2022\_Rating\_Curve

Jake Cavaiani

9/2/2022

# The purpose of this script is to read in pressure transducer data from 2022 and generate rating curves to ultimately generate predicted continuous discharge records throughout the whole season

# Step 1) Read in most updated merged files from each site from the google drive (DoD Project->2022 AK sensors->Discharge->site->“merged\_date\_file” )

knitr::opts\_chunk$set(message = FALSE, warnings = FALSE)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.7 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0   
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

##   
## Attaching package: 'lubridate'

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

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday, week,  
## yday, year

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:purrr':  
##   
## transpose

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':  
##   
## alpha, rescale

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

## here() starts at C:/Users/Karen Jorgenson/Documents/Github/DoD\_Discharge

##   
## Attaching package: 'cowplot'

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

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

## Loading required package: ggpp

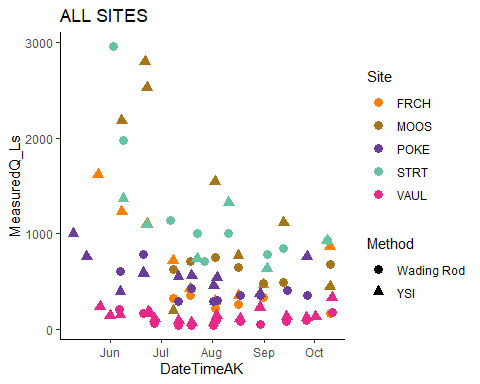
##   
## Attaching package: 'ggpp'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

### ALL Sites

# Import data from google drive #  
discharge.2022 <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vQR6HHHDpnxc6DmNHfdNLR9-dgDHR5Imt0Ve4\_t2DzIF18\_8D3O2da5zcWQzJUSoFQfaetPZDeXZ610/pub?gid=0&single=true&output=csv"  
QSummary.2022 <- read.csv(url(discharge.2022))  
  
### Format Time ###  
QSummary.2022$Date <- mdy(QSummary.2022$Date)  
QSummary.2022$DateTimeAK <- as.POSIXct(paste(QSummary.2022$Date, QSummary.2022$Time), format = "%Y-%m-%d %H:%M", tz = "America/Anchorage")  
QSummary.2022$DateTimeAK <- lubridate::round\_date(QSummary.2022$DateTimeAK, "60 minutes")  
  
### ALL Sites ###  
QSummary.2022 %>% ggplot() +  
 geom\_point(aes(x=DateTimeAK, y=MeasuredQ\_Ls, color=Site, shape=Method), size=3) +  
 theme\_classic() +  
 scale\_color\_manual(values=c("#FF7F00","#A6761D", "#6A3D9A", "#66C2A5", "#E7298A")) + ggtitle("ALL SITES")

## Warning: Removed 10 rows containing missing values (geom\_point).

 Here is all the discrete discharge measurements for 2022

Eielson atmospheric pressure data will be used to fill gaps in air pressure.

eielson.atmo.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vQjJbkVdOseFUsTFMDXaHRoKkPiFt6Uvc-9R3tkFZxKFjqSmPPgZzexdbhezBr\_qYgGUdVN6ywYR28c/pub?output=csv"  
  
eielson.atmo.2022 <- read.csv(url(eielson.atmo.2022.url), skip = 6)   
eielson.atmo.2022 <- eielson.atmo.2022[-1,]  
  
names(eielson.atmo.2022) <- c("Site", "DateTimeAK", "sea\_level\_pressure", "AirPressure")  
  
eielson.atmo.2022$DateTimeAK <- mdy\_hm(eielson.atmo.2022$DateTimeAK)  
  
eielson.atmo.2022$DateTimeAK <- lubridate::round\_date(eielson.atmo.2022$DateTimeAK, "15 minutes")  
  
eielson.atmo.2022$DateTimeAK <- force\_tz(eielson.atmo.2022$DateTimeAK, "America/Anchorage")  
  
eielson.atmo.2022$AirPressure <- as.numeric(eielson.atmo.2022$AirPressure)

## Warning: NAs introduced by coercion

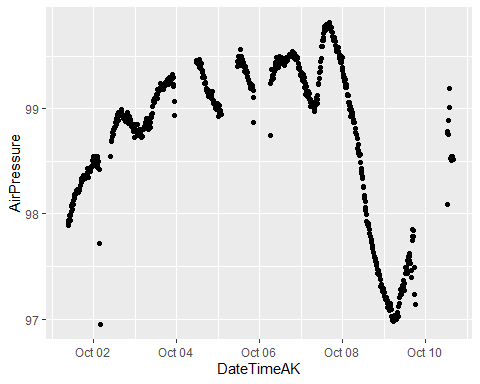
eielson.atmo.2022$AirPressure <- eielson.atmo.2022$AirPressure\*3.38639 # converting from inHG to kPa  
  
## conversion to elevation at each site  
# MOOS elevation is 574 feet  
# POKE 760.96 ft  
# STRT 820 ft  
# FRCH 601 ft   
# VAUL 688.8 ft  
  
eielson.atmo.2022.compare <- eielson.atmo.2022  
eielson.atmo.2022.compare$sea\_level\_pressure <- as.numeric(eielson.atmo.2022.compare$sea\_level\_pressure)  
eielson.atmo.2022.compare$mmHG <- eielson.atmo.2022.compare$sea\_level\_pressure \* 25.44 # converting to mmHG  
  
   
# MOOS   
eielson.atmo.2022.compare$mmHGcorrectedMOOS <- eielson.atmo.2022.compare$mmHG - (2.5\*574/100)   
eielson.atmo.2022.compare$AirPressureCorrectedMOOS <- eielson.atmo.2022.compare$mmHGcorrectedMOOS \* 0.133322 # converting this to kPA to compare with MOOS PT  
  
# POKE   
eielson.atmo.2022.compare$mmHGcorrectedPOKE <- eielson.atmo.2022.compare$mmHG - (2.5\*760.96/100)   
eielson.atmo.2022.compare$AirPressureCorrectedPOKE <- eielson.atmo.2022.compare$mmHGcorrectedPOKE \* 0.133322 # converting this to kPA   
  
# STRT  
eielson.atmo.2022.compare$mmHGcorrectedSTRT <- eielson.atmo.2022.compare$mmHG - (2.5\*820/100)   
eielson.atmo.2022.compare$AirPressureCorrectedSTRT <- eielson.atmo.2022.compare$mmHGcorrectedSTRT \* 0.133322 # converting this to kPA   
  
# FRCH  
eielson.atmo.2022.compare$mmHGcorrectedFRCH <- eielson.atmo.2022.compare$mmHG - (2.5\*601/100)   
eielson.atmo.2022.compare$AirPressureCorrectedFRCH <- eielson.atmo.2022.compare$mmHGcorrectedFRCH \* 0.133322 # converting this to kPA   
  
# VAUL  
eielson.atmo.2022.compare$mmHGcorrectedVAUL <- eielson.atmo.2022.compare$mmHG - (2.5\*688.8/100)  
eielson.atmo.2022.compare$AirPressureCorrectedVAUL <- eielson.atmo.2022.compare$mmHGcorrectedVAUL \* 0.133322 # converting this to kPA

### MOOS

This is the raw atmospheric pressure that is located at MOOS. I am going to clean this up a little bit:

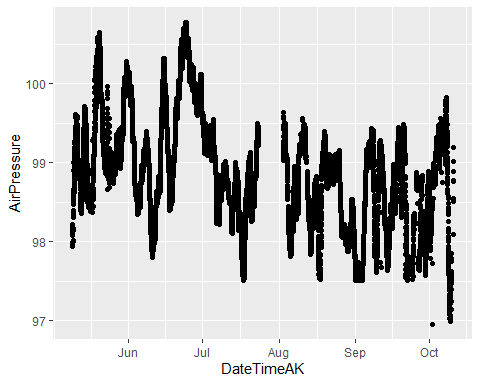
# cleaning erroneous points   
moos.atmo.2022 <- moos.atmo.2022 %>%  
 mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-05-01" & DateTimeAK <= "2022-10-01" & AirPressure < 97.5, NA, .)))  
  
moos.atmo.2022 <- moos.atmo.2022 %>%  
 mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-05-15" & DateTimeAK <= "2022-05-20" & AirPressure < 98.2, NA, .)))  
  
moos.atmo.2022 <- moos.atmo.2022 %>%  
 mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-05-18" & DateTimeAK <= "2022-05-20" & AirPressure < 99.1, NA, .)))  
  
moos.atmo.2022 <- moos.atmo.2022 %>%  
 mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-08-10" & DateTimeAK <= "2022-08-12" & AirPressure > 99.5, NA, .)))  
  
moos.atmo.2022 <- moos.atmo.2022 %>%  
 mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-08-10" & DateTimeAK <= "2022-08-12" & AirPressure < 99, NA, .)))  
  
moos.atmo.2022 <- moos.atmo.2022 %>%  
 mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-08-10" & AirPressure < 96.8, NA, .)))  
  
moos.atmo.2022 <- moos.atmo.2022 %>% filter(DateTimeAK <= "2022-10-10 12:30:00") %>% mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-10-3"& DateTimeAK <= "2022-10-8" & AirPressure < 98.5, NA, .)))   
  
moos.atmo.2022 <- moos.atmo.2022 %>% mutate(across(c(AirPressure),   
 ~ifelse(DateTimeAK >= "2022-10-2"& DateTimeAK <= "2022-10-3" & AirPressure < 98, NA, .)))   
  
  
moos.atmo.2022 %>% filter(DateTimeAK <= "2022-10-15" & DateTimeAK >= "2022-10-01") %>% ggplot(aes(x = DateTimeAK, y = AirPressure)) +  
 geom\_point()

## Warning: Removed 225 rows containing missing values (geom\_point).



moos.atmo.2022 %>% ggplot(aes(x = DateTimeAK, y = AirPressure)) +  
 geom\_point()

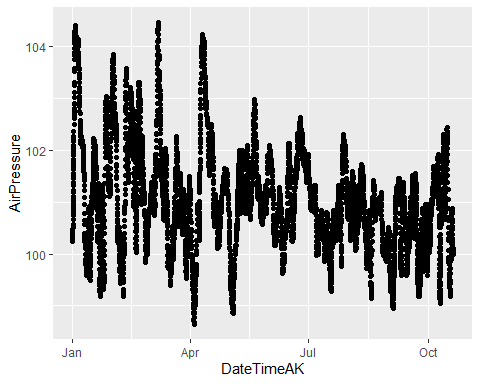
## Warning: Removed 1064 rows containing missing values (geom\_point).

 This is a little better

# Eielson Atmospheric Pressure

ggplot(eielson.atmo.2022, aes(x = DateTimeAK, y = AirPressure)) +  
 geom\_point()

## Warning: Removed 4 rows containing missing values (geom\_point).

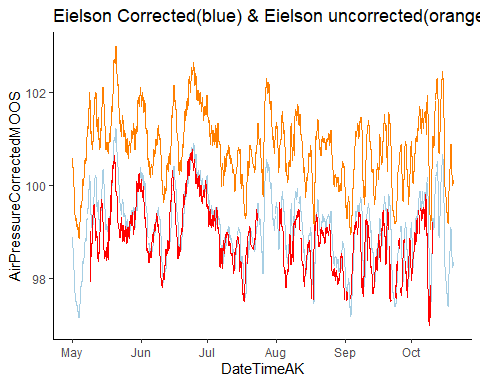


moos.atmo.2022.compare <- moos.atmo.2022

This is the uncorrected air pressure comparison between Eielson and MOOS

atmo.pt.2022.1 <- left\_join(eielson.atmo.2022.compare, moos.atmo.2022.compare, by = "DateTimeAK")  
  
atmo.pt.2022.1 %>% filter(DateTimeAK >= "2022-05-01") %>% ggplot(aes(x = DateTimeAK, y = AirPressureCorrectedMOOS)) +  
 geom\_line(aes(x=DateTimeAK, y=AirPressureCorrectedMOOS), color="#A6CEE3") +  
 geom\_line(aes(x=DateTimeAK, y=AirPressure.x), color="#FF7F00") +  
 geom\_line(aes(x=DateTimeAK, y=AirPressure.y), color = "red") +  
 theme\_classic() +  
 ggtitle("Eielson Corrected(blue) & Eielson uncorrected(orange) & MOOS(red) Atmo P")

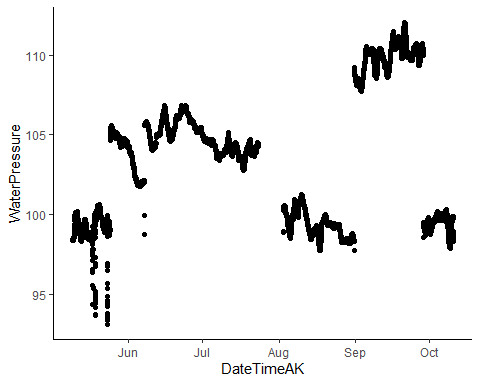
## Warning: Removed 599 row(s) containing missing values (geom\_path).

 Looks like we have to use the corrected pressure to get what we want for our rating curves.

We will only use pressure transducer 2 at Moose because it is more complete.

ggplot(moos.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure)) +  
 geom\_point() +  
 theme\_classic()

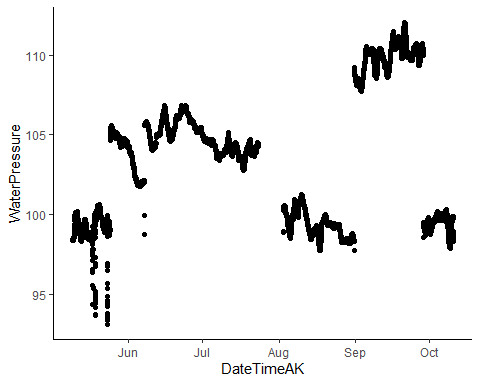
## Warning: Removed 15 rows containing missing values (geom\_point).



This is PT2 that has the step changes due to the cleaning and then it has a gap at the end of July to beginning of August was because the memory filled up.

# adjusting/cleaning moos.stream.two  
  
ggplot(moos.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure)) +  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 15 rows containing missing values (geom\_point).



moos.stream.two.2022 <- moos.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-05-09" & DateTimeAK <= "2022-05-25", NA, .))) # PT wasnt placed into the stream until 5/25  
  
moos.stream.two.2022 <- moos.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-06-01" & DateTimeAK <= "2022-07-01" & WaterPressure < 101, NA, .))) %>% filter(WaterPressure > 90)  
  
# shfiting the step change   
moos.2.before <- moos.stream.two.2022[c(1:3817), ] # 6/7 @ 14:05 we cleaned the PVC housing so we are shifting the previous data up to match after the cleaning   
moos.2.after <- moos.stream.two.2022[c(3818:40000), ]  
moos.stream.two.2022[3818, 2] - moos.stream.two.2022[3817, 2] # 3.573

## [1] 3.573

moos.2.before$WaterPressure <- moos.2.before[, 2] + 3.573  
moos.stream.two.2022 <- full\_join(moos.2.before, moos.2.after)

## Joining, by = c("Site", "WaterPressure", "TempC", "DateTimeAK")

moos.2.before2 <- moos.stream.two.2022[c(1:25434), ]   
moos.2.mid2 <- moos.stream.two.2022[c(25435:33497), ]  
moos.2.after2 <- moos.stream.two.2022[c(33498:40000), ]  
moos.stream.two.2022[25435, 2] - moos.stream.two.2022[25434, 2] # 10.926

## [1] 10.926

moos.2.mid2$WaterPressure <- moos.2.mid2[, 2] - 10.926  
moos.stream.two.2022.2 <- full\_join(moos.2.before2, moos.2.mid2)

## Joining, by = c("Site", "WaterPressure", "TempC", "DateTimeAK")

moos.stream.two.2022 <- full\_join(moos.stream.two.2022.2, moos.2.after2)

## Joining, by = c("Site", "WaterPressure", "TempC", "DateTimeAK")

## Bring the last chunk up half way?  
moos.2.before3 <- moos.stream.two.2022[c(1:17096), ]   
moos.2.after3 <- moos.stream.two.2022[c(17097:40000), ]  
moos.stream.two.2022[17096, 2] - moos.stream.two.2022[17097, 2] # 5.389

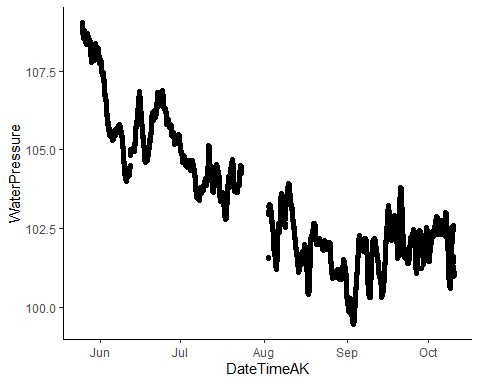
## [1] 5.389

moos.2.after3$WaterPressure <- moos.2.after3[, 2] + 5.389/2  
moos.stream.two.2022 <- full\_join(moos.2.before3, moos.2.after3)

## Joining, by = c("Site", "WaterPressure", "TempC", "DateTimeAK")

ggplot(data = moos.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure)) +  
 geom\_point() +  
 theme\_classic()

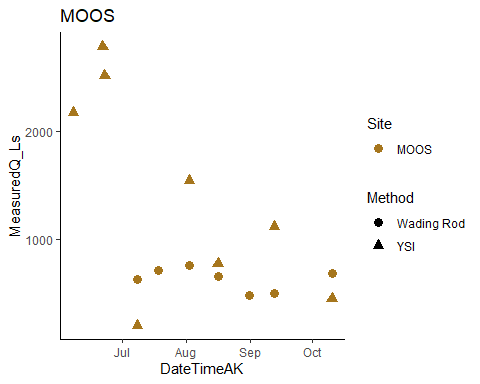
## Warning: Removed 3023 rows containing missing values (geom\_point).

 This looks much better

# joining two cleaned water pressure together   
moos.final.pressure.2022 <- moos.stream.two.2022  
  
  
moos.final.pressure.2022 <- moos.final.pressure.2022[,-c(3,5,7)] # removing columns that are unnecessary. I only need both water pressures and mean pressure with a DateTime column   
  
  
eielson.atmo.2022 <- eielson.atmo.2022[,-c(1,3)] # i dont need site, Temp and sea\_level\_pressure column  
  
eielson.atmo.2022.compare <- eielson.atmo.2022.compare[ , -which(names(eielson.atmo.2022.compare) %in% c("Site","TempC", "sea\_level\_pressure", "AirPressure", "mmHG", "mmHGcorrectedMOOS","mmHGcorrectedPOKE","mmHGcorrectedSTRT","mmHGcorrectedFRCH","mmHGcorrectedVAUL"))] # removing columns I do not need  
  
### Using mostly our air pressure data with gap filled in by Eilson:  
atmo.pt.2022.1$moos.combo <- ifelse(atmo.pt.2022.1$AirPressure.y %in% NA, atmo.pt.2022.1$AirPressureCorrectedMOOS, atmo.pt.2022.1$AirPressure.y )  
  
#join the atmospheric and water pressure together  
MOOS.2022 <- left\_join(atmo.pt.2022.1, moos.final.pressure.2022, by = "DateTimeAK")  
  
  
# Water pressure - atmospheric pressure  
 MOOS.2022$difference <- MOOS.2022$WaterPressure - MOOS.2022$moos.combo  
  
  
names(MOOS.2022)[names(MOOS.2022) == 'Site.x'] <- 'Site'  
MOOS.2022$Site <- "MOOS"

### Plot MOOS measured Q  
QSummary.MO.2022 <- QSummary.2022 %>% filter(Site =="MOOS")  
  
ggplot(QSummary.MO.2022) +  
 geom\_point(aes(x = DateTimeAK, y = MeasuredQ\_Ls, shape = Method, color = Site), size=3) +  
 theme\_classic() +  
 scale\_color\_manual(values=c("#A6761D")) +   
 ggtitle("MOOS")

## Warning: Removed 2 rows containing missing values (geom\_point).



# trying to merge by nearest date if we have an offset point   
MOOS.2022.dt <- setDT(MOOS.2022)  
MOOS.2022.dt <- subset(MOOS.2022.dt, DateTimeAK < "2022-08-31 06:00:00") # removing rows that had dates corresponding to end of record that messed up the rolling nearest function   
QSummary.MO.2022.dt <- QSummary.MO.2022  
  
Moose1comb.2022 <- MOOS.2022.dt[QSummary.MO.2022.dt, on = "DateTimeAK", roll = 'nearest']  
  
  
# Remove very low Q points  
Moose1comb.2022 <- Moose1comb.2022[-c(4:6),]  
  
## ONLY using PT 2  
MOOS1.lm.2022 <- lm(Moose1comb.2022$MeasuredQ\_Ls ~ Moose1comb.2022$difference)  
summary(MOOS1.lm.2022)

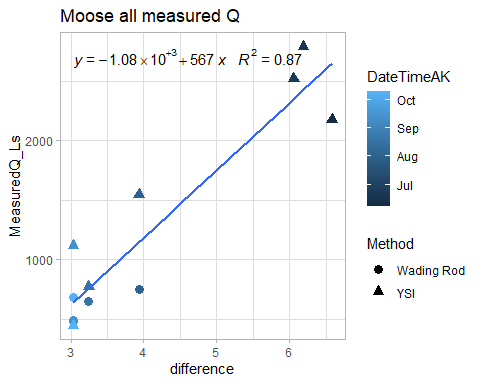
##   
## Call:  
## lm(formula = Moose1comb.2022$MeasuredQ\_Ls ~ Moose1comb.2022$difference)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -469.93 -166.50 -49.56 222.40 476.08   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1083.7 293.4 -3.694 0.00415 \*\*   
## Moose1comb.2022$difference 567.2 69.1 8.209 9.39e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 320.3 on 10 degrees of freedom  
## (2 observations deleted due to missingness)  
## Multiple R-squared: 0.8708, Adjusted R-squared: 0.8578   
## F-statistic: 67.38 on 1 and 10 DF, p-value: 9.392e-06

# plot rating curve   
moos.formula = y~x  
  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Moose1comb.2022) +  
 geom\_point(aes(color = DateTimeAK, shape = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = moos.formula) +  
 stat\_poly\_eq(formula = moos.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 theme\_light() +  
 ggtitle("Moose all measured Q")

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 2 rows containing non-finite values (stat\_poly\_eq).

## Warning: Removed 2 rows containing missing values (geom\_point).



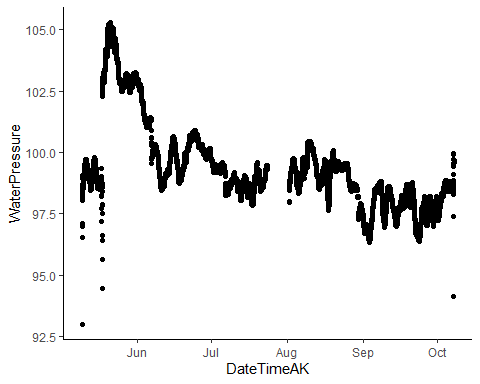
We have all 6 wading rod points here

### POKE

poke.stream.one.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vR8p1pdbkQBMo54XaZinGwxyzojCnlkXeZid3EFvmt9v31PFvpsa1DlWClj8aG0kkJIuU57WXl-cq7Q/pub?output=csv"  
poke.stream.two.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vR-wJjf3rO8eC7bMpQZgwqFLHFQpf5qCInyWtUF1PqaCB8Z\_EoM1cTHRUmWypjftREIx1rP0V6zYJxL/pub?output=csv"  
  
# load in url  
poke.stream.one.2022 <- read.csv(url(poke.stream.one.2022.url), skip = 1)   
poke.stream.two.2022 <- read.csv(url(poke.stream.two.2022.url), skip = 1)   
  
# cleaning df to be able to interpret and merge  
poke.stream.one.2022 <- poke.stream.one.2022[, -c(5:18)] # removing columns that arent date/abs pressure and temp  
  
poke.stream.two.2022 <- poke.stream.two.2022[, -c(5:9)] # removing columns that arent date/abs pressure and temp  
  
# changing to AK time (It reads in as GMT but it is actually AKST )  
poke.stream.one.2022$DateTimeAK <- mdy\_hms(poke.stream.one.2022$Date.Time..GMT.08.00)  
  
poke.stream.two.2022$DateTimeAK <- mdy\_hms(poke.stream.two.2022$Date.Time..GMT.08.00)  
  
# round date to 5 minute intervals  
poke.stream.one.2022$DateTimeAK <- lubridate::round\_date(poke.stream.one.2022$DateTimeAK, "5 minutes")  
poke.stream.two.2022$DateTimeAK <- lubridate::round\_date(poke.stream.two.2022$DateTimeAK, "5 minutes")  
  
# cleaning off original datetime   
poke.stream.one.2022 <- poke.stream.one.2022[, -c(2)] # removing columns that aren't date/abs pressure and temp  
poke.stream.two.2022 <- poke.stream.two.2022[, -c(2)] # removing columns that arent   
  
  
names(poke.stream.one.2022) <- c("Site", "WaterPressure", "TempC", "DateTimeAK")  
names(poke.stream.two.2022) <- c("Site", "WaterPressure", "TempC", "DateTimeAK")

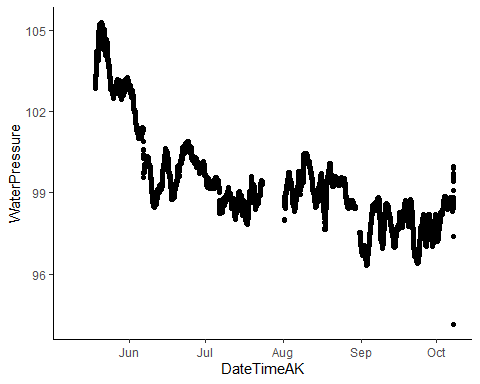
ggplot(poke.stream.one.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 15 rows containing missing values (geom\_point).

 This is PT1….step change is air pressure and then placed in the stream. Let me clip that out

#adjusting/cleaning poke.stream.one  
poke.stream.one.2022 <- poke.stream.one.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-05-09" & DateTimeAK <= "2022-05-18", NA, .))) # PT wasnt placed into the stream until 5/17  
  
poke.stream.one.2022 <- poke.stream.one.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-08-27" & DateTimeAK <= "2022-08-31" & WaterPressure < 98.3, NA, .))) %>% filter( DateTimeAK <= "2022-10-07 15:30:00")  
  
poke.stream.one.2022 <- poke.stream.one.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-10-05" & DateTimeAK <= "2022-10-07" & WaterPressure < 98, NA, .)))  
  
ggplot(poke.stream.one.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

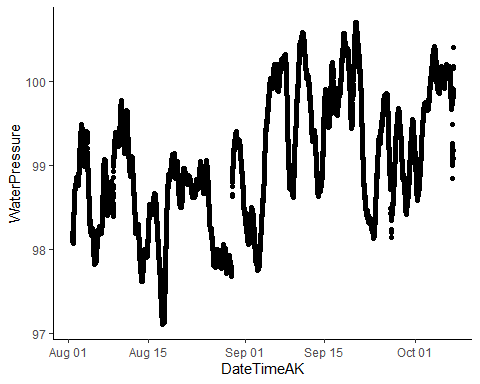
## Warning: Removed 3088 rows containing missing values (geom\_point).



Clean PT2

poke.stream.two.2022 <- poke.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-09-17" & WaterPressure < 98, NA, .))) %>%  
 filter(DateTimeAK <= "2022-10-07 15:35:00")  
  
ggplot(poke.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

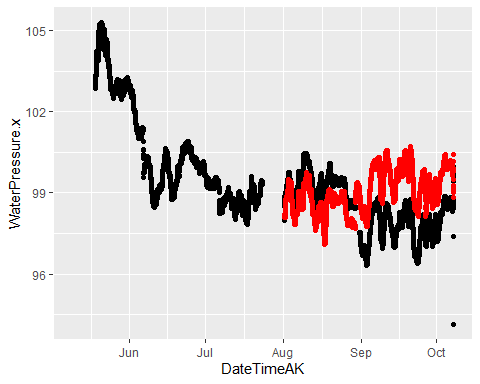
## Warning: Removed 26 rows containing missing values (geom\_point).

 This is PT2…we only have data from 8/1 because of our musical chairs situation. We got two PTs back from onset so we deployed a second PT at POKE.

# merge to one   
poke.final.pressure.2022 <- left\_join(poke.stream.one.2022, poke.stream.two.2022, by = c("DateTimeAK"))  
poke.final.pressure.2022$MeanPressure <- rowMeans(poke.final.pressure.2022[,c(2,6)], na.rm = TRUE)  
  
poke.final.pressure.2022 %>% ggplot() + geom\_point(aes(DateTimeAK, WaterPressure.x)) + geom\_point(aes(DateTimeAK, WaterPressure.y), color = "red")

## Warning: Removed 3100 rows containing missing values (geom\_point).

## Warning: Removed 21718 rows containing missing values (geom\_point).

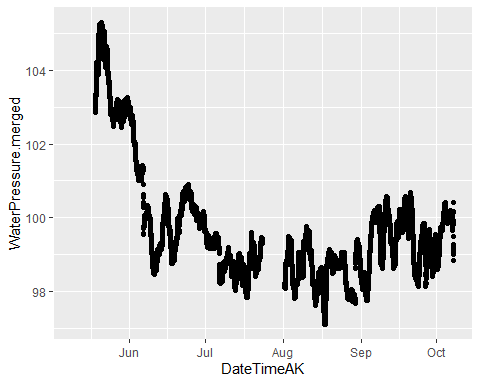
 It looks like we should use PT2 before August 1st (black) and PT1 after August 1st (red).

## Use PT2 before August 1st and PT1 after August 1st  
poke.final.pressure.2022 <- poke.final.pressure.2022 %>%  
 mutate(WaterPressure.merged = ifelse(DateTimeAK <= "2022-08-01", WaterPressure.x, WaterPressure.y))   
  
any(!is.na(poke.final.pressure.2022$WaterPressure.merged))

## [1] TRUE

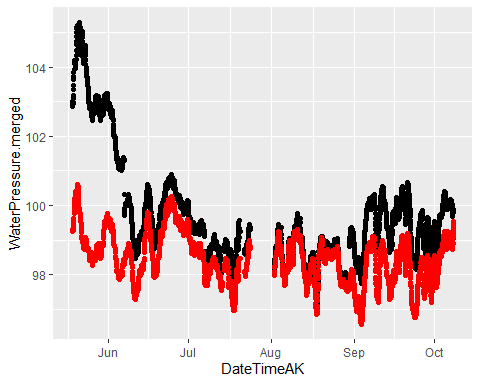
poke.final.pressure.2022 %>% ggplot() + geom\_point(aes(DateTimeAK, WaterPressure.merged))

## Warning: Removed 2604 rows containing missing values (geom\_point).



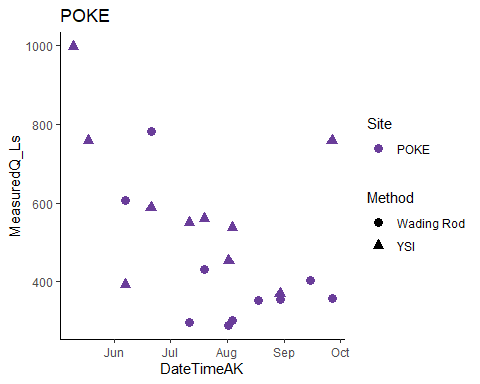
# I'm too irritated to fix time zone problem right now so I will do it manually  
poke.final.pressure.2022.2 <- poke.final.pressure.2022 %>% mutate(DateTimeAK = DateTimeAK + 28800\*2) #Shift time by 16 hours - twice the difference between AK time and GMT...  
  
# Join the two atmospheric and water pressure together  
POKE.2022 <- left\_join(eielson.atmo.2022.compare, poke.final.pressure.2022.2, by = "DateTimeAK") %>% filter(!WaterPressure.merged %in% NA)  
  
# Water pressure - atmospheric pressure   
POKE.2022$difference <- POKE.2022$WaterPressure.merged - POKE.2022$AirPressureCorrectedPOKE  
  
  
# Plot air and water pressure  
POKE.2022 %>% #filter(DateTimeAK > "2022-08-01" & DateTimeAK < "2022-08-15") %>%  
 ggplot(aes(DateTimeAK, WaterPressure.merged)) + geom\_point() + geom\_point(aes(DateTimeAK, AirPressureCorrectedPOKE), color = "red")

## Warning: Removed 86 rows containing missing values (geom\_point).



### Filter POKE ###  
QSummary.PO.2022 <- QSummary.2022 %>% filter(Site =="POKE")  
  
ggplot(QSummary.PO.2022) +  
 geom\_point(aes(x = DateTimeAK, y = MeasuredQ\_Ls, shape = Method, color = Site), size=3) +  
 theme\_classic() +  
 scale\_color\_manual(values=c("#6A3D9A")) +   
 ggtitle("POKE")

## Warning: Removed 2 rows containing missing values (geom\_point).



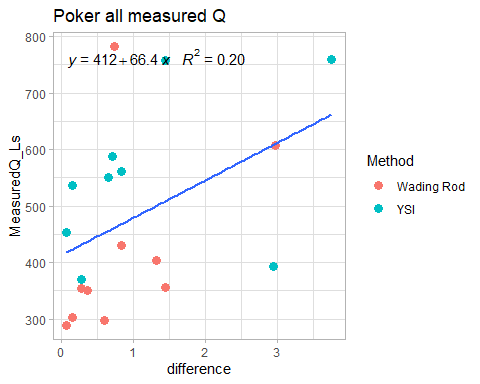
# Merge by nearest date if we have an offset point   
POKE.2022.dt <- setDT(POKE.2022)  
QSummary.PO.2022.dt <- QSummary.PO.2022 %>% filter(DateTimeAK > "2022-05-17")  
  
Poker1comb.2022 <- POKE.2022.dt[QSummary.PO.2022.dt, on = "DateTimeAK", roll = 'nearest']  
  
#Poker1comb.2022[15,9] <- 0.2580529  
  
POKE1.lm.2022 <- lm(Poker1comb.2022$MeasuredQ\_Ls ~ Poker1comb.2022$difference)

### Rating Curve ###  
poke.formula <- y ~ x  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Poker1comb.2022) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = poke.formula) +  
 stat\_poly\_eq(formula = poke.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 # xlim(-2, 2) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Poker all measured Q")

## Warning: Removed 1 rows containing non-finite values (stat\_smooth).

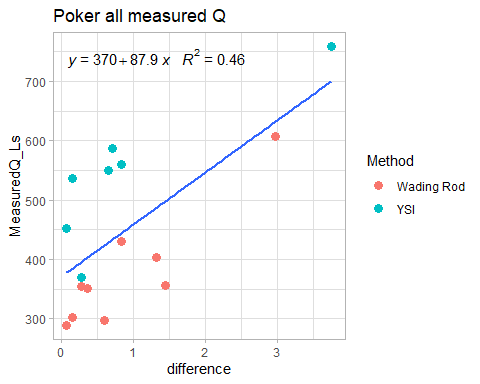
## Warning: Removed 1 rows containing non-finite values (stat\_poly\_eq).

## Warning: Removed 1 rows containing missing values (geom\_point).

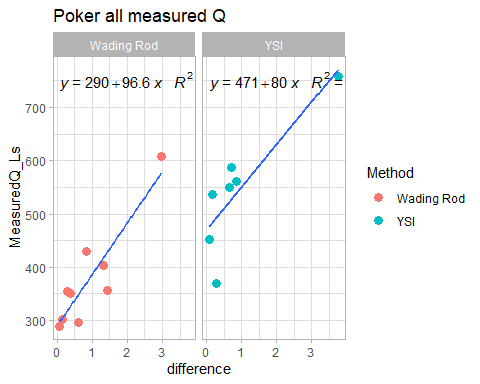
 This is the rating curve. We are missing 5/9 and 5/17 YSI because we didn’t have our Pressure Transducers in yet.

I will remove three points.

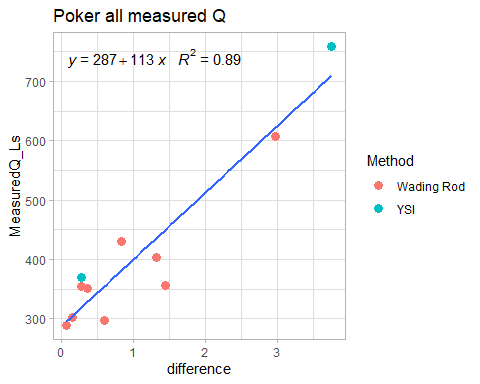
# Remove three points  
Poker1comb.2022.2 <- Poker1comb.2022 %>% filter(MeasuredQ\_Ls < 700 | difference > 3)%>% filter(MeasuredQ\_Ls > 400 | difference < 2)  
### Rating Curve ###  
poke.formula <- y ~ x  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Poker1comb.2022.2) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = poke.formula) +  
 stat\_poly\_eq(formula = poke.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 # xlim(-2, 2) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Poker all measured Q")

 I will try curves for wading rod and YSI seperately

### Rating Curve ###  
poke.formula <- y ~ x  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Poker1comb.2022.2) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = poke.formula) +  
 stat\_poly\_eq(formula = poke.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 theme\_light() +  
 ggtitle("Poker all measured Q") + facet\_wrap(~Method)

 The wading rod curve looks better than the YSI curve. I will try adding just a couple YSI points to the wading rod curve.

Poker1comb.2022.3 <- Poker1comb.2022.2 %>% filter(MeasuredQ\_Ls < 450 | difference > 2)  
### Rating Curve ###  
poke.formula <- y ~ x  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Poker1comb.2022.3) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = poke.formula) +  
 stat\_poly\_eq(formula = poke.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 theme\_light() +  
 ggtitle("Poker all measured Q")

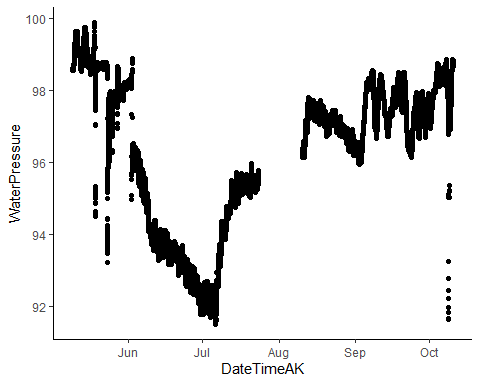
 That looks nice! Did we cheat at all?

### STRT

strt.stream.one.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vS4XSmuumKD-j1aDhpA1gfyfrpqUiJUTuvcP9UGrgagvIrzlGYWk71hl0zAC8g8GHqOm1ONjWOpO\_So/pub?output=csv" # WL  
strt.stream.two.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vRhNn42-Jr1Y3aTFfMPiRxKrNZZjHeBBxkCmVcLVPQEKBf4qg\_1Pw4nyUBr3mDwtEE8NKGIbS7kNSrS/pub?output=csv" #WR  
strt.atmo.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vQxotkNZ6zyvUi51XOjNPumqSvwGfsZMJc7I6-QXBELHY7msBzAaqvEHV41A88Lt0I1Ga-l1343Q1cr/pub?output=csv"  
# load in data   
strt.stream.one.2022 <- read.csv(url(strt.stream.one.2022.url), skip = 1)   
strt.stream.two.2022 <- read.csv(url(strt.stream.two.2022.url), skip = 1)   
strt.atmo.2022 <- read.csv(url(strt.atmo.2022.url), skip = 1)   
  
# clean for merging purposes  
strt.stream.one.2022 <- strt.stream.one.2022[, -c(4,6:19)] # removing columns that arent date/abs pressure and temp  
  
strt.stream.two.2022 <- strt.stream.two.2022[, -c(5:12)] # removing columns that arent date/abs pressure and temp  
  
strt.atmo.2022 <- strt.atmo.2022[,-c(4:9)]  
  
# changing to AK time (It reads in as GMT but it is actually AKST )  
strt.stream.one.2022$DateTimeAK <- mdy\_hms(strt.stream.one.2022$Date.Time..GMT.08.00)  
  
strt.stream.two.2022$DateTimeAK <- mdy\_hms(strt.stream.two.2022$Date.Time..GMT.08.00)  
strt.atmo.2022$DateTimeAK <- mdy\_hms(strt.atmo.2022$Date.Time..GMT.08.00)  
  
# round date to 5 minute intervals  
strt.stream.one.2022$DateTimeAK <- lubridate::round\_date(strt.stream.one.2022$DateTimeAK, "5 minutes")  
strt.stream.two.2022$DateTimeAK <- lubridate::round\_date(strt.stream.two.2022$DateTimeAK, "5 minutes")  
  
# cleaning off original datetime   
strt.stream.one.2022 <- strt.stream.one.2022[, -c(2)] # removing columns that aren't date/abs pressure and temp  
strt.stream.two.2022 <- strt.stream.two.2022[, -c(2)] # removing columns that arent   
strt.atmo.2022 <- strt.atmo.2022[,-c(2)]  
  
names(strt.stream.one.2022) <- c("Site","WaterPressure", "TempC", "DateTimeAK")  
  
names(strt.stream.two.2022) <- c("Site","WaterPressure", "TempC", "DateTimeAK")  
  
names(strt.atmo.2022) <- c("Site", "AirPressureSTRT", "DateTimeAK")

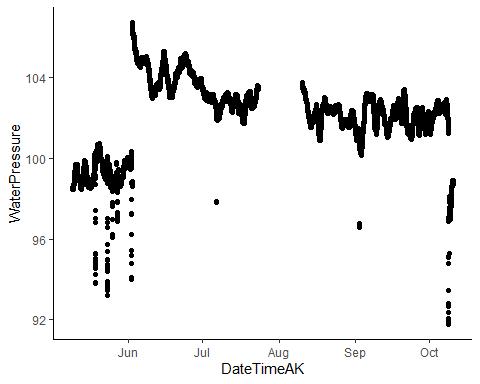
ggplot(strt.stream.one.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 16 rows containing missing values (geom\_point).

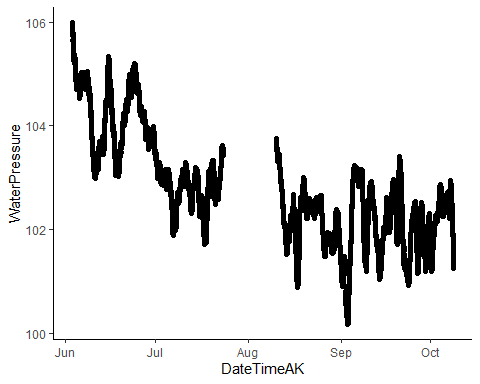
 This is PT1 for STRT raw. Looking at how low the PT data is I am NOT going to trust this one and take it out of our analysis

ggplot(strt.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 6 rows containing missing values (geom\_point).

 This is raw PT2 for STRT. I need to remove out of water points (anything before 6/2) and clean any erroneous points

#adjusting/cleaning STRT 2  
strt.stream.two.2022 <- strt.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-05-09" & DateTimeAK <= "2022-06-03", NA, .))) # PT wasnt placed into the stream until 6/3  
  
strt.stream.two.2022 <- strt.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-05-09" & DateTimeAK <= "2022-10-03" &  
 WaterPressure < 100, NA, .))) %>% # PT wasnt placed into the stream until 6/3  
 filter(WaterPressure > 100)  
  
ggplot(strt.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

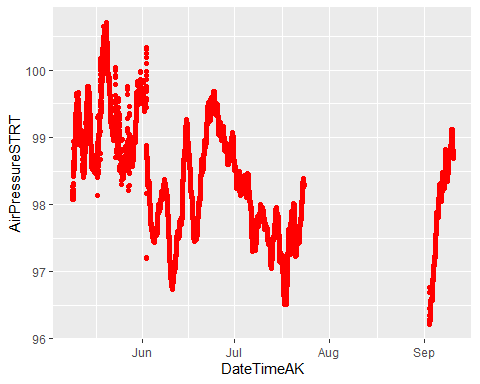
 This looks better

# merge to one   
strt.final.pressure.2022 <- strt.stream.two.2022 # the first PT is crap

Clean and plot our measured air pressure

strt.atmo.2022 <- strt.atmo.2022 %>% filter(AirPressureSTRT > 96)  
  
strt.atmo.2022 <- strt.atmo.2022 %>%  
 mutate(across(c(AirPressureSTRT),   
 ~ifelse(DateTimeAK <= "2022-06-01" & AirPressureSTRT < 98, NA, .))) %>% filter( DateTimeAK <= "2022-09-10")  
  
strt.atmo.2022 %>% ggplot() + geom\_point(aes(DateTimeAK, AirPressureSTRT), color = "red")

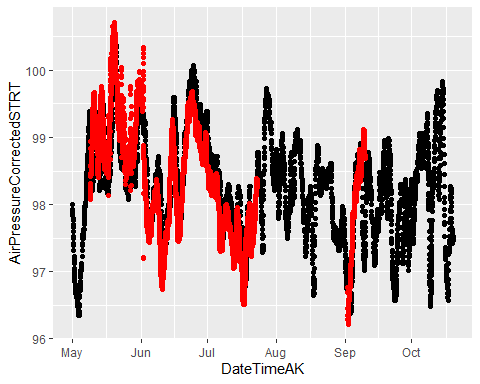
## Warning: Removed 77 rows containing missing values (geom\_point).



# Plot STRT measured and Eilson air pressure  
STRT.atmo.2022.compare <- full\_join(eielson.atmo.2022.compare, strt.atmo.2022, by = "DateTimeAK")  
  
STRT.atmo.2022.compare %>% filter(DateTimeAK > "2022-05-01") %>% ggplot() + geom\_point(aes(DateTimeAK, AirPressureCorrectedSTRT)) +   
 geom\_point(aes(DateTimeAK, AirPressureSTRT), color = "red")

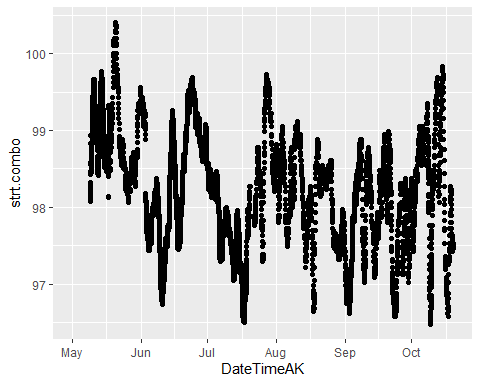
## Warning: Removed 21312 rows containing missing values (geom\_point).

## Warning: Removed 3461 rows containing missing values (geom\_point).



## Use measured air pressure when possible  
STRT.atmo.2022.compare <- STRT.atmo.2022.compare %>% select(DateTimeAK, AirPressureCorrectedSTRT, AirPressureSTRT )  
  
 ### Using mostly our air pressure data with gap filled in by Eilson:  
STRT.atmo.2022.compare$strt.combo <- c()  
STRT.atmo.2022.compare<- STRT.atmo.2022.compare %>%  
 mutate(strt.combo = ifelse(DateTimeAK <= "2022-05-18" | DateTimeAK >= "2022-06-03", AirPressureSTRT, AirPressureCorrectedSTRT))  
  
STRT.atmo.2022.compare<- STRT.atmo.2022.compare %>%  
 mutate(across(c(strt.combo),   
 ~ifelse(DateTimeAK >= "2022-07-18" & DateTimeAK <= "2022-09-03" | DateTimeAK >= "2022-09-08", AirPressureCorrectedSTRT, .)))  
  
STRT.atmo.2022.compare %>% filter(DateTimeAK > "2022-05-01") %>% ggplot() + geom\_point(aes(DateTimeAK, strt.combo)) #+ geom\_point(aes(DateTimeAK, AirPressureCorrectedSTRT), color = "red", cex = 0.5)

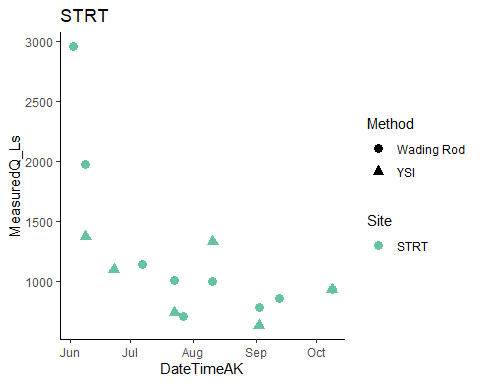
## Warning: Removed 6516 rows containing missing values (geom\_point).



# join the two atmospheric and water pressure together  
STRT.2022 <- left\_join(STRT.atmo.2022.compare, strt.final.pressure.2022, by = "DateTimeAK")  
  
# Water pressure - atmospheric pressure   
STRT.2022$difference <- STRT.2022$WaterPressure - STRT.2022$strt.combo  
  
STRT.2022 <- STRT.2022[ , -which(names(STRT.2022) %in% c("TempC"))] # removing columns I do not need  
STRT.2022$Site <- "STRT"

### Filter STRT ###  
QSummary.ST.2022 <- QSummary.2022 %>% filter(Site =="STRT")  
  
ggplot(QSummary.ST.2022) +  
 geom\_point(aes(x = DateTimeAK, y = MeasuredQ\_Ls, shape = Method, color = Site), size=3) +  
 theme\_classic() +  
 scale\_color\_manual(values=c("#66C2A5")) +   
 ggtitle("STRT")

## Warning: Removed 3 rows containing missing values (geom\_point).



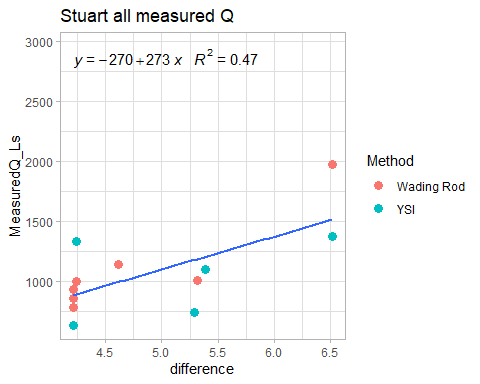
# trying to merge by nearest date if we have an offset point   
STRT.2022.dt <- setDT(STRT.2022)  
STRT.2022.dt <- subset(STRT.2022.dt, DateTimeAK < "2022-09-02 05:00:00") # removing rows that had dates corresponding to end of record that messed up the rolling nearest function   
QSummary.ST.2022.dt <- QSummary.ST.2022  
  
Stuart1comb.2022 <- STRT.2022.dt[QSummary.ST.2022.dt, on = "DateTimeAK", roll = 'nearest']  
Stuart1comb.2022[c(1,2),9] <- 6.463909 # merging by nearest difference value manually  
Stuart1comb.2022[c(10,11),9] <- 3.910157  
Stuart1comb.2022[c(12,13),9] <- 3.714323  
  
  
STRT1.lm.2022 <- lm(Stuart1comb.2022$MeasuredQ\_Ls ~ Stuart1comb.2022$difference)

### Filter STRT ###  
strt.formula <- y ~ x  
  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Stuart1comb.2022) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = strt.formula) +  
 stat\_poly\_eq(formula = strt.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Stuart all measured Q")

## Warning: Removed 5 rows containing non-finite values (stat\_smooth).

## Warning: Removed 5 rows containing non-finite values (stat\_poly\_eq).

## Warning: Removed 5 rows containing missing values (geom\_point).

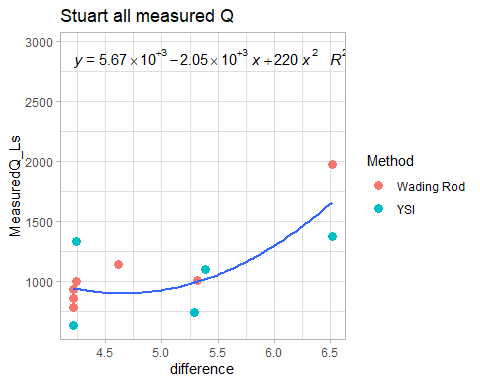
 Stuart looks like a quadratic relationship might fit better

STRT1.lm.2022 <- lm(Stuart1comb.2022$MeasuredQ\_Ls ~ Stuart1comb.2022$difference + Stuart1comb.2022$difference^2)  
  
strt.formula <- y ~ poly(x, 2, raw = TRUE)  
  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Stuart1comb.2022) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = strt.formula) +  
 stat\_poly\_eq(formula = strt.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Stuart all measured Q")

## Warning: Removed 5 rows containing non-finite values (stat\_smooth).

## Warning: Removed 5 rows containing non-finite values (stat\_poly\_eq).

## Warning: Removed 5 rows containing missing values (geom\_point).

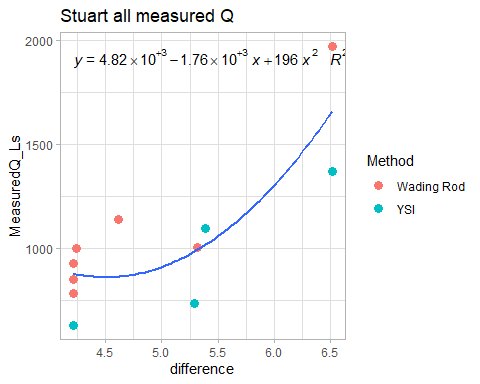
 Lefts try removing that low difference, high measured Q YSI point:

Stuart1comb.2022 <- Stuart1comb.2022 %>% filter(difference > 4.5 | MeasuredQ\_Ls <1250)  
STRT1.lm.2022 <- lm(Stuart1comb.2022$MeasuredQ\_Ls ~ Stuart1comb.2022$difference + Stuart1comb.2022$difference^2)  
  
strt.formula <- y ~ poly(x, 2, raw = TRUE)  
  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = Stuart1comb.2022) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = strt.formula) +  
 stat\_poly\_eq(formula = strt.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Stuart all measured Q")

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 2 rows containing non-finite values (stat\_poly\_eq).

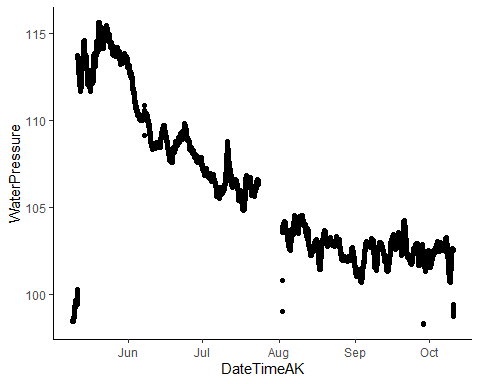
## Warning: Removed 2 rows containing missing values (geom\_point).

 ### FRCH ###

frch.stream.one.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vRkT5GIhFY4nLblzPi4Upc8y6Pk2k4vXLSGAkFyMhLowsCmct0fZ5rxm-vd85RfFr7YLrCuuSiOqQbL/pub?output=csv"  
frch.stream.two.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vRiDAbzBj\_jHk1Hd3iM7fFvzMrRCSF9fbKTcmQD\_dUFGOjoIQuBZBKV4KdNhfevLOq0Udch0LbWiKXw/pub?output=csv"  
  
# load in url  
frch.stream.one.2022 <- read.csv(url(frch.stream.one.2022.url), skip = 1)   
frch.stream.two.2022 <- read.csv(url(frch.stream.two.2022.url), skip = 1)   
  
# clean for merging purposes  
frch.stream.one.2022 <- frch.stream.one.2022[, -c(5:13)] # removing columns that arent date/abs pressure and temp  
  
frch.stream.two.2022 <- frch.stream.two.2022[, -c(5:14)] # removing columns that arent date/abs pressure and temp  
  
# changing to AK time (It reads in as GMT but it is actually AKST )  
frch.stream.one.2022$DateTimeAK <- mdy\_hms(frch.stream.one.2022$Date.Time..GMT.08.00)  
  
frch.stream.two.2022$DateTimeAK <- mdy\_hms(frch.stream.two.2022$Date.Time..GMT.08.00)  
  
# round date to 5 minute intervals  
frch.stream.one.2022$DateTimeAK <- lubridate::round\_date(frch.stream.one.2022$DateTimeAK, "15 minutes")  
frch.stream.two.2022$DateTimeAK <- lubridate::round\_date(frch.stream.two.2022$DateTimeAK, "15 minutes")  
  
# cleaning off original datetime   
frch.stream.one.2022 <- frch.stream.one.2022[, -c(2)] # removing columns that aren't date/abs pressure and temp  
frch.stream.two.2022 <- frch.stream.two.2022[, -c(2)] # removing columns that arent   
  
names(frch.stream.one.2022) <- c("Site", "WaterPressure", "TempC", "DateTimeAK")  
  
names(frch.stream.two.2022) <- c("Site", "WaterPressure", "TempC", "DateTimeAK")  
  
  
# merge to one   
frch.final.pressure.2022 <- left\_join(frch.stream.one.2022, frch.stream.two.2022, by = c("DateTimeAK"))  
frch.final.pressure.2022$MeanPressure <- rowMeans(frch.final.pressure.2022[,c(2,6)], na.rm = TRUE)

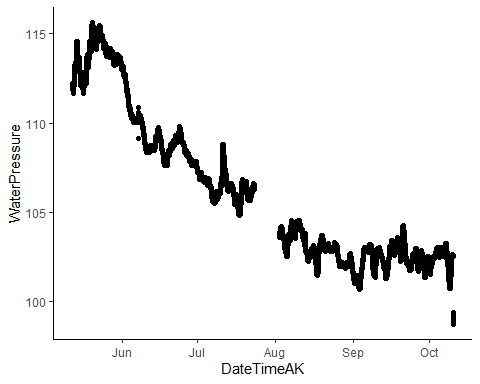
ggplot(frch.stream.one.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 15 rows containing missing values (geom\_point).



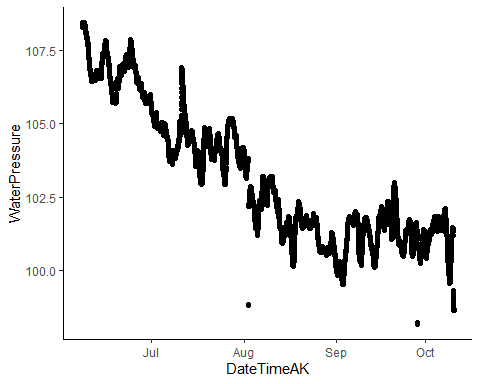
This looks like the best one yet…we need to remove the beginning out of water points

frch.stream.one.2022 <- frch.stream.one.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-05-09" & DateTimeAK <= "2022-05-12", NA, .))) # PT wasnt placed into the stream until 5/12  
  
frch.stream.one.2022 <- frch.stream.one.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-08-01" & DateTimeAK <= "2022-08-15" &  
 WaterPressure < 102, NA, .))) %>% filter(WaterPressure > 98.4) # PT wasnt placed into the stream until 5/12  
  
ggplot(frch.stream.one.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

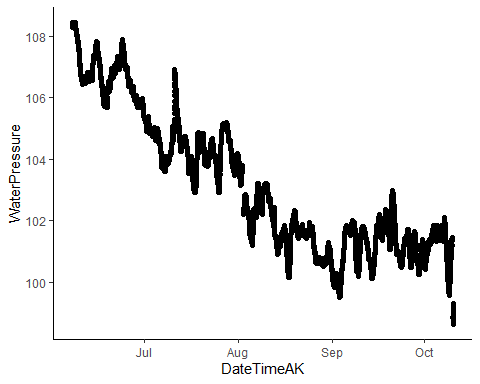
 This looks better

ggplot(frch.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 18 rows containing missing values (geom\_point).

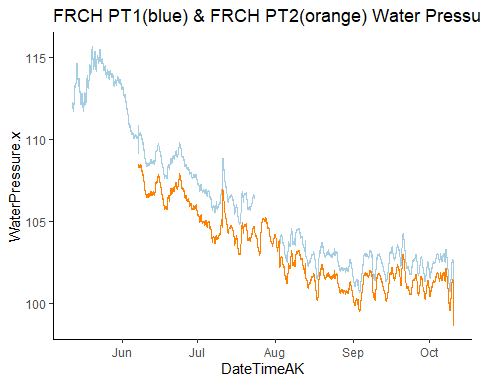
 PT2 also looks good…let me remove that one point in august that is below everything else

frch.stream.two.2022 <- frch.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-08-01" & DateTimeAK <= "2022-08-15" &  
 WaterPressure < 100, NA, .))) %>% filter(WaterPressure > 98.4)# PT wasnt placed into the stream until 5/12  
  
ggplot(frch.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()



# Checking closeness between two PTs  
atmo.pt.frch <- full\_join(frch.stream.one.2022, frch.stream.two.2022, by = "DateTimeAK")  
  
ggplot(aes(x = DateTimeAK, y = WaterPressure.x), data = atmo.pt.frch) +  
 geom\_line(aes(x=DateTimeAK, y=WaterPressure.x), data = atmo.pt.frch, color="#A6CEE3") +  
 geom\_line(aes(x=DateTimeAK, y=WaterPressure.y), data = atmo.pt.frch, color="#FF7F00") +  
 theme\_classic() +  
 ggtitle("FRCH PT1(blue) & FRCH PT2(orange) Water Pressure")

## Warning: Removed 7533 row(s) containing missing values (geom\_path).



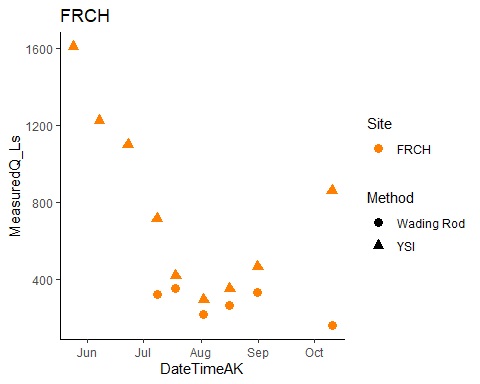
# They look the same and PT1 is more complete, so we will use that one.

frch.final.pressure.2022 <- frch.stream.one.2022

# join the two atmospheric and water pressure together  
FRCH.2022 <- left\_join(eielson.atmo.2022.compare, frch.final.pressure.2022, by = "DateTimeAK")  
  
  
# Water pressure - atmospheric pressure   
FRCH.2022$difference <- FRCH.2022$WaterPressure - FRCH.2022$AirPressureCorrectedFRCH  
  
names(FRCH.2022)[names(FRCH.2022) == 'Site.x'] <- 'Site'  
  
FRCH.2022$Site <- "FRCH"

### Filter FRCH ###  
QSummary.FR.2022 <- QSummary.2022 %>% filter(Site =="FRCH")  
  
ggplot(QSummary.FR.2022) +  
 geom\_point(aes(x = DateTimeAK, y = MeasuredQ\_Ls, shape = Method, color = Site), size=3) +  
 theme\_classic() +  
 scale\_color\_manual(values=c("#FF7F00")) +   
 ggtitle("FRCH")

## Warning: Removed 1 rows containing missing values (geom\_point).



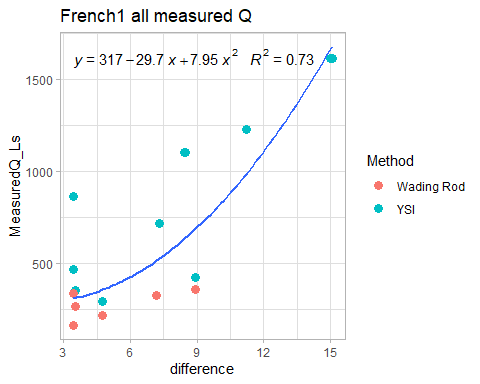
# trying to merge by nearest date if we have an offset point   
FRCH.2022.dt <- setDT(FRCH.2022)  
FRCH.2022.dt <- subset(FRCH.2022.dt, DateTimeAK < "2022-08-31 03:00:00") # removing rows that had dates corresponding to end of record that messed up the rolling nearest function   
QSummary.FR.2022.dt <- QSummary.FR.2022  
  
French1comb.2022 <- FRCH.2022.dt[QSummary.FR.2022.dt, on = "DateTimeAK", roll = 'nearest']  
  
FRCH1.lm.2022 <- lm(French1comb.2022$MeasuredQ\_Ls ~ French1comb.2022$difference)

# rating curve #   
frch.formula <- y ~ poly(x, 2, raw = TRUE)  
  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = French1comb.2022) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = frch.formula) +  
 stat\_poly\_eq(formula = frch.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("French1 all measured Q")

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).

## Warning: Removed 3 rows containing non-finite values (stat\_poly\_eq).

## Warning: Removed 3 rows containing missing values (geom\_point).

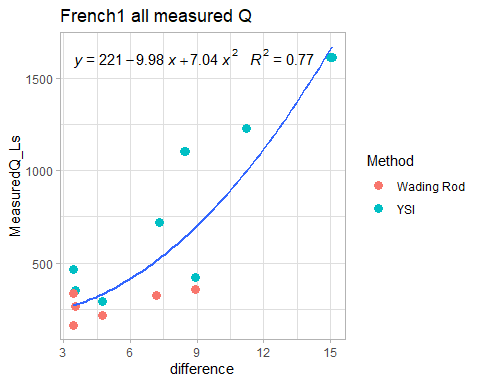
 Lets try removing the low difference, high measured Q point:

French1comb.2022 <- French1comb.2022 %>% filter(difference > 4 | MeasuredQ\_Ls < 500)  
  
# rating curve #   
frch.formula <- y ~ poly(x, 2, raw = TRUE)  
  
ggplot(aes(x = difference, y = MeasuredQ\_Ls), data = French1comb.2022) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = frch.formula) +  
 stat\_poly\_eq(formula = frch.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("French1 all measured Q")

## Warning: Removed 3 rows containing non-finite values (stat\_smooth).

## Warning: Removed 3 rows containing non-finite values (stat\_poly\_eq).

## Warning: Removed 3 rows containing missing values (geom\_point).

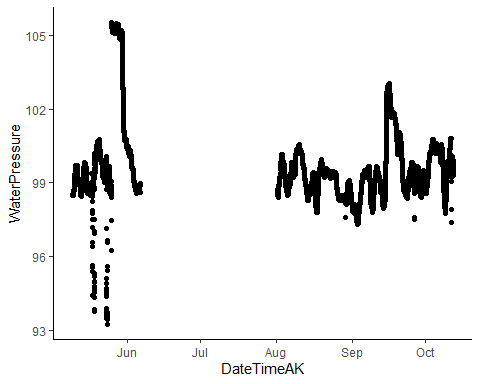


### VAUL

vaul.stream.one.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vQf78OhtBtjvs6HmnEVPJ0PVYSlpIjVAom5VIv12LhZDMbezaFb-LJCGMZUoyVjLJI-pqizM-Be2hbl/pubhtml"  
vaul.stream.two.2022.url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vSR7qqdF5BXklT1ocG9bLLUwJ07ha8qmzAVRPaUjm7mUq12VptmViI9NJcW1-jO4cu0KtHBLl90A4DZ/pub?output=csv"  
  
# load in url  
#vaul.stream.one.2022 <- read.csv(url(vaul.stream.one.2022.url), skip = 1)   
vaul.stream.two.2022 <- read.csv(url(vaul.stream.two.2022.url), skip = 1)   
  
# clean for merging purposes  
#vaul.stream.one.2022 <- vaul.stream.one.2022[, -c(5:13)] # removing columns that arent date/abs pressure and temp  
  
vaul.stream.two.2022 <- vaul.stream.two.2022[, -c(5:13)] # removing columns that arent date/abs pressure and temp  
  
# changing to AK time (It reads in as GMT but it is actually AKST )  
  
vaul.stream.two.2022$DateTimeAK <- mdy\_hms(vaul.stream.two.2022$Date.Time..GMT.08.00)  
  
vaul.stream.two.2022$DateTimeAK <- force\_tz(vaul.stream.two.2022$DateTimeAK, "America/Anchorage")  
  
# round date to 5 minute intervals  
#vaul.stream.one.2022$DateTimeAK <- lubridate::round\_date(vaul.stream.one.2022$DateTimeAK, "5 minutes")  
vaul.stream.two.2022$DateTimeAK <- lubridate::round\_date(vaul.stream.two.2022$DateTimeAK, "5 minutes")  
  
# cleaning off original datetime   
#vaul.stream.one.2022 <- vaul.stream.one.2022[, -c(2)] # removing columns that aren't date/abs pressure and temp  
vaul.stream.two.2022 <- vaul.stream.two.2022[, -c(2)] # removing columns that arent   
  
#names(vaul.stream.one.2022) <- c("Site", "WaterPressure", "TempC", "DateTimeAK")  
  
names(vaul.stream.two.2022) <- c("Site", "WaterPressure", "TempC", "DateTimeAK")

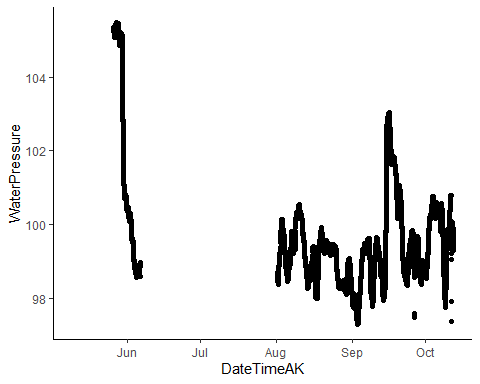
ggplot(vaul.stream.two.2022, aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

## Warning: Removed 14 rows containing missing values (geom\_point).

 This is the raw record of VAUL pressure. The beginning is airpressure and then there is a gap as we had a relaunch issue.

vaul.stream.two.2022 <- vaul.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-05-09" & DateTimeAK <= "2022-05-26", NA, .))) # PT wasnt placed into the stream until 5/26  
  
vaul.stream.two.2022 <- vaul.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-08-15" & DateTimeAK <= "2022-08-31" &  
 WaterPressure < 98, NA, .))) # PT wasnt placed into the stream until 5/12  
  
vaul.stream.two.2022 <- vaul.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-09-30" & DateTimeAK <= "2022-10-01" &  
 WaterPressure < 100, NA, .)))  
  
vaul.stream.two.2022 <- vaul.stream.two.2022 %>%  
 mutate(across(c(WaterPressure),   
 ~ifelse(DateTimeAK >= "2022-10-15" &  
 WaterPressure < 99, NA, .)))  
  
vaul.stream.two.2022 %>% ggplot( aes(x = DateTimeAK, y = WaterPressure))+  
 geom\_point() +  
 theme\_classic()

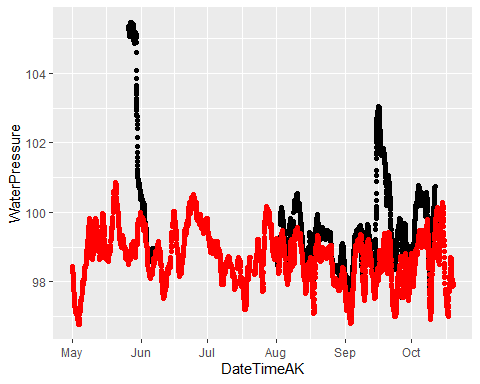
## Warning: Removed 5257 rows containing missing values (geom\_point).

 That is better. Crazy how fast the water level comes down at VAUL

# Join the atmospheric and water pressure together  
vaul.final.pressure.2022 <- vaul.stream.two.2022  
# Eielson atmosphere data  
VAUL.2022 <- left\_join(eielson.atmo.2022.compare, vaul.final.pressure.2022 , by = "DateTimeAK")  
  
# Water pressure - atmospheric pressure   
VAUL.2022$difference <- VAUL.2022$WaterPressure - VAUL.2022$AirPressureCorrectedVAUL  
  
VAUL.2022 <- VAUL.2022[ , -which(names(VAUL.2022) %in% c("TempC"))] # removing columns I do not need  
  
VAUL.2022$Site <- "VAUL"  
  
# Plot air and water pressure  
VAUL.2022 %>% filter(DateTimeAK > "2022-05-01") %>% ggplot(aes(DateTimeAK, WaterPressure)) + geom\_point() + geom\_point(aes(DateTimeAK, AirPressureCorrectedVAUL), color = "red")

## Warning: Removed 2948 rows containing missing values (geom\_point).

## Warning: Removed 115 rows containing missing values (geom\_point).

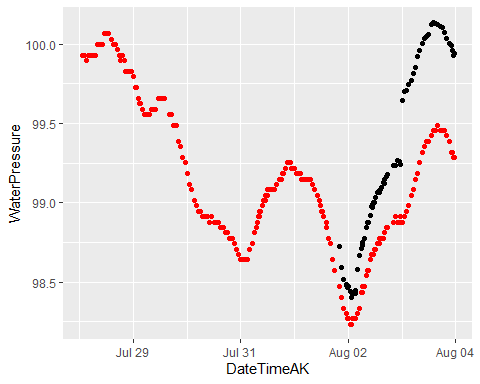


The Eielson air pressure appear shifted… Our pressure date time zone must be off… I can’t find any issue with how the date time zones are reading in. I don’t know what else to try to I will manually shift the time of the pressure data for now.

vaul.final.pressure.2022.2 <- vaul.final.pressure.2022  
vaul.final.pressure.2022.2$DateTimeAK <- vaul.final.pressure.2022.2$DateTimeAK + 28800 #Shift time by 8 hours - the difference between AK time and GMT  
  
VAUL.2022.2 <- left\_join(eielson.atmo.2022.compare, vaul.final.pressure.2022.2, by = "DateTimeAK")  
  
# Water pressure - atmospheric pressure   
VAUL.2022.2$difference <- VAUL.2022.2$WaterPressure - VAUL.2022.2$AirPressureCorrectedVAUL  
  
  
# Plot air and water pressure  
VAUL.2022.2 %>% filter(DateTimeAK > "2022-07-28" & DateTimeAK < "2022-08-04") %>% ggplot(aes(DateTimeAK, WaterPressure)) + geom\_point() + geom\_point(aes(DateTimeAK, AirPressureCorrectedVAUL), color = "red")

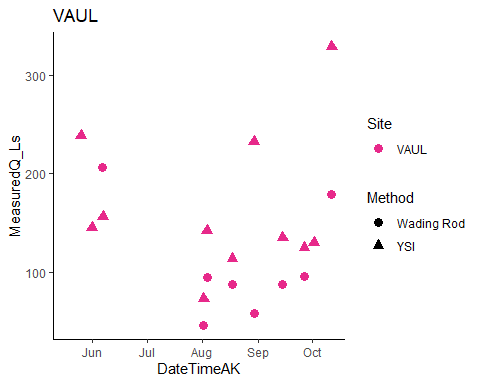
## Warning: Removed 158 rows containing missing values (geom\_point).

## Warning: Removed 3 rows containing missing values (geom\_point).

 That’s better. Onward.

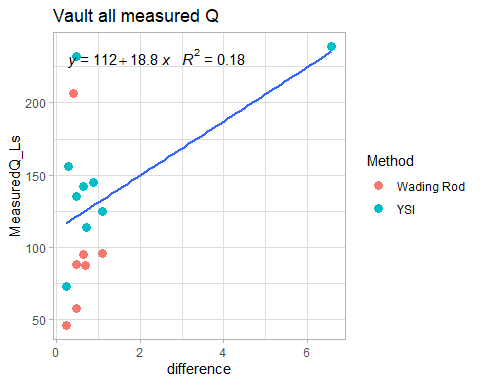
### Filter VAUL ###  
# Remove measurements during pressure gap!  
QSummary.VA.2022 <- QSummary.2022 %>% filter(Site =="VAUL") %>% filter(DateTimeAK < "2022-06-07" | DateTimeAK > "2022-08-01")  
  
ggplot(QSummary.VA.2022) +  
 geom\_point(aes(x = DateTimeAK, y = MeasuredQ\_Ls, shape = Method, color = Site), size=3) +  
 theme\_classic() +  
 scale\_color\_manual(values=c("#E7298A")) +   
 ggtitle("VAUL")

## Warning: Removed 1 rows containing missing values (geom\_point).

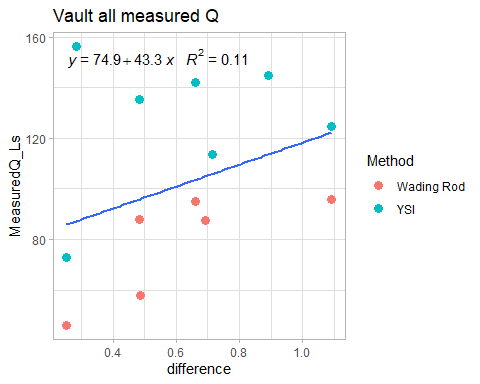


# Merge by nearest date if we have an offset point   
VAUL.2022.dt <- setDT(VAUL.2022.2) %>% filter(!WaterPressure %in% NA)  
QSummary.VA.2022.dt <- QSummary.VA.2022 %>% filter(DateTimeAK > "2022-05-25" & DateTimeAK < "2022-10-01")  
  
Vault1comb.2022 <- VAUL.2022.dt[QSummary.VA.2022.dt, on = "DateTimeAK", roll = 'nearest']  
  
  
VAUL1.lm.2022 <- lm(Vault1comb.2022$MeasuredQ\_Ls ~ Vault1comb.2022$difference)

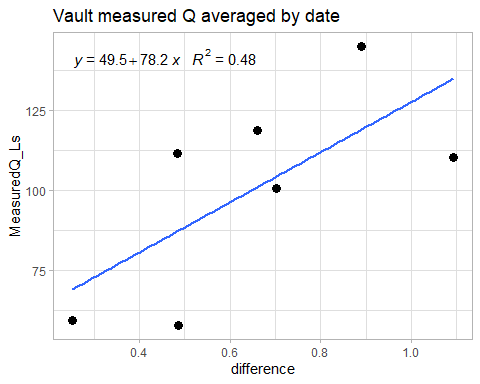
# rating curve #   
vaul.formula <- y ~ x  
  
Vault1comb.2022 %>% ggplot(aes(x = difference, y = MeasuredQ\_Ls)) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = vaul.formula) +  
 stat\_poly\_eq(formula = vaul.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Vault all measured Q")

 That very high difference seems accurate, but necessary to remove for curve. The August 29th point also looks much too high, and 6/6 wading rod was much higher than YSI.

# Remove high difference point  
Vault1comb.2022.2 <- Vault1comb.2022 %>% filter(difference < 6, Notes != "May be high", DateTimeAK != "2022-06-06 10:00:00")  
  
# rating curve #   
vaul.formula <- y ~ x  
  
Vault1comb.2022.2 %>% ggplot(aes(x = difference, y = MeasuredQ\_Ls)) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = vaul.formula) +  
 stat\_poly\_eq(formula = vaul.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 theme\_light() +  
 ggtitle("Vault all measured Q")

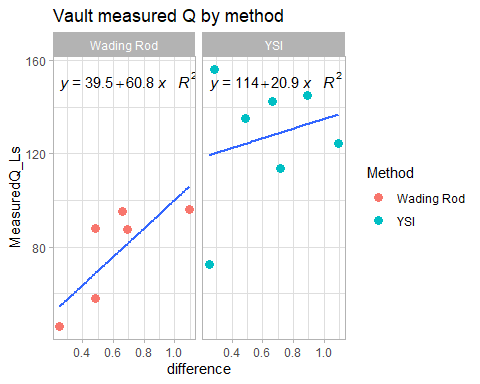
 Let’s try curve with wading rod and YSI averaged by date. I removed one point.

Vault1comb.2022.3 <- Vault1comb.2022.2 %>% group\_by(Date) %>% summarise(MeasuredQ\_Ls = mean(MeasuredQ\_Ls),  
difference = mean(difference))  
  
# rating curve #   
vaul.formula <- y ~ x  
  
Vault1comb.2022.3 %>% filter(MeasuredQ\_Ls < 150) %>% ggplot(aes(x = difference, y = MeasuredQ\_Ls)) +  
 geom\_point(size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = vaul.formula) +  
 stat\_poly\_eq(formula = vaul.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 theme\_light() +  
 ggtitle("Vault measured Q averaged by date")



Let’s try separate curves for YSI and wading rod

# rating curve #   
vaul.formula <- y ~ x  
  
Vault1comb.2022.2 %>% ggplot(aes(x = difference, y = MeasuredQ\_Ls)) +  
 geom\_point(aes(color = Method), size = 3) +  
 geom\_smooth(method = "lm", se=FALSE, formula = vaul.formula) +  
 stat\_poly\_eq(formula = vaul.formula,  
 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),  
 parse = TRUE) +  
 #xlim(216, 216.4) +  
 #ylim(0,1500) +  
 theme\_light() +  
 ggtitle("Vault measured Q by method") + facet\_wrap(~Method)

 It looks like just using only the wading rods creates our best curve.