Modified ALCC

Adrian A. Correndo

03/20/2020

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

Instructions for users

1. Load your soil test value (STV) and relative yield (RY) data as vectors of a data.frame.
2. Specify into the function -ALCC()- which vectors correspond to RY and STV.
3. Specify your relative yield target (e.g. 90%) and desired confidence level (e.g. 0.95 for 1 - alpha(0.05)). Used for the estimation of critical soil test value (CSTV) and lower and upper confidence limits.
4. Run.
5. Check results and adjust plot as desired.
6. 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.

References

Correndo, A.A., F. Salvagiotti, F.O. García, y F.H. Gutiérrez-Boem. 2017. A modification of the arcsine-log calibration curve for analysing soil test value - relative yield relationships. Crop & Pasture Science 68 (3): 297-304, doi: 10.10171/CP16444

Last update: 03-20-2022

1. Libraries

```
# Install if needed
# install.packages("tidyverse")
# tidyverse contains "dplyr" and "tidyr" for data wrangling and "ggplot2" for creating plots
library(tidyverse)
```

2. Data

```r
# 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")

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

3. ALCC function, definition

```r
ALCC <- function(RY, STV, target, confidence){
  n = length(RY) # Sample size
  df = n - 2 # Degrees of freedom
  prob = 1-((1-confidence)/1/2) # Probability for t-dist
  tvalue = qt(p=prob, df = df) # Student-t value
  #arcsine = function(RY)asin(sqrt({{RY}}/100))
  arc_RY = asin(sqrt(RY/100)) - asin(sqrt(target/100)) # RY transformation (centered to target)
  ln_STV = log(STV) # STV natural log transformation
  r = cor(ln_STV, arc_RY) # Pearson correlation (r)
  p_value = cor.test(ln_STV,arc_RY)$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
  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((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 new_RY
  fitted_STV = exp(fitted_Line) # Fitted ln_STV for new_RY
  # Outcome
  results <- as.data.frame(list("r" = r,
                                "p_value" = p_value,
                                "CSTV" = CSTV,
                                "LL" = CSTV_lower,
                                "UL" = CSTV_upper,
                                "RY.fitted" = new_RY,
                                "STV.fitted" = fitted_STV))   %>%
    tidyr::nest(Curve = c("RY.fitted", "STV.fitted"))

  return(results)
}
```

4. Fit examples

```
# Fit the ALCC model
# RY target = 90%, confidence level = 0.95, replace with your desired values
fit_example_1 = ALCC(RY = data_1$RY,STV = data_1$STV, target=90,confidence = 0.95)
fit_example_2 = ALCC(RY = data_2$RY,STV = data_2$STV, target=90,confidence = 0.95)

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

head(fit_examples)
```

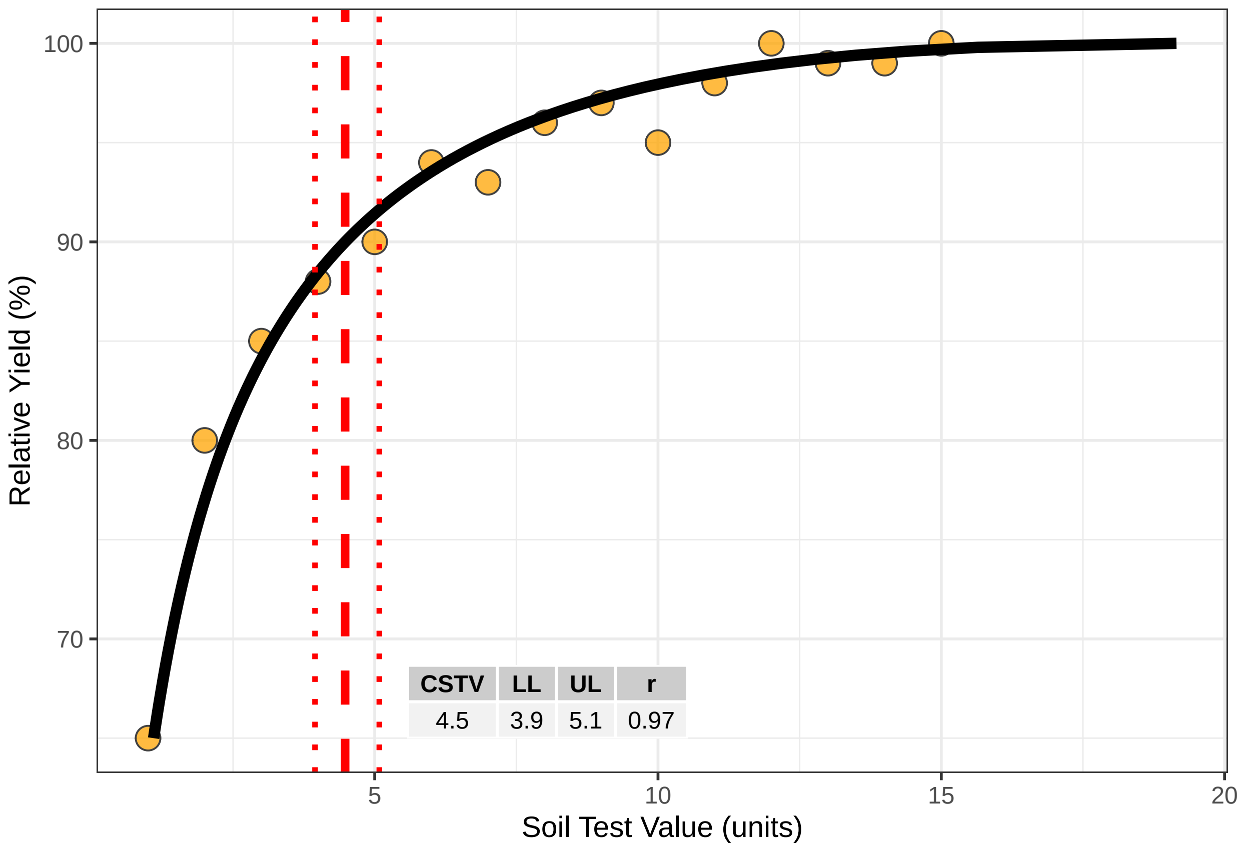
```
## # A tibble: 2 x 8
##   id    data                   r  p_value  CSTV    LL    UL Curve
##   <chr> <list>             <dbl>    <dbl> <dbl> <dbl> <dbl> <list>
## 1 1     <tibble [15 x 2]>  0.968 3.30e- 9  4.48  3.95  5.08 <tibble [176 x 2]>
## 2 2     <tibble [137 x 2]> 0.716 7.31e-23 23.3  21.6  25.1  <tibble [441 x 2]>
```

5. Plots

5.1. Example 1

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

# Plot
data_1 %>% ggplot()+
  # Points
  geom_point(aes(x = STV, y = RY), fill = "orange", shape = 21, size = 4, alpha = 0.75)+
  # Fitted ALCC
  geom_line(data = curve_example1, aes(x= STV.fitted, y = RY.fitted), size = 2)+
  # Critical value
  geom_vline(xintercept = fit_example_1$CSTV, col = "red", size = 1.5, linetype = "dashed")+
  # Confidence limits
  geom_vline(xintercept = fit_example_1$LL, col = "red", size = 1, linetype = "dotted")+
  geom_vline(xintercept = fit_example_1$UL, col = "red", size = 1, linetype = "dotted")+
  # Axis titles
  labs(x = "Soil Test Value (units)", y = "Relative Yield (%)")+
  theme_bw()+
  theme()+
  # Annotate critical values data
  ggpp::annotate(geom = "table", y = min(data_1$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))
                 )
```

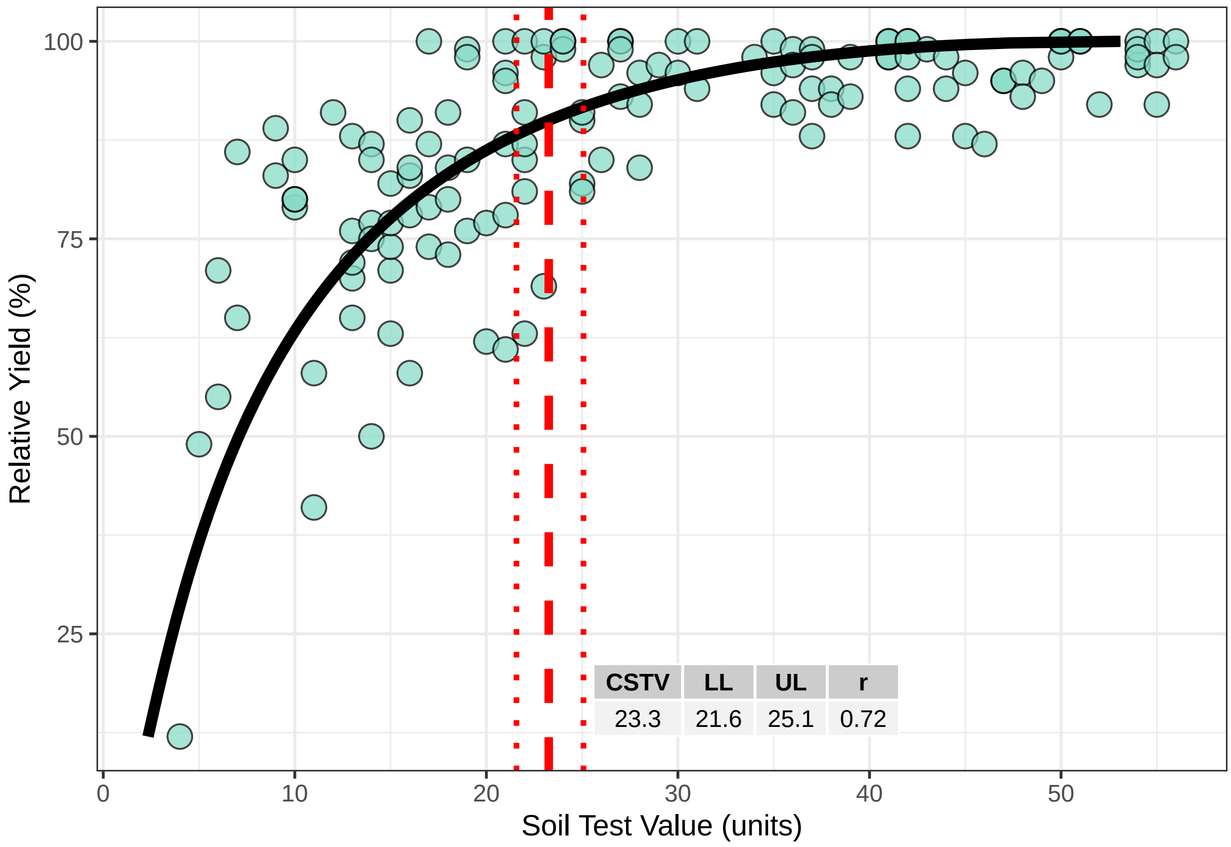


CSTV	LL	UL	r
4.5	3.9	5.1	0.97

5.1. Example 1

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

# Plot
data_2 %>% ggplot()+
  # Points
  geom_point(aes(x = STV, y = RY), fill = "#88dbc8", shape = 21, size = 4, alpha = 0.75)+
  # Fitted ALCC
  geom_line(data = curve_example2, aes(x= STV.fitted, y = RY.fitted), size = 2)+
  # Critical value
  geom_vline(xintercept = fit_example_2$CSTV, col = "red", size = 1.5, linetype = "dashed")+
  # Confidence limits
  geom_vline(xintercept = fit_example_2$LL, col = "red", size = 1, linetype = "dotted")+
  geom_vline(xintercept = fit_example_2$UL, col = "red", size = 1, linetype = "dotted")+
  # Axis titles
  labs(x = "Soil Test Value (units)", y = "Relative Yield (%)")+
  theme_bw()+
  theme()+
  # Annotate critical values data
  ggpp::annotate(geom = "table", y = min(data_2$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))
                   )
```



CSTV	LL	UL	r
23.3	21.6	25.1	0.72