Validation Analytics Guideline
2024

Contents

Introduction

We designed this document to assist the Use Case teams in understanding the validation data set and the related standard analytics including, data preparation techniques, and the visualization of validation data. We pay specific attention to the parameters measuring effectively Agronomic gain key performance indicators (KPIs). The KPIs of each Use Case will be integrated into the CGIAR Excellence in Agronomy (EiA) Initiative Impact Dashboard.

While TRANSFORM is ready to analyze the validation data for display on the impact dashboard it is vital that Use Case teams engages in data cleaning to, for example, verify the validity of field set-ups and data submissions. Nevertheless, we strongly encourage Use Cases to run the analysis in parallel for their own insights and to ascertain the validity of the results.

The KPI documentation (Saito et al., 2022) is accessible via this link. The KPI guide provides a description of various KPIs and how to calculate them. They cover land productivity and its stability, resource use efficiency and soil health and are used across geographies, farming systems, and research and development (R&D) stages (like validation and piloting stages) (Saito et al., 2022)

The graphic below shows the various stages of the data flow during validation exercise. The highlighted section (in red) is what will be highlighted in detail in this analytics document.

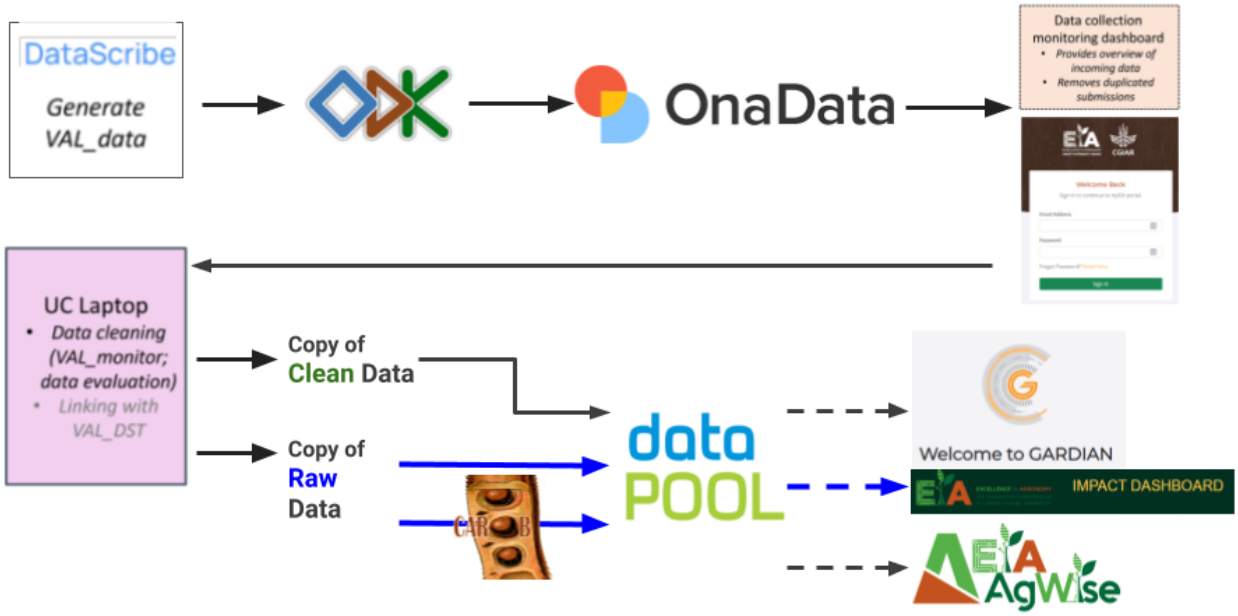


Figure 1: Validation data flow stages in the EiA ecosystem: Collected validation data is cleaned and transformed then passed to the data pool and finaly impact dashboard for visualization

Validation Analysis

Use Cases perform a validation exercise to test whether their Minimum Viable Product (MVP) meets the requirements (performance metric) that the Use Case postulated for each KPI. For more information on the protocol for validation exercises see Kreye et al., 2023. The following steps are required before any analysis and visualization can be performed on the validation data.

KPI Parameters

Revisit your protocol and the your KPIs, Which KPIs did you select and what are the performance metrices that you expect your MVP to meet for each of these? The list of main KPIs considered in EiA Use Cases comprises Yield, Yield stability, Profit, Nutrient-use efficiency, NUE for N, NUE for P, NUE for K, Water productivity, Labor productivity, Soil organic carbon, yield-scaled GHGs and Product quality.

Below is a description of the parameters for the calculations of yield, profitability (land productivity) and resource use efficiency and the required parameters for their KPI calculations (Saito et al., 2021). The following table is copied from the KPI guide, which provides comprehensive insights on the KPIs.

KPI	Unit	Detailed Indicator	Required Variables Description
Land productivity and its stability	kg/ha	Primary product harvested yield (referred to as yield)	Weight of primary harvested crop product
			Area of the plot where the trial was conducted
		Secondary product harvested yield	Weight of secondary harvested product
			Area of the plot where the trial was conducted
	US$/ha or Local currency/ha	Profit or cost–benefit balance	Gross revenue
			Total production cost
Resource use efficiency	kg (yield)/kg (nutrient input) or kg (nutrient in yield)/kg (nutrient input)	Nutrient-use efficiency (e.g. nitrogen, phosphorus)	Weight of primary harvested crop product
			Area of the plot where the trial was conducted
			Nutrient applied to a crop via inorganic fertilizer
			Nutrient applied to the crop via organic input
	kg (yield)/m3 (water input [rainfall + irrigation])	Water productivity	Weight of primary harvested crop product
			Area of the plot where the trial was conducted
			Total amount of irrigated water to the plot
			Total amount of rainfall water to the plot
	kg (yield)/ work-day	Labor productivity	Weight of primary harvested crop product
			Area of the plot where the trial was conducted
			Total number of person-day dedicated to the trial

Data Cleaning

Real data from the use cases will come in slightly different formats due to adjustments from the standard fieldbooks or ODK forms. *Data cleaning* entails the process of identifying and correcting errors, inconsistencies, or duplicates in a data set. In this guide we will not focus on the data cleaning steps, as these can differ from case to case depending on the purpose and the interest. However, it is important to notice that different cleaning purposes might result in different final results. Below is an example of data cleaning as performed in the Data Collection Monitoring Tool. Use cases that are collecting their data via the tool and downloading their data from there for analysis are already running some cleaning on their data. Below are examples steps.

An initial data cleaning step would be to subset the data to relevant variables. For this and the following examples we will make use of the tidyverse framework.

```
# Load the necessary R packages
library(tidyverse)

# Read the data
file <- "./samples/sampledata/sample_yield_data.csv"
raw.data <- read.csv(file)

# Variables to be used
vars <- c("household", "enumerator", "region", "plot_yield_T2", "plot_length_T2",
    "plot_width_T2", "plot_yield_T1", "plot_length_T1", "plot_width_T1", "riceSystem",
    "moisture_T2", "moisture_T1")

sub.data <- raw.data %>%
    select(vars)

# Remove duplicated household entries
clean.data <- sub.data %>%
    distinct(household, .keep_all = TRUE)

# Filter out moisture higher than 20%
valid.data <- clean.data %>%
    filter(across(c(moisture_T2, moisture_T1), ~. <= 20))
```

Use the above snippet as an example and substitute as per your own specific needs. For example, substitute the variables in the `vars` with the list of your own variables that are needed from your data set. In any case, the Use Case team must perform data cleaning to calculate the required standard parameters.

Additionally, the data can also be converted into a tidy data structure, for more explicit and standard format. For example:

```
# Transform data from wide to long format
tidy.data <- valid.data %>%
  pivot_longer(
    cols = c(plot_yield_T2, plot_length_T2, plot_width_T2, moisture_T2,
             plot_yield_T1, plot_length_T1, plot_width_T1, moisture_T1),
    names_to = c("variable", "treatment"),
    names_pattern = "(.*)_(T[12])",
    values_to = "value"
  )%>%
pivot_wider(
  names_from = variable,
  values_from = value
)
```

For other examples of data cleaning, please see an example script on this Github link.

Data cleaning – what to look at:

The obvious: are there any outliers? In general, applies to all input data. Review your data collection form and check all parameters that are relevant. Yield will be a first choice, but there are also other parameters that help understanding whether the output KPI data are good enough for the validation analysis. Among these are: number of plants per plot, plot dimensions, spacing, confirmations of inputs, plot management (like weeding) as well as the evaluation of the side-by-side comparison by the EA for stresses (group name) can help the assessment whether or not to include the data of a specific farm. However, do make an effort to scrutinize carefully whether data can be rightfully excluded. Keep a log with rules for exclusion.

Data Transformations

Very likely, you will need to transform your data to be able to report on your KPIs. For example, the KPI is dry matter primary yield in kg per ha and the use case has recorded the fresh yield of the product in kg), the plot length and width (m), and also the moisture content (%). From this the use case can calculate the KPI as:

$$DryYield(\frac{kg}{ha}) = \frac{FreshWeight(kg)*(1-(\frac{Moisture}{100}))}{\frac{PlotLength(m)*PlotWidth(m)}{10000}}$$

In another use case, the required KPI may be primary fresh yield in kg per ha following this calculation:

$$Yield(\frac{kg}{ha}) = \frac{FreshWeight(kg)}{\frac{PlotLength(m)*PlotWidth(m)}{10000}}$$

In essence, different KPIs require different variables (explained in table 1) which require specific unit conversions and data transformations. On your data, this will require some additional steps of organizing (and transforming) it to get to the final KPI calculations.

```
# Calculate Primary Productivity Harvested Yield (PPHY)
yield.vars <- c("household", "enumerator", "region", "plot_yield_T2", "plot_length_T2",
    "plot_width_T2", "riceSystem", "moisture_T2")
pphy.data <- clean.data %>%
    select(yield.vars)

# Generate Primary Productivity Harvested Yield (PPHY)
pphy <- (pphy.data$plot_yield_T2 * (pphy.data$moisture_T2/100))/((pphy.data$plot_length_T2 *
    pphy.data$plot_width_T2)/10000)

# Create table with transformation
kpi.data <- cbind(pphy.data, pphy) %>%
    select("household", "region", "riceSystem", "pphy")
```

The example above can be implemented for all other KPI metrics as needed by the use case. Following sections will focus on demonstrating visualization examples as produced in the EiA's impact dashboard.

Visualization

Description of terms used:

`Treatment 1 (T1): site-specific recommendation`

`Treatment 2 (T2): control`

`Treatment 3 (T3): blanket recommendation`

`Change: change in yield, profit or other KPI being measured. (Treatment 1 - Treatment 2)`

General view indicates the simple descriptive analysis of the data to show distribution (via bar chart) and difference in the different treatments (via pie chart).

The Detailed view delves into more statistical analysis to compare the distribution using a scatter plot and also show difference in the various treatments using cumulative distribution plots and boxplots to visualize distribution of nutrient-use efficiency and water-use efficiency.

I. KPI Land Productivity and its Stability

The following code illustrates how to visualize Primary Product Harvested Yield, Secondary Product Harvested Yield, and Profit.

General View

To visualize the results we will make use of wide use of 'tidyverse' libraries which offer a lot of flexibility to visualize data. There are other alternatives such as 'plotly'. To install these tools you will need to run `install.packages("<package-name>")` for each package before running the following examples.

The below block of code initializes the visualization graphics to define an object (`them2`) which controls things such as background color, chart grid lines, legends and titles, etc.

```
# Load libraries
library(tidyr)
library(dplyr)
library(tools)
library(ggplot2)
library(plotly)

# ggplot theme; can be adjusted as
# required.
them2 <- theme(panel.background = element_rect(fill = "white"),
    plot.background = element_rect(fill = "white",
        color = NA), panel.grid.major = element_blank(),
    plot.title = element_text(color = "#CD7F32",
        size = 13), strip.text.x = element_text(size = 10,
        color = "black", face = "bold"),
    axis.text = element_text(color = "black",
        face = "bold", size = 10), axis.title.x = element_text(margin = unit(c(5,
        0, 0, 0), "mm")), axis.title = element_text(color = "black",
        size = 12), legend.title = element_blank(),
    legend.text = element_text(size = 12),
    legend.background = element_rect(fill = "white"),
    panel.border = element_blank(), axis.line.x = element_blank(),
    axis.line.y = element_blank())
```

A preliminary visualization to compare the results from the MVP is to check the average treatment (control vs. MVP) yield responses. The below example shows how to construct a barplot displaying the average yield response to 3 treatments (T1, T2, T3) in 3 different regions, using the visualization style defined in `them2`.

```
  # Define the path to the file location
  file <- "./samples/sampledata/sample_Yieldiff.csv"

  # Read the data
  dataHm.pp1 <- read.csv(file)

  #Variables of interest
  subset_df <- dataHm.pp1[, c("household", "region","riceSystem",
                              "T2_yieldha","T1_yieldha","T3_yieldha", "eSSR","incrSSR" )]
subset_df <- distinct(subset_df)

  # Group by region and calculate the averages
  averages_by_state <- subset_df %>%
    group_by(region) %>%
    summarise(
      T2 = mean(T2_yieldha, na.rm = TRUE),
      T1 = mean(T1_yieldha, na.rm = TRUE),
      T3 = mean(T3_yieldha, na.rm = TRUE)
    )
  averages_by_state <- averages_by_state %>%
    pivot_longer(!region, names_to = "Treatment", values_to = "Average")

  #plot data
  ggplot(
    averages_by_state,
    aes(
      fill=Treatment,
      y=Average, x=region
    )) +
    geom_bar(position="dodge", stat="identity") +
    xlab("Location") +
    ylab(" Average grain yield (t/ha)") +
    scale_fill_manual(values = c("T2" = "#004080",
                                 "T1" = "#4caf50",
                                 "T3" = "#c26e60")) +
    #theme as defined at start
    them2
```

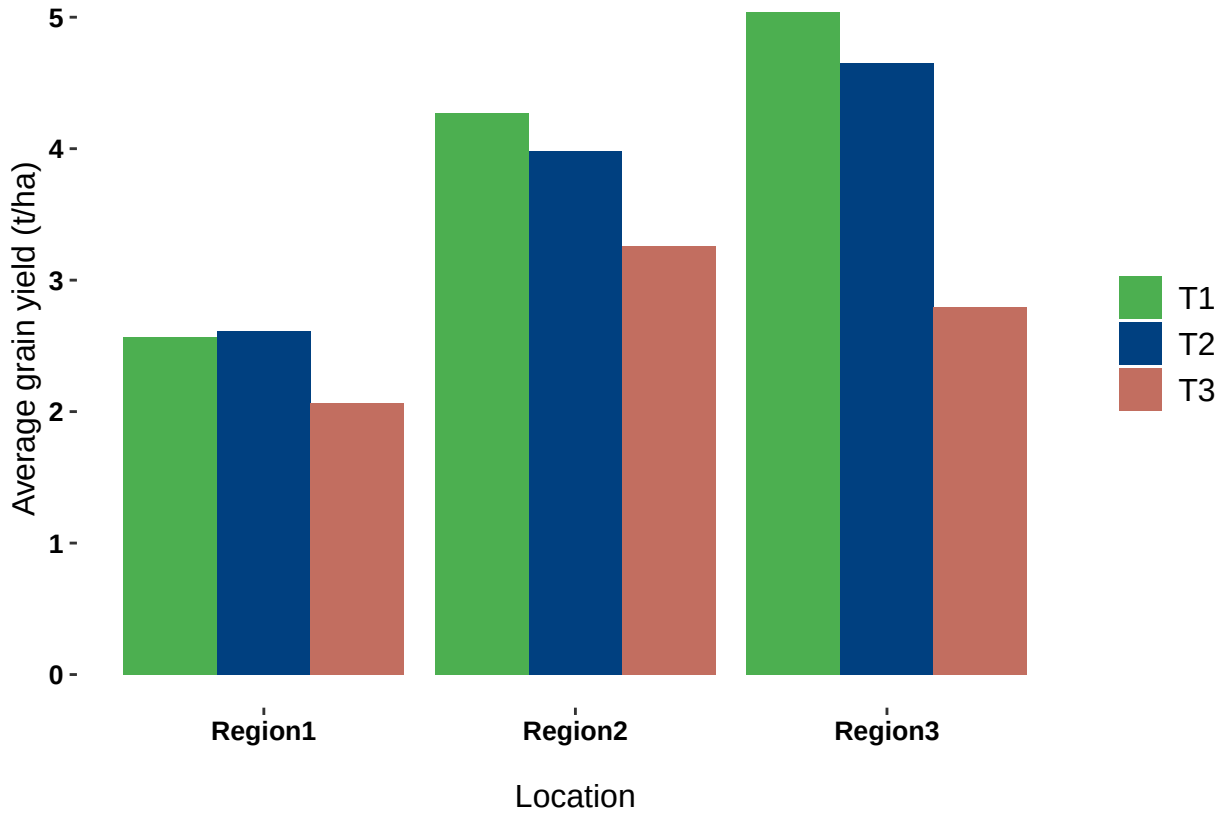


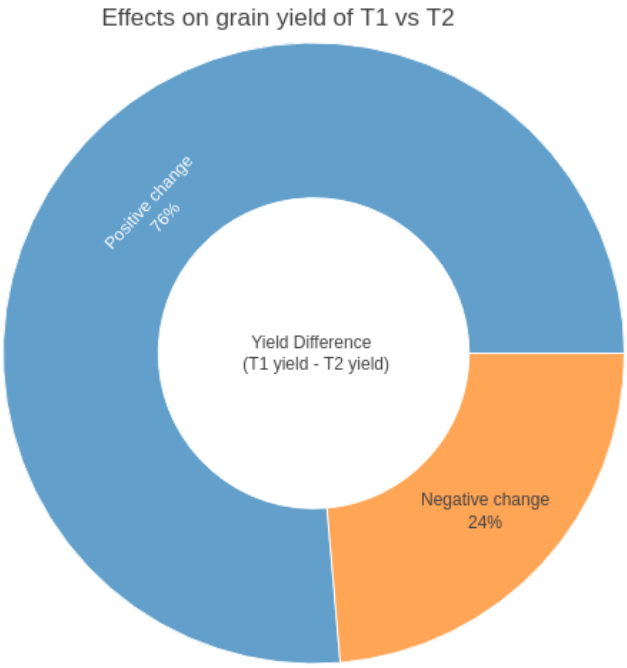
Figure 2: Yield distribution

Additional comparison of the treatments (control vs. MVP, etc) showing the percentage of trials resulting in benefits in terms of increased yield responses to the validation experiment. Pie charts are easy to interpret and summarize results in aggregated relative numbers.

```
# Pie plot to show  difference (positive vs negative)
# incrSSR  is the change variable (T1 - T2)

# Calculate negative and positive change percentage values
x <- (subset_df[! is.na(subset_df$incrSSR),] )$incrSSR
xi <- x[x<0]
xj <- x[x>0]
pos <- (length(xj)/length(x))*100
neg <- (length(xi)/length(x))*100
ds <- data.frame(labels = c("Yield Difference <br> (T1 yield - T2 yield)",
                            "Positive change", "Negative change"),
                 values = c(NA, pos, neg))

# Plot the data
plot_ly(data = ds,
         labels = ~labels,
         values = ~values,
         parents = c("", "Yield Difference <br> (T1 yield - T2 yield)",
                     "Yield Difference <br> (T1 yield - T2 yield)"),
         type = "sunburst",
         branchvalues = 'total',
         textinfo = "label+percent entry",
         hoverinfo = "text",
         hovertext = paste("% of farmers experiencing<br>",
                           tolower(ds$labels), "from T1")) %>%
  layout(title = 'Effects on grain yield of T1 vs T2')
```


Effects on grain yield of T1 vs T2
Positive change
76%
Yield Difference
(T1 yield - T2 yield)
Negative change
24%

Detailed View

This section provides the example code blocks in order to produce more detailed visualization of the results. These include scatter plots showing the explicit distribution of the observations or cumulative distributions (CDF) to visualize primary yield, secondary yield or profit distributions.

Below provides the code necessary to generate a scatter plot of the grain yield (tonne/ha) observed in the T1 and T3 compared to T2 (control). This section provides the example code blocks in order to produce more detailed visualization of the results. These include scatter plots showing the explicit distribution of the observations or cumulative distributions (CDF) to visualize primary yield, secondary yield or profit distributions.

```
# Detail-scatter plot

# Define the path to the file location
file <- "./samples/sampledata/sample_yield_data.csv"

# Read the data
dataHm.pp1 <- read.csv(file)

# Transform data accordingly
dataHm.pp1 <- dataHm.pp1 %>%
    mutate(riceSystem = case_when(riceSystem == "rainfedLowland" ~ "Rainfed lowland",
        riceSystem == "irrigated" ~ "Irrigated", TRUE ~ riceSystem))
dataHm.pp1$region <- toTitleCase(dataHm.pp1$region)
dataHm.pp1 <- dataHm.pp1[!is.na(dataHm.pp1$T2_yieldha), ]

# Plot the data
  ggplot(
    dataHm.pp1,
    aes(
      T2_yieldha, T1_yieldha
      )
    ) +
    geom_point(size = 1) +
    geom_abline(slope = 1, intercept = 0, size = 0.5, colour = "grey") +
    facet_wrap(~region) +
    xlab("Grain yield (t/ha) of the control (T2)") +
    ylab("Grain yield (t/ha) of the MVP (T1)") +
    labs(title = "Yield distribution") +
    #theme as defined above
    them2
```

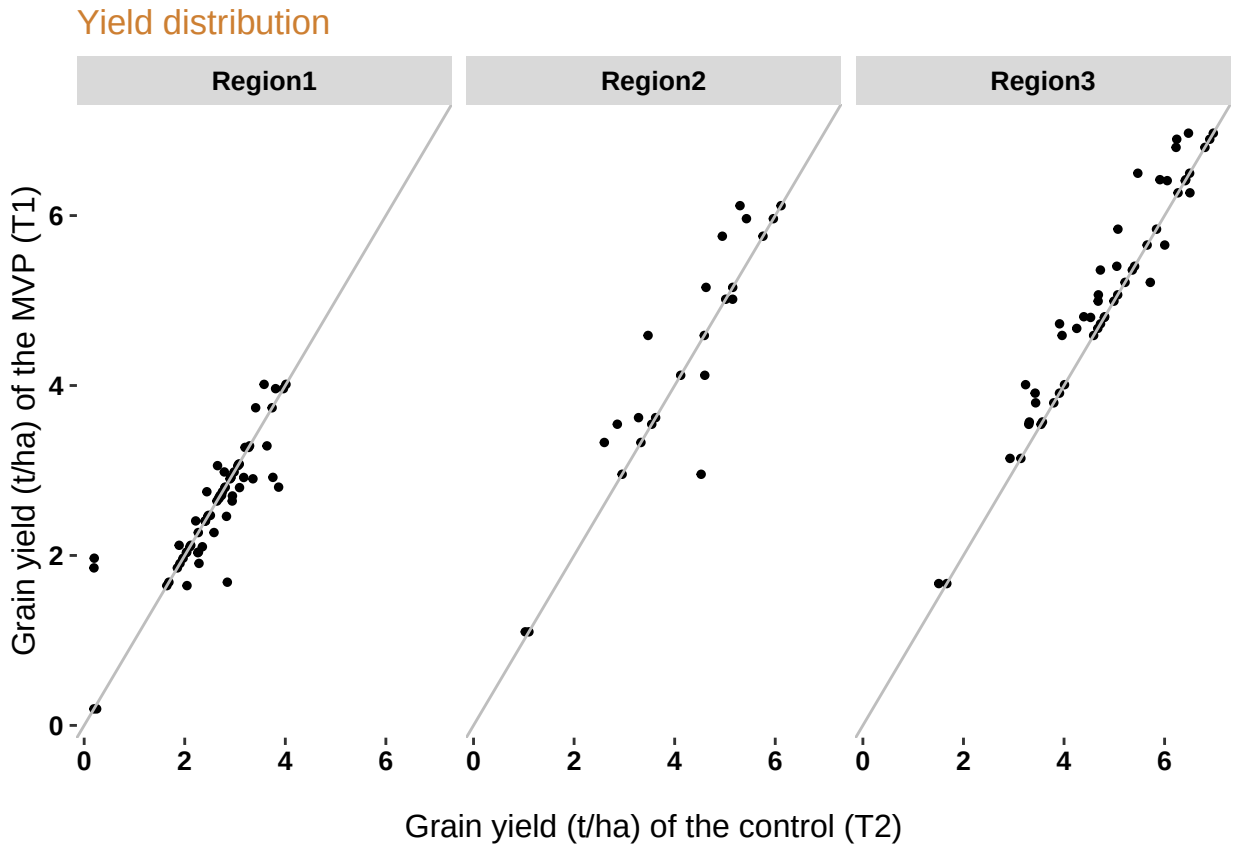


Figure 3: Yield distribution

Cumulative distributions allow to see which probability of the sample distribute across negative and positive impacts. Positive values indicate better performance of the technology (T1) vs the reference treatment (T2), and negative values indicate trials that have worse results for the new technology (T1) vs the reference treatment (T2). We add the upper and lower boundaries of a 95% confidence interval to better understand how treatments compare and across the probability distributions.

```
# eSSR is the difference in yield between the treatments (T1-T2)
# riceSystem used to compare different attributes can also be landscapes.

# Define the path to the file location
file <- "./samples/sampledata/sample_Yieldiff.csv"
# Read the data
yieldiff <- read.csv(file)

# Calculate and add the empirical cumulative probability from the
# empirical cumulative distribution function (eCDF) and add the upper
# and lower 95% confidence intervals
yieldiff <- yieldiff[,c("region", "riceSystem", "eSSR")] %>%
  group_by(region, riceSystem) %>%
  arrange(region, riceSystem, eSSR) %>%
  mutate(N = n(),
         ecdf = ecdf(eSSR)(eSSR),
         sd = sqrt(log(2/0.05)/(2*N)),
         lower = pmax(1:N/N-sd, 0),
         upper = pmin(1:N/N+sd, 1))

# Plot the data
ggplot(yieldiff, aes(eSSR, ecdf, colour=riceSystem)) +
  geom_point() +
  geom_ribbon(aes(ymin = lower,
                  ymax = upper),
              alpha=.2) +
  geom_vline(xintercept = 0, size = 0.5, colour = "grey")+
  xlab("Yield difference (T1 - T2) (t/ha)") +
  labs(title="Comparison of yields for site-specific (T1) and control (T2)")+
  ylab("Cumulative probability") +
  facet_wrap(~region)+
  them2
```

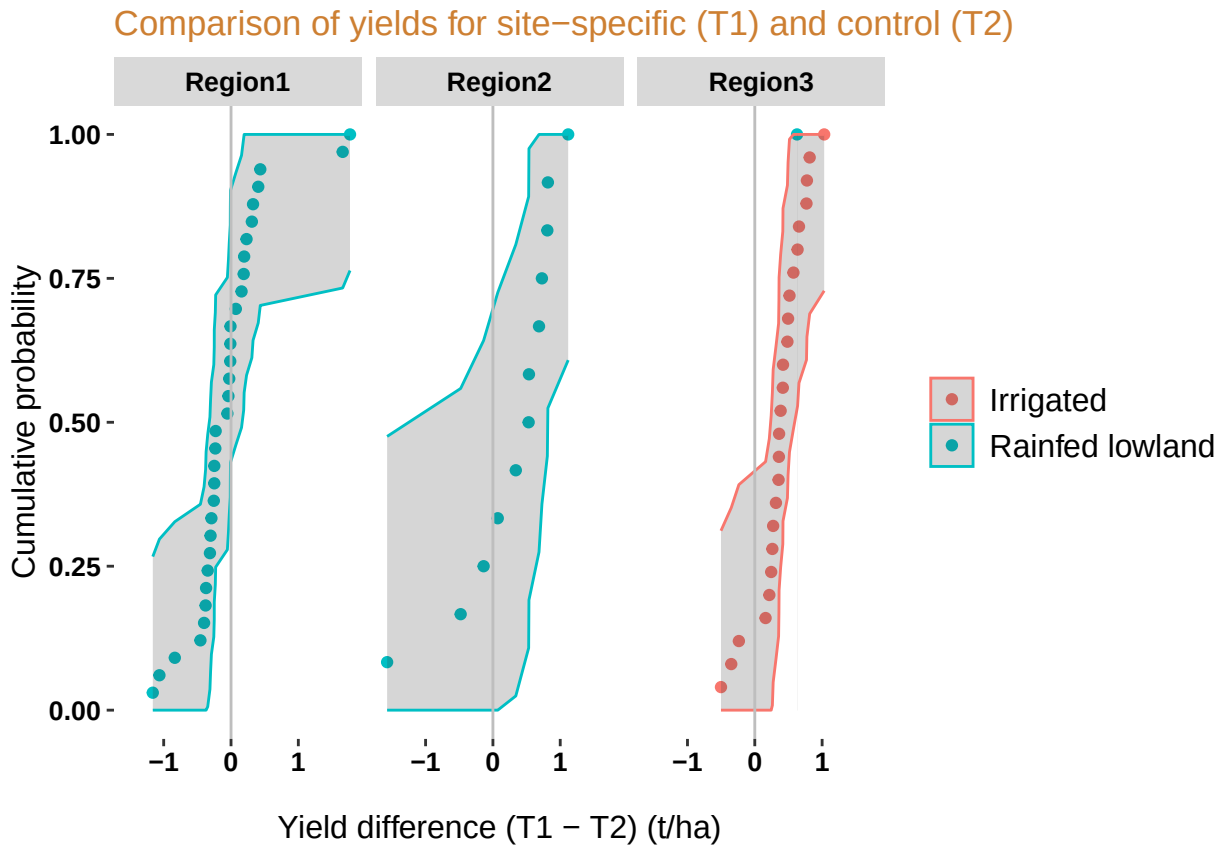


Figure 4: Comparison of yields for site-specific (T1) and control (T2)

II. KPI Resource Use Efficiency

The following code illustrates how to visualize Nutrient-Use Efficiency, Water-Use Efficiency and Labor Productivity

General Overview

```
# Bar plot to show distribution for various treatments across regions

# This example shows distribution of nitrogen-use efficiency.

# Define the path to the file location
file <- "./samples/sampledata/sample_nue_data.csv"

# Read the data
dataHNm.nn1 <- read.csv(file)

# Variables of interest
nue_df <- dataHNm.nn1[, c( "Region", "plot" , "useN" )]

# Transform data and calculate average by group
nue_df <- distinct(nue_df)
nue_by_state <- nue_df %>%
  group_by(Region, plot) %>%
  summarise(avg_useN = mean(useN, na.rm = TRUE))

# Plot the data
ggplot(
  nue_by_state,
  aes(fill = plot, y = avg_useN, x = Region)) +
  geom_bar(position = "dodge", stat = "identity") +
  xlab("Location") +
  ylab("Average (Kg grain per kg applied N))") +
  scale_fill_manual(values = c("T2" = "#004080", "T1" = "#4caf50", "T3" = "#c26e60")) +
  them2
```

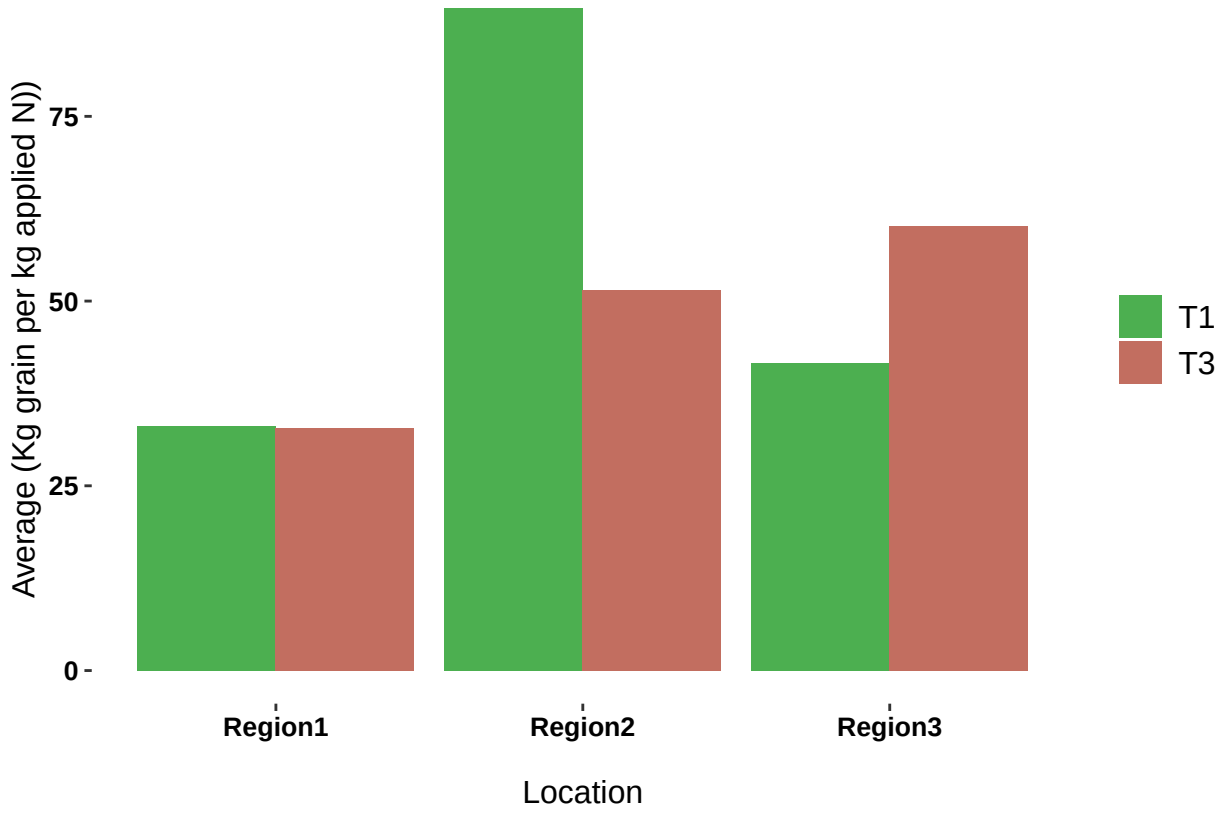


Figure 5: Nitrogen Use Efficiency

Distribution: Nutrient-Use Efficiency, Water-Use Efficiency and Labor Productivity

```
# Box plot- nitrogen use efficiency

ggplot(dataHNm.nn1, aes(plot, useN, fill = plot)) +
  geom_boxplot()+
  facet_grid(riceSystem~Region)+
  xlab("Plot") +
  ylab("Kg grain per kg applied N") +
  them2
```

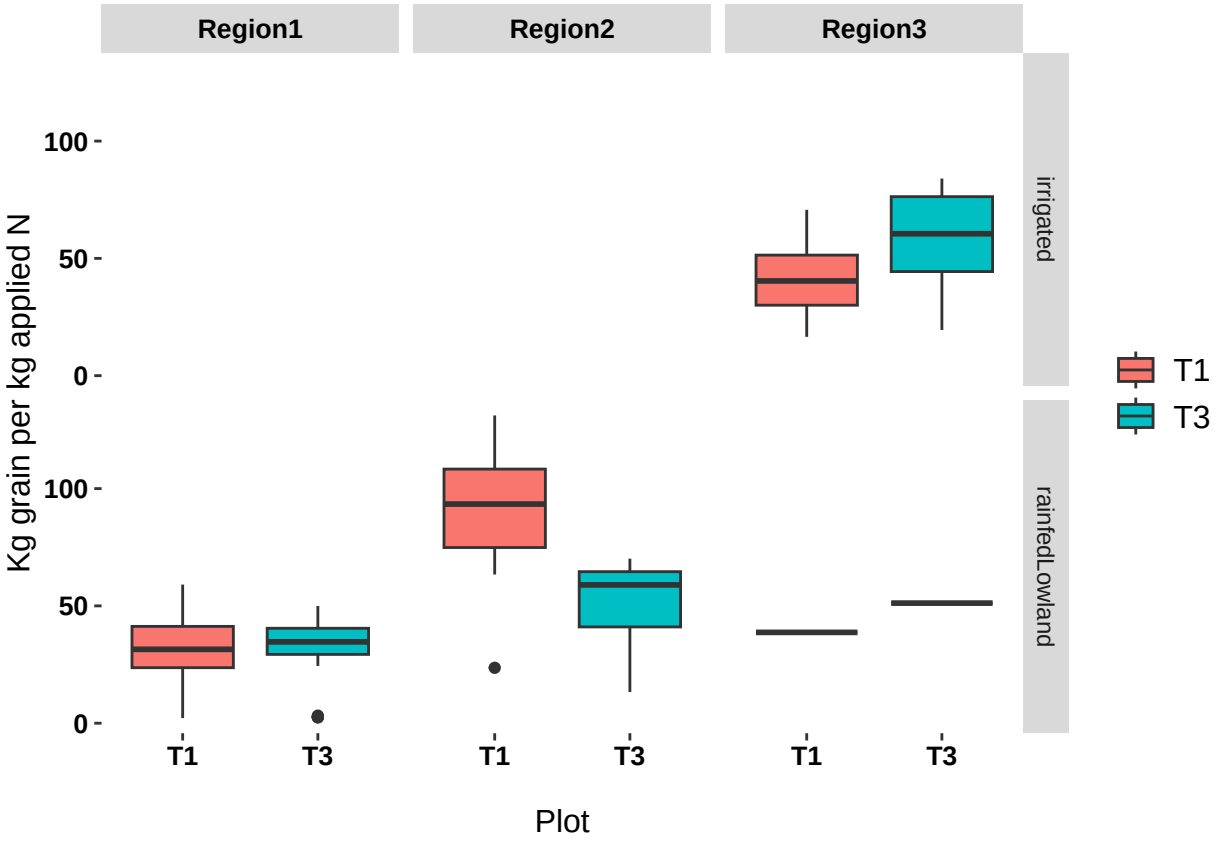


Figure 6: Nitrogen Use Efficiency

References

C.Kreye, R.Flor, R.Manners, C.Aubert , 2023. Protocol for the validation exercise. Excellence in Agronomy Initiative, CGIAR, 21p.

Saito K, Johnson J-M, Hauser S, Corbeels M, Devkota M and Casimero M. 2022. Guideline for measuring agronomic gain key performance indicators in on-farm trials, v. 1. Africa Rice Center, Abidjan, Côte d'Ivoire.

Contributors

Regina Kilwenge (International Institute of Tropical Agriculture [IITA]), Abigail Elmido-Mabilangan (International Rice Research Institute) and Christine Kreye (International Institute of Tropical Agriculture [IITA])

This work was financially supported by the Excellence in Agronomy for Sustainable Intensification and Climate Change Adaptation Initiative.

Suggested Citation: