem aktuellen Experiment. Die “ps” Daten stammen aus dem Sommer mit der Algenart Pseudococcomyxa simplex. Die Daten “sb” stammen aus dem abgebrochenen Experiment mit der Algenart Stichococcus bacillaris.

aktuell <- read\_delim("2023-10-02\_Kultivierungsdaten.csv", ",")  
ps <- read\_delim("2023-08-15\_Kultivierungsdaten.csv", ",")  
sb <- read\_delim("2023-09-19\_Kultivierungsdaten.csv", ",")

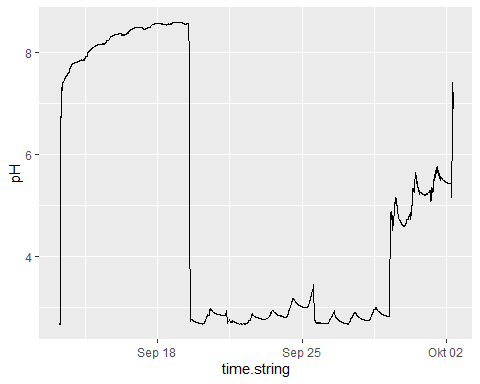
## Plots zur allgemeinen Übersicht

Plots um zu sehen, wie sich der pH-Wert bei den drei Experimenten entwickelt hat.

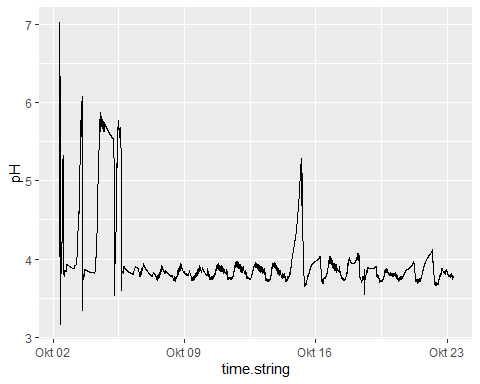
ggplot(ps, aes(time.string, pH)) + geom\_line()



ggplot(sb, aes(time.string, pH)) + geom\_line()

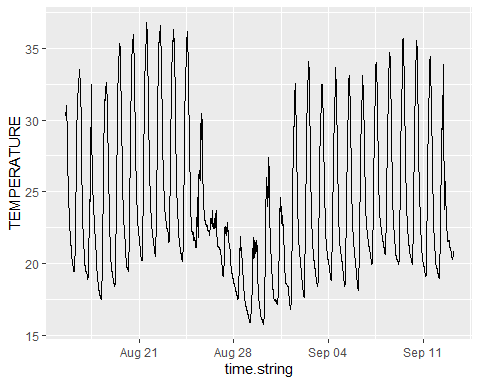


ggplot(aktuell, aes(time.string, pH)) + geom\_line()

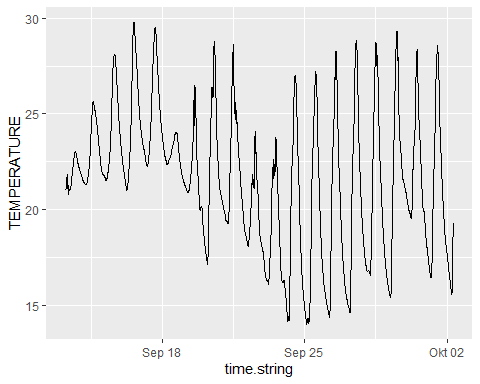


Plots um zu sehen, wie sich die Temperatur in den drei Experimenten entwickelt hat.

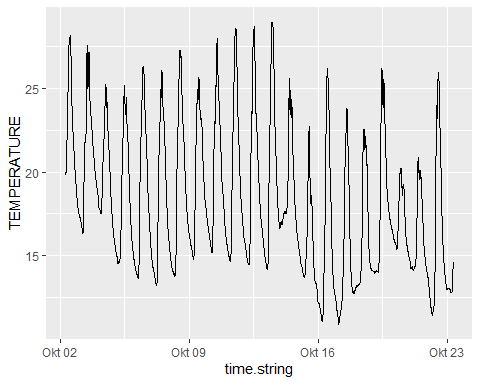
ggplot(ps, aes(time.string, TEMPERATURE)) + geom\_line()



ggplot(sb, aes(time.string, TEMPERATURE)) + geom\_line()



ggplot(aktuell, aes(time.string, TEMPERATURE)) + geom\_line()



## Neustes Experiment

Wir versuchen diese Daten so anzupassen, dass Auswertungen möglich werden. Dafür wird eine neue Dataframe mit jenen Spalten erstellt, welche wir auswerten möchten. Das Datum muss als solches formatiert werden, damit Analysen möglich werden.

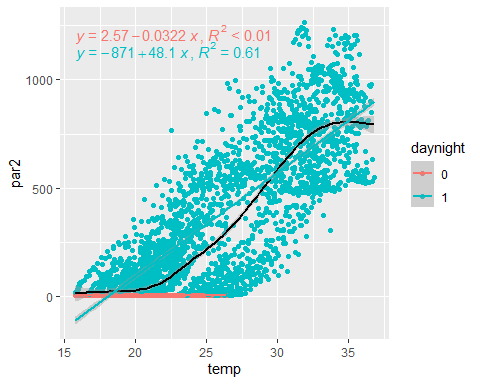
ps\_neu <- data.frame(DateTime = as.POSIXct(as.character(ps$time.string), format = "%Y-%m-%d %H:%M:%OS"))  
  
ps\_neu <- ps\_neu |>   
 mutate(par1 = ps$PAR.1,   
 par2 = ps$PAR.2,  
 pardiff = par2 - par1,  
 temp = ps$TEMPERATURE,  
 pH = ps$pH,  
 daynight = as.factor(ps$daynight))

Nun gruppieren wir die Daten nach Tag und Nacht, um bessere Aussagen über Zusammenhänge machen zu können.

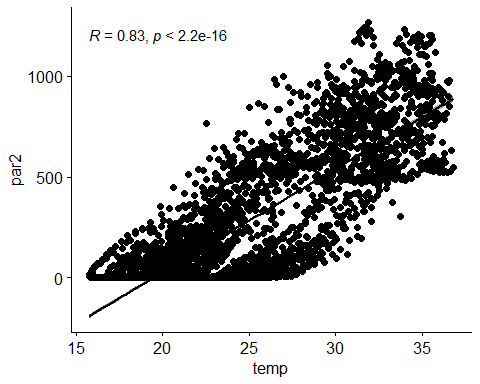
sum\_ps\_neu <- ps\_neu |>   
 group\_by(daynight)

Nun erstellen wir Plots um den Zusammenhang zwischen der Strahlung und der Temperatur zu sehen.

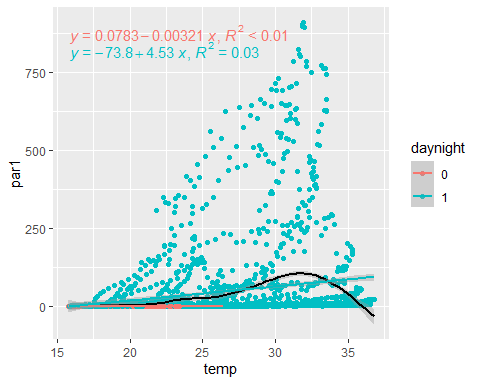
ggplot(sum\_ps\_neu, aes(temp, par2, color = daynight)) +   
 geom\_point() +   
 geom\_smooth(col = "black") +  
 stat\_poly\_line() +  
 stat\_poly\_eq(use\_label(c("eq", "R2")))



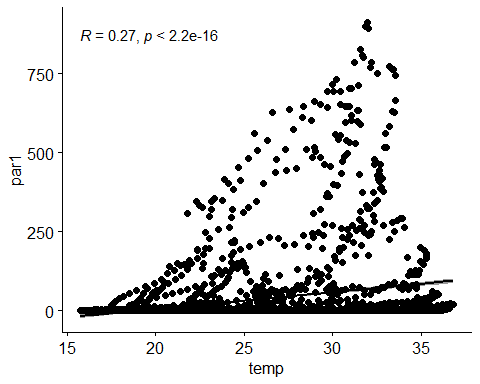
ggscatter(sum\_ps\_neu, x = "temp", y = "par2",   
 add = "reg.line", conf.int = TRUE,   
 cor.coef = TRUE, cor.method = "pearson" )



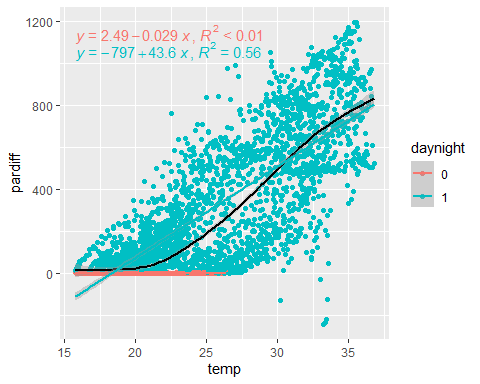
ggplot(sum\_ps\_neu, aes(temp, par1, color = daynight)) +   
 geom\_point() +   
 geom\_smooth(col = "black") +  
 stat\_poly\_line() +  
 stat\_poly\_eq(use\_label(c("eq", "R2")))



ggscatter(sum\_ps\_neu, x = "temp", y = "par1",   
 add = "reg.line", conf.int = TRUE,   
 cor.coef = TRUE, cor.method = "pearson" )



ggplot(sum\_ps\_neu, aes(temp, pardiff, color = daynight)) +   
 geom\_point() +   
 geom\_smooth(col = "black") +  
 stat\_poly\_line(geom = "smooth") +  
 stat\_poly\_eq(use\_label(c("eq", "R2")))



ggscatter(sum\_ps\_neu, x = "temp", y = "pardiff",   
 add = "reg.line", conf.int = TRUE,   
 cor.coef = TRUE, cor.method = "pearson" )

