Modified ALCC

Adrian A. Correndo

03/24/2022

This code was prepared as a tutorial for potential users of the Modified Arcsine-log Calibration Curve (modALCC) detailed in Correndo et al. (2017).

Instructions for users

1. Load your dataframe with soil test value (STV) and relative yield (RY) data.
2. Specify the following arguments into the function -modALCC()-:

(i). 'data' (optional),

(ii). soil test value 'STV' and relative yield 'RY',

(iii). 'target' of relative yield (e.g. 90%),

(iv). desired confidence level (e.g. 0.95 for 1 - alpha(0.05)). Used for the estimation of critical soil test value (CSTV) lower and upper limits.

3. Run and check results in a data.frame.
4. Check residuals plot, and warnings related to potential leverage points.
5. Adjust curve plots as desired.

Please, refer any question to Adrian Correndo, correndo@agro.uba.ar - correndo@ksu.edu.

Note: RY should be expressed relative to a maximum in order to obtain values bounded at %100. Otherwise, arcsine transformation doesn't work. If RY values $> 100\%$ are found, the function will cap them up to 100% and will display a warning about this.

References

Correndo, A.A., Salvagiotti, F., García, F.O. and Gutiérrez-Boem, F.H., 2017. A modification of the arcsine–log calibration curve for analysing soil test value–relative yield relationships. Crop and Pasture Science, 68(3), pp.297-304. https://doi.org/10.1071/CP16444

Last update: 03-24-2022

1. Libraries

```
# Install if needed
# install.packages("easypackages")
# install.packages("devtools")
library(easypackages) # Helps to load packages and install & load them if they are not installed yet.
library(devtools)
packages("readxl") # Open xlsx files
packages("tidyverse", "ggpmisc") # Data wrangling and plots
packages("smatr") # SMA regression analysis for reference
packages("agridat") # For cotton example dataset
```

2. Datasets

```
# Example 1 dataset
data_1 = data.frame("RY" = c(65,80,85,88,90,94,93,96,97,95,98,100,99,99,100),
                    "STV" = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))


# Example 2 dataframe. Imported from csv file
# It can be easily replaced with your own csv file
data_2 = read.csv(file = "data_test.csv")

# Example 3 dataframe. Imported from xlsx file
data_3 = readxl::read_xlsx(path ="data_3.xlsx", sheet = 1)

# Create nested structure as example of multiple datasets
data.all = bind_rows(data_1, data_2, data_3, .id = "id") %>%
  tidyr::nest(data = c("STV", "RY"))
```

3. modALCC() function

```
modALCC <- function(data=NULL, RY, STV, target, confidence){

  # Add a function to cap if there are RY values > 100
  ry <- rlang::eval_tidy(data = data, rlang::quo(ifelse({{RY}} > 100, 100, as.double({{RY}})) ))
  n <- length(ry) # Sample size
  df <- n - 2 # Degrees of freedom
  prob <- 1-((1-confidence)/2) # Probability for t-dist
  tvalue <- qt(p=prob, df = df) # Student-t value
  arc_RY <- asin(sqrt(ry/100)) - asin(sqrt(target/100)) # RY transformation (centered to target)
  ln_STV <- rlang::eval_tidy(data = data, rlang::quo(log({{STV}}) ) ) # STV natural log transformation
  r <- cor(ln_STV, arc_RY, method = "pearson") # Pearson correlation (r)
  p_value <- cor.test(ln_STV,arc_RY, method = "pearson")$p.value # p-value of r
  slope <- sd(ln_STV)/sd(arc_RY) # SMA slope for ln_STV ~ arc_RY
  intercept <- mean(ln_STV) - (mean(arc_RY)*slope) # Intercept
  SMA_line <- intercept + slope * arc_RY # Fitted ln_STV for observed RY
  CSTV <- exp(intercept) # Critical STV for specified RY-target and confidence (1-alpha)
  MSE <- sum((SMA_line-ln_STV)^2)/df # Mean Square Error of ln_STV
  SSx <- sum((mean(arc_RY)-arc_RY)^2)  # Sum of Squares of arc_RY
  SE_int <- sqrt(MSE*((1/n)+ ((mean(arc_RY)^2)/SSx)))  # Standard Error intercept
  CSTV_lower <- exp(intercept - (tvalue * SE_int))  # Lower limit of CSTV
  CSTV_upper <- exp(intercept + (tvalue * SE_int)) # Upper limit of CSTV
  new_RY <- seq(min(ry),100, by=0.2) # New RY vector up to %100 to fit curve
  new_arc_RY <- asin(sqrt(new_RY/100)) - asin(sqrt(target/100)) # Transforming new_RY vector
  fitted_Line <- intercept + slope * new_arc_RY # Fitted ln_STV for curve plot
  fitted_STV <- exp(fitted_Line) # Fitted ln_STV for new_RY
  residuals <- ln_STV - SMA_line # Residuals of SMA Regression
  fitted_axis <- ln_STV + slope * arc_RY # Fitted axis to check SMA residuals
  target <- target # Target RY to show on summary
  confidence <- confidence # Confidence level to show on summary
  # Critical STV for RY = 90 & 100
  arc_ry_100 <- asin(sqrt(ry/100)) - asin(sqrt(1))
  cstv.100 <- exp(mean(ln_STV) - (mean(arc_ry_100)*(sd(ln_STV)/sd(arc_ry_100))))
  arc_ry_90 <- asin(sqrt(ry/100)) - asin(sqrt(90/100))
  cstv.90 <- exp(mean(ln_STV) - (mean(arc_ry_90)*(sd(ln_STV)/sd(arc_ry_90))))
  # Count cases with STV > x2 cstv90 and STV > cstv100
  n.90x2 <- rlang::eval_tidy(data=data, rlang::quo(length(which({{STV}} > (2*cstv.90))) ) )
  n.100 <- rlang::eval_tidy(data=data, rlang::quo(length(which({{STV}} > cstv.100)) ) )
  # Outcome
  results <- as.data.frame(list("n" = n, "r" = r, "target" = target,"CSTV" = CSTV,
    "LL" = CSTV_lower,"UL" = CSTV_upper,"confidence" = confidence,"p_value" = p_value,
    "CSTV90" = cstv.90, "n.90x2" = n.90x2,"CSTV100" = cstv.100,"n.100" = n.100)) %>%
    bind_cols(., as.data.frame(list("RY.fitted" = new_RY, "STV.fitted" = fitted_STV))%>% tidyr::nest(Cu
    bind_cols(.,  as.data.frame(list("ln_STV" = ln_STV, "arc_RY" = arc_RY, "SMA_line" = SMA_line,"residu

  # WARNINGS
  rlang::eval_tidy(data = data, rlang::quo(if (max({{RY}}) > 100) {
  warning("One or more original RY values exceeded 100%. All RY values greater
          than 100% have been capped to 100%.", call. = FALSE) } ) )

  if (results$n.100 > 0) {warning(paste0(n.100," STV points exceeded the CSTV for 100%.
  Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
```

```
re-run the modALCC(), and check results."), call. = FALSE) }

if (results$n.90x2 > 0) {warning(paste0(n.90x2," STV points exceeded two-times (2x)
the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
removing extreme points, re-run the modALCC(), and check results."), call. = FALSE) }

return(results) }
```

4. Fit examples

4.1. Fit ALCC models individually

```
# RY target = 90%, confidence level = 0.95, replace with your desired values

# Data 1
# Using dataframe
fit_example_1 = modALCC(data = data_1, RY = RY, STV = STV, target=90, confidence = 0.95)
```

```
## Warning: 7 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```
# Alternative using the vectors
#fit_example_1 = ALCC(RY = data_1$RY,STV = data_1$STV, target=90,confidence = 0.95)

fit_example_1
```

```
##    n         r target     CSTV       LL       UL confidence      p_value
## 1 15 0.9682908     90 4.478476 3.947041 5.081463       0.95 3.296044e-09
##     CSTV90 n.90x2  CSTV100 n.100
## 1 4.478476      7 19.15054     0
##
## 1 65.000000, 65.200000, 65.400000, 65.600000, 65.800000, 66.000000, 66.200000, 66.400000, 66.600000,
##
## 1 0.00000000, 0.69314718, 1.09861229, 1.38629436, 1.60943791, 1.79175947, 1.94591015, 2.07944154, 2.
```

```
# Data 2
fit_example_2 = modALCC(data = data_2, RY = RY, STV = STV, target=90, confidence = 0.95)
```

```
## Warning: 9 STV points exceeded the CSTV for 100%.
##   Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
##   re-run the modALCC(), and check results.

## Warning: 22 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```
fit_example_2
```

```
##     n         r target     CSTV       LL       UL confidence      p_value
## 1 137 0.7164928     90 23.25457 21.57156 25.06888       0.95 7.314913e-23
##     CSTV90 n.90x2  CSTV100 n.100
## 1 23.25457     22 53.10299     9
##
## 1 12.000000, 12.200000, 12.400000, 12.600000, 12.800000, 13.000000, 13.200000, 13.400000, 13.600000,
##
## 1 1.386294361, 1.609437912, 1.791759469, 1.791759469, 1.945910149, 1.945910149, 2.197224577, 2.19722
```

```r
# Data 3
fit_example_3 = modALCC(data = data_3, RY = RY, STV = STV, target=90, confidence = 0.95)
```

```
## Warning: 3 STV points exceeded the CSTV for 100%.
##   Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
##   re-run the modALCC(), and check results.

## Warning: 3 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```r
fit_example_3
```

```
##     n         r target     CSTV       LL       UL confidence      p_value
## 1 107 0.3735166     90 21.76416 19.35018 24.47929       0.95 7.410056e-05
##     CSTV90 n.90x2 CSTV100 n.100
## 1 21.76416      3 38.7896     3
##
## 1 25.000000, 25.200000, 25.400000, 25.600000, 25.800000, 26.000000, 26.200000, 26.400000, 26.600000,
##
## 1 1.6094379124, 1.7917594692, 1.7917594692, 1.9459101491, 1.9459101491, 2.0794415417, 2.0794415417, 1
```

4.2. Fit multiple ALCC models with mapping

```r
# Run multiple examples at once with map()
fit_examples = data.all %>%
  mutate(modALCC = map(data, ~ modALCC(RY = .$RY, STV = .$STV, target=90, confidence = 0.95))) %>%
  unnest(., cols = c("modALCC"))
```

```
## Warning: 7 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.

## Warning: 9 STV points exceeded the CSTV for 100%.
##   Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
##   re-run the modALCC(), and check results.

## Warning: 22 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.

## Warning: 3 STV points exceeded the CSTV for 100%.
##   Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
##   re-run the modALCC(), and check results.

## Warning: 3 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```
head(fit_examples)
```

```
## # A tibble: 3 x 16
##   id    data         n     r target  CSTV    LL    UL confidence  p_value CSTV90
##   <chr> <list>   <int> <dbl>  <dbl> <dbl> <dbl> <dbl>      <dbl>    <dbl>  <dbl>
## 1 1     <tibble>    15 0.968     90  4.48  3.95  5.08       0.95 3.30e- 9   4.48
## 2 2     <tibble>   137 0.716     90 23.3  21.6  25.1        0.95 7.31e-23  23.3
## 3 3     <tibble>   107 0.374     90 21.8  19.4  24.5        0.95 7.41e- 5  21.8
## # ... with 5 more variables: n.90x2 <int>, CSTV100 <dbl>, n.100 <int>,
## #   Curve <list>, SMA <list>
```

```
# Alternative with group_map, this does not required nested data.
fit_all = bind_rows(data_1, data_2, data_3, .id = "id") %>%
  group_by(id) %>%
  group_map(~ modALCC(data = ., RY = RY, STV = STV, target = 90, confidence = 0.95))
```

```
## Warning: 7 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```
## Warning: 9 STV points exceeded the CSTV for 100%.
##   Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
##   re-run the modALCC(), and check results.
```

```
## Warning: 22 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```
## Warning: 3 STV points exceeded the CSTV for 100%.
##   Risk of leverage. You may consider a sensitivity analysis by removing extreme points,
##   re-run the modALCC(), and check results.
```

```
## Warning: 3 STV points exceeded two-times (2x)
##   the CSTV for 90%. Risk of leverage. You may consider a sensitivity analysis by
##   removing extreme points, re-run the modALCC(), and check results.
```

```
head(fit_all)
```

```
## [[1]]
##    n         r target     CSTV       LL       UL confidence      p_value
## 1 15 0.9682908     90 4.478476 3.947041 5.081463       0.95 3.296044e-09
##     CSTV90 n.90x2  CSTV100 n.100
## 1 4.478476      7 19.15054     0
##
## 1 65.000000, 65.200000, 65.400000, 65.600000, 65.800000, 66.000000, 66.200000, 66.400000, 66.600000,
##
## 1 0.00000000, 0.69314718, 1.09861229, 1.38629436, 1.60943791, 1.79175947, 1.94591015, 2.07944154, 2.1
##
## [[2]]
##    n         r target     CSTV       LL       UL confidence      p_value
```

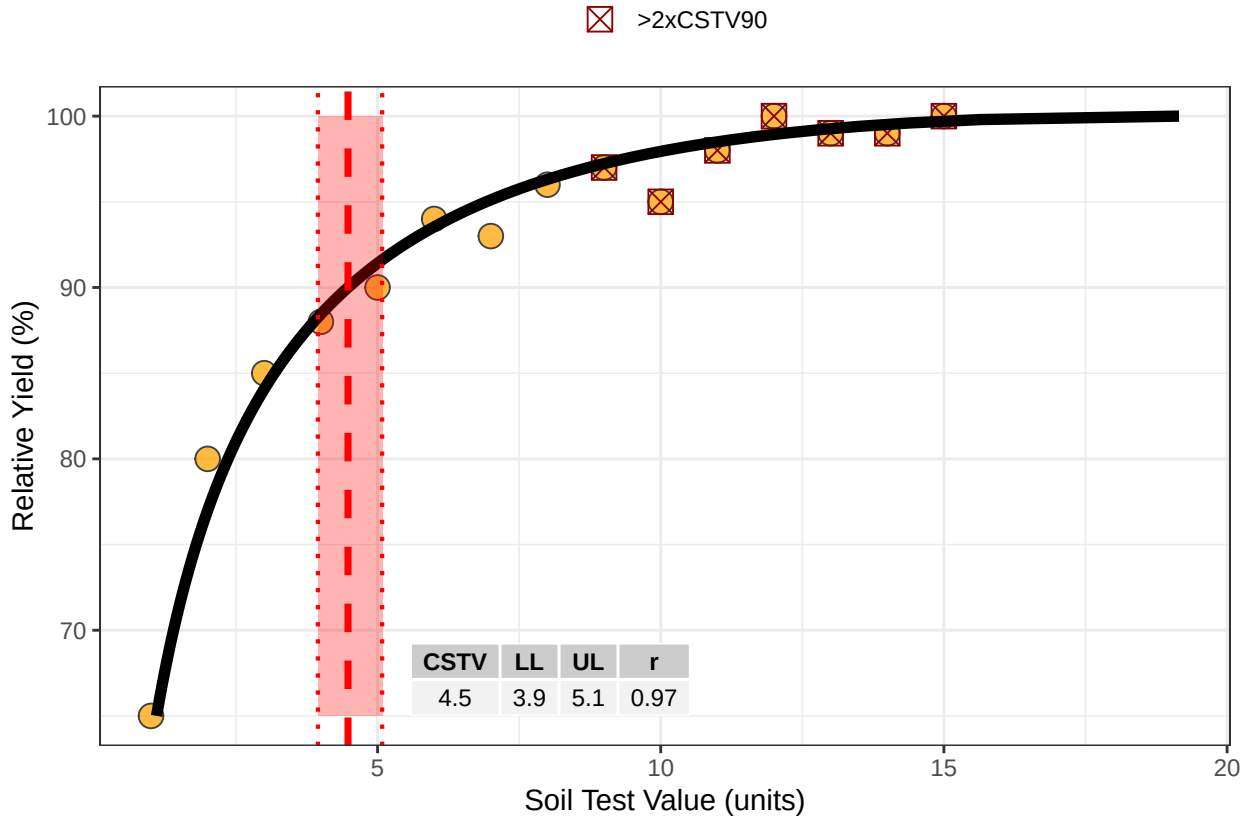
```
## 1 137 0.7164928     90 23.25457 21.57156 25.06888       0.95 7.314913e-23
##     CSTV90 n.90x2  CSTV100 n.100
## 1 23.25457     22 53.10299     9
##
## 1 12.000000, 12.200000, 12.400000, 12.600000, 12.800000, 13.000000, 13.200000, 13.400000, 13.600000,
##
## 1 1.386294361, 1.609437912, 1.791759469, 1.791759469, 1.945910149, 1.945910149, 2.197224577, 2.19722
##
## [[3]]
##     n         r target     CSTV       LL       UL confidence      p_value
## 1 107 0.3735166     90 21.76416 19.35018 24.47929       0.95 7.410056e-05
##     CSTV90 n.90x2 CSTV100 n.100
## 1 21.76416      3 38.7896     3
##
## 1 25.000000, 25.200000, 25.400000, 25.600000, 25.800000, 26.000000, 26.200000, 26.400000, 26.600000,
##
## 1 1.6094379124, 1.7917594692, 1.7917594692, 1.9459101491, 1.9459101491, 2.0794415417, 2.0794415417, 1
```

5. Plots

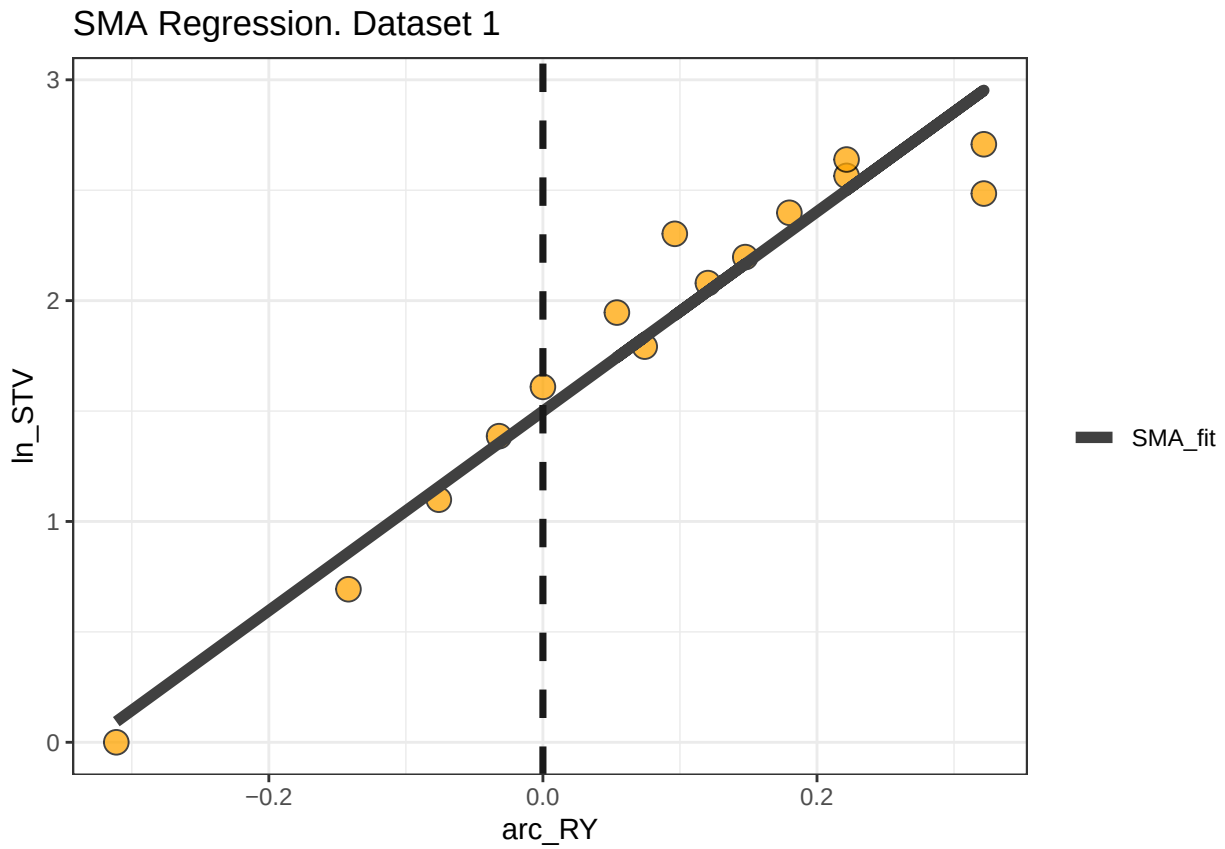
Examples using ggplot

5.1. Example 1

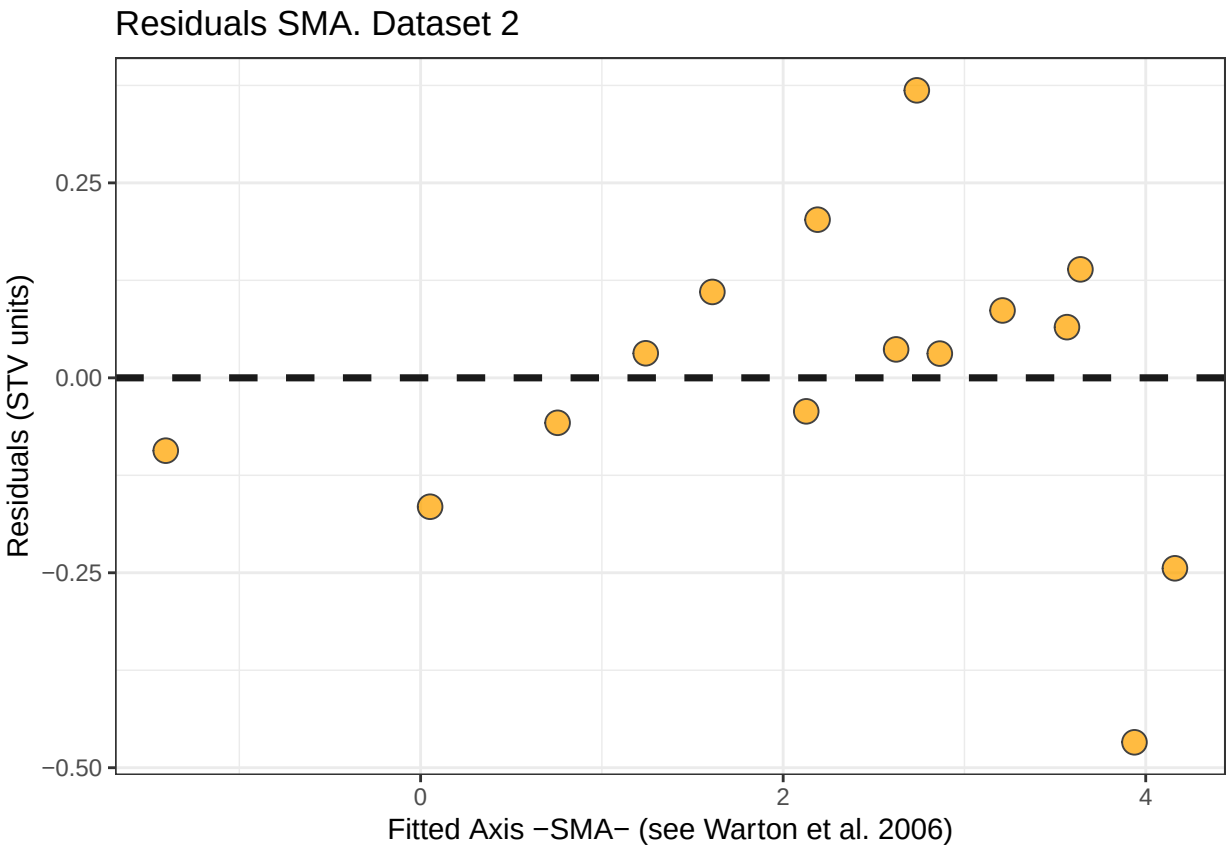
```
# Extracting curve data as a data.frame to plot
curve_example1 = fit_example_1 %>% unnest(., cols = Curve)

# Plot
data_1 %>%
  # Want to remove leverage points?
  #dplyr::filter(STV < fit_example_1$CSTV100) %>%
  #dplyr::filter(STV < 2*fit_example_1$CSTV90) %>%
  ggplot()+
  # Points
  geom_point(aes(x = STV, y = RY), fill = "orange", shape = 21, size = 4, alpha = 0.75)+
  # Highlight potential leverage points >2xCSTV90
  geom_point(data = data_1 %>% dplyr::filter(STV > 2*fit_example_1$CSTV90),
             aes(x = STV, y = RY, shape = ">2xCSTV90"), col = "dark red", size = 4, alpha = 1)+
  # Highlight potential leverage points >2xCSTV90
  #geom_point(data = data_1 %>% dplyr::filter(STV > fit_example_1$CSTV100),
   #           aes(x = STV, y = RY, shape = ">CSTV100"), col = "dark red", size = 4, alpha = 1)+
  scale_shape_manual(name = "", values = c(7,10))+
  # Fitted ALCC
  geom_line(data = curve_example1, aes(x= STV.fitted, y = RY.fitted), size = 2)+
  # Critical value
  geom_vline(xintercept = fit_example_1$CSTV, col = "red", size = 1.25, linetype = "dashed")+
  # Confidence limits
  # Lines
  geom_vline(xintercept = fit_example_1$LL, col = "red", size = 0.75, linetype = "dotted")+
  geom_vline(xintercept = fit_example_1$UL, col = "red", size = 0.75, linetype = "dotted")+
  # Shade
  ggpp::annotate(geom = "rect", xmin = fit_example_1$LL, xmax = fit_example_1$UL,
                 ymin = min(data_1$RY), ymax = 100, alpha = .3, fill = "red")+
  # Axis titles
  labs(x = "Soil Test Value (units)", y = "Relative Yield (%)")+
  theme_bw()+
  theme(legend.position = "top")+
  # Annotate critical values data
  ggpp::annotate(geom = "table", y = min(data_1$RY), x = fit_example_1$UL + 0.5, hjust= 0, vjust = 0,
                 label = fit_example_1 %>% dplyr::select(CSTV, LL, UL, r) %>%
                   mutate_at(.vars = c("r"), ~round(.,2)) %>%
                   mutate_at(.vars = c("CSTV","LL","UL"), ~round(.,1))
                   )
```




```
# SMA regression

SMA_example1 = fit_example_1 %>%  unnest(., cols = SMA)

SMA_example1 %>%
  ggplot(aes(x = arc_RY, y = ln_STV))+
  ggtitle("SMA Regression. Dataset 1")+
  geom_point(shape=21, fill = "orange", size = 4, alpha = 0.75)+
  #SMA Line
  geom_path(aes(x=arc_RY, y = SMA_line, linetype = "SMA_fit"), size = 2, col = "grey25")+
  scale_linetype_manual(name="", values = c("solid"))+
  #Critical value
  geom_vline(xintercept = 0, col = "grey10", size = 1.25, linetype = "dashed")+
  theme_bw()+
  # Axis titles
  labs(y = "ln_STV", y = "asin(sqrt(RY))-centered")
```

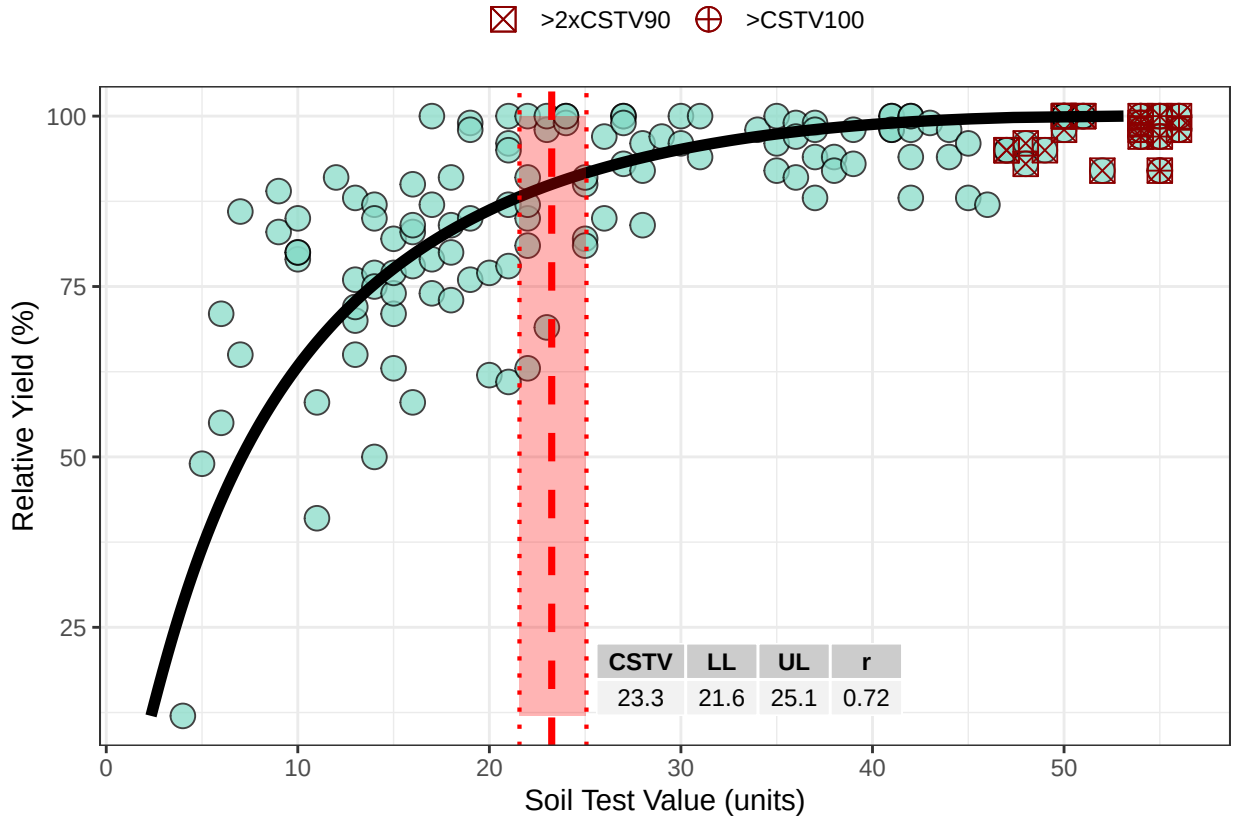



```
# Residuals plot

SMA_example1 %>%
  ggplot(aes(x = fitted_axis, y = residuals))+
  ggtitle("Residuals SMA. Dataset 2")+
  geom_point(shape=21, fill = "orange", size = 4, alpha = 0.75)+
  geom_hline(yintercept = 0, col = "grey10", size = 1.25, linetype = "dashed")+
  theme_bw()+
  # Axis titles
  labs(x = "Fitted Axis -SMA- (see Warton et al. 2006)", y = "Residuals (STV units)")
```

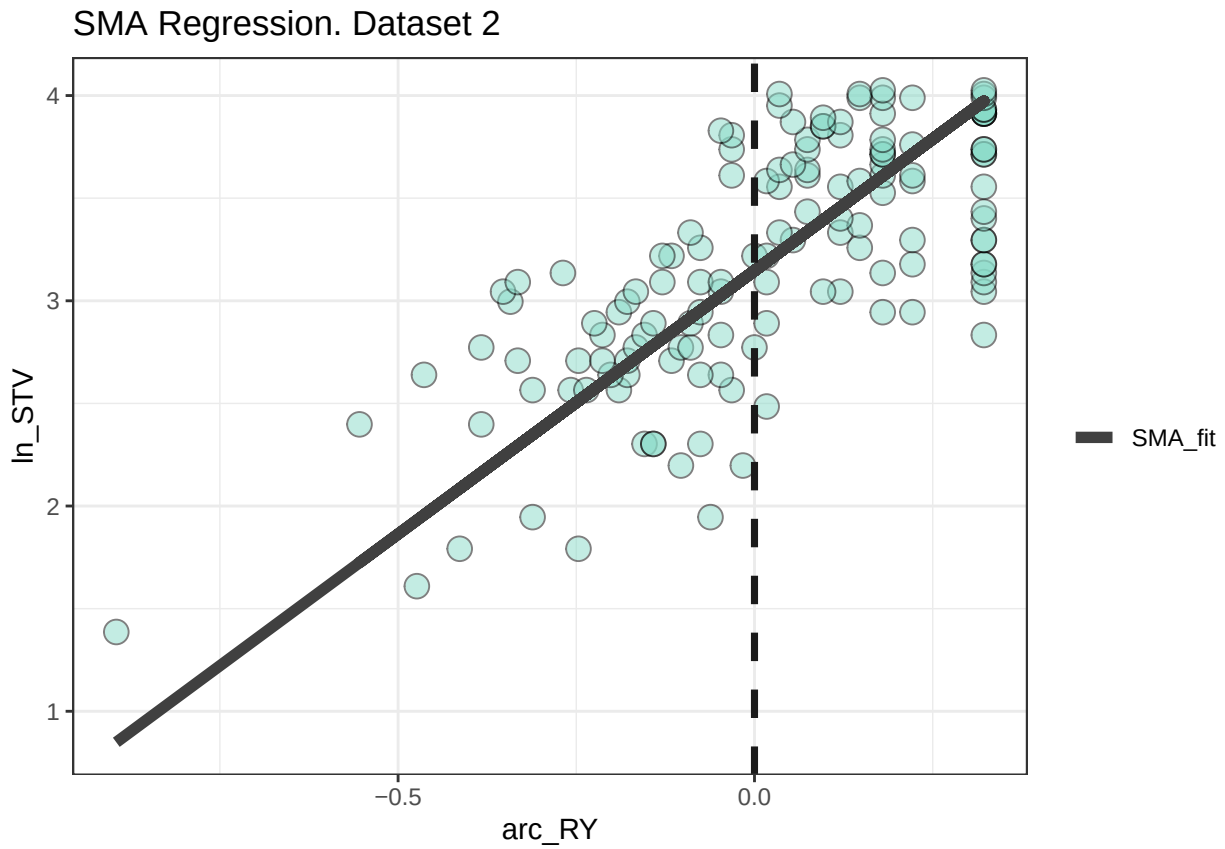

Residuals SMA. Dataset 2
Residuals (STV units)
0.25
0.00
−0.25
−0.50
0
2
4
Fitted Axis −SMA− (see Warton et al. 2006)

5.2. Example 2

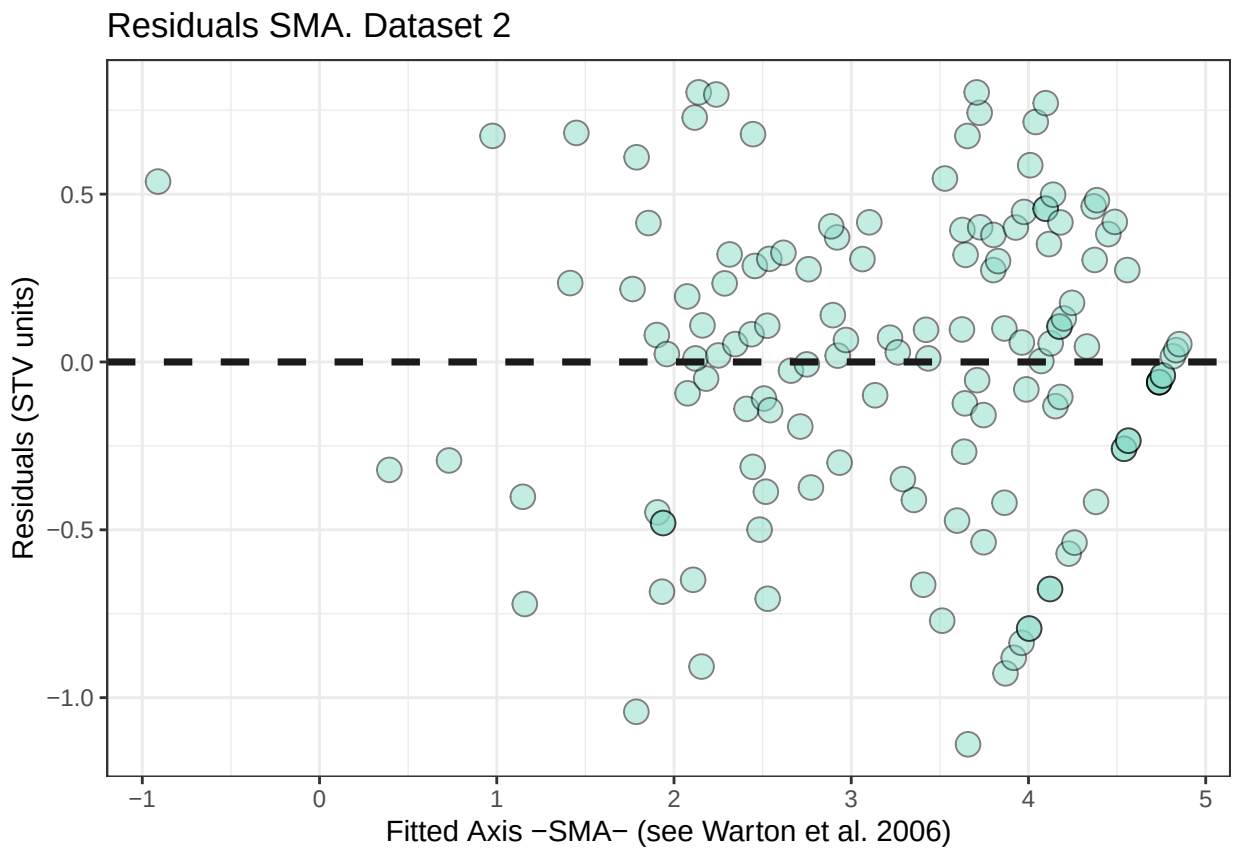
```
# Extracting curve data as a data.frame to plot
curve_example2 = fit_example_2 %>% unnest(., cols = Curve)

# Plot
data_2 %>% ggplot()+
  # Want to remove leverage points?
  #dplyr::filter(STV < fit_example_2$CSTV100) %>%
  #dplyr::filter(STV < 2*fit_example_2$CSTV90) %>%
  # Points
  geom_point(aes(x = STV, y = RY), fill = "#88dbc8", shape = 21, size = 4, alpha = 0.75)+
  # Highlight potential leverage points >2xCSTV90
  geom_point(data = data_2 %>% dplyr::filter(STV > 2*fit_example_2$CSTV90),
             aes(x = STV, y = RY, shape = ">2xCSTV90"), col = "dark red", size = 4, alpha = 1)+
  # Highlight potential leverage points >CSTV100
  geom_point(data = data_2 %>% dplyr::filter(STV > fit_example_2$CSTV100),
             aes(x = STV, y = RY, shape = ">CSTV100"), col = "dark red", size = 4, alpha = 1)+
  scale_shape_manual(name = "", values = c(7,10))+
  # Fitted ALCC
  geom_line(data = curve_example2, aes(x= STV.fitted, y = RY.fitted), size = 2)+
  # Critical value
  geom_vline(xintercept = fit_example_2$CSTV, col = "red", size = 1.25, linetype = "dashed")+
  # Confidence limits
  geom_vline(xintercept = fit_example_2$LL, col = "red", size = 0.75, linetype = "dotted")+
  geom_vline(xintercept = fit_example_2$UL, col = "red", size = 0.75, linetype = "dotted")+
  ggpp::annotate(geom = "rect", xmin = fit_example_2$LL, xmax = fit_example_2$UL,
                 ymin = min(data_2$RY), ymax = 100, alpha = .3, fill = "red")+
  # Axis titles
  labs(x = "Soil Test Value (units)", y = "Relative Yield (%)")+
  theme_bw()+
  theme(legend.position = "top")+
  # Annotate critical values data
  ggpp::annotate(geom = "table", y = min(data_2$RY), x = fit_example_2$UL + 0.5, hjust= 0, vjust = 0,
                 label = fit_example_2 %>% dplyr::select(CSTV, LL, UL, r) %>%
                   mutate_at(.vars = c("r"), ~round(.,2)) %>%
                   mutate_at(.vars = c("CSTV","LL","UL"), ~round(.,1))
                   )
```




```
# SMA regression

SMA_example2 = fit_example_2 %>%  unnest(., cols = SMA)

SMA_example2 %>%
  ggplot(aes(x = arc_RY, y = ln_STV))+
  ggtitle("SMA Regression. Dataset 2")+
  geom_point(shape=21, fill = "#88dbc8", size = 4, alpha = 0.5)+
  #SMA Line
  geom_path(aes(x=arc_RY, y = SMA_line, linetype = "SMA_fit"), size = 2, col = "grey25")+
  scale_linetype_manual(name="", values = c("solid"))+
  #Critical value
  geom_vline(xintercept = 0, col = "grey10", size = 1.25, linetype = "dashed")+
  theme_bw()+
  # Axis titles
  labs(y = "ln_STV", y = "asin(sqrt(RY))-centered")
```




```
# Residuals plot

resid_example2 = fit_example_2 %>%  unnest(., cols = SMA)

resid_example2 %>%
  ggplot(aes(x = fitted_axis, y = residuals))+
  ggtitle("Residuals SMA. Dataset 2")+
  geom_point(shape = 21, fill = "#88dbc8", size = 4, alpha = 0.5)+
  geom_hline(yintercept = 0, col = "grey10", size = 1.25, linetype = "dashed")+
  theme_bw()+
  # Axis titles
  labs(x = "Fitted Axis -SMA- (see Warton et al. 2006)", y = "Residuals (STV units)")
```


Residuals SMA. Dataset 2
Residuals (STV units)
0.5
0.0
−0.5
−1.0
−1
0
1
2
3
4
5
Fitted Axis −SMA− (see Warton et al. 2006)