Box Cox Data Modelling

Jerry Wu

9/19/2023

```
full_data <- readRDS('data/full_data.rds')
daily_full <- readRDS('data/daily_full.rds')
```

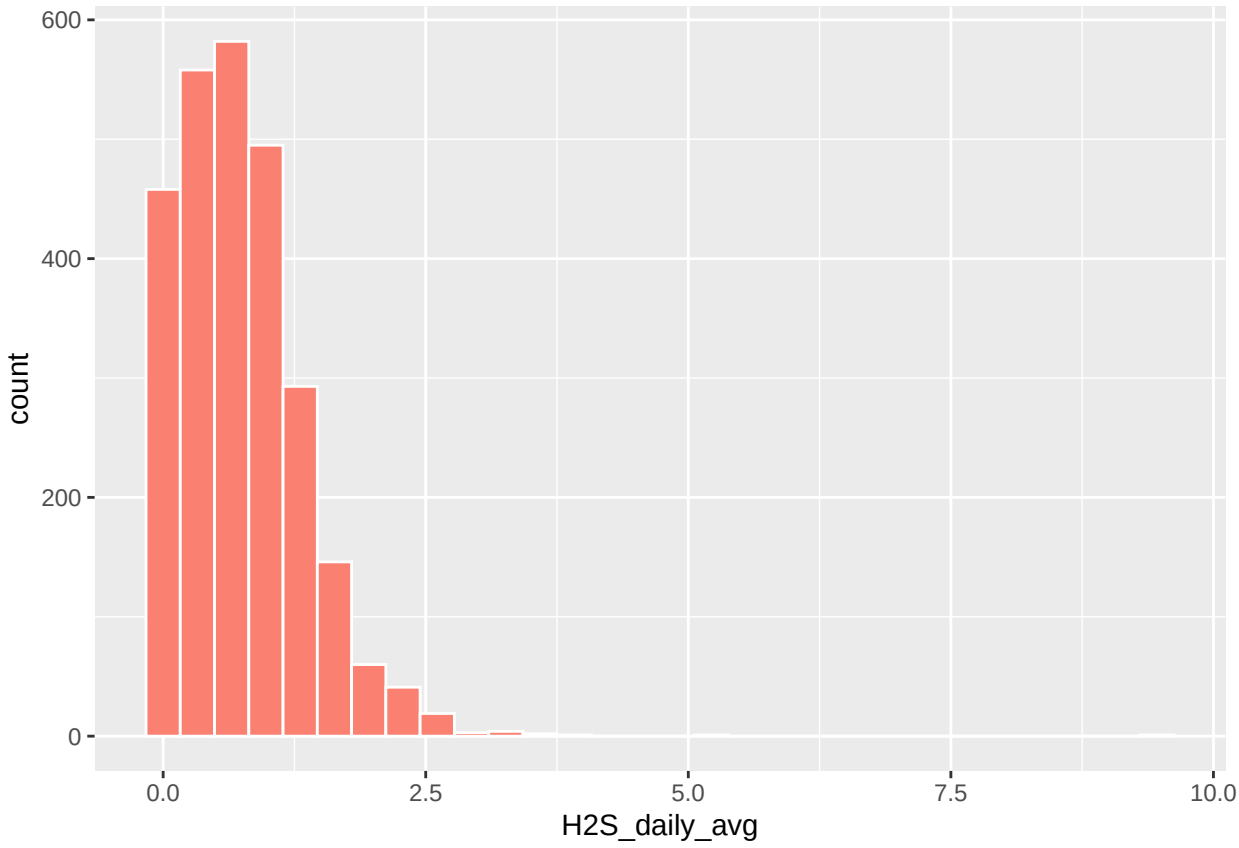
Data setup

```
# get train data set for daily average H2S
daily_avg_train_sincefeb2022 <- daily_full[complete.cases(daily_full),] %>%
  filter(day >= '2022-02-01') %>%
  select(H2S_daily_avg, month, year, weekday, MinDist,
         wd_avg, ws_avg, daily_downwind_ref, capacity, dist_wrp, mon_utm_x,
         mon_utm_y, day, monthly_oil_1km, monthly_gas_1km, active_1km, daily_downwind_wrp,
         elevation, EVI, num_odor_complaints, dist_dc, avg_temp,  avg_hum, precip)
```

Explore H2S concentration

```
ggplot(daily_avg_train_sincefeb2022, aes(x = H2S_daily_avg)) +
  geom_histogram(fill = "salmon", col = "white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

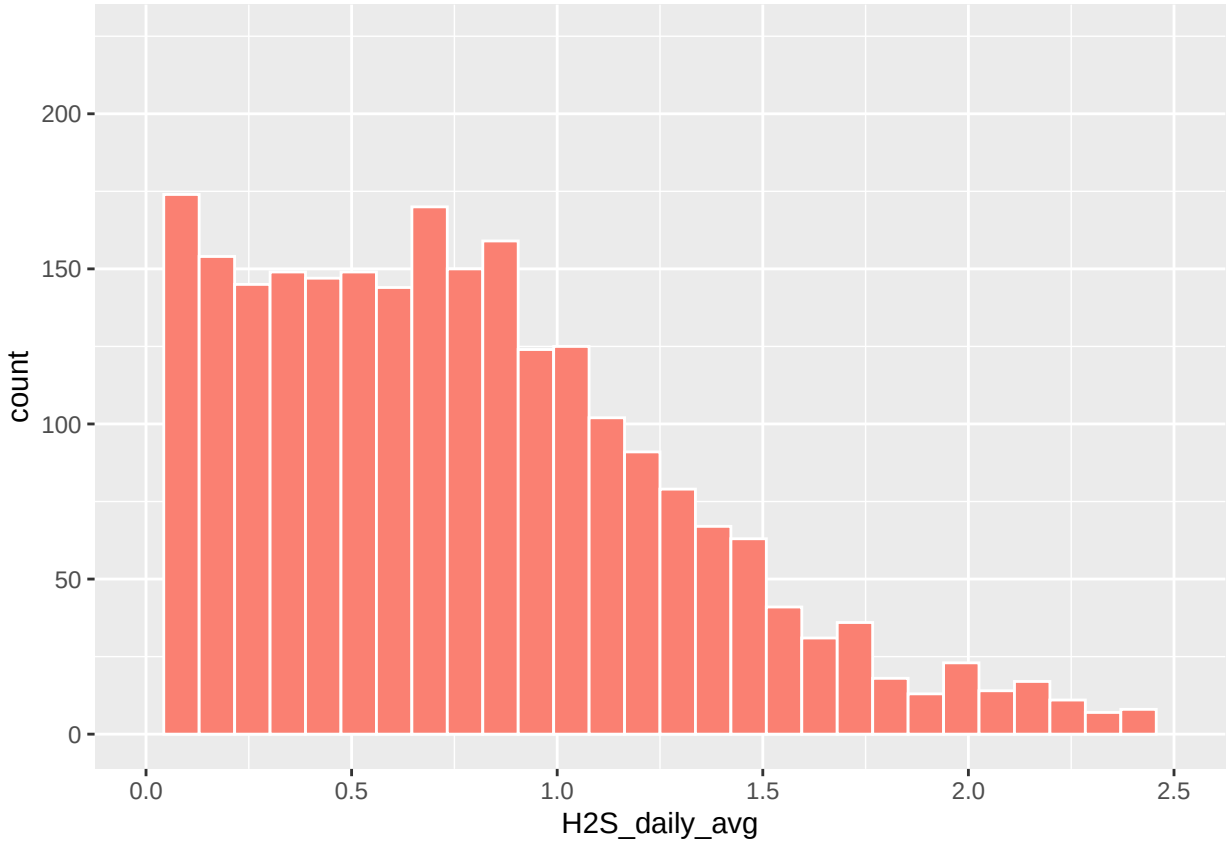



```
ggplot(daily_avg_train_sincefeb2022, aes(x = H2S_daily_avg)) +
  geom_histogram(fill = "salmon", col = "white") +
  scale_x_continuous(limits = c(0, 2.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 25 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



GAM

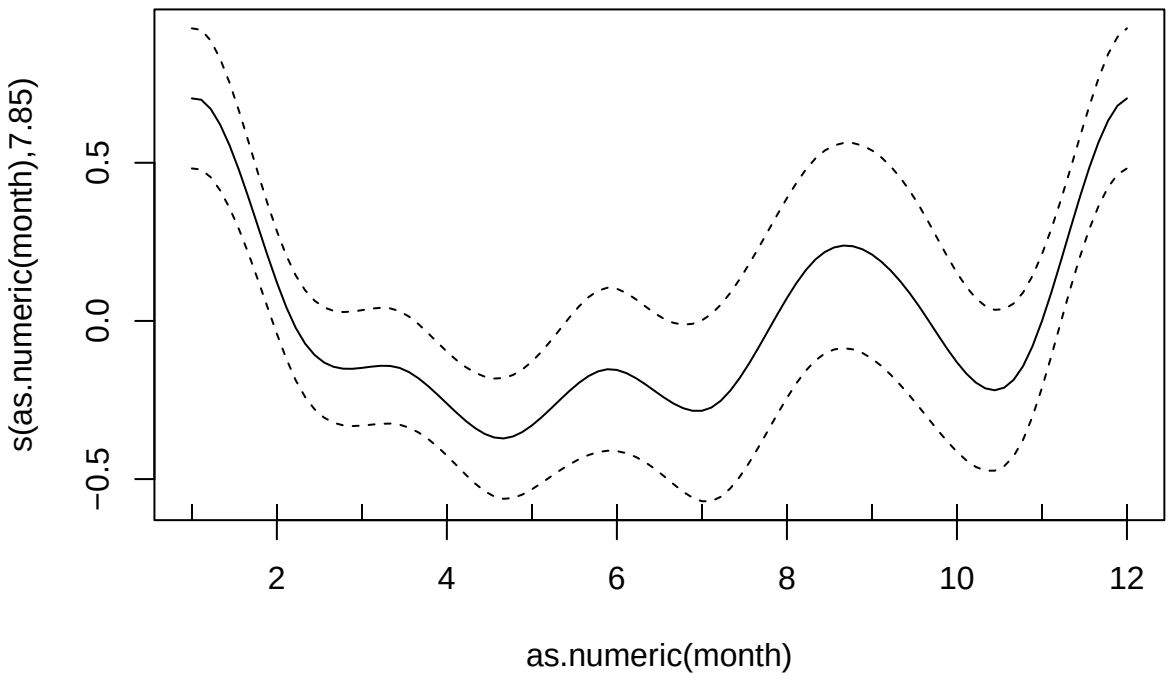
Since Feb 2022

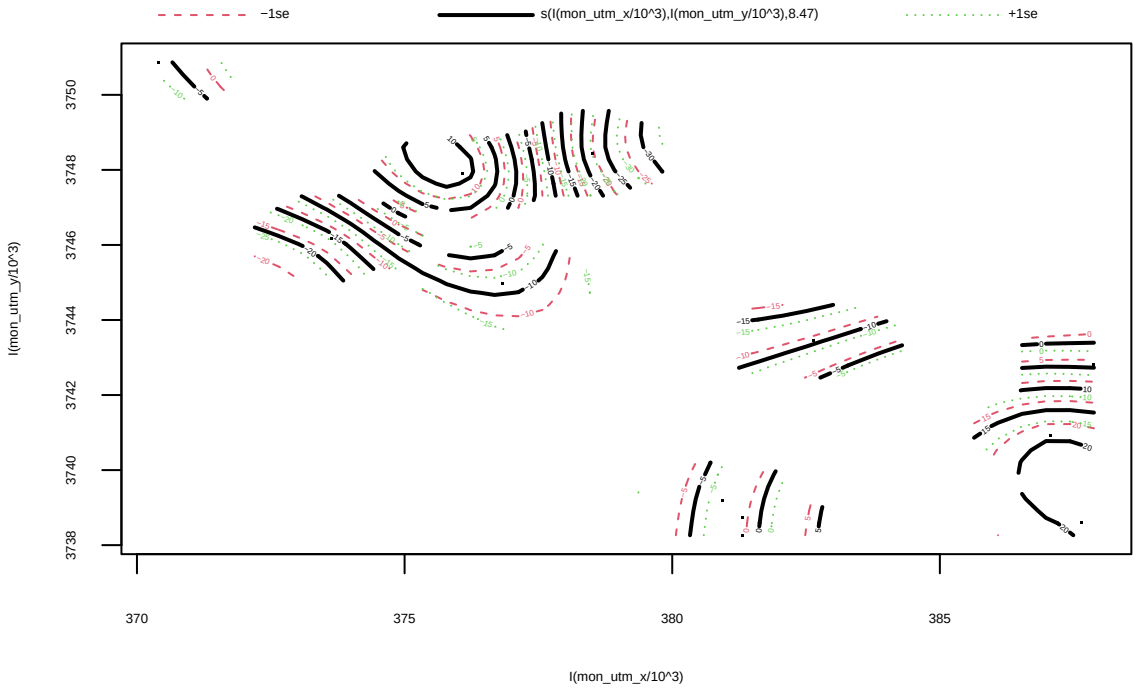
```
# Since feb 2022
h2s_da_model_f <- gam(H2S_daily_avg~s(as.numeric(month),bs='cc') + year + as.character(weekday) +
                        wd_avg + ws_avg + daily_downwind_ref + capacity +
                         I(1/dist_wrp^2) + I(1/MinDist^2) +
                        s(I(mon_utm_x/10^3), I(mon_utm_y/10^3), bs='tp', k = 10) +
                       te(I(mon_utm_x/10^3), I(mon_utm_y/10^3), as.numeric(day),
                          k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                        monthly_oil_1km + monthly_gas_1km + active_1km +
                       daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                       I(1/dist_dc^2) + avg_temp + avg_hum + precip,
                     data = daily_avg_train_sincefeb2022)
summary(h2s_da_model_f)
```

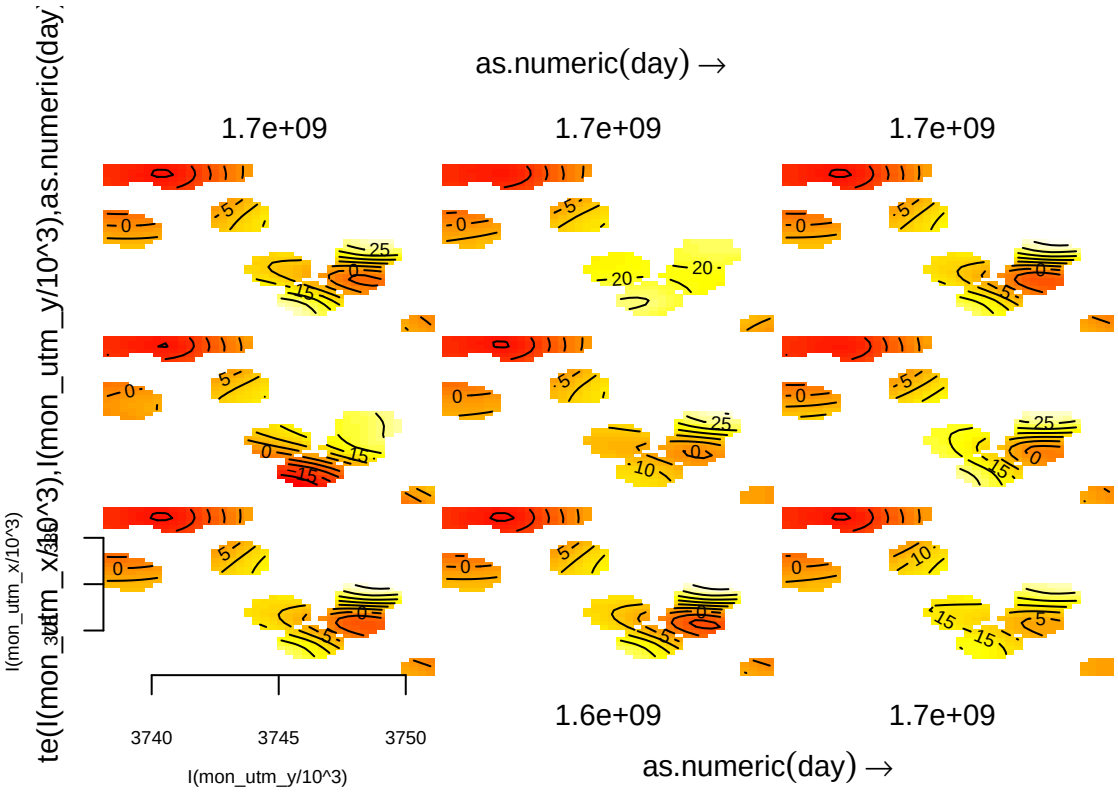
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## H2S_daily_avg ~ s(as.numeric(month), bs = "cc") + year + as.character(weekday) +
##     wd_avg + ws_avg + daily_downwind_ref + capacity + I(1/dist_wrp^2) +
##     I(1/MinDist^2) + s(I(mon_utm_x/10^3), I(mon_utm_y/10^3),
```

```
##     bs = "tp", k = 10) + te(I(mon_utm_x/10^3), I(mon_utm_y/10^3),
##     as.numeric(day), k = c(10, 10), d = c(2, 1), bs = c("tp",
##         "cc")) + monthly_oil_1km + monthly_gas_1km + active_1km +
##     daily_downwind_wrp + elevation + EVI + num_odor_complaints +
##     I(1/dist_dc^2) + avg_temp + avg_hum + precip
##
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -4.696e+00  7.111e-01  -6.603 4.88e-11 ***
## year2023                    4.435e-01  1.232e-01   3.600 0.000324 ***
## as.character(weekday)Mon   -8.809e-02  2.550e-02  -3.454 0.000561 ***
## as.character(weekday)Sat   -9.022e-02  2.540e-02  -3.552 0.000389 ***
## as.character(weekday)Sun   -2.061e-01  2.542e-02  -8.108 7.92e-16 ***
## as.character(weekday)Thu   -3.686e-02  2.542e-02  -1.450 0.147173
## as.character(weekday)Tue   -8.548e-03  2.527e-02  -0.338 0.735230
## as.character(weekday)Wed    2.998e-02  2.555e-02   1.173 0.240810
## wd_avg                      3.508e-04  9.030e-05   3.885 0.000105 ***
## ws_avg                     -7.855e-02  6.026e-03 -13.035  < 2e-16 ***
## daily_downwind_ref          1.567e-01  3.302e-02   4.744 2.21e-06 ***
## capacity                    1.003e-02  1.427e-03   7.031 2.63e-12 ***
## I(1/dist_wrp^2)             2.122e-07  1.133e-07   1.873 0.061128 .
## I(1/MinDist^2)             -3.629e-05  4.990e-06  -7.273 4.64e-13 ***
## monthly_oil_1km             1.611e-04  4.233e-05   3.805 0.000145 ***
## monthly_gas_1km            -1.306e-04  2.429e-04  -0.538 0.590730
## active_1km                 -9.257e-03  1.728e-02  -0.536 0.592319
## daily_downwind_wrp          6.698e-02  3.329e-02   2.012 0.044311 *
## elevation                  -2.960e-02  1.022e-02  -2.897 0.003804 **
## EVI                        -7.131e-01  1.698e-01  -4.200 2.76e-05 ***
## num_odor_complaints         2.552e-02  1.252e-02   2.038 0.041693 *
## I(1/dist_dc^2)              2.080e-05  9.122e-06   2.281 0.022659 *
## avg_temp                    1.206e-02  2.685e-03   4.491 7.42e-06 ***
## avg_hum                    -5.599e-03  7.027e-04  -7.968 2.42e-15 ***
## precip                     -7.076e-02  3.637e-02  -1.945 0.051851 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                        edf Ref.df     F
## s(as.numeric(month))                                 7.854  8.000 13.30
## s(I(mon_utm_x/10^3),I(mon_utm_y/10^3))               8.471  8.471 23.66
## te(I(mon_utm_x/10^3),I(mon_utm_y/10^3),as.numeric(day)) 75.878 76.000 23.14
##                                                       p-value
## s(as.numeric(month))                                   <2e-16 ***
## s(I(mon_utm_x/10^3),I(mon_utm_y/10^3))                 <2e-16 ***
## te(I(mon_utm_x/10^3),I(mon_utm_y/10^3),as.numeric(day))  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 114/118
## R-sq.(adj) =  0.666   Deviance explained =   68%
## GCV = 0.12669  Scale est. = 0.12128   n = 2664
```

```
plot(h2s_da_model_f)
```

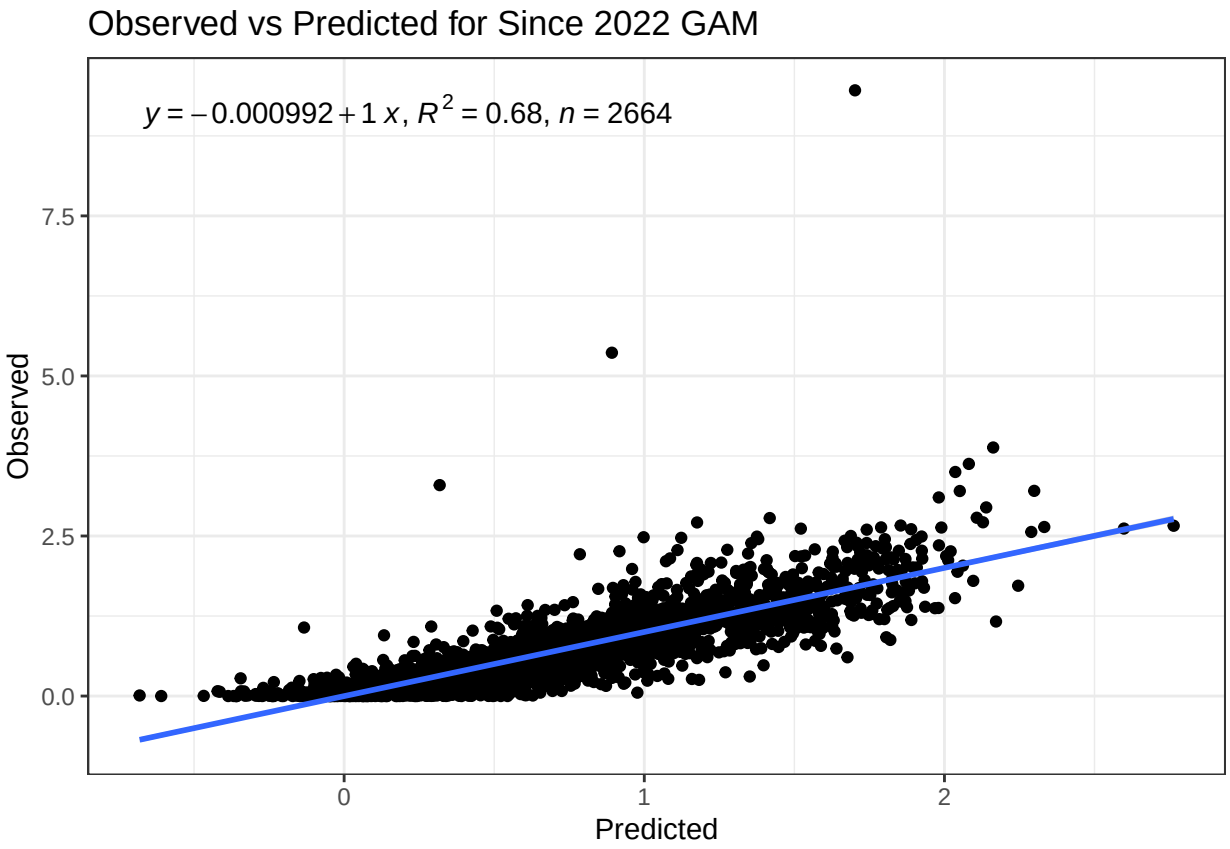

s(as.numeric(month),7.85)
0.5
0.0
−0.5
2
4
6
8
10
12
as.numeric(month)


−1se
s(I(mon_utm_x/10^3),I(mon_utm_y/10^3),8.47)
+1se
I(mon_utm_y/10^3)
3738
3740
3742
3744
3746
3748
3750
370
375
380
385
I(mon_utm_x/10^3)



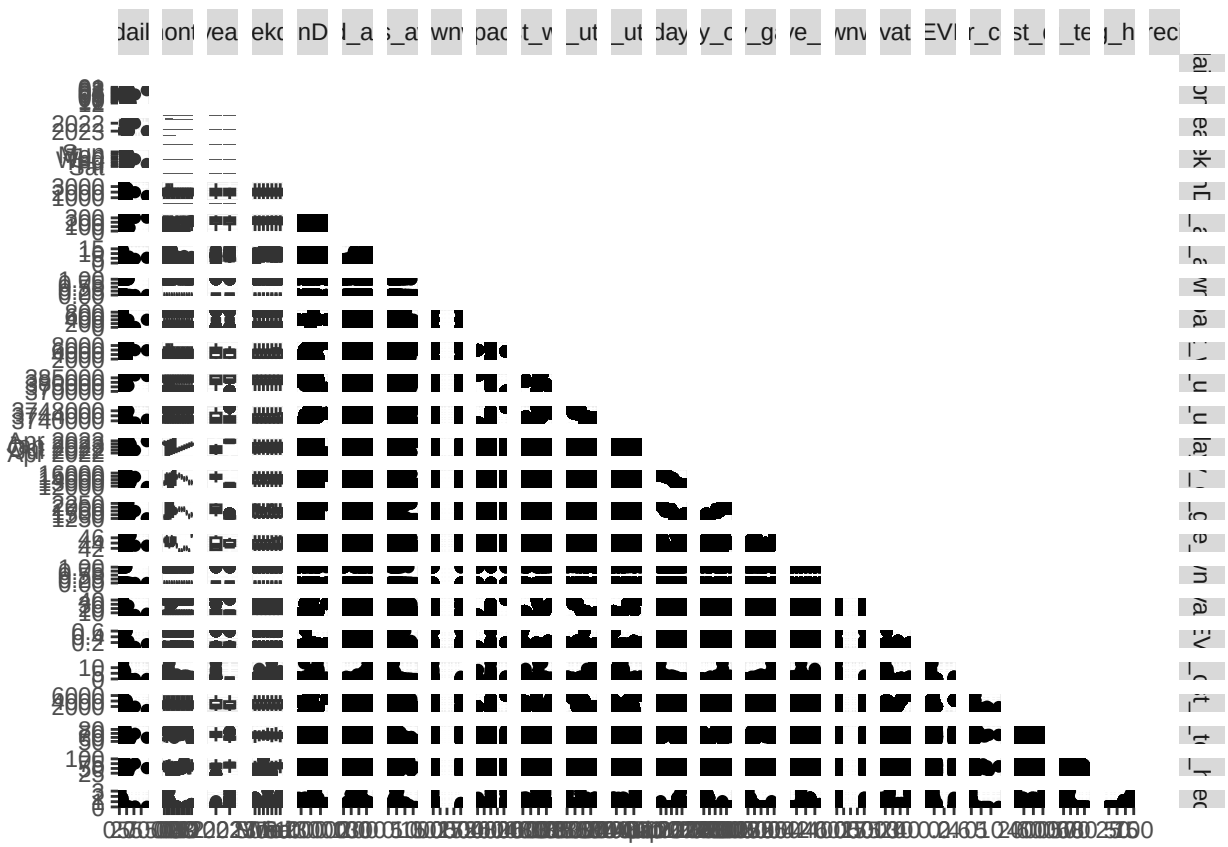
Condition 1

```
# Check Condition 1
ggplot(tibble(obs = daily_avg_train_sincefeb2022$H2S_daily_avg, pred = fitted(h2s_da_model_f)),
                          aes(x = pred, y = obs)) +
                    geom_point() +
                    stat_poly_line() +
                    stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                    labs(y = 'Observed', x = 'Predicted',
                         title = 'Observed vs Predicted for Since 2022 GAM') +
                    theme_bw()
```



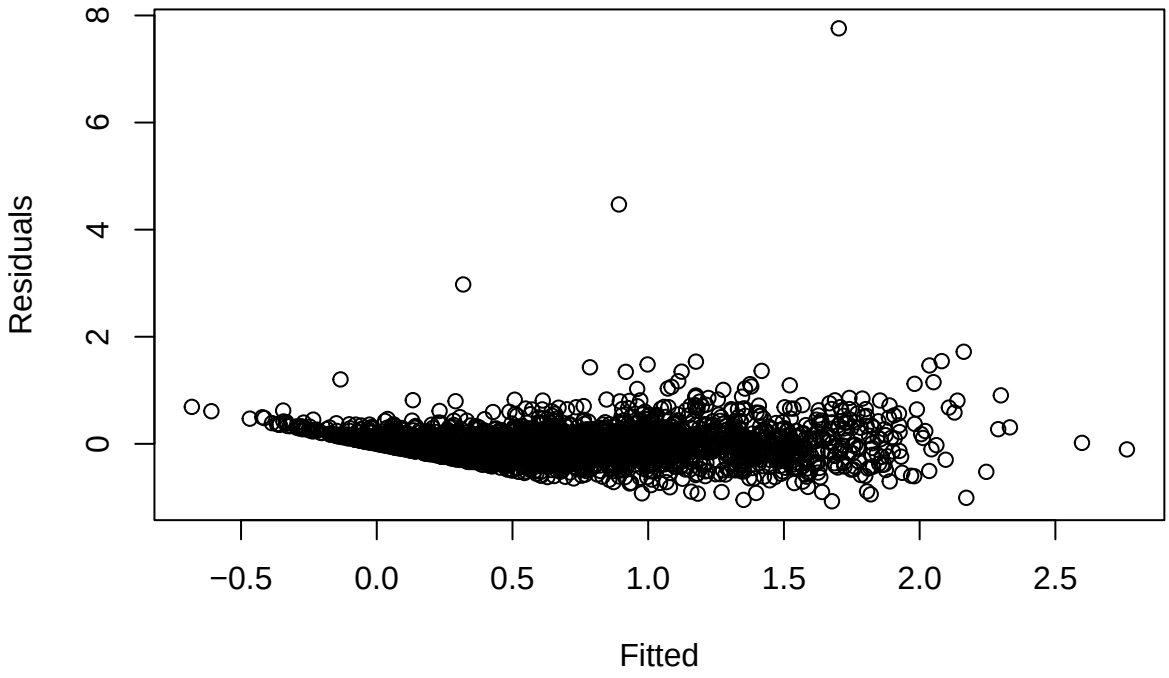
Condition 2

```
# Check condition 2
pair <- ggpairs(daily_avg_train_sincefeb2022, lower=list(continuous=ggally_points, combo=ggally_box_no_
pair
```



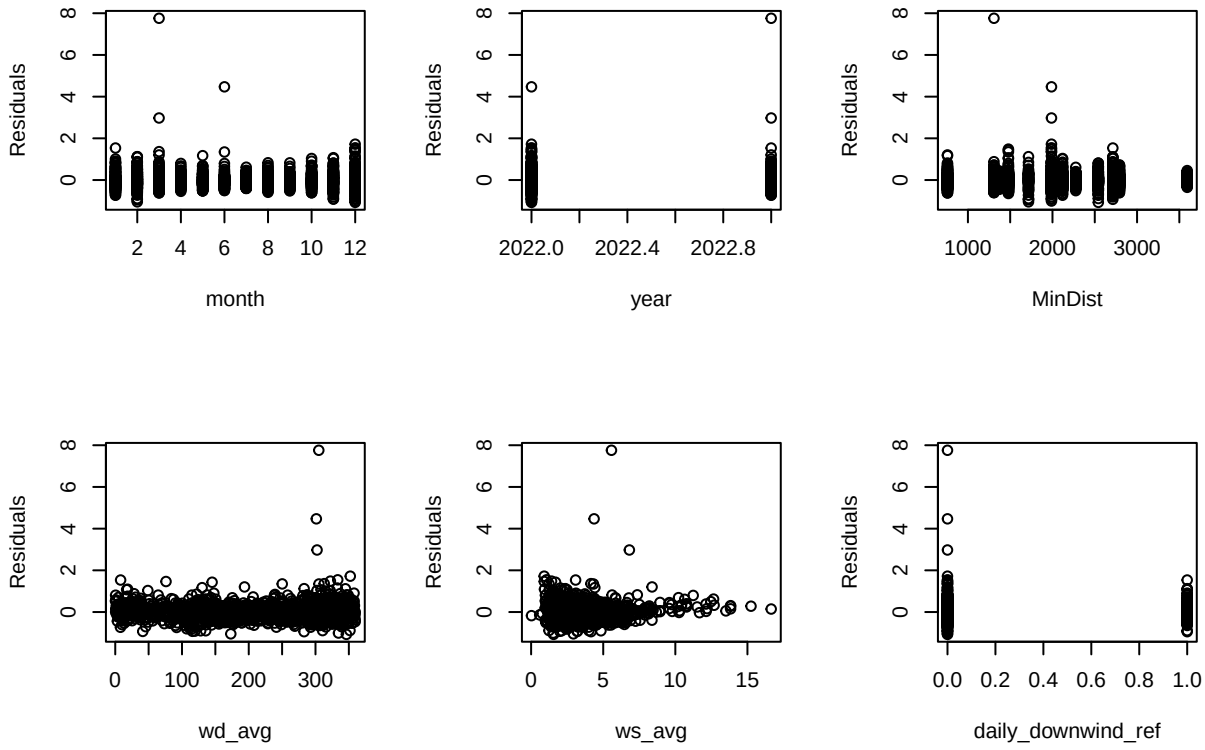
Residual Plots

```
# Residual plots
rs_full <- h2s_da_model_f$residuals
plot(rs_full~fitted(h2s_da_model_f), xlab="Fitted", ylab="Residuals")
```

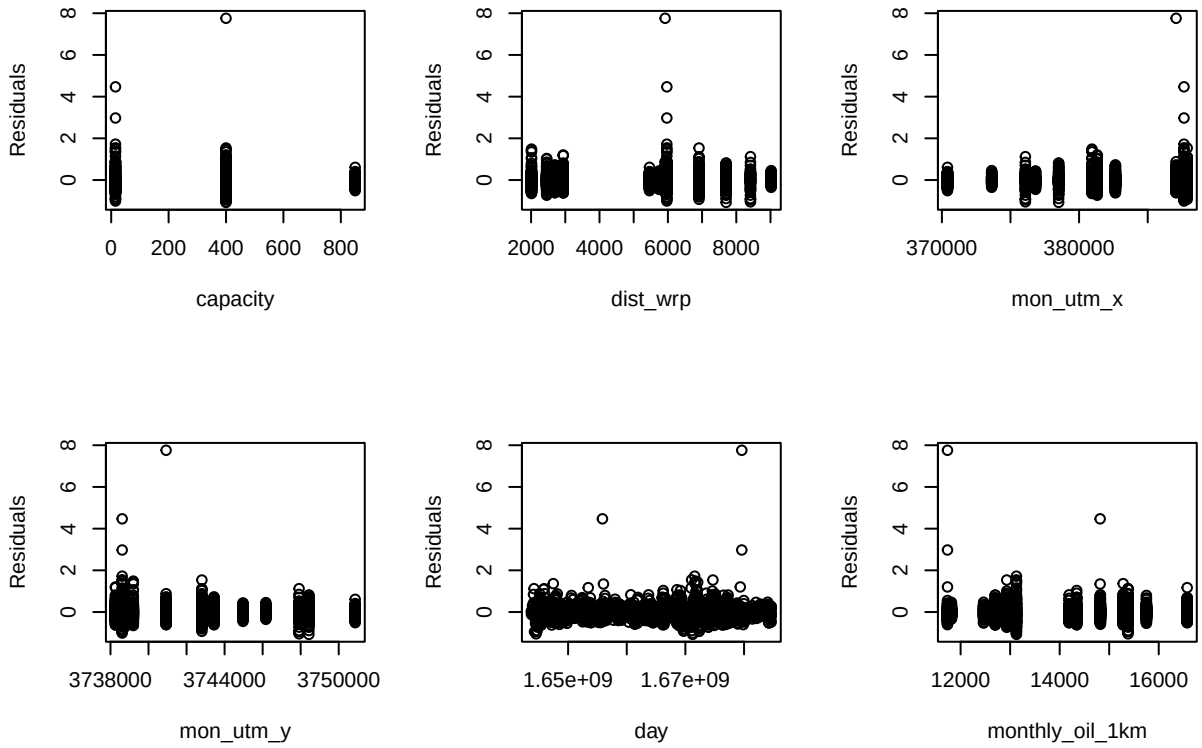



```
par(mfrow=c(2,3))
for(i in c(2:3, 5:length(daily_avg_train_sincefeb2022))){
  print(i)
  plot(rs_full~unlist(daily_avg_train_sincefeb2022[,i]), xlab=names(daily_avg_train_sincefeb2022)[i], yl
}
```

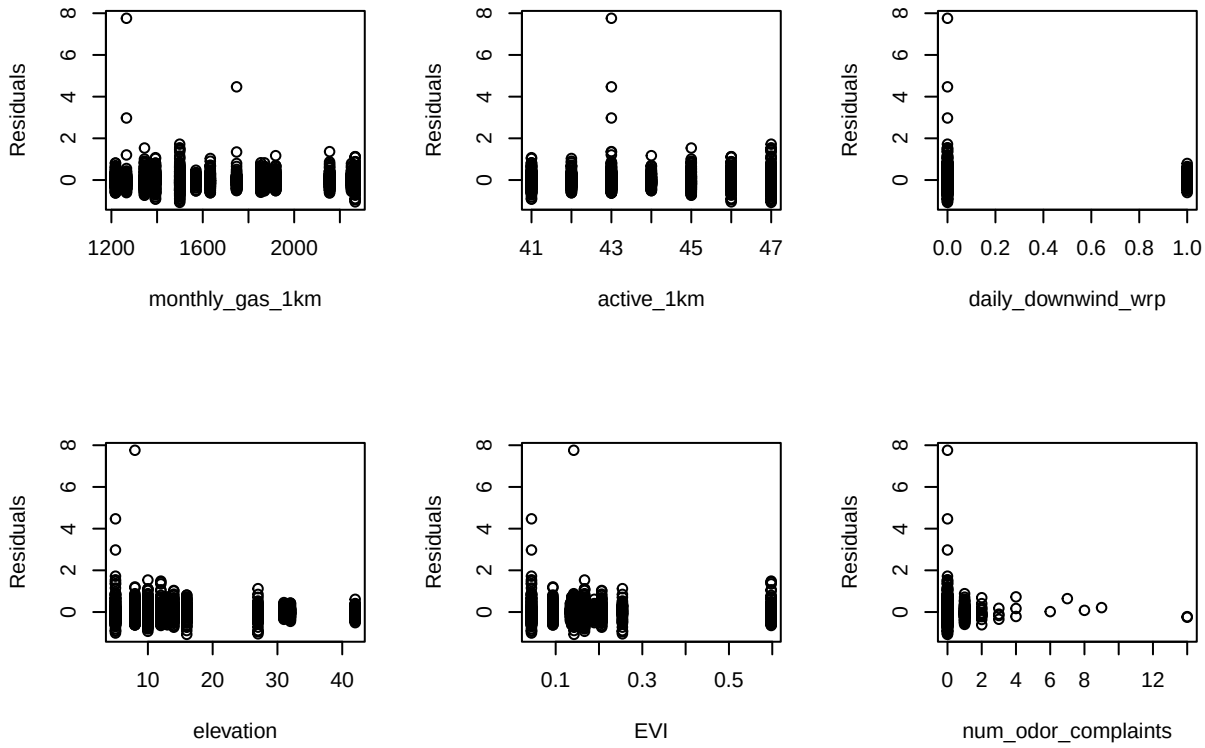
```
## [1] 2
## [1] 3
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```



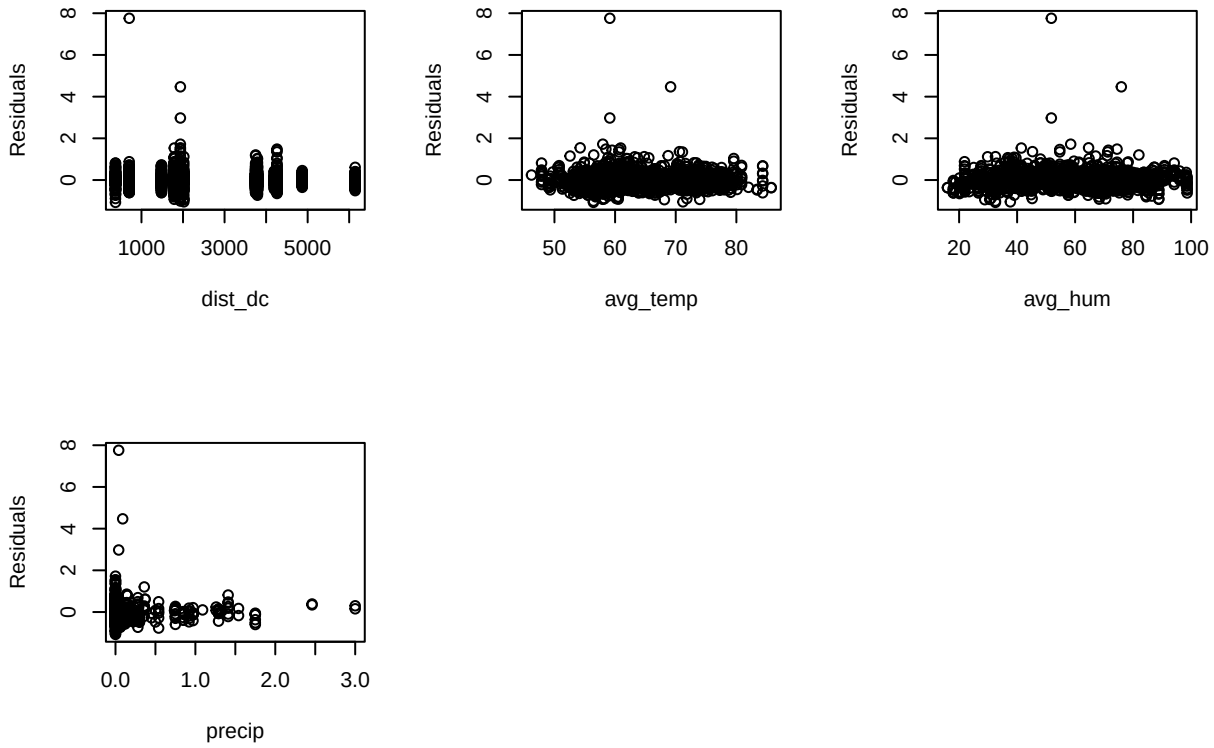

```
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
```



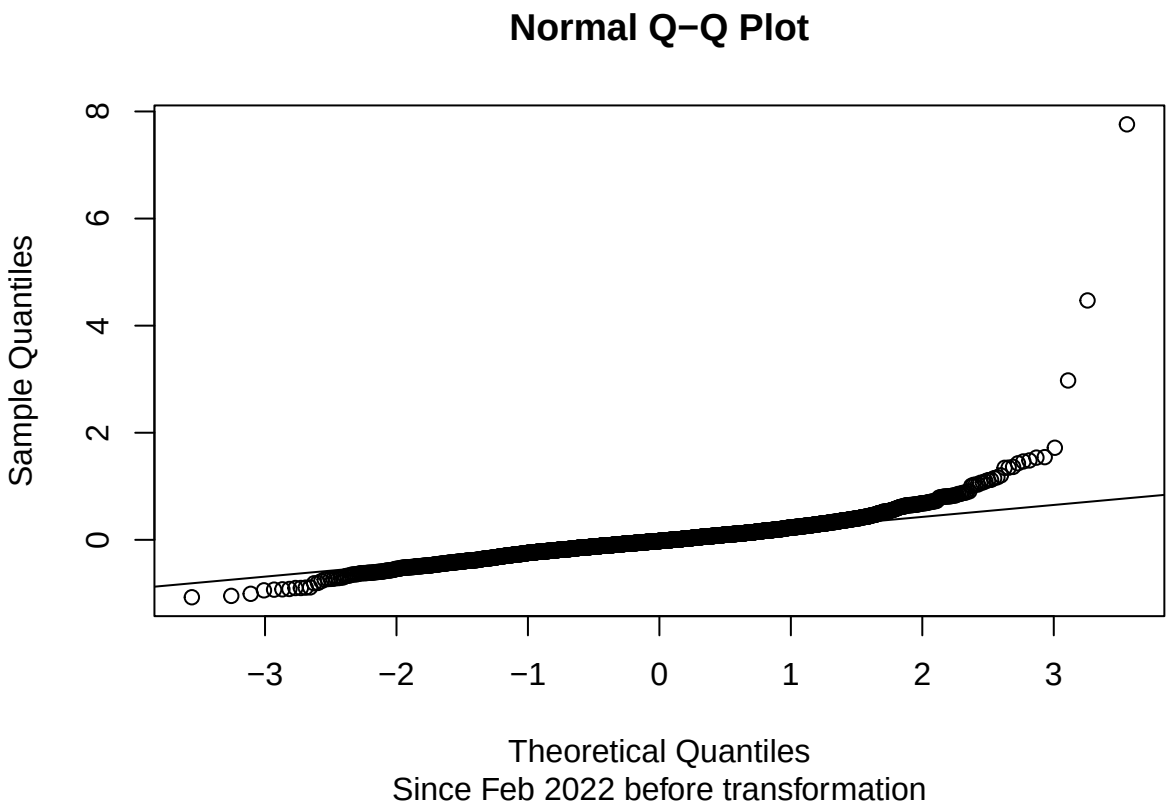

```
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```



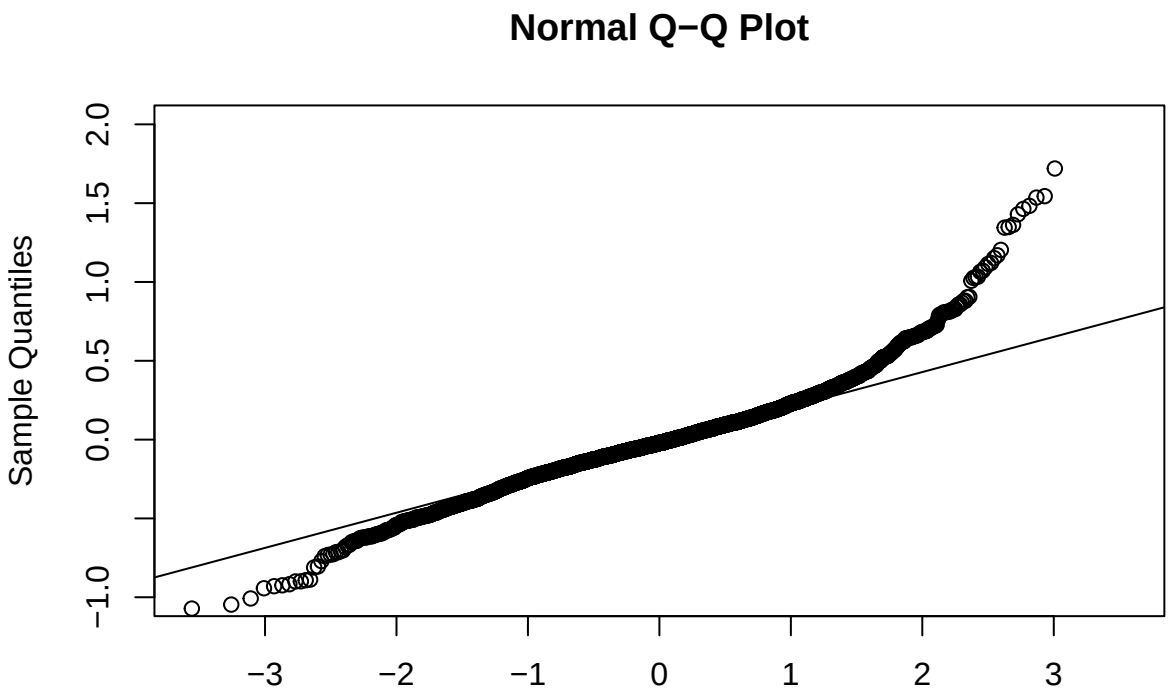

```
## [1] 21
## [1] 22
## [1] 23
## [1] 24
```




```
# Normal Q-Q Plot
stats::qqnorm(rs_full, sub = "Since Feb 2022 before transformation")
stats::qqline(rs_full)
```




```
stats::qqnorm(rs_full, sub = "Since Feb 2022 before transformation w.o zoomed in",
              ylim = c(-1, 2))
stats::qqline(rs_full)
```



Since Feb 2022 before transformation w.o zoomed in

Box Cox Transformations Since Feb 2022

```
# try transformations
# On both response and predictors
transform_all <- readRDS('rfiles/transform_all.rds')
#transform_all <- powerTransform(as.matrix(daily_avg_train_sincefeb2022 %>%
#                                            select(-c(month, day, year, weekday))),
#                                family = "bcnPower")
#saveRDS(transform_all, 'rfiles/transform_all.rds')
summary(transform_all)
```

```
## bcnPower transformation to Multinormality
##
## Estimated power, lambda
##                      Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## H2S_daily_avg          0.3219       0.330       0.2881       0.3557
## MinDist               -0.4742      -0.500      -0.6708      -0.2775
## wd_avg                 2.3111       2.311       2.0991       2.5232
## ws_avg                -1.6057      -1.606      -1.7590      -1.4525
## daily_downwind_ref    -3.0000      -3.000      -3.1145      -2.8855
## capacity               0.6753       0.675       0.5850       0.7655
## dist_wrp               0.1919       0.192       0.0858       0.2980
## mon_utm_x             -1.5206      -1.521      -1.5288      -1.5123
## mon_utm_y             -0.8946      -0.895      -0.9160      -0.8732
## monthly_oil_1km       -0.2475       0.000      -0.6627       0.1676
```

```
## monthly_gas_1km       -0.6881      -0.688      -0.8551      -0.5211
## active_1km            -3.0000      -3.000      -3.9091      -2.0909
## daily_downwind_wrp    -3.0000      -3.000      -3.1145      -2.8855
## elevation             -0.3980      -0.398      -0.4645      -0.3315
## EVI                   -1.6754      -1.675      -1.7753      -1.5756
## num_odor_complaints   -3.0000      -3.000      -3.1144      -2.8856
## dist_dc                0.9797       1.000       0.9291       1.0304
## avg_temp              -0.4824      -0.500      -0.7532      -0.2117
## avg_hum                2.4582       2.458       2.2184       2.6979
## precip                -3.0000      -3.000      -3.1307      -2.8693
##
## Estimated location, gamma
##                         Est gamma Std Err. Wald Lower Bound Wald Upper Bound
## H2S_daily_avg              0.1000       NA               NA               NA
## MinDist                 3592.7799       NA               NA               NA
## wd_avg                   359.4889       NA               NA               NA
## ws_avg                     7.2474       NA               NA               NA
## daily_downwind_ref         0.1000       NA               NA               NA
## capacity                 458.2879       NA               NA               NA
## dist_wrp                   0.1000       NA               NA               NA
## mon_utm_x                  0.2249       NA               NA               NA
## mon_utm_y            1437729.3535       NA               NA               NA
## monthly_oil_1km       16574.9995       NA               NA               NA
## monthly_gas_1km           0.1000       NA               NA               NA
## active_1km                0.1000       NA               NA               NA
## daily_downwind_wrp        0.1000       NA               NA               NA
## elevation                 0.1000       NA               NA               NA
## EVI                       0.1000       NA               NA               NA
## num_odor_complaints       0.1000       NA               NA               NA
## dist_dc                   0.1000       NA               NA               NA
## avg_temp                  0.1000       NA               NA               NA
## avg_hum                  98.6464       NA               NA               NA
## precip                    0.1000       NA               NA               NA
##
## Likelihood ratio tests about transformation parameters
##                                                                          LRT df pval
## LR test, lambda = (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0) 39675.88 20    0
## LR test, lambda = (1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1) 84208.44 20    0
```

```
# On only predictors
transform_pred <- readRDS('rfiles/transform_pred.rds')
# transform_pred <- powerTransform(as.matrix(daily_avg_train_sincefeb2022 %>%
#                                              select(-c(month, day, year, weekday,
#                                                        H2S_daily_avg))),
#                                  family = "bcnPower")
# saveRDS(transform_pred, 'rfiles/transform_pred.rds')
summary(transform_pred)
```

```
## Warning in sqrt(diag(object$invHess)): NaNs produced
```

```
## bcnPower transformation to Multinormality
##
## Estimated power, lambda
##                      Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## wd_avg                  2.3006       2.301       2.0884       2.5129
```

```
## ws_avg                -1.5494      -1.549      -1.7074      -1.3913
## daily_downwind_ref    -3.0000      -3.000      -3.1145      -2.8855
## capacity               0.8330       0.833       0.7426       0.9234
## dist_wrp               0.3365       0.330       0.2244       0.4485
## mon_utm_x             -1.7820      -1.782      -1.7844      -1.7795
## mon_utm_y             -0.4800      -0.480          NaN          NaN
## monthly_oil_1km       -0.2598       0.000      -0.6790       0.1594
## monthly_gas_1km       -0.7286      -0.729      -0.8954      -0.5618
## active_1km            -3.0000      -3.000      -3.9087      -2.0913
## daily_downwind_wrp    -3.0000      -3.000      -3.1145      -2.8855
## elevation             -0.1503      -0.150      -0.2178      -0.0828
## EVI                   -1.5314      -1.531      -1.6248      -1.4379
## num_odor_complaints   -3.0000      -3.000      -3.1144      -2.8856
## dist_dc                0.8907       0.891       0.8402       0.9412
## avg_temp              -0.5132      -0.500      -0.7835      -0.2429
## avg_hum                2.5338       2.534       2.2904       2.7771
## precip                -3.0000      -3.000      -3.1307      -2.8693
##
## Estimated location, gamma
##                         Est gamma Std Err. Wald Lower Bound Wald Upper Bound
## wd_avg                   359.4889       NA                NA                NA
## ws_avg                     7.2473       NA                NA                NA
## daily_downwind_ref         0.1000       NA                NA                NA
## capacity                 458.2879       NA                NA                NA
## dist_wrp                   0.1000       NA                NA                NA
## mon_utm_x                  0.2255       NA                NA                NA
## mon_utm_y            1437729.2528       NA                NA                NA
## monthly_oil_1km       16574.9995       NA                NA                NA
## monthly_gas_1km           0.1000       NA                NA                NA
## active_1km                0.1000       NA                NA                NA
## daily_downwind_wrp        0.1000       NA                NA                NA
## elevation                 0.1000       NA                NA                NA
## EVI                       0.1000       NA                NA                NA
## num_odor_complaints       0.1000       NA                NA                NA
## dist_dc                   0.1000       NA                NA                NA
## avg_temp                  0.1000       NA                NA                NA
## avg_hum                  98.6464       NA                NA                NA
## precip                    0.1000       NA                NA                NA
##
## Likelihood ratio tests about transformation parameters
##                                                                      LRT df pval
## LR test, lambda = (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0) 39060.66 18    0
## LR test, lambda = (1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1) 81738.35 18    0
```

```r
# On only response
transform_resp <- readRDS('rfiles/transform_resp.rds')
# transform_resp <- powerTransform(as.matrix(daily_avg_train_sincefeb2022 %>%
#                                              select(H2S_daily_avg)),
#                                  family = "bcnPower")
# saveRDS(transform_resp, 'rfiles/transform_resp.rds')
summary(transform_resp)
```

```
## bcnPower transformation to Normality
##
## Estimated power, lambda
```

```
##                Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## H2S_daily_avg    0.3701        0.37       0.3337       0.4065
##
## Location gamma was fixed at its lower bound
##                Est gamma Std Err. Wald Lower Bound Wald Upper Bound
## H2S_daily_avg       0.1       NA               NA               NA
##
## Likelihood ratio tests about transformation parameters
##                              LRT df pval
## LR test, lambda = (0)  402.2188  1    0
## LR test, lambda = (1) 1122.5811  1    0
```

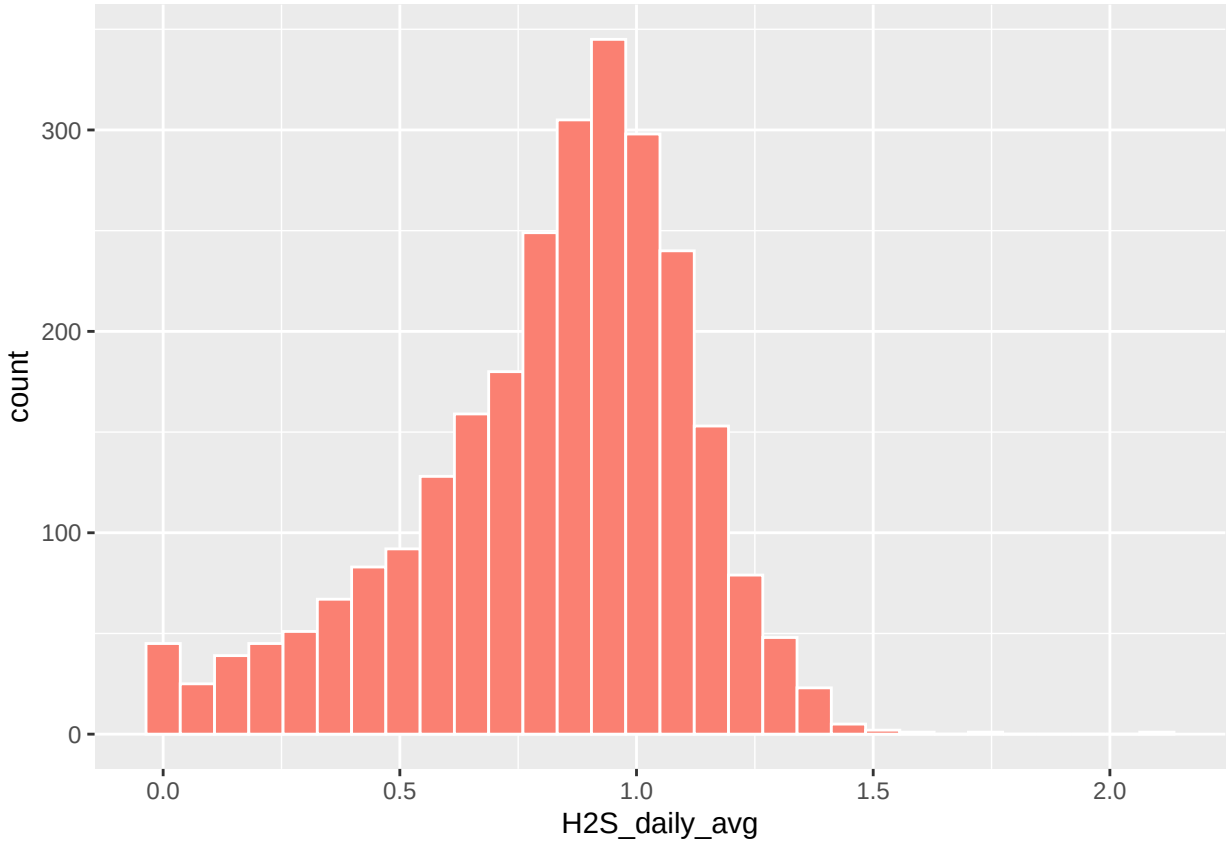
```
# This is quite similar to transforming both

# This suggests a 0.37 power transformation
```

Tranformed Pred/Resp

```
# Apply the suggested power transformations for transforming both predictor and outcome
# get train data set for daily average H2S
daily_avg_train_sincefeb2022_both_trans <- daily_avg_train_sincefeb2022 %>%
  mutate(H2S_daily_avg = H2S_daily_avg^0.33,
         wd_avg = wd_avg^2.311,
         ws_avg = ws_avg^-1.606,
         capacity = capacity^0.675,
         dist_wrp = dist_wrp^0.192,
         MinDist = MinDist^-0.500,
         mon_utm_x = mon_utm_x^-1.521,
         mon_utm_y = mon_utm_y^-0.895,
         monthly_oil_1km = log(monthly_oil_1km),
         monthly_gas_1km = monthly_gas_1km^-0.688,
         active_1km = active_1km^-3,
         elevation = elevation^-0.398,
         EVI = EVI^-1.675,
         num_odor_complaints = (num_odor_complaints + 1)^-3,
         avg_temp = avg_temp^-0.5,
         avg_hum = avg_hum^2.458,
         precip = (precip + 1)^-3,
         )
```

```
ggplot(daily_avg_train_sincefeb2022_both_trans, aes(x = H2S_daily_avg)) +
  geom_histogram(fill = "salmon", col = "white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

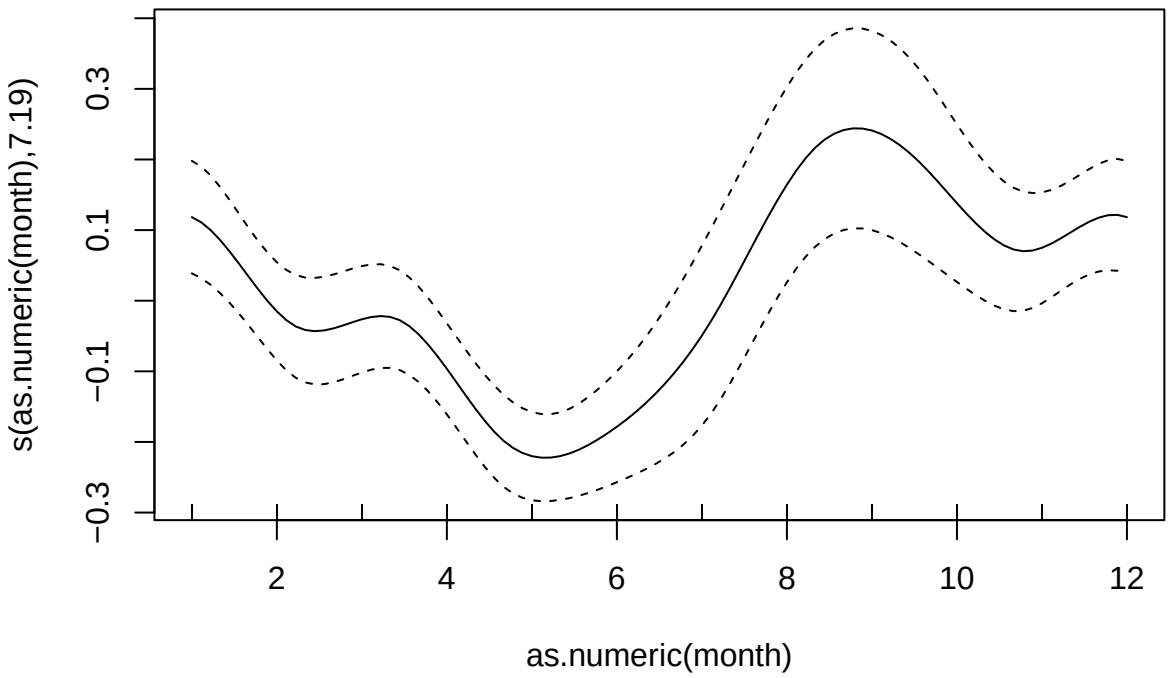


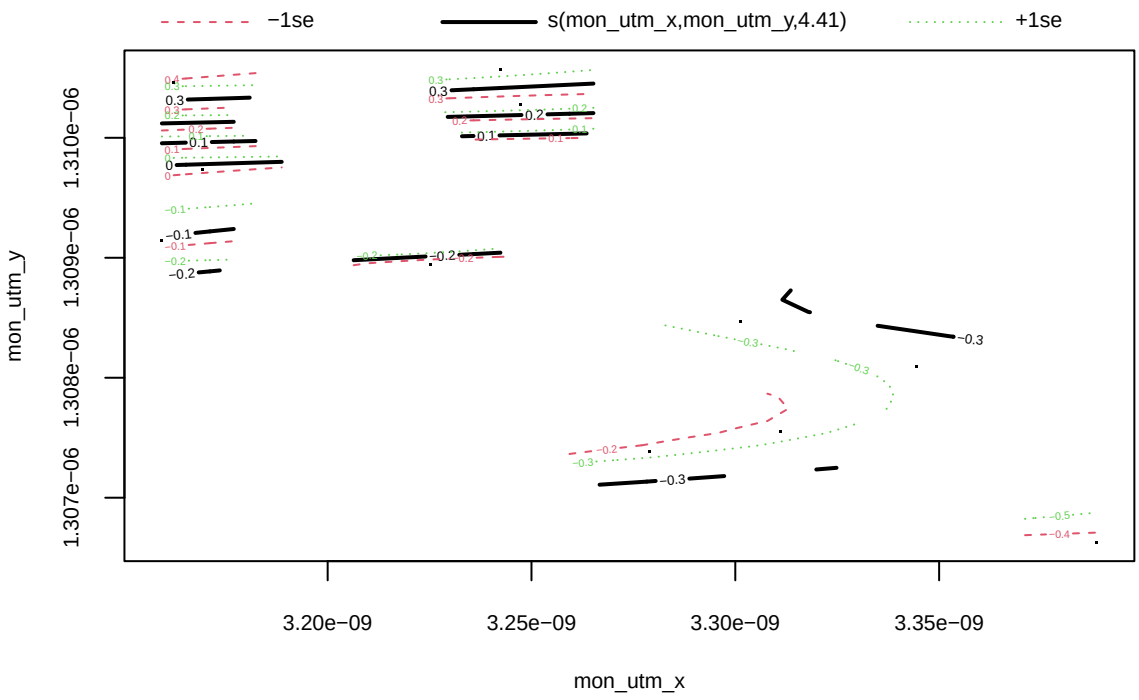

```
# Since feb 2022
h2s_da_model_f_trans <- gam(H2S_daily_avg~s(as.numeric(month),bs='cc') + year + as.character(weekday) +
                         wd_avg + ws_avg + daily_downwind_ref + capacity +
                          dist_wrp + MinDist +
                         s(mon_utm_x, mon_utm_y, bs='tp', k = 10) +
                        te(mon_utm_x, mon_utm_y, as.numeric(day),
                           k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                         monthly_oil_1km + monthly_gas_1km + active_1km +
                        daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                        dist_dc + avg_temp + avg_hum + precip,
                      data = daily_avg_train_sincefeb2022_both_trans)
summary(h2s_da_model_f_trans)
```

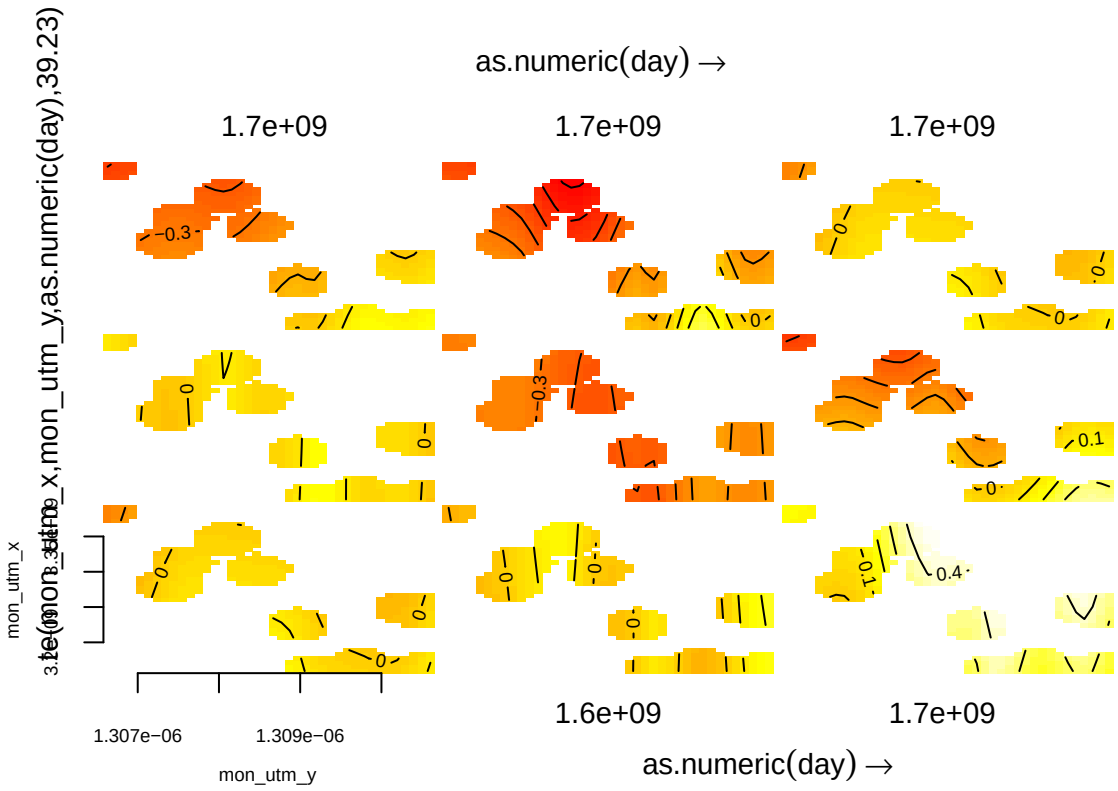
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## H2S_daily_avg ~ s(as.numeric(month), bs = "cc") + year + as.character(weekday) +
##     wd_avg + ws_avg + daily_downwind_ref + capacity + dist_wrp +
##     MinDist + s(mon_utm_x, mon_utm_y, bs = "tp", k = 10) + te(mon_utm_x,
##     mon_utm_y, as.numeric(day), k = c(10, 10), d = c(2, 1), bs = c("tp",
##         "cc")) + monthly_oil_1km + monthly_gas_1km + active_1km +
##     daily_downwind_wrp + elevation + EVI + num_odor_complaints +
##     dist_dc + avg_temp + avg_hum + precip
##
```

```
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.736e-02  8.687e-03   1.999 0.045747 *
## year2023                   -3.762e-02  4.580e-02  -0.821 0.411532
## as.character(weekday)Mon -2.335e-02  1.221e-02  -1.913 0.055915 .
## as.character(weekday)Sat -2.730e-02  1.222e-02  -2.235 0.025517 *
## as.character(weekday)Sun -9.646e-02  1.223e-02  -7.885 4.58e-15 ***
## as.character(weekday)Thu -5.938e-03  1.221e-02  -0.486 0.626687
## as.character(weekday)Tue -2.579e-03  1.215e-02  -0.212 0.831861
## as.character(weekday)Wed -2.251e-03  1.225e-02  -0.184 0.854159
## wd_avg                      5.061e-08  1.650e-08   3.066 0.002189 **
## ws_avg                      3.506e-04  4.864e-04   0.721 0.471116
## daily_downwind_ref          3.393e-02  1.550e-02   2.189 0.028653 *
## capacity                    1.791e-02  9.251e-04  19.357  < 2e-16 ***
## dist_wrp                    1.242e-01  1.829e-02   6.788 1.41e-11 ***
## MinDist                    -8.071e-01  8.371e-02  -9.642  < 2e-16 ***
## monthly_oil_1km             6.674e-02  1.145e-02   5.828 6.30e-09 ***
## monthly_gas_1km             9.535e-07  5.356e-04   0.002 0.998580
## active_1km                 -6.254e-06  1.089e-06  -5.742 1.04e-08 ***
## daily_downwind_wrp          8.589e-03  1.581e-02   0.543 0.587052
## elevation                  -3.006e+00  2.444e-01 -12.303  < 2e-16 ***
## EVI                         5.423e-03  3.109e-04  17.441  < 2e-16 ***
## num_odor_complaints        -2.469e-02  1.599e-02  -1.544 0.122779
## dist_dc                    -1.562e-04  5.843e-06 -26.742  < 2e-16 ***
## avg_temp                    6.387e-02  1.306e-02   4.890 1.07e-06 ***
## avg_hum                    -4.893e-06  2.623e-07 -18.656  < 2e-16 ***
## precip                      8.300e-02  2.334e-02   3.556 0.000384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                            edf Ref.df       F p-value
## s(as.numeric(month))                     7.186   8.00   5.989  <2e-16 ***
## s(mon_utm_x,mon_utm_y)                   4.411   4.45 130.006  <2e-16 ***
## te(mon_utm_x,mon_utm_y,as.numeric(day)) 39.233  72.00  17.143  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 84/118
## R-sq.(adj) =  0.676   Deviance explained = 68.4%
## GCV = 0.028812  Scale est. = 0.028046  n = 2664
```

```
plot(h2s_da_model_f_trans)
```


s(as.numeric(month),7.19)
0.3
0.1
−0.1
−0.3
2
4
6
8
10
12
as.numeric(month)


−1se
s(mon_utm_x,mon_utm_y,4.41)
+1se
mon_utm_y
1.307e−06
1.308e−06
1.309e−06
1.310e−06
3.20e−09
3.25e−09
3.30e−09
3.35e−09
mon_utm_x

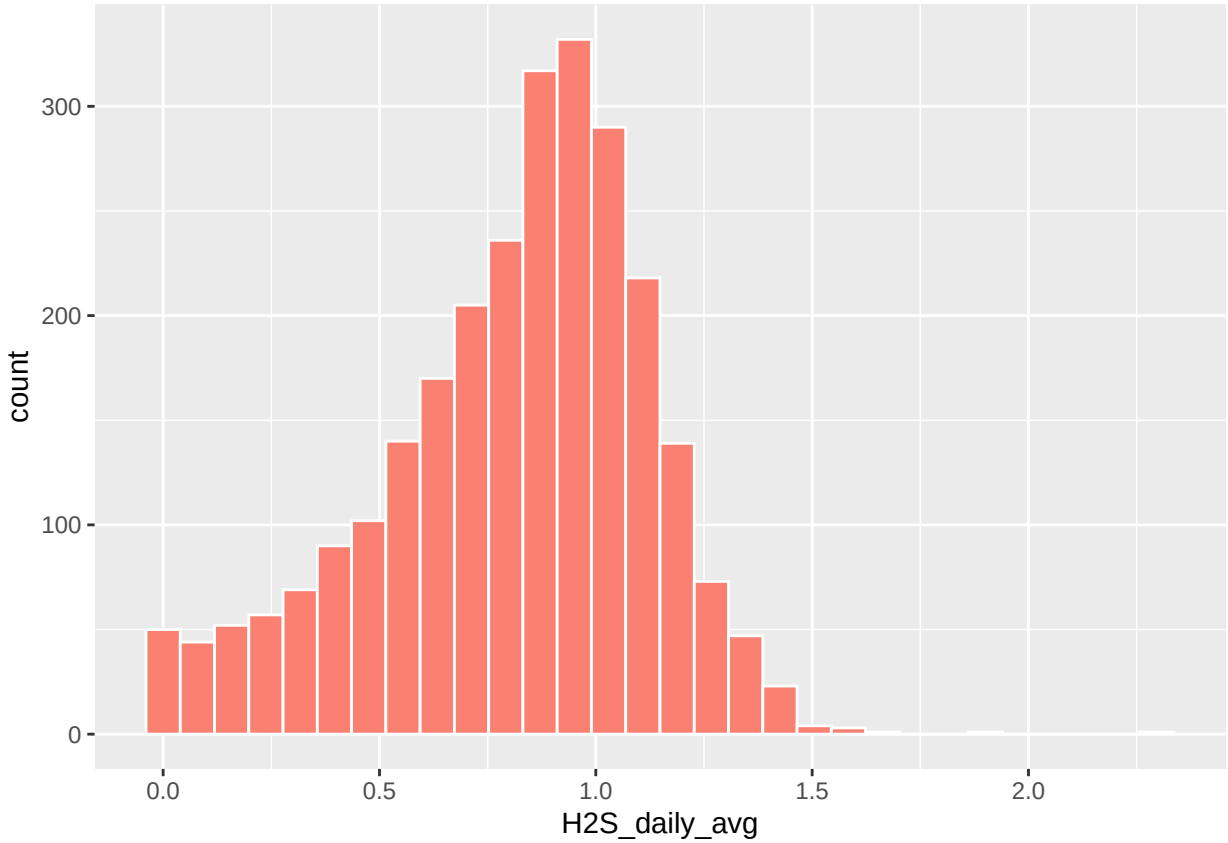


Tranformed Resp^0.37

```
# Apply the suggested power transformations for transforming both predictor and outcome
# get train data set for daily average H2S
daily_avg_train_sincefeb2022_trans_resp <- daily_avg_train_sincefeb2022 %>%
  mutate(H2S_daily_avg = H2S_daily_avg^0.37)
```

```
ggplot(daily_avg_train_sincefeb2022_trans_resp, aes(x = H2S_daily_avg)) +
  geom_histogram(fill = "salmon", col = "white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

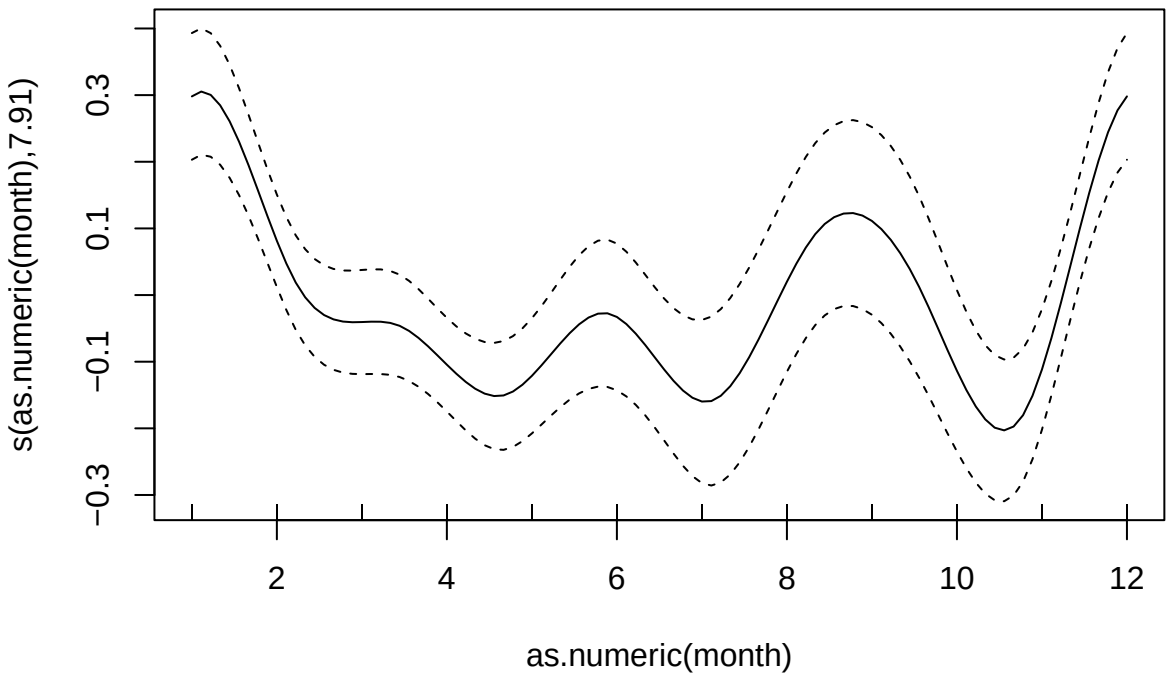



```
h2s_da_model_f_trans_resp <- gam(H2S_daily_avg~s(as.numeric(month),bs='cc') + year + as.character(weekda
                          wd_avg + ws_avg + daily_downwind_ref + capacity +
                           I(1/dist_wrp^2) + I(1/MinDist^2) +
                          s(I(mon_utm_x/10^3), I(mon_utm_y/10^3), bs='tp', k = 10) +
                         te(I(mon_utm_x/10^3), I(mon_utm_y/10^3), as.numeric(day),
                            k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                          monthly_oil_1km + monthly_gas_1km + active_1km +
                         daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                         I(1/dist_dc^2) + avg_temp + avg_hum + precip,
                       data = daily_avg_train_sincefeb2022_trans_resp)
summary(h2s_da_model_f_trans_resp)
```

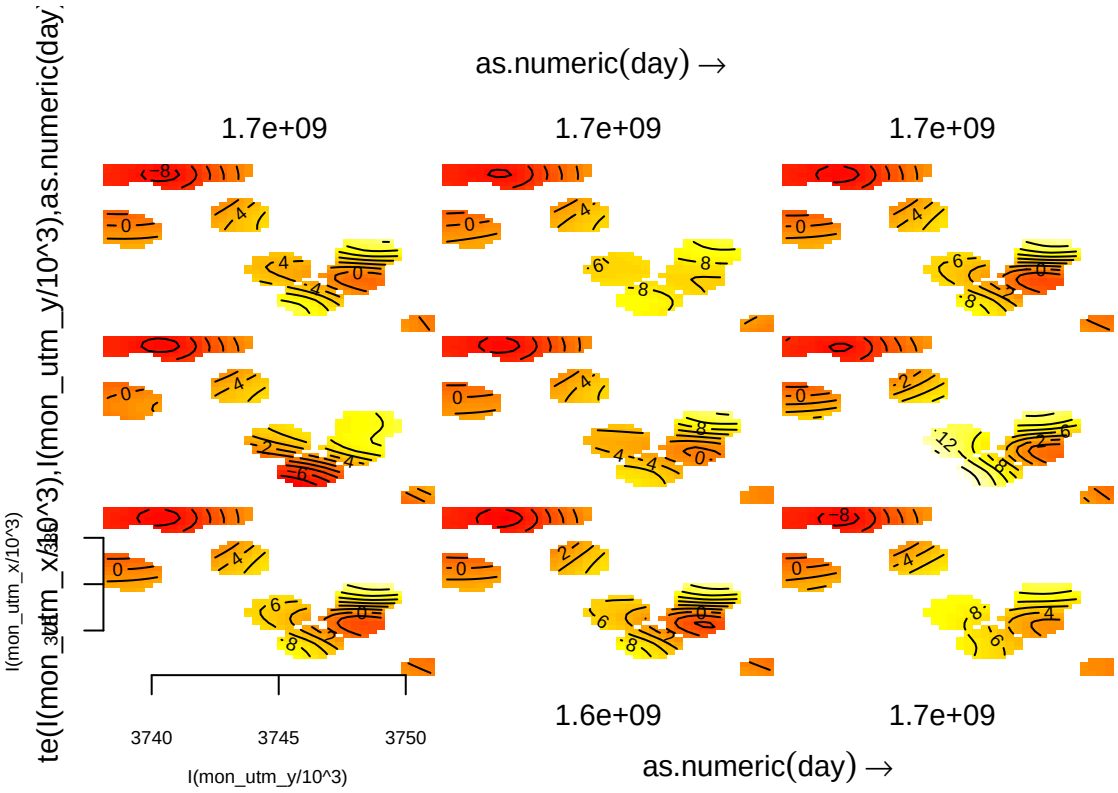
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## H2S_daily_avg ~ s(as.numeric(month), bs = "cc") + year + as.character(weekday) +
##     wd_avg + ws_avg + daily_downwind_ref + capacity + I(1/dist_wrp^2) +
##     I(1/MinDist^2) + s(I(mon_utm_x/10^3), I(mon_utm_y/10^3),
##     bs = "tp", k = 10) + te(I(mon_utm_x/10^3), I(mon_utm_y/10^3),
##     as.numeric(day), k = c(10, 10), d = c(2, 1), bs = c("tp",
##         "cc")) + monthly_oil_1km + monthly_gas_1km + active_1km +
##     daily_downwind_wrp + elevation + EVI + num_odor_complaints +
##     I(1/dist_dc^2) + avg_temp + avg_hum + precip
##
```

```
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2.032e+00  3.038e-01  -6.691 2.72e-11 ***
## year2023                    2.569e-01  5.266e-02   4.878 1.14e-06 ***
## as.character(weekday)Mon -3.541e-02  1.089e-02  -3.251 0.001163 **
## as.character(weekday)Sat -3.210e-02  1.085e-02  -2.960 0.003106 **
## as.character(weekday)Sun -9.900e-02  1.086e-02  -9.118  < 2e-16 ***
## as.character(weekday)Thu -1.272e-02  1.086e-02  -1.172 0.241213
## as.character(weekday)Tue  3.786e-03  1.079e-02   0.351 0.725753
## as.character(weekday)Wed  1.611e-02  1.091e-02   1.476 0.139938
## wd_avg                     1.295e-04  3.856e-05   3.358 0.000798 ***
## ws_avg                    -4.584e-02  2.573e-03 -17.817  < 2e-16 ***
## daily_downwind_ref         4.505e-02  1.410e-02   3.195 0.001417 **
## capacity                   6.577e-03  6.097e-04  10.786  < 2e-16 ***
## I(1/dist_wrp^2)            2.324e-07  4.832e-08   4.810 1.60e-06 ***
## I(1/MinDist^2)            -2.019e-05  2.128e-06  -9.489  < 2e-16 ***
## monthly_oil_1km            7.036e-05  1.819e-05   3.868 0.000112 ***
## monthly_gas_1km            3.169e-04  1.038e-04   3.052 0.002298 **
## active_1km                -3.304e-02  7.420e-03  -4.453 8.82e-06 ***
## daily_downwind_wrp         2.015e-02  1.422e-02   1.418 0.156433
## elevation                  7.133e-03  4.365e-03   1.634 0.102366
## EVI                       -3.184e-01  7.251e-02  -4.391 1.17e-05 ***
## num_odor_complaints        1.170e-02  5.348e-03   2.187 0.028840 *
## I(1/dist_dc^2)             1.914e-05  3.889e-06   4.922 9.10e-07 ***
## avg_temp                   4.552e-03  1.146e-03   3.971 7.37e-05 ***
## avg_hum                   -3.213e-03  3.000e-04 -10.709  < 2e-16 ***
## precip                    -5.246e-02  1.554e-02  -3.377 0.000745 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                            edf Ref.df     F
## s(as.numeric(month))                                     7.913  8.000 17.14
## s(I(mon_utm_x/10^3),I(mon_utm_y/10^3))                   8.471  8.471 31.39
## te(I(mon_utm_x/10^3),I(mon_utm_y/10^3),as.numeric(day)) 75.827 76.000 35.46
##                                                           p-value
## s(as.numeric(month))                                       <2e-16 ***
## s(I(mon_utm_x/10^3),I(mon_utm_y/10^3))                     <2e-16 ***
## te(I(mon_utm_x/10^3),I(mon_utm_y/10^3),as.numeric(day))    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 114/118
## R-sq.(adj) =  0.774   Deviance explained = 78.4%
## GCV = 0.023105  Scale est. = 0.022119  n = 2664
```

```
plot(h2s_da_model_f_trans_resp)
```


s(as.numeric(month),7.91)
−0.3
−0.1
0.1
0.3
2
4
6
8
10
12
as.numeric(month)

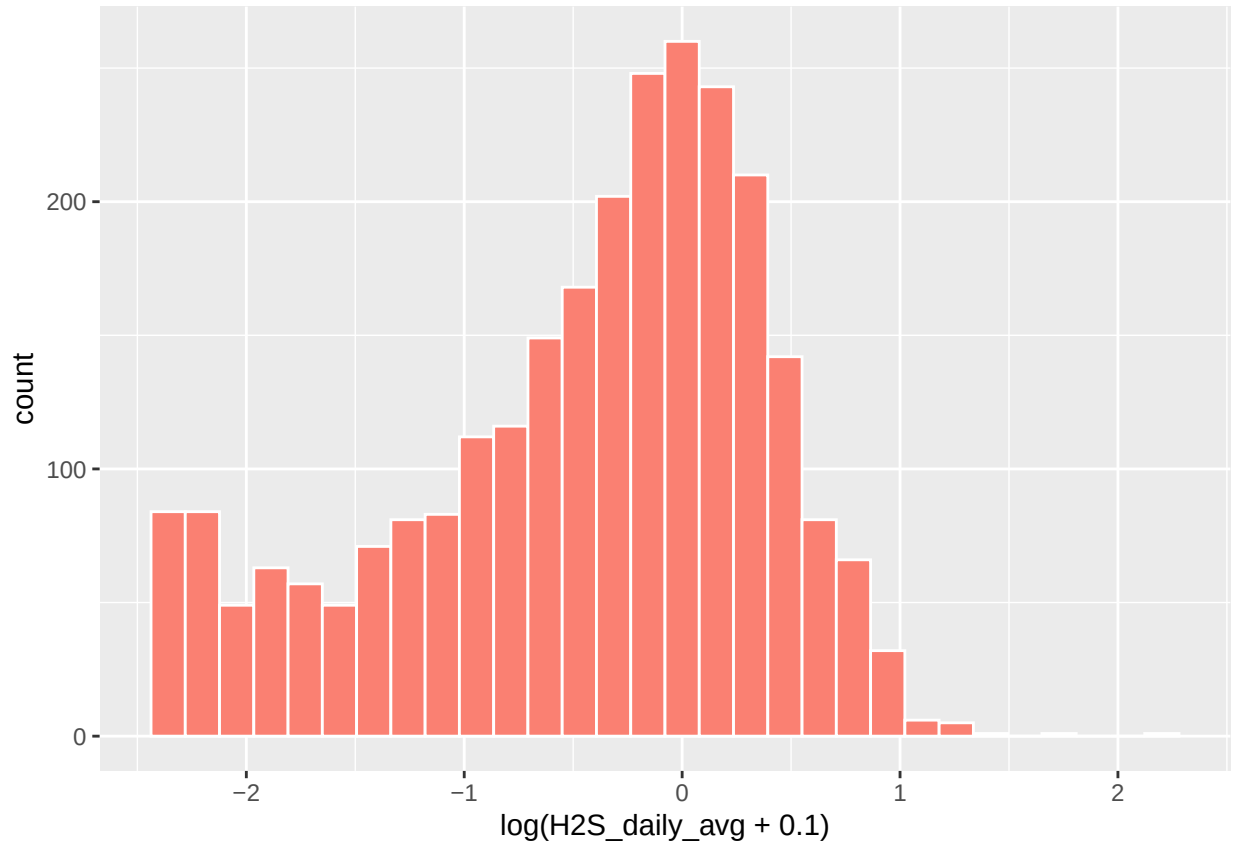

−1se
s(I(mon_utm_x/10^3),I(mon_utm_y/10^3),8.47)
+1se
I(mon_utm_y/10^3)
3738
3740
3742
3744
3746
3748
3750
370
375
380
385
I(mon_utm_x/10^3)



Tranformed log-Resp

```
ggplot(daily_avg_train_sincefeb2022, aes(x = log(H2S_daily_avg+0.1))) +
  geom_histogram(fill = "salmon", col = "white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

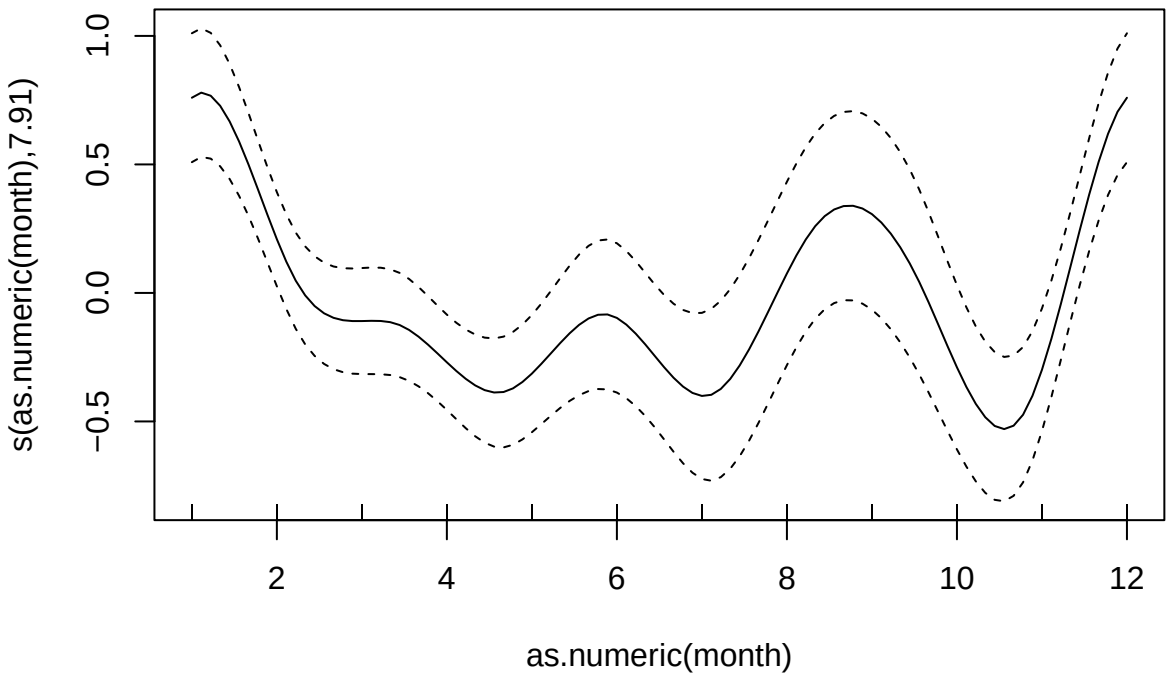


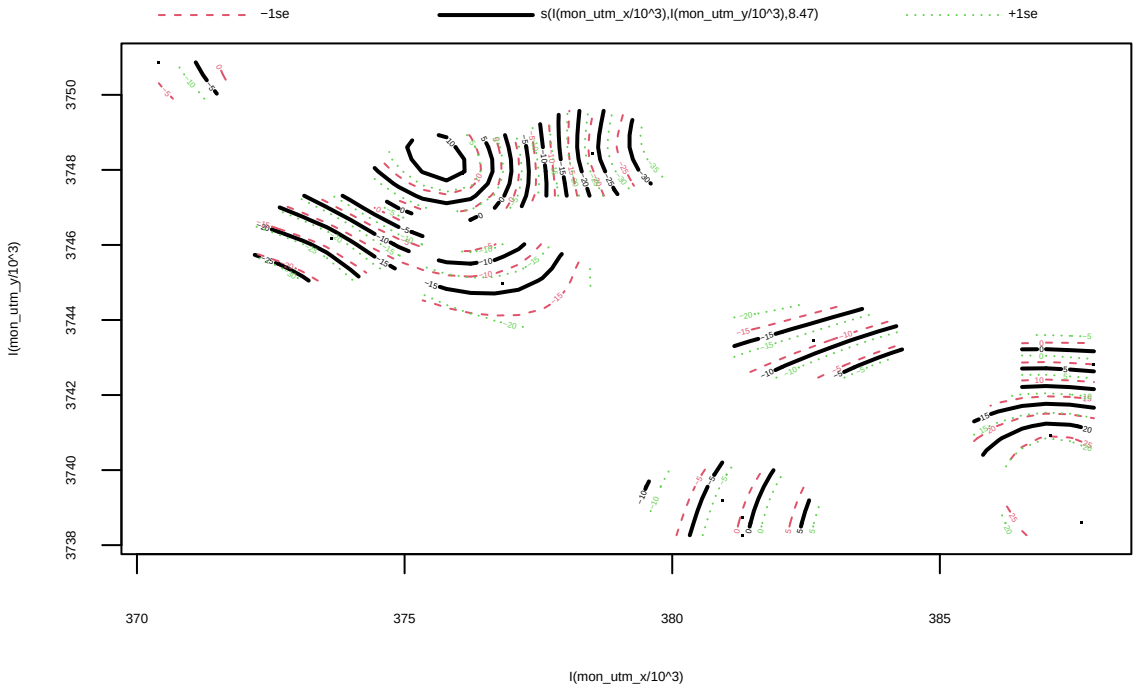

```
h2s_da_model_f_logresp <- gam(I(log(H2S_daily_avg + 0.1))~s(as.numeric(month),bs='cc') + year + as.chara
                          wd_avg + ws_avg + daily_downwind_ref + capacity +
                           I(1/dist_wrp^2) + I(1/MinDist^2) +
                          s(I(mon_utm_x/10^3), I(mon_utm_y/10^3), bs='tp', k = 10) +
                         te(I(mon_utm_x/10^3), I(mon_utm_y/10^3), as.numeric(day),
                            k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                          monthly_oil_1km + monthly_gas_1km + active_1km +
                         daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                         I(1/dist_dc^2) + avg_temp + avg_hum + precip,
                       data = daily_avg_train_sincefeb2022)
summary(h2s_da_model_f_logresp)
```

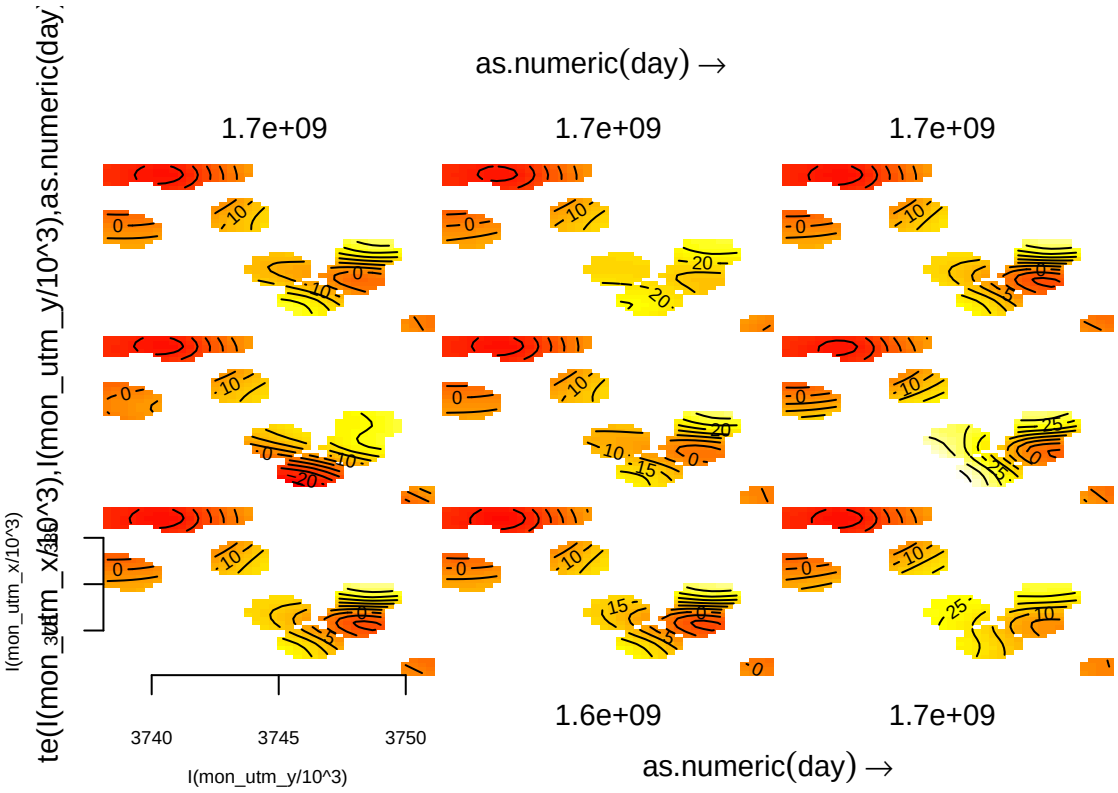
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## I(log(H2S_daily_avg + 0.1)) ~ s(as.numeric(month), bs = "cc") +
##     year + as.character(weekday) + wd_avg + ws_avg + daily_downwind_ref +
##     capacity + I(1/dist_wrp^2) + I(1/MinDist^2) + s(I(mon_utm_x/10^3),
##     I(mon_utm_y/10^3), bs = "tp", k = 10) + te(I(mon_utm_x/10^3),
##     I(mon_utm_y/10^3), as.numeric(day), k = c(10, 10), d = c(2,
##         1), bs = c("tp", "cc")) + monthly_oil_1km + monthly_gas_1km +
##     active_1km + daily_downwind_wrp + elevation + EVI + num_odor_complaints +
##     I(1/dist_dc^2) + avg_temp + avg_hum + precip
##
```

```
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -7.022e+00  8.020e-01  -8.755  < 2e-16 ***
## year2023                    6.721e-01  1.390e-01   4.834 1.42e-06 ***
## as.character(weekday)Mon   -9.463e-02  2.876e-02  -3.291 0.001013 **
## as.character(weekday)Sat   -8.033e-02  2.864e-02  -2.805 0.005071 **
## as.character(weekday)Sun   -2.518e-01  2.867e-02  -8.785  < 2e-16 ***
## as.character(weekday)Thu   -3.250e-02  2.866e-02  -1.134 0.256908
## as.character(weekday)Tue    1.340e-02  2.850e-02   0.470 0.638379
## as.character(weekday)Wed    4.040e-02  2.882e-02   1.402 0.161023
## wd_avg                      3.865e-04  1.018e-04   3.795 0.000151 ***
## ws_avg                     -1.241e-01  6.794e-03 -18.267  < 2e-16 ***
## daily_downwind_ref          1.029e-01  3.724e-02   2.763 0.005767 **
## capacity                    1.552e-02  1.610e-03   9.641  < 2e-16 ***
## I(1/dist_wrp^2)             7.099e-07  1.276e-07   5.566 2.88e-08 ***
## I(1/MinDist^2)             -5.976e-05  5.617e-06 -10.638  < 2e-16 ***
## monthly_oil_1km             1.739e-04  4.800e-05   3.624 0.000296 ***
## monthly_gas_1km             9.319e-04  2.742e-04   3.399 0.000687 ***
## active_1km                 -9.251e-02  1.958e-02  -4.724 2.43e-06 ***
## daily_downwind_wrp          4.822e-02  3.754e-02   1.284 0.199089
## elevation                   2.166e-02  1.153e-02   1.880 0.060279 .
## EVI                        -9.855e-01  1.915e-01  -5.147 2.85e-07 ***
## num_odor_complaints         2.941e-02  1.412e-02   2.082 0.037402 *
## I(1/dist_dc^2)              5.639e-05  1.027e-05   5.492 4.37e-08 ***
## avg_temp                    1.228e-02  3.026e-03   4.058 5.10e-05 ***
## avg_hum                    -8.260e-03  7.920e-04 -10.429  < 2e-16 ***
## precip                     -1.252e-01  4.102e-02  -3.053 0.002287 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                            edf Ref.df     F
## s(as.numeric(month))                                     7.909  8.000 16.55
## s(I(mon_utm_x/10^3),I(mon_utm_y/10^3))                   8.471  8.471 30.23
## te(I(mon_utm_x/10^3),I(mon_utm_y/10^3),as.numeric(day)) 75.816 76.000 37.44
##                                                           p-value
## s(as.numeric(month))                                       <2e-16 ***
## s(I(mon_utm_x/10^3),I(mon_utm_y/10^3))                     <2e-16 ***
## te(I(mon_utm_x/10^3),I(mon_utm_y/10^3),as.numeric(day))    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 114/118
## R-sq.(adj) =   0.78   Deviance explained =   79%
## GCV = 0.16109  Scale est. = 0.15421   n = 2664
```

```
plot(h2s_da_model_f_logresp)
```

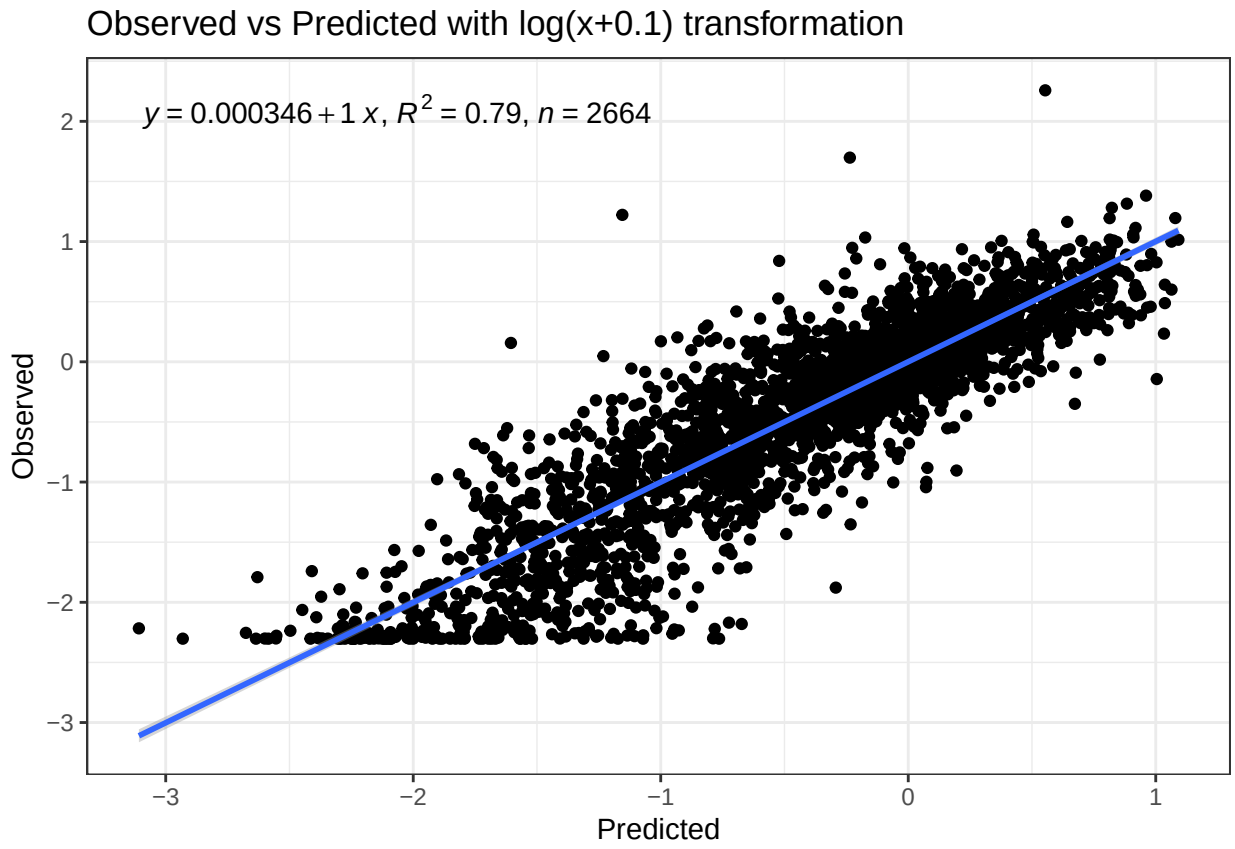

s(as.numeric(month),7.91)
1.0
0.5
0.0
−0.5
2
4
6
8
10
12
as.numeric(month)


−1se
s(I(mon_utm_x/10^3),I(mon_utm_y/10^3),8.47)
+1se
I(mon_utm_y/10^3)
3738
3740
3742
3744
3746
3748
3750
370
375
380
385
I(mon_utm_x/10^3)



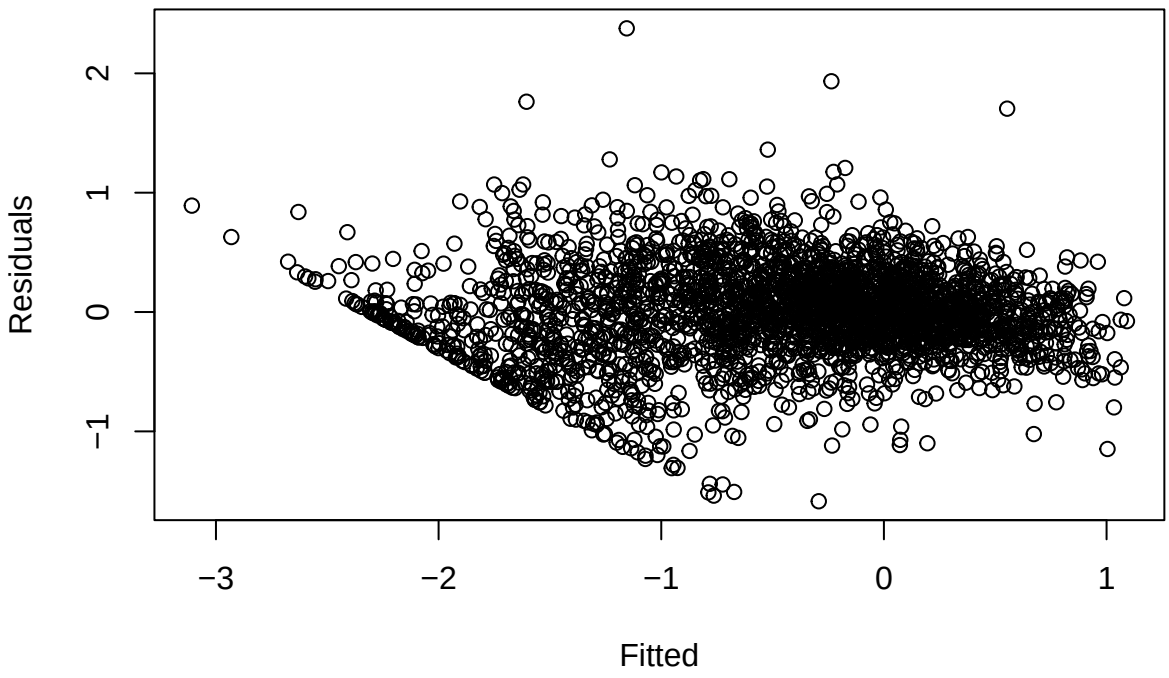
Condition 1

```
# Check Condition 1
ggplot(tibble(obs = log(daily_avg_train_sincefeb2022$H2S_daily_avg + 0.1), pred = fitted(h2s_da_model_f
                        aes(x = pred, y = obs)) +
                    geom_point() +
                    stat_poly_line() +
                    stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                    labs(y = 'Observed', x = 'Predicted',
                        title = 'Observed vs Predicted with log(x+0.1) transformation') +
                    theme_bw()
```

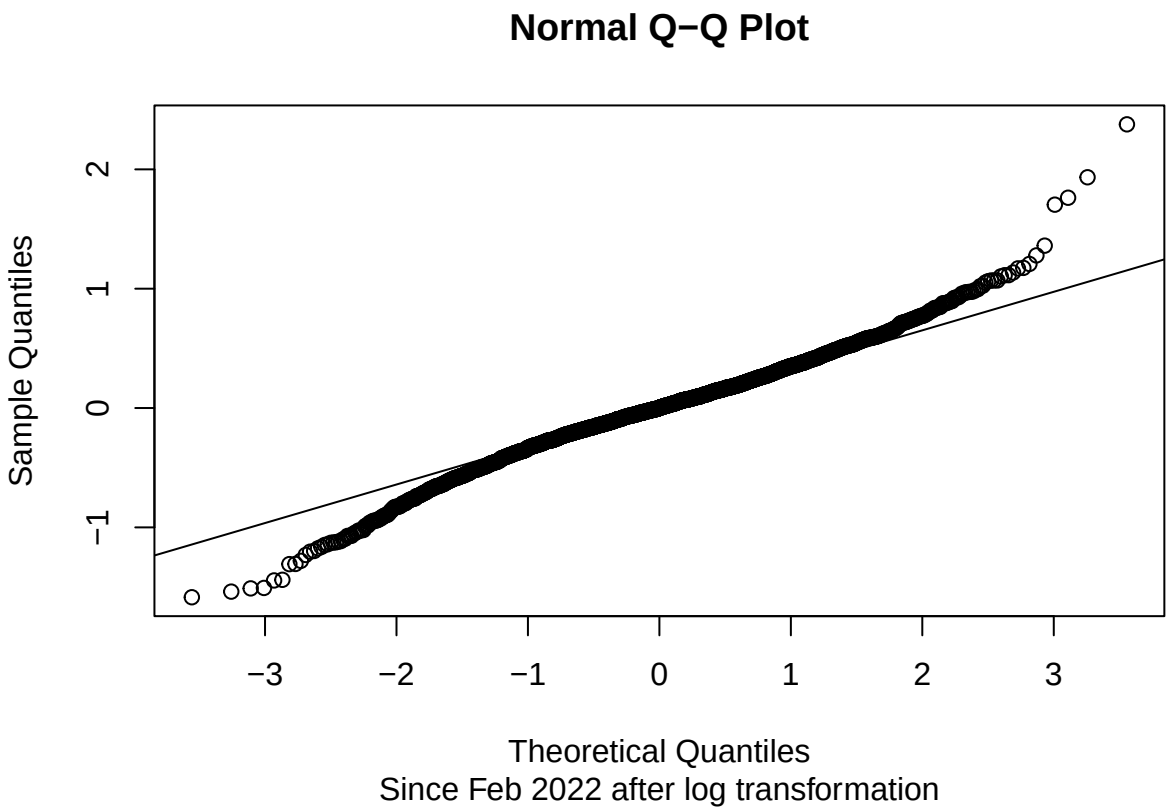


Residual Plots

```
# Residual plots
rs_full <- h2s_da_model_f_logresp$residuals
plot(rs_full~fitted(h2s_da_model_f_logresp), xlab="Fitted", ylab="Residuals")
```




```
# Normal Q-Q Plot
stats::qqnorm(rs_full, sub = "Since Feb 2022 after log transformation")
stats::qqline(rs_full)
```



10-fold CV

```
adj_r2 <- function(r2, n, p){
  return(1 - (1-r2)*(n - 1)/(n - p - 1))
}
```

```
set.seed(1)
random_seeds <- floor(runif(3, min=0, max=1)*100)
```

```
result_10cv <- tibble(Model = character(),
                  '10CV AVG Train R-Sq' = numeric(),
                  '10CV AVG Test R-Sq' = numeric(),
                  'Test RMSE' = numeric())

validation_result_10cv <- tibble(Model = character(),
                            'Coef' = numeric(),
                            'R-Sq' = numeric())

# model 1: Log H2S
obs_10cv_log_trans <- c()
pred_10cv_log_trans <- c()
adj_r2_log_trans <- c()
adj_r2_test_log_trans <- c()

set.seed(random_seeds[1])
```

```r
folds <- createFolds(seq(1, nrow(daily_avg_train_sincefeb2022)), k = 10)

for (fold in seq(1, 10)) {
  test <- daily_avg_train_sincefeb2022[folds[[fold]], ]
  train <- anti_join(daily_avg_train_sincefeb2022, test, by = join_by(dist_dc, day))

  h2s_da_model_10cv <- gam(I(log(H2S_daily_avg + 0.1))~s(as.numeric(month),bs='cc') + year + as.charact
                           wd_avg + ws_avg + daily_downwind_ref + capacity +
                            I(1/dist_wrp^2) + I(1/MinDist^2) +
                           s(I(mon_utm_x/10^3), I(mon_utm_y/10^3), bs='tp', k = 10) +
                          te(I(mon_utm_x/10^3), I(mon_utm_y/10^3), as.numeric(day),
                             k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                           monthly_oil_1km + monthly_gas_1km + active_1km +
                          daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                          I(1/dist_dc^2) + avg_temp + avg_hum + precip,
                            data = train)

  predictions <- predict(h2s_da_model_10cv, newdata = test)
  r2_test <- R2(test$H2S_daily_avg, predictions)
  adj_r2_test <- adj_r2(r2_test, summary(h2s_da_model_10cv)$n, summary(h2s_da_model_10cv)$np)

  obs_10cv_log_trans <- append(obs_10cv_log_trans, test %>% pull(H2S_daily_avg))
  pred_10cv_log_trans <- append(pred_10cv_log_trans, predictions)
  adj_r2_log_trans <- append(adj_r2_log_trans, summary(h2s_da_model_10cv)$r.sq)
  adj_r2_test_log_trans <- append(adj_r2_test_log_trans, adj_r2_test)
}

result_10cv <- rbind(result_10cv, tibble(Model = 'Since Feb 2022 Log Transform',
                              '10CV AVG Train R-Sq' = mean(adj_r2_log_trans),
                              '10CV AVG Test R-Sq' = mean(adj_r2_test_log_trans),
                              'Test RMSE' = RMSE(pred_10cv_log_trans, obs_10cv_log_trans)))

log_h2s_10cv_obs_vs_pred_plot <- ggplot(tibble(obs = obs_10cv_log_trans, pred = pred_10cv_log_trans),
                             aes(x = pred, y = obs)) +
                         geom_point() +
                         stat_poly_line() +
                         stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                         labs(y = 'Observed', x = 'Predicted',
                              title = 'Observed vs Predicted for Since 2022 GAM 10CV Log H2S') +
                         theme_bw()

validation_result_10cv <- rbind(validation_result_10cv,
                            tibble(Model = 'Since Feb 2022 Log Transform',
                                   'Coef' = summary(lm(obs_10cv_log_trans ~
                                                       pred_10cv_log_trans))$coefficients[2, 1],
                                   'R-Sq' = summary(lm(obs_10cv_log_trans ~
                                                       pred_10cv_log_trans))$r.squared))

# model 2: Transform Pred and Resp
obs_10cv_both_trans <- c()
pred_10cv_both_trans <- c()
adj_r2_both_trans <- c()
adj_r2_test_both_trans <- c()
```

```r
set.seed(random_seeds[2])
folds <- createFolds(seq(1, nrow(daily_avg_train_sincefeb2022_both_trans)), k = 10)

for (fold in seq(1, 10)) {
  test <- daily_avg_train_sincefeb2022_both_trans[folds[[fold]], ]
  train <- anti_join(daily_avg_train_sincefeb2022_both_trans, test, by = join_by(dist_dc, day))

  h2s_da_model_10cv <- gam(H2S_daily_avg~s(as.numeric(month),bs='cc') + year + as.character(weekday) +
                           wd_avg + ws_avg + daily_downwind_ref + capacity +
                           dist_wrp + MinDist +
                           s(mon_utm_x, mon_utm_y, bs='tp', k = 10) +
                          te(mon_utm_x, mon_utm_y, as.numeric(day),
                             k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                           monthly_oil_1km + monthly_gas_1km + active_1km +
                          daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                          dist_dc + avg_temp + avg_hum + precip,
                        data = daily_avg_train_sincefeb2022_both_trans)

  predictions <- predict(h2s_da_model_10cv, newdata = test)
  r2_test <- R2(test$H2S_daily_avg, predictions)
  adj_r2_test <- adj_r2(r2_test, summary(h2s_da_model_10cv)$n, summary(h2s_da_model_10cv)$np)

  obs_10cv_both_trans <- append(obs_10cv_both_trans, test %>% pull(H2S_daily_avg))
  pred_10cv_both_trans <- append(pred_10cv_both_trans, predictions)
  adj_r2_both_trans <- append(adj_r2_both_trans, summary(h2s_da_model_10cv)$r.sq)
  adj_r2_test_both_trans <- append(adj_r2_test_both_trans, adj_r2_test)
}

result_10cv <- rbind(result_10cv, tibble(Model = 'Since Feb 2022 Both Transform',
                                '10CV AVG Train R-Sq' = mean(adj_r2_both_trans),
                                '10CV AVG Test R-Sq' = mean(adj_r2_test_both_trans),
                                'Test RMSE' = RMSE(pred_10cv_both_trans, obs_10cv_both_trans)))

both_trans_10cv_obs_vs_pred_plot <- ggplot(tibble(obs = obs_10cv_both_trans, pred = pred_10cv_both_trans
                             aes(x = pred, y = obs)) +
                        geom_point() +
                        stat_poly_line() +
                        stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                        labs(y = 'Observed', x = 'Predicted',
                             title = 'Observed vs Predicted for Since 2022 GAM 10CV both transform') +
                        theme_bw()

validation_result_10cv <- rbind(validation_result_10cv,
                           tibble(Model = 'Since Feb 2022 Both Transform',
                                  'Coef' = summary(lm(obs_10cv_both_trans ~
                                                      pred_10cv_both_trans))$coefficients[2, 1],
                                  'R-Sq' = summary(lm(obs_10cv_both_trans ~
                                                      pred_10cv_both_trans))$r.squared))

# model 3: Transform response according to boxcox
obs_10cv_resp_trans <- c()
pred_10cv_resp_trans <- c()
adj_r2_resp_trans <- c()
```

```
adj_r2_test_resp_trans <- c()

set.seed(random_seeds[3])
folds <- createFolds(seq(1, nrow(daily_avg_train_sincefeb2022_trans_resp)), k = 10)

for (fold in seq(1, 10)) {
  test <- daily_avg_train_sincefeb2022_trans_resp[folds[[fold]], ]
  train <- anti_join(daily_avg_train_sincefeb2022_trans_resp, test, by = join_by(dist_dc, day))

  h2s_da_model_10cv <- gam(H2S_daily_avg~s(as.numeric(month),bs='cc') + year + as.character(weekday) +
                           wd_avg + ws_avg + daily_downwind_ref + capacity +
                            I(1/dist_wrp^2) + I(1/MinDist^2) +
                           s(I(mon_utm_x/10^3), I(mon_utm_y/10^3), bs='tp', k = 10) +
                          te(I(mon_utm_x/10^3), I(mon_utm_y/10^3), as.numeric(day),
                             k=c(10,10),d=c(2,1),bs=c('tp','cc')) +
                           monthly_oil_1km + monthly_gas_1km + active_1km +
                          daily_downwind_wrp + elevation + EVI + num_odor_complaints +
                          I(1/dist_dc^2) + avg_temp + avg_hum + precip,
                        data = daily_avg_train_sincefeb2022_trans_resp)

  predictions <- predict(h2s_da_model_10cv, newdata = test)
  r2_test <- R2(test$H2S_daily_avg, predictions)
  adj_r2_test <- adj_r2(r2_test, summary(h2s_da_model_10cv)$n, summary(h2s_da_model_10cv)$np)

  obs_10cv_resp_trans <- append(obs_10cv_resp_trans, test %>% pull(H2S_daily_avg))
  pred_10cv_resp_trans <- append(pred_10cv_resp_trans, predictions)
  adj_r2_resp_trans <- append(adj_r2_resp_trans, summary(h2s_da_model_10cv)$r.sq)
  adj_r2_test_resp_trans <- append(adj_r2_test_resp_trans, adj_r2_test)
}

result_10cv <- rbind(result_10cv, tibble(Model = 'Since Feb 2022 Resp Transform',
                                '10CV AVG Train R-Sq' = mean(adj_r2_resp_trans),
                                '10CV AVG Test R-Sq' = mean(adj_r2_test_resp_trans),
                                'Test RMSE' = RMSE(pred_10cv_resp_trans, obs_10cv_resp_trans)))

resp_trans_10cv_obs_vs_pred_plot <- ggplot(tibble(obs = obs_10cv_resp_trans, pred = pred_10cv_resp_trans
                             aes(x = pred, y = obs)) +
                        geom_point() +
                        stat_poly_line() +
                        stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                        labs(y = 'Observed', x = 'Predicted',
                             title = 'Observed vs Predicted for Since 2022 GAM 10CV resp H2S') +
                        theme_bw()

validation_result_10cv <- rbind(validation_result_10cv,
                           tibble(Model = 'Since Feb 2022 Resp Transform',
                                  'Coef' = summary(lm(obs_10cv_resp_trans ~
                                                      pred_10cv_resp_trans))$coefficients[2, 1],
                                  'R-Sq' = summary(lm(obs_10cv_resp_trans ~
                                                      pred_10cv_resp_trans))$r.squared))

validation_result_10cv
```

```
## # A tibble: 3 x 3
```

```
##   Model                          Coef `R-Sq`
##   <chr>                         <dbl>  <dbl>
## 1 Since Feb 2022 Log Transform  0.622  0.591
## 2 Since Feb 2022 Both Transform 1.03   0.685
## 3 Since Feb 2022 Resp Transform 1.00   0.784
```

XGBoost: H2S^0.37

```
validation_result <- tibble(Model = character(),
                            'Coef' = character(),
                            'R-Sq' = numeric())

xgb_result <- tibble(Model = character(),
                 '10CV Train R-Sq' = numeric(),
                 '10CV Test R-Sq' = numeric(),
                 '10CV Test RMSE' = numeric())

daily_avg_train_sincefeb2022_trans_resp <- daily_avg_train_sincefeb2022_trans_resp %>%
  mutate(daily_downwind_ref = as.integer(daily_downwind_ref),
         daily_downwind_wrp = as.integer(daily_downwind_wrp))

train <- fastDummies::dummy_cols(daily_avg_train_sincefeb2022_trans_resp %>%
                   select(-c(day)) %>%
                   mutate(MinDist = 1/(MinDist^2),
                          dist_wrp = 1/(dist_wrp^2),
                          weekday = as.character(weekday)),
                   remove_selected_columns = TRUE)

# Try for a continuous month
tune_grid <- expand.grid(nrounds = c(100, 200, 500),
                         max_depth = c(3, 4, 5),
                         eta = c(0.1, 0.3),
                         gamma = c(0.01, 0.001),
                         colsample_bytree = c(0.5, 1),
                         min_child_weight = 0,
                         subsample = c(0.5, 0.75, 1))

# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv",
                        number=10,
                        verboseIter=TRUE,
                        search='grid',
                        savePredictions = 'final')

fit.xgb_da_resp_trans <- readRDS('rfiles/fit.xgb_da_resp_trans.rds')
# fit.xgb_da_resp_trans <- train(H2S_daily_avg~.,
#                   method = 'xgbTree',
#                   data = train,
#                   trControl=control,
#                   tuneGrid = tune_grid,
#                   tuneLength = 10, importance=TRUE, verbosity = 0, verbose=FALSE)
# saveRDS(fit.xgb_da_resp_trans, 'rfiles/fit.xgb_da_resp_trans.rds')
```

```r
getTrainPerf(fit.xgb_da_resp_trans)
```

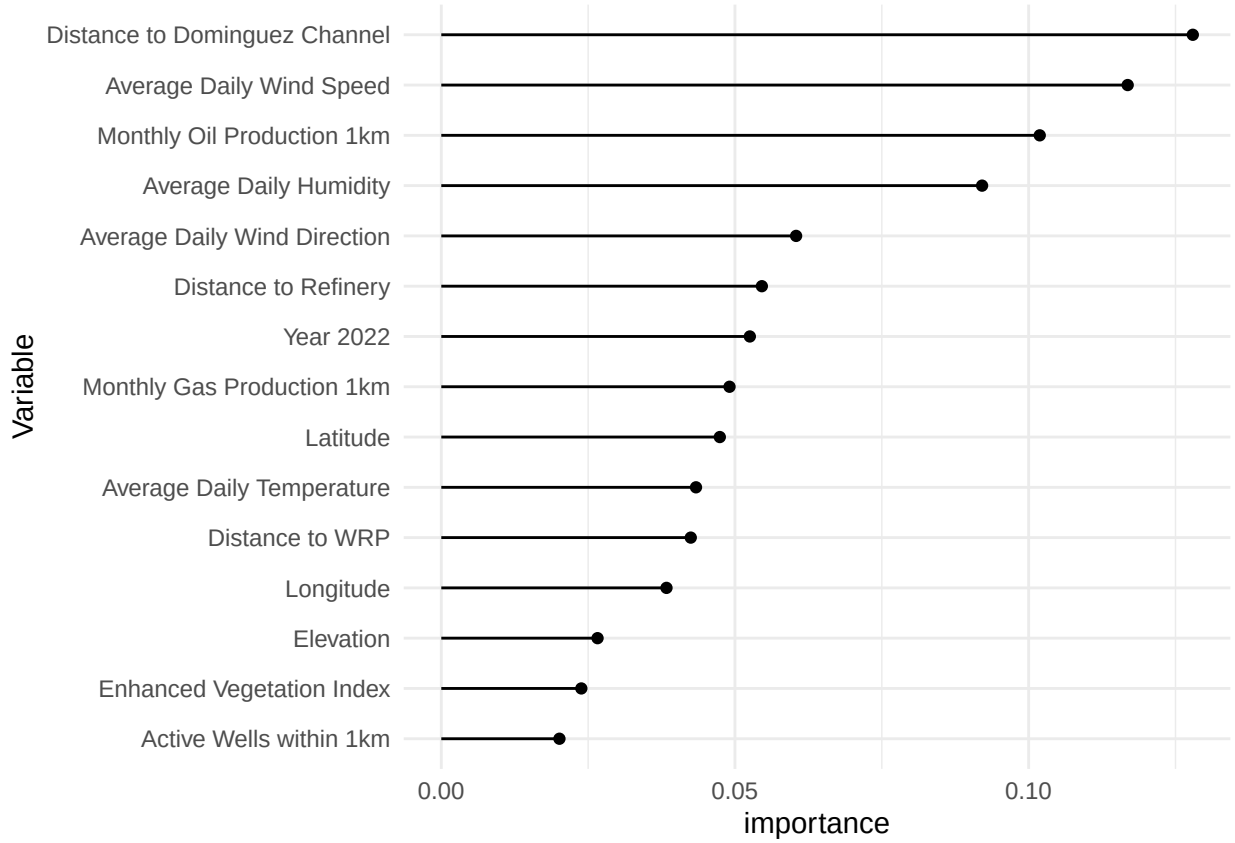
```
##   TrainRMSE TrainRsquared   TrainMAE  method
## 1 0.1237656     0.8435917 0.08549404 xgbTree
```

```r
fit.xgb_da_resp_trans$finalModel
```

```
## ##### xgb.Booster
## Handle is invalid! Suggest using xgb.Booster.complete
## raw: 1.2 Mb
## call:
##   xgboost::xgb.train(params = list(eta = param$eta, max_depth = param$max_depth,
##     gamma = param$gamma, colsample_bytree = param$colsample_bytree,
##     min_child_weight = param$min_child_weight, subsample = param$subsample),
##     data = x, nrounds = param$nrounds, verbose = FALSE, objective = "reg:squarederror",
##     importance = TRUE, verbosity = 0)
## params (as set within xgb.train):
##   eta = "0.1", max_depth = "5", gamma = "0.001", colsample_bytree = "0.5", min_child_weight = "0", su
## # of features: 40
## niter: 500
## nfeatures : 40
## xNames : MinDist wd_avg ws_avg daily_downwind_ref capacity dist_wrp mon_utm_x mon_utm_y monthly_oil_
## problemType : Regression
## tuneValue :
##     nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 78     500         5 0.1 0.001              0.5                0      0.75
## obsLevels : NA
## param :
##  $importance
## [1] TRUE
##
## $verbosity
## [1] 0
##
## $verbose
## [1] FALSE
```

```r
names <- c("Longitude" = "mon_utm_x", "Latitude" = "mon_utm_y",
        "Distance to Refinery" = "MinDist", "Angle to Refinery" = "Converted_Angle",
        "Active Wells within 1km" = "active_1km",
        "Monthly Oil Production 1km" = "monthly_oil_1km",
        "Monthly Gas Production 1km" = "monthly_gas_1km",
        "Distance to WRP" = "dist_wrp",
        "WRP Capacity" = "dist_wrp",
        "Angle to WRP" = "wrp_angle",
        "Distance to Dominguez Channel" = "dist_dc",
        "Average Daily Temperature" = "avg_temp",
        "Average Daily Humidity" = "avg_hum",
        "Daily Precipitation" = "precip",
        "Average Daily Wind Speed" = "ws_avg",
        "Average Daily Wind Direction" = "wd_avg",
        "Downwind Refinery" = "daily_downwind_ref",
        "Downwind WRP" = "daily_downwind_wrp",
        "Elevation" = "elevation",
        "Enhanced Vegetation Index" = "EVI",
```

```
        "Number of Daily Odor Complaints" = "num_odor_complaints",
        "Year 2022" = "year_2022",
        "Year 2023" = "year_2023",
        "January" = "month_01",
        "February" = "month_02",
        "March" = "month_03",
        "April" = "month_04",
        "May" = "month_05",
        "June" = "month_06",
        "July" = "month_07",
        "August" = "month_08",
        "September" = "month_09",
        "October" = "month_10",
        "November" = "month_11",
        "December" = "month_12",
        "Monday" = "weekday_Mon",
        "Tuesday" = "weekday_Tue",
        "Wednesday" = "weekday_Wed",
        "Thursday" = "weekday_Thu",
        "Friday" = "weekday_Fri",
        "Saturday" = "weekday_Sat",
        "Sunday" = "weekday_Sun")

imp<-varImp(fit.xgb_da_resp_trans,scale=FALSE)

# rename variables
imp <- tibble(variable = rownames(imp$importance), importance = imp$importance$Overall) %>%
    pivot_wider(names_from = variable,
                values_from = importance) %>%
    rename(any_of(names)) %>%
    pivot_longer(cols = everything(),names_to = 'variable', values_to = 'importance')

imp %>%
  top_n(15, importance) %>%
  ggplot(aes(x=reorder(variable, importance), y=importance)) +
  geom_point() +
  geom_segment(aes(x=variable,xend=variable,y=0,yend=importance)) +
  ylab("importance") +
  xlab("Variable") +
  coord_flip() +
  theme_minimal()
```



Fold Performance

```
# Here, we compute the R2 and RMSE for each fold and take the average
fold_stat <- fit.xgb_da_resp_trans$pred %>% group_by(Resample) %>%
  summarise(R2 = R2(pred, obs), RMSE = RMSE(pred, obs))
test_r2 <- mean(fold_stat$R2)
test_rmse <- mean(fold_stat$RMSE)
```

```
resp_trans_xgb_sincefeb2022_obs_vs_pred_plot <- ggplot(tibble(obs = fit.xgb_da_resp_trans$pred$obs,
                                                              pred = fit.xgb_da_resp_trans$pred$pred),
                                                       aes(x = pred, y = obs)) +
                                                  geom_point() +
                                                  labs(y = 'Observed', x = 'Predicted',
                                                       title = 'Observed vs Predicted for Since 2022 XGBoost Resp Transform') +
                                                  stat_poly_line() +
                                                  stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                                                  theme_bw()
```

```
validation_result <- rbind(validation_result,
                           tibble(Model = 'Since Feb 2022 Resp Transform',
                                  'Coef' = summary(lm(fit.xgb_da_resp_trans$pred$obs ~
                                                      fit.xgb_da_resp_trans$pred$pred))$coefficients[1
                                  'R-Sq' = summary(lm(fit.xgb_da_resp_trans$pred$obs ~
                                                      fit.xgb_da_resp_trans$pred$pred))$r.squared))
```

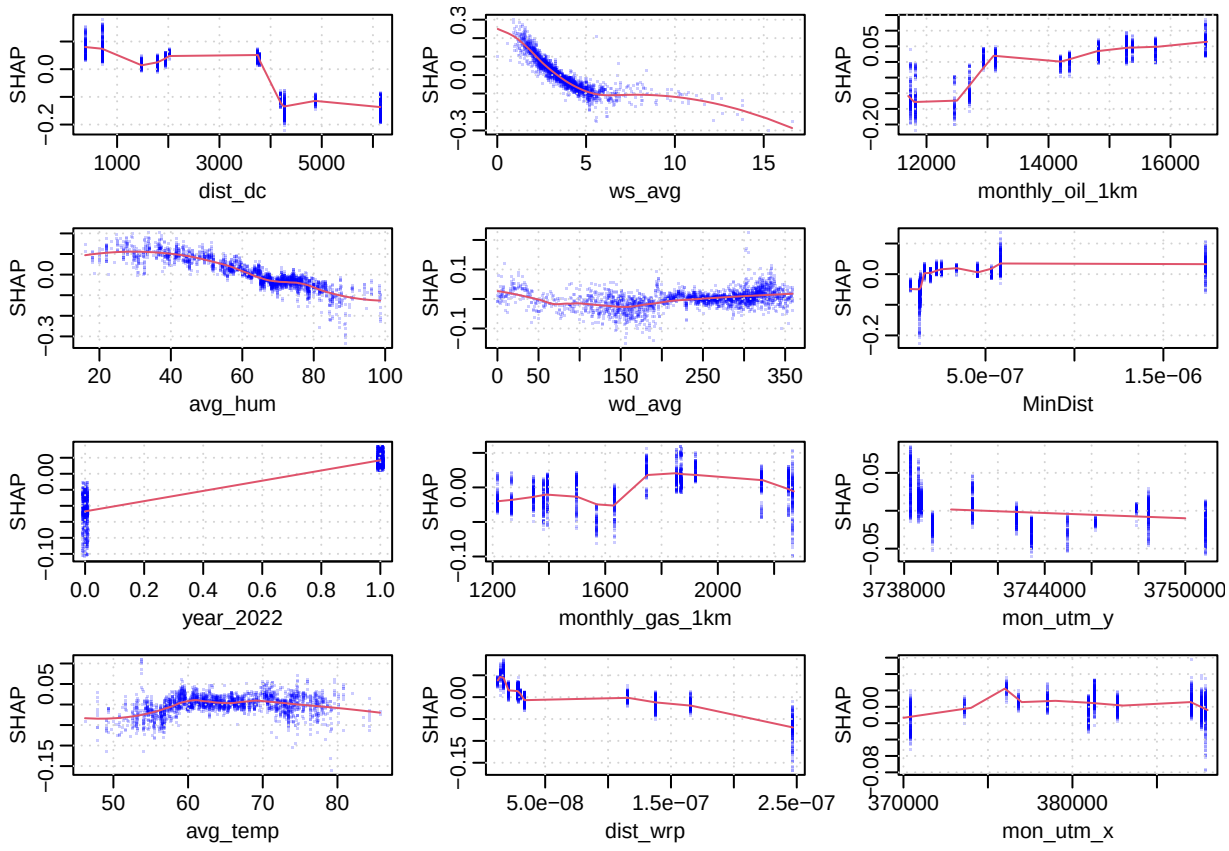
SHAP

```
matrix <- train %>% select(-H2S_daily_avg)
matrix <- as.matrix(sapply(matrix, as.numeric))
xgb.plot.shap(data = matrix,
              model = fit.xgb_da_resp_trans$finalModel, top_n = 12, n_col = 3)
```

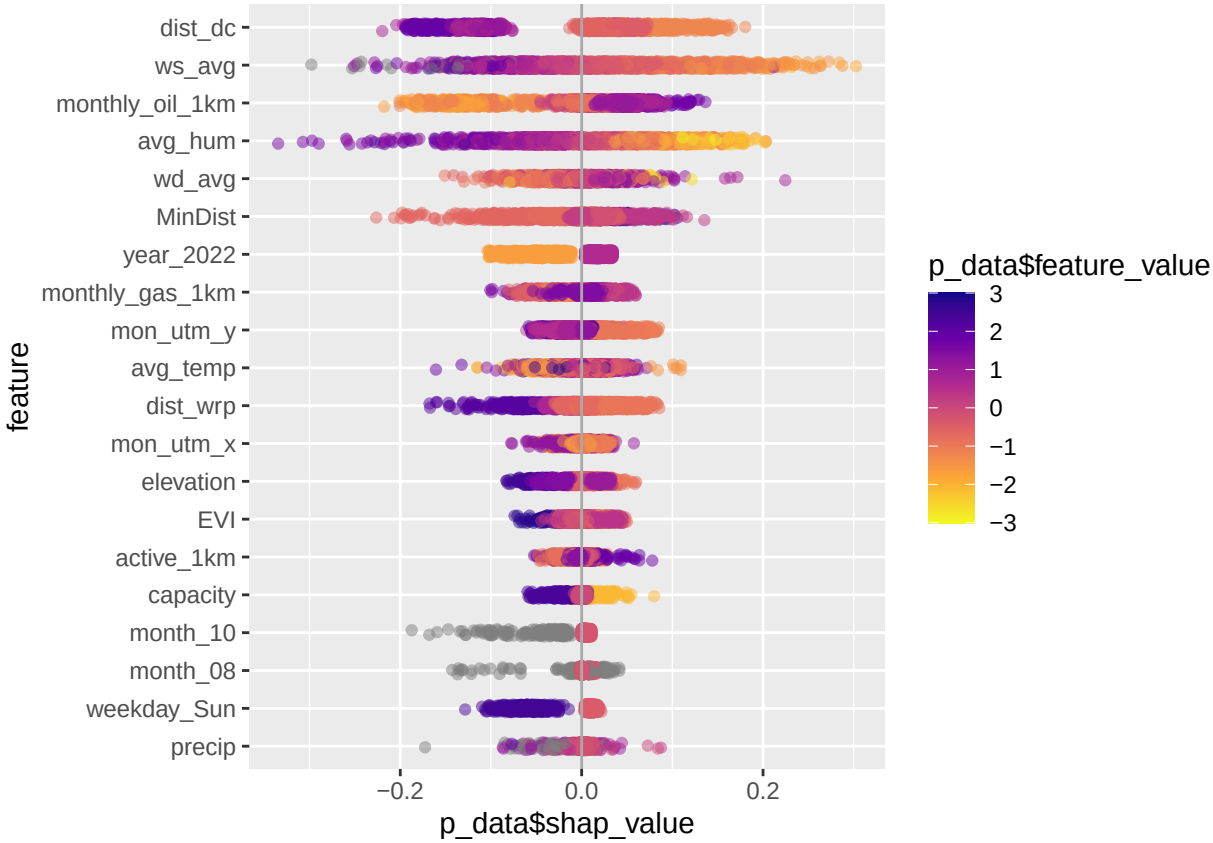
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 3.83e+05
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 4000
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 0
```




```
xgb.ggplot.shap.summary(data = matrix,
              model = fit.xgb_da_resp_trans$finalModel, top_n = 20)
```




```
train_adj_r2 <- adj_r2(getTrainPerf(fit.xgb_da_resp_trans)$TrainRsquared,
                       nrow(fit.xgb_da_resp_trans$trainingData),
                       fit.xgb_da_resp_trans$finalModel$nfeatures)
test_adj_r2 <- adj_r2(test_r2,
                       nrow(fit.xgb_da_resp_trans$trainingData),
                       fit.xgb_da_resp_trans$finalModel$nfeatures)

xgb_result <- rbind(xgb_result,
                    tibble(Model = 'Since Feb 2022 Resp Transform',
                           '10CV Train R-Sq' = train_adj_r2,
                           '10CV Test R-Sq' = test_adj_r2,
                           '10CV Test RMSE' = test_rmse))
```

XGBoost: log(H2S+0.1)

```
train <- daily_avg_train_sincefeb2022 %>%
  mutate(daily_downwind_ref = as.integer(daily_downwind_ref),
         daily_downwind_wrp = as.integer(daily_downwind_wrp)) %>%
  mutate(H2S_daily_avg = log(H2S_daily_avg+0.1))

train <- fastDummies::dummy_cols(train %>%
                    select(-c(day)) %>%
                    mutate(MinDist = 1/(MinDist^2),
                           dist_wrp = 1/(dist_wrp^2),
                           weekday = as.character(weekday)),
```

```
                  remove_selected_columns = TRUE)

# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv",
                        number=10,
                        verboseIter=TRUE,
                        search='grid',
                        savePredictions = 'final')

fit.xgb_da_log_h2s <- readRDS('rfiles/fit.xgb_da_log_h2s.rds')
# fit.xgb_da_log_h2s <- train(H2S_daily_avg~.,
#                   method = 'xgbTree',
#                   data = train,
#                   trControl=control,
#                   tuneGrid = tune_grid,
#                   tuneLength = 10, importance=TRUE, verbosity = 0, verbose=FALSE)
# saveRDS(fit.xgb_da_log_h2s, 'rfiles/fit.xgb_da_log_h2s.rds')
```

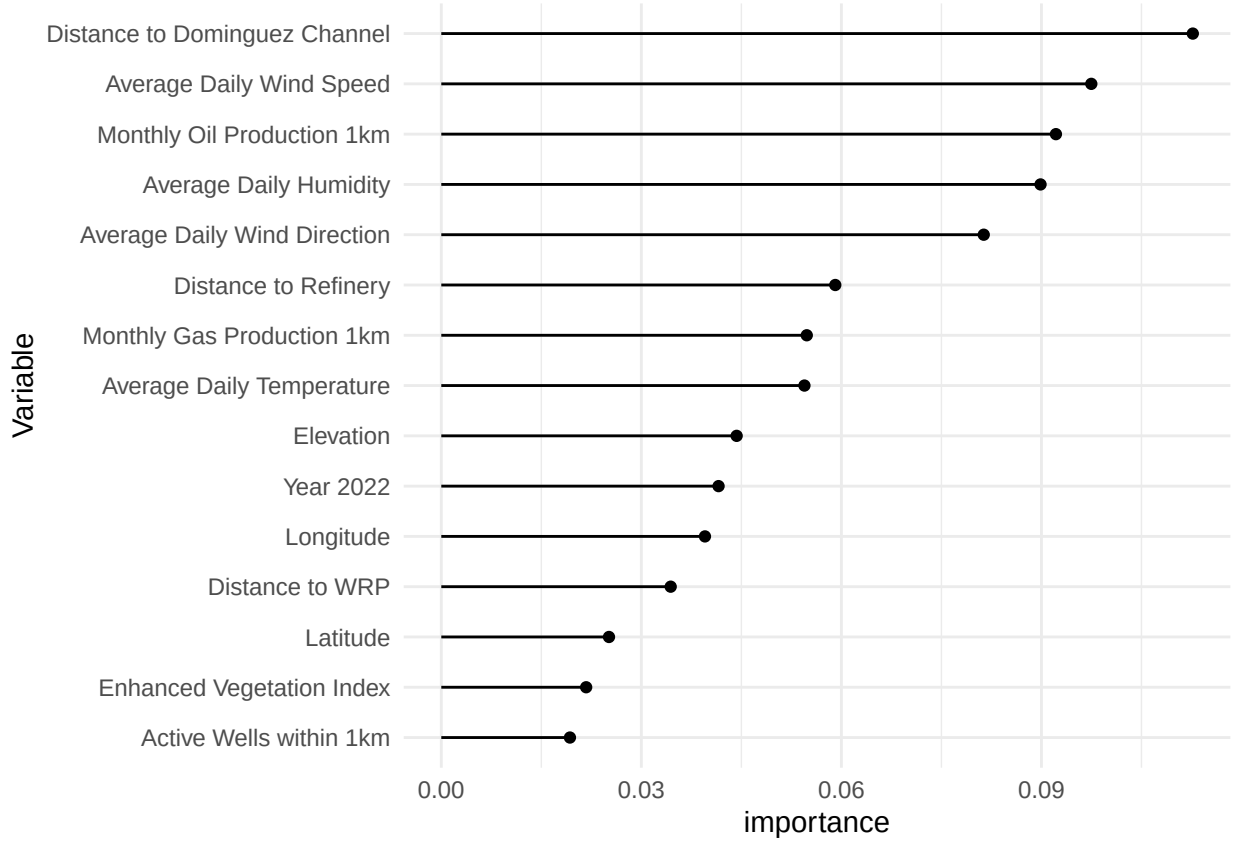
```
getTrainPerf(fit.xgb_da_log_h2s)
```

```
##   TrainRMSE TrainRsquared  TrainMAE  method
## 1 0.3363177     0.8395088 0.2348152 xgbTree
```

```
fit.xgb_da_log_h2s$finalModel
```

```
## ##### xgb.Booster
## Handle is invalid! Suggest using xgb.Booster.complete
## raw: 1.2 Mb
## call:
##   xgboost::xgb.train(params = list(eta = param$eta, max_depth = param$max_depth,
##     gamma = param$gamma, colsample_bytree = param$colsample_bytree,
##     min_child_weight = param$min_child_weight, subsample = param$subsample),
##     data = x, nrounds = param$nrounds, verbose = FALSE, objective = "reg:squarederror",
##     importance = TRUE, verbosity = 0)
## params (as set within xgb.train):
##   eta = "0.1", max_depth = "5", gamma = "0.01", colsample_bytree = "0.5", min_child_weight = "0", su
## # of features: 40
## niter: 500
## nfeatures : 40
## xNames : MinDist wd_avg ws_avg daily_downwind_ref capacity dist_wrp mon_utm_x mon_utm_y monthly_oil_
## problemType : Regression
## tuneValue :
##     nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 93     500         5 0.1  0.01              0.5                0       0.5
## obsLevels : NA
## param :
##  $importance
## [1] TRUE
##
## $verbosity
## [1] 0
##
## $verbose
## [1] FALSE
```

```
imp<-varImp(fit.xgb_da_log_h2s,scale=FALSE)

# rename variables
imp <- tibble(variable = rownames(imp$importance), importance = imp$importance$Overall) %>%
    pivot_wider(names_from = variable,
                values_from = importance) %>%
    rename(any_of(names)) %>%
    pivot_longer(cols = everything(),names_to = 'variable', values_to = 'importance')

imp %>%
  top_n(15, importance) %>%
  ggplot(aes(x=reorder(variable, importance), y=importance)) +
  geom_point() +
  geom_segment(aes(x=variable,xend=variable,y=0,yend=importance)) +
  ylab("importance") +
  xlab("Variable") +
  coord_flip() +
  theme_minimal()
```



Fold Performance

```
# Here, we compute the R2 and RMSE for each fold and take the average
fold_stat <- fit.xgb_da_log_h2s$pred %>% group_by(Resample) %>%
  summarise(R2 = R2(pred, obs), RMSE = RMSE(pred, obs))
test_r2 <- mean(fold_stat$R2)
```

```
test_rmse <- mean(fold_stat$RMSE)
```

```
log_h2s_xgb_sincefeb2022_obs_vs_pred_plot <- ggplot(tibble(obs = fit.xgb_da_log_h2s$pred$obs,
                                                           pred = fit.xgb_da_log_h2s$pred$pred),
                            aes(x = pred, y = obs)) +
                        geom_point() +
                        labs(y = 'Observed', x = 'Predicted',
                            title = 'Observed vs Predicted for Since 2022 XGBoost Log H2S') +
                        stat_poly_line() +
                        stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                        theme_bw()
```

```
validation_result <- rbind(validation_result,
                           tibble(Model = 'Since Feb 2022 Log Transform',
                                  'Coef' = summary(lm(fit.xgb_da_log_h2s$pred$obs ~
                                                      fit.xgb_da_log_h2s$pred$pred))$coefficients[2,
                                  'R-Sq' = summary(lm(fit.xgb_da_log_h2s$pred$obs ~
                                                      fit.xgb_da_log_h2s$pred$pred))$r.squared))
```

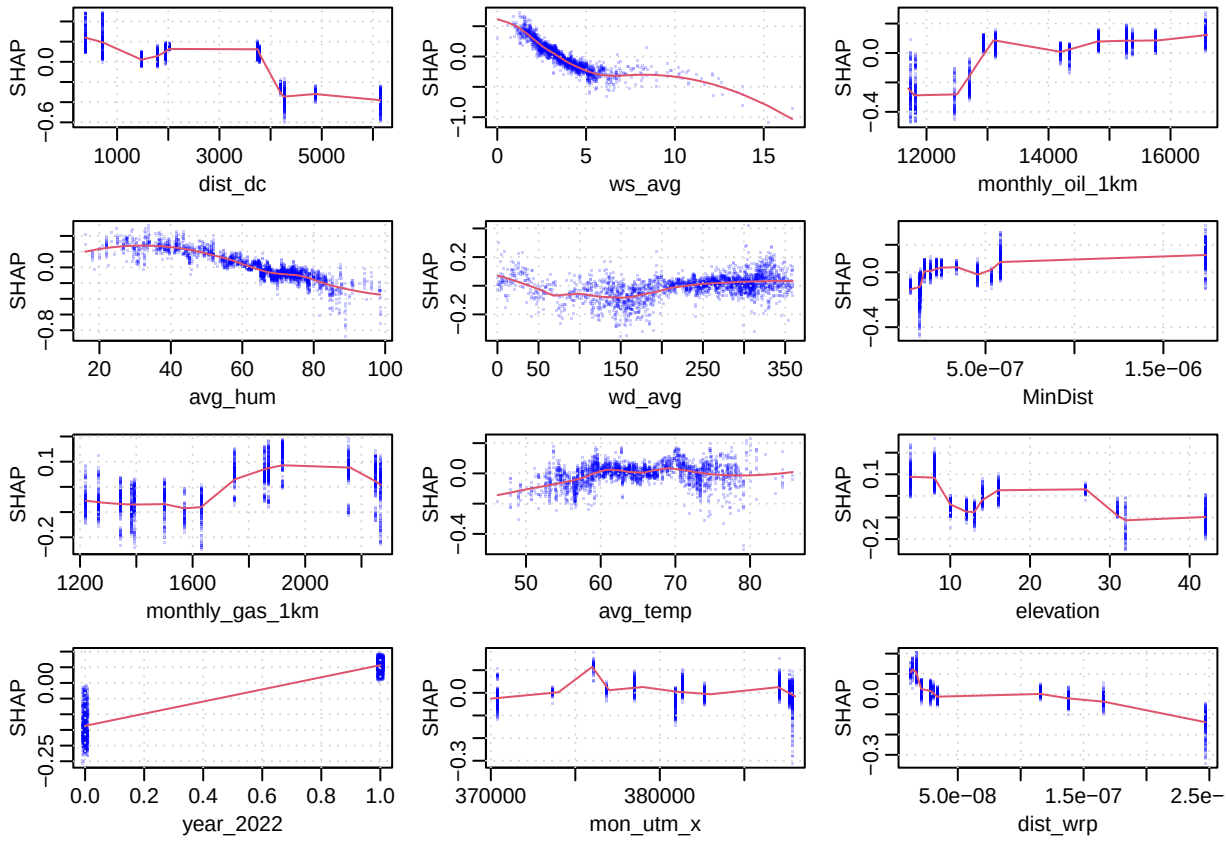
SHAP

```
matrix <- train %>% select(-H2S_daily_avg)
matrix <- as.matrix(sapply(matrix, as.numeric))
xgb.plot.shap(data = matrix,
              model = fit.xgb_da_log_h2s$finalModel, top_n = 12, n_col = 3)
```

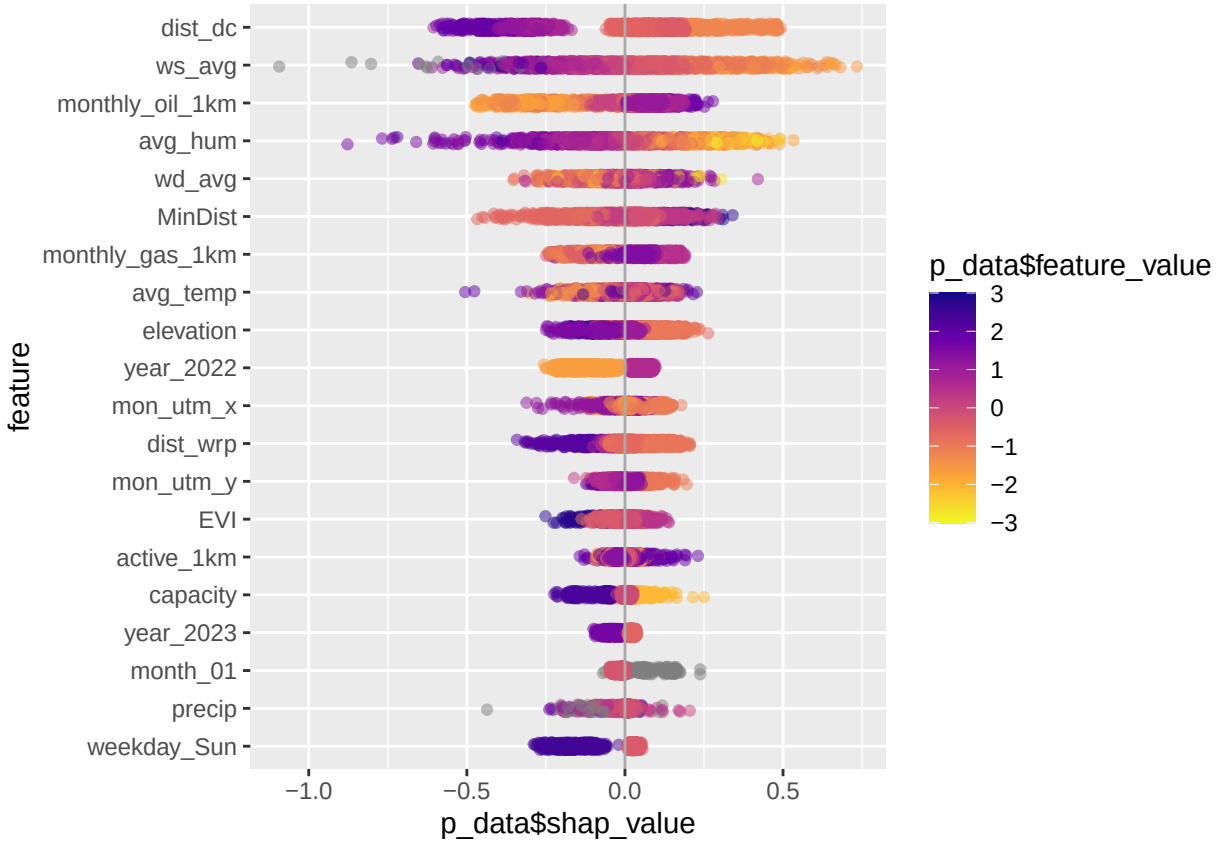
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 3.83e+05
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 4000
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 0
```




```
xgb.ggplot.shap.summary(data = matrix,
              model = fit.xgb_da_log_h2s$finalModel, top_n = 20)
```




```
train_adj_r2 <- adj_r2(getTrainPerf(fit.xgb_da_log_h2s)$TrainRsquared,
                       nrow(fit.xgb_da_log_h2s$trainingData),
                       fit.xgb_da_log_h2s$finalModel$nfeatures)
test_adj_r2 <- adj_r2(test_r2,
                       nrow(fit.xgb_da_log_h2s$trainingData),
                       fit.xgb_da_log_h2s$finalModel$nfeatures)

xgb_result <- rbind(xgb_result,
                    tibble(Model = 'Since Feb 2022 Log Transform',
                           '10CV Train R-Sq' = train_adj_r2,
                           '10CV Test R-Sq' = test_adj_r2,
                           '10CV Test RMSE' = test_rmse))
```

XGBoost: Both Transform

```
train <- fastDummies::dummy_cols(daily_avg_train_sincefeb2022_both_trans %>%
                   select(-c(day)) %>%
                   mutate(MinDist = 1/(MinDist^2),
                          dist_wrp = 1/(dist_wrp^2),
                          weekday = as.character(weekday)),
                   remove_selected_columns = TRUE)

# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv",
                        number=10,
```

```
                        verboseIter=TRUE,
                        search='grid',
                        savePredictions = 'final')

fit.xgb_da_both_trans <- readRDS('rfiles/fit.xgb_da_both_trans.rds')
# fit.xgb_da_both_trans <- train(H2S_daily_avg~.,
#                   method = 'xgbTree',
#                   data = train,
#                   trControl=control,
#                   tuneGrid = tune_grid,
#                   tuneLength = 10, importance=TRUE, verbosity = 0, verbose=FALSE)
# saveRDS(fit.xgb_da_both_trans, 'rfiles/fit.xgb_da_both_trans.rds')
```

```
getTrainPerf(fit.xgb_da_both_trans)
```

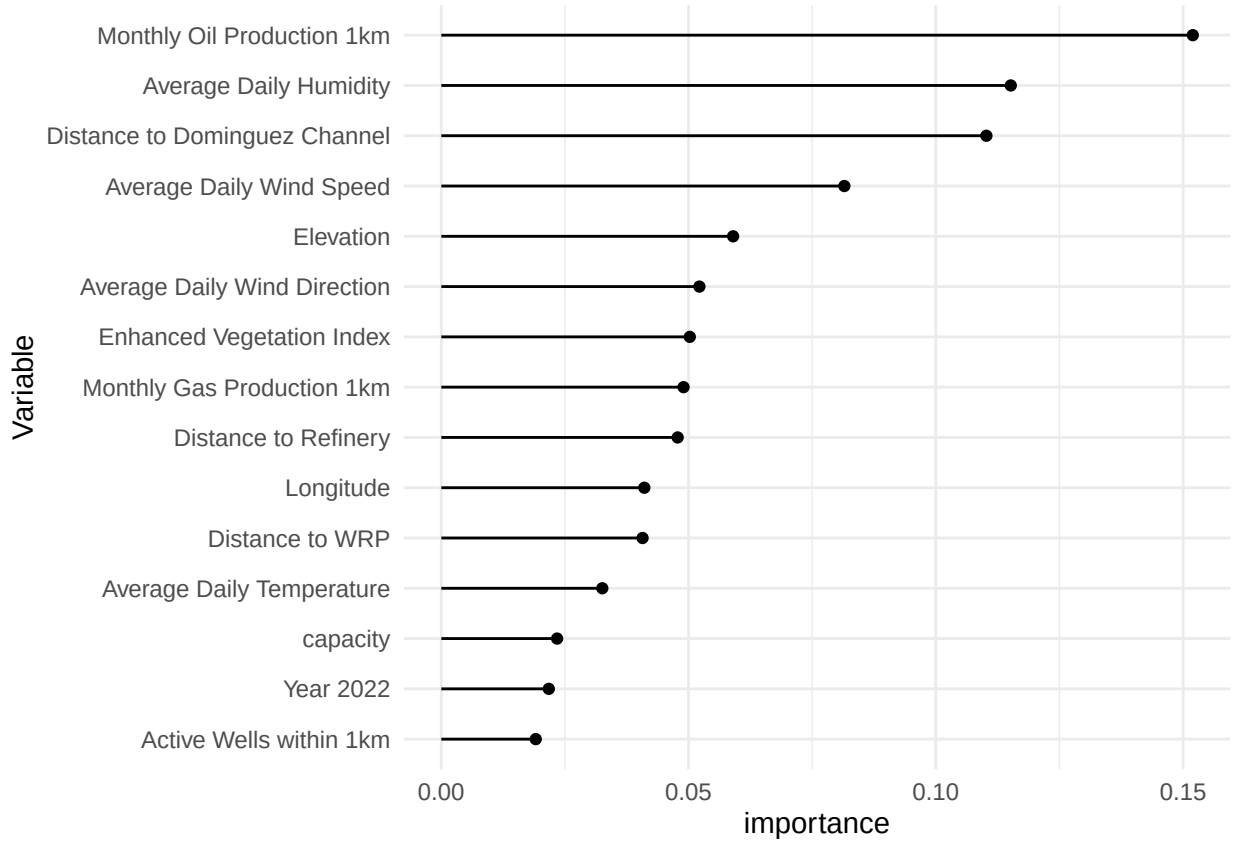
```
##   TrainRMSE TrainRsquared   TrainMAE  method
## 1 0.1203762     0.8341171 0.08215884 xgbTree
```

```
fit.xgb_da_both_trans$finalModel
```

```
## ##### xgb.Booster
## Handle is invalid! Suggest using xgb.Booster.complete
## raw: 1.2 Mb
## call:
##   xgboost::xgb.train(params = list(eta = param$eta, max_depth = param$max_depth,
##     gamma = param$gamma, colsample_bytree = param$colsample_bytree,
##     min_child_weight = param$min_child_weight, subsample = param$subsample),
##     data = x, nrounds = param$nrounds, verbose = FALSE, objective = "reg:squarederror",
##     importance = TRUE, verbosity = 0)
## params (as set within xgb.train):
##   eta = "0.1", max_depth = "5", gamma = "0.001", colsample_bytree = "0.5", min_child_weight = "0", su
## # of features: 40
## niter: 500
## nfeatures : 40
## xNames : MinDist wd_avg ws_avg daily_downwind_ref capacity dist_wrp mon_utm_x mon_utm_y monthly_oil_
## problemType : Regression
## tuneValue :
##     nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 78     500         5 0.1 0.001              0.5                0      0.75
## obsLevels : NA
## param :
##  $importance
## [1] TRUE
##
## $verbosity
## [1] 0
##
## $verbose
## [1] FALSE
```

```
imp<-varImp(fit.xgb_da_both_trans,scale=FALSE)

# rename variables
imp <- tibble(variable = rownames(imp$importance), importance = imp$importance$Overall) %>%
    pivot_wider(names_from = variable,
```

```
                values_from = importance) %>%
    rename(any_of(names)) %>%
    pivot_longer(cols = everything(),names_to = 'variable', values_to = 'importance')

imp %>%
  top_n(15, importance) %>%
  ggplot(aes(x=reorder(variable, importance), y=importance)) +
  geom_point() +
  geom_segment(aes(x=variable,xend=variable,y=0,yend=importance)) +
  ylab("importance") +
  xlab("Variable") +
  coord_flip() +
  theme_minimal()
```



Fold Performance

```
# Here, we compute the R2 and RMSE for each fold and take the average
fold_stat <- fit.xgb_da_both_trans$pred %>% group_by(Resample) %>%
  summarise(R2 = R2(pred, obs), RMSE = RMSE(pred, obs))
test_r2 <- mean(fold_stat$R2)
test_rmse <- mean(fold_stat$RMSE)
```

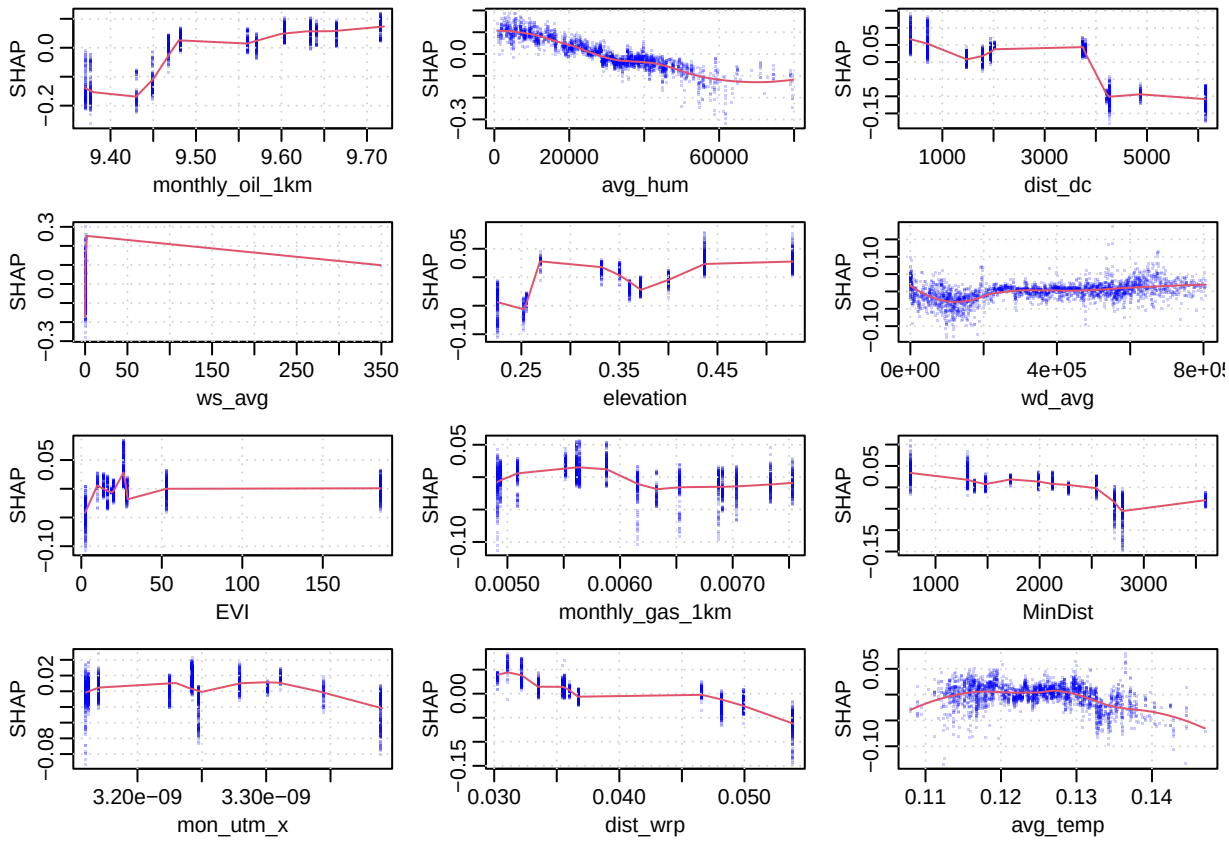
```
both_trans_xgb_sincefeb2022_obs_vs_pred_plot <- ggplot(tibble(obs = fit.xgb_da_both_trans$pred$obs,
                                                              pred = fit.xgb_da_both_trans$pred$pred),
                                                       aes(x = pred, y = obs)) +
                                                  geom_point() +
```

```
                        labs(y = 'Observed', x = 'Predicted',
                             title = 'Observed vs Predicted for Since 2022 XGBoost Resp Transform') +
                        stat_poly_line() +
                        stat_poly_eq(use_label(c("eq", "R2", "n"))) +
                        theme_bw()
```

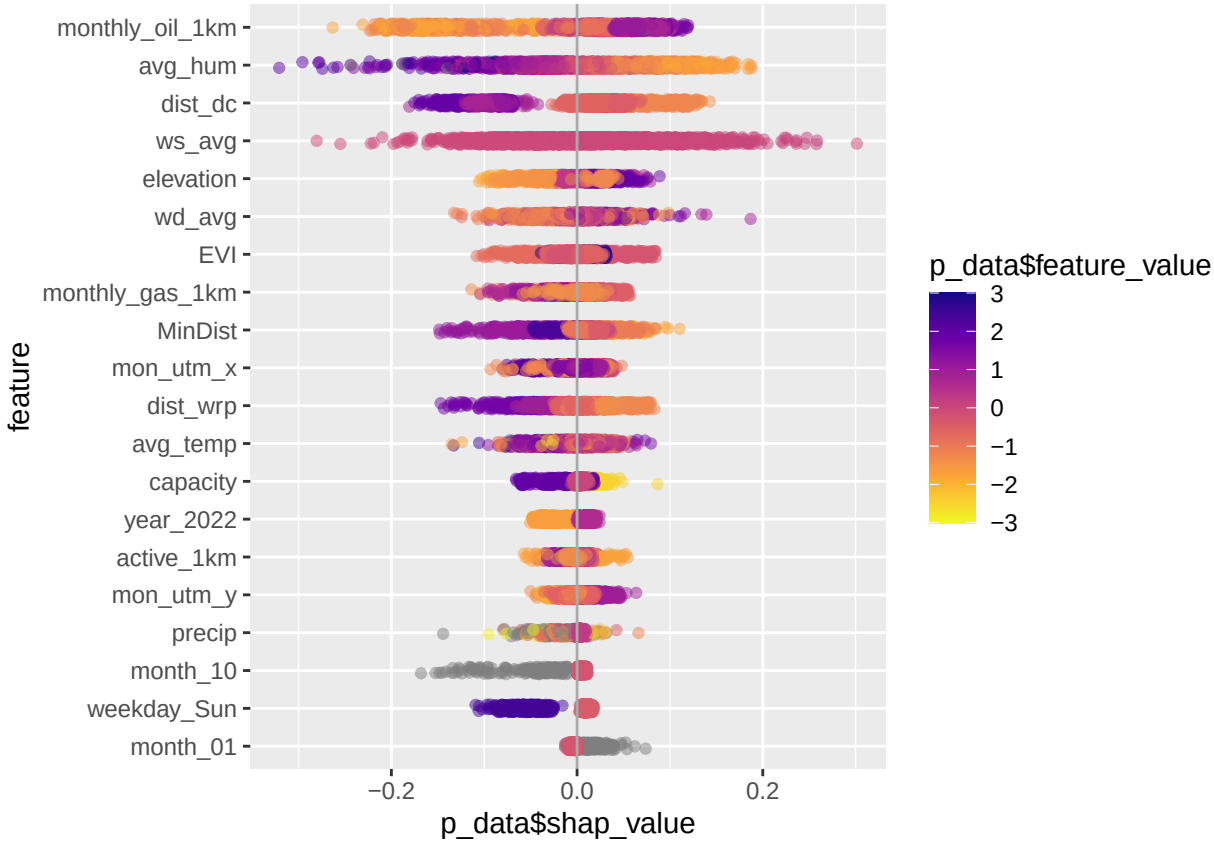
```
validation_result <- rbind(validation_result,
                           tibble(Model = 'Since Feb 2022 Both Transform',
                                  'Coef' = summary(lm(fit.xgb_da_both_trans$pred$obs ~
                                                      fit.xgb_da_both_trans$pred$pred))$coefficients[1
                                  'R-Sq' = summary(lm(fit.xgb_da_both_trans$pred$obs ~
                                                      fit.xgb_da_both_trans$pred$pred))$r.squared))
```

SHAP

```
matrix <- train %>% select(-H2S_daily_avg)
matrix <- as.matrix(sapply(matrix, as.numeric))
xgb.plot.shap(data = matrix,
              model = fit.xgb_da_both_trans$finalModel, top_n = 12, n_col = 3)
```




```
xgb.ggplot.shap.summary(data = matrix,
              model = fit.xgb_da_both_trans$finalModel, top_n = 20)
```




```
train_adj_r2 <- adj_r2(getTrainPerf(fit.xgb_da_both_trans)$TrainRsquared,
                       nrow(fit.xgb_da_both_trans$trainingData),
                       fit.xgb_da_both_trans$finalModel$nfeatures)
test_adj_r2 <- adj_r2(test_r2,
                       nrow(fit.xgb_da_both_trans$trainingData),
                       fit.xgb_da_both_trans$finalModel$nfeatures)

xgb_result <- rbind(xgb_result,
                    tibble(Model = 'Since Feb 2022 Both Transform',
                           '10CV Train R-Sq' = train_adj_r2,
                           '10CV Test R-Sq' = test_adj_r2,
                           '10CV Test RMSE' = test_rmse))
```

Result Comparison

Comparison

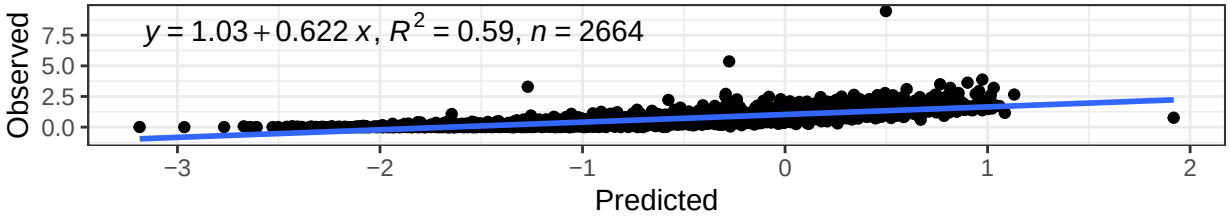
```
knitr::kable(tibble(Model = c('Since Feb 2022',
                              'Since Feb 2022 Log Transform',
                              'Since Feb 2022 Both Transform',
                              'Since Feb 2022 Resp Transform'),
                    'Train R-sq' = c(0.67,
                                     round(summary(h2s_da_model_f_logresp)$r.sq, 2),
                                     round(summary(h2s_da_model_f_trans)$r.sq, 2),
```

```
                                    round(summary(h2s_da_model_f_trans_resp)$r.sq, 2))) %>%
              left_join(rbind(result_10cv, tibble(Model = 'Since Feb 2022',
                                                  '10CV AVG Train R-Sq' = 0.664,
                                                  '10CV AVG Test R-Sq' = 0.665,
                                                  'Test RMSE' = 0.357)), join_by(Model)) %>%
              left_join(rbind(xgb_result, tibble(Model = 'Since Feb 2022',
                                                  '10CV Train R-Sq' = 0.777,
                                                  '10CV Test R-Sq' = 0.777,
                                                  '10CV Test RMSE' = 0.283)), join_by(Model)),
            digits = 3)
```

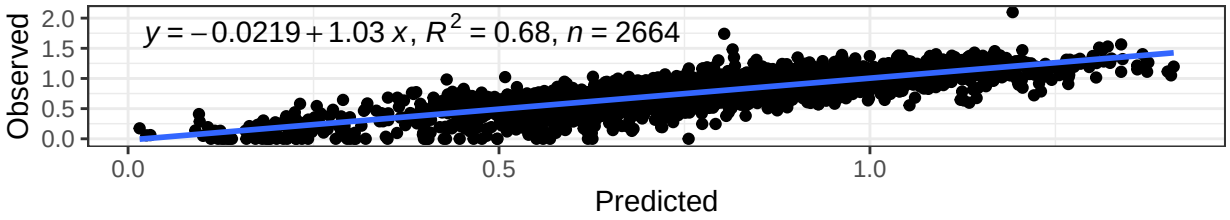
Model	Train R-sq	10CV AVG Train R-Sq	10CV AVG Test R-Sq	Test RMSE	10CV Train R-Sq	10CV Test R-Sq	10CV Test RMSE
Since Feb 2022	0.67	0.664	0.665	0.357	0.777	0.777	0.283
Since Feb 2022 Log Transform	0.78	0.780	0.590	1.295	0.837	0.837	0.336
Since Feb 2022 Both Transform	0.68	0.676	0.669	0.165	0.832	0.832	0.120
Since Feb 2022 Resp Transform	0.77	0.774	0.774	0.146	0.841	0.841	0.124

```
ggarrange(log_h2s_10cv_obs_vs_pred_plot,
          both_trans_10cv_obs_vs_pred_plot,
          resp_trans_10cv_obs_vs_pred_plot,
          labels = c("1", "2", "3"),
          nrow = 3)
```

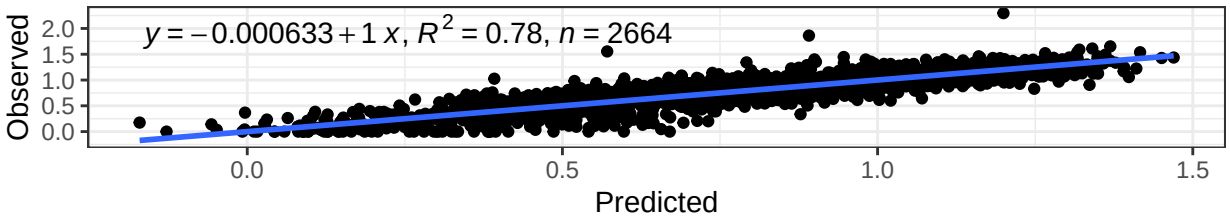
1 Observed vs Predicted for Since 2022 GAM 10CV Log H2S



2 Observed vs Predicted for Since 2022 GAM 10CV both transform

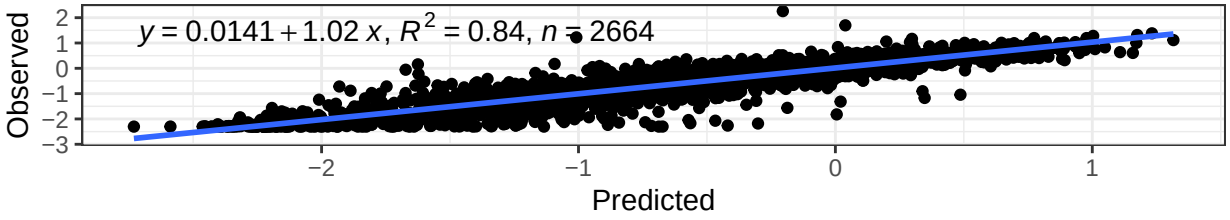


3 Observed vs Predicted for Since 2022 GAM 10CV resp H2S

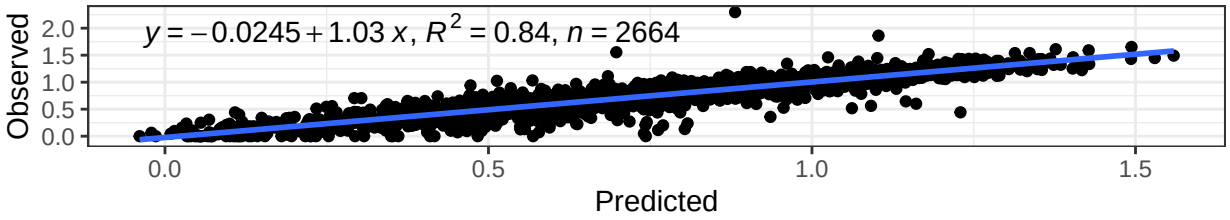



```
ggarrange(log_h2s_xgb_sincefeb2022_obs_vs_pred_plot,
          resp_trans_xgb_sincefeb2022_obs_vs_pred_plot,
          both_trans_xgb_sincefeb2022_obs_vs_pred_plot,
          labels = c("1", "2", "3"),
          nrow = 3)
```

1 Observed vs Predicted for Since 2022 XGBoost Log H2S



2 Observed vs Predicted for Since 2022 XGBoost Resp Transform



3 Observed vs Predicted for Since 2022 XGBoost Resp Transform