

# Figure2

严婉莹

2025-05-19

```
# Load required libraries
require(readxl); # for reading Excel files

require(data.table); # for efficient data manipulation

require(ggplot2); # for plotting

library(ggpubr) # for combining multiple plots

# Read the data from the Excel file (Figure2 sheet)
metaresult_group<- readxl::read_xlsx('F:/研究生/研究生课程/数据驱动与可重复性研究/小组
作业/Source Data.xlsx',sheet = "Figure2")
metaresult_group <- as.data.table(metaresult_group) # Convert data to a data.table for better pe
rformance

# Subset the data where 'Vari' is 'NUE' and 'Group type' is 'Data'
mydata <- metaresult_group[Vari == 'NUE' & `Group type` == 'Data']

# Create the first plot (p1) for the 'Data' group
p1 <- ggplot(data = mydata,
  aes(x = Management, y = mean, shape = Group, fill = Group)) +
  # Add horizontal dashed line at y=0
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.3) +
  # Add error bars for the confidence intervals
  geom_errorbar(position = position_dodge(0.7), aes(ymin = ci.lb, ymax = ci.ub), width = 0.3, s
ize = 0.8) +
  # Add points for each data, with custom size and stroke
  geom_point(position = position_dodge(0.7), size = 4, stroke = 0.5) +
  # Set custom shapes for different groups
  scale_shape_manual(values = c("Meta-analytical data (ROM)" = 21, "Primary data (ROM)" =
24)) +
  # Add text labels (n) above the upper bound of confidence intervals
  geom_text(aes(x = Management, y = ci.ub + 6, label = n),
    position = position_dodge(width = 0.7), vjust = 0, hjust = 0.5,
    size = 4.5, check_overlap = FALSE) +
  # Set custom x-axis order and labels for the management practices
  scale_x_discrete(limits = rev(c("Reduced tillage", "No tillage", "Crop rotation",
    "Cover cropping", "Residue retention", "Fertilizer timing",
    "Fertilizer rate", "Fertilizer placement", "Organic fertilizer",
    "Combined fertilizer", "Enhanced efficiency")),
    labels = rev(c("RT", "ZT", "ROT", "CC", "RES", "RFT", "RFR", "RFP", "OF", "CF",
    "EE")))) +
  # Set y-axis limits and breaks
  scale_y_continuous(limits = c(-70, 120), breaks = c(-50, 0, 50, 100)) +
```

```

# Add axis labels and set the color of the legend title
labs(x = "Management practice", y = "Relative change of NUEr (%)", colour = 'black') +
# Apply white background theme and customize axis titles and text
theme_bw() +
theme(legend.title = element_blank(),
      legend.direction = "horizontal",
      legend.position = c(0.3, 0.1),
      legend.key = element_rect(fill = "white", size = 1.5),
      legend.key.width = unit(0.4, "lines"),
      legend.key.height = unit(0.5, "lines"),
      legend.background = element_blank(),
      legend.text = element_text(colour = 'black', size = 18),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title = element_text(size = 20, colour = 'black', face = 'bold'),
      axis.title.x = element_blank(), # Hide x-axis title
      axis.text.y = element_text(colour = 'black', size = 22),
      axis.text.x = element_text(colour = 'black', size = 22, hjust = 0.5, vjust = 0.5))

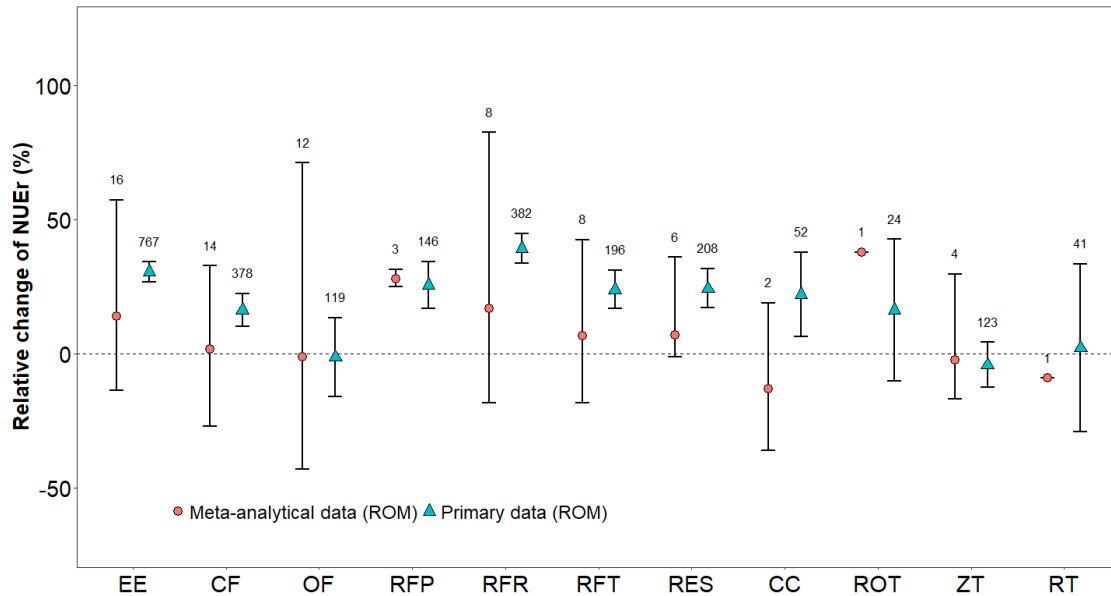
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

p1 # Display the first plot

```



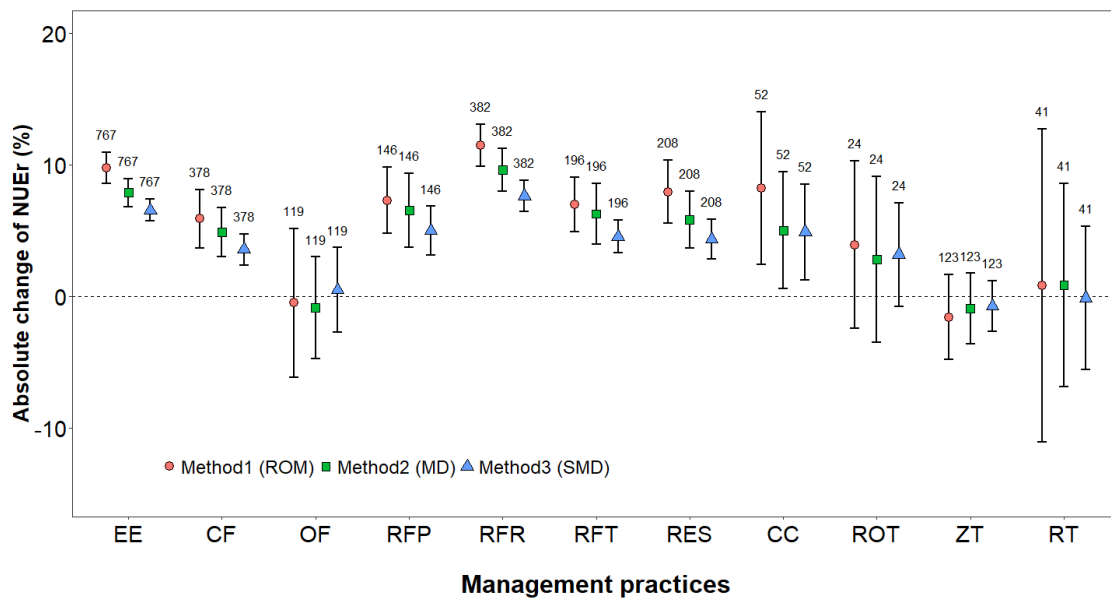
```
# Subset the data where 'Vari' is 'NUE' and 'Group type' is 'Method'
mydata <- metaresult_group[Vari == 'NUE' & `Group type` == 'Method']

# Create the second plot (p2) for the 'Method' group
p2 <- ggplot(data = mydata,
  aes(x = Management, y = mean, shape = Group, fill = Group)) +
  # Add horizontal dashed line at y=0
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.3) +
  # Add error bars for the confidence intervals
  geom_errorbar(position = position_dodge(0.7), aes(ymin = ci.lb, ymax = ci.ub), width = 0.3, size = 0.8) +
  # Add points for each data, with custom size and stroke
  geom_point(position = position_dodge(0.7), size = 4, stroke = 0.5) +
  # Set custom shapes for different groups
  scale_shape_manual(values = c("Method1 (ROM)" = 21, "Method2 (MD)" = 22, "Method3 (SMD)" = 24)) +
  # Add text labels (n) above the upper bound of confidence intervals
  geom_text(aes(x = Management, y = ci.ub + 1, label = n),
    position = position_dodge(width = 0.7), vjust = 0,
    hjust = 0.5, size = 4.5, check_overlap = FALSE) +
  # Set custom x-axis order and labels for the management practices
  scale_x_discrete(limits = rev(c("Reduced tillage", "No tillage", "Crop rotation",
    "Cover cropping", "Residue retention", "Fertilizer timing",
    "Fertilizer rate", "Fertilizer placement", "Organic fertilizer",
    "Combined fertilizer", "Enhanced efficiency")),
    labels = rev(c("RT", "ZT", "ROT", "CC", "RES", "RFT", "RFR", "RFP", "OF", "CF",
    "EE")))) +
  # Set y-axis limits and breaks
  scale_y_continuous(limits = c(-15, 20), breaks = c(-10, 0, 10, 20)) +
  # Add axis labels and set the color of the legend title
  labs(x = "\nManagement practices", y = "Absolute change of NUEr (%) ", colour = 'black') +
  # Apply white background theme and customize axis titles and text
```

```

theme_bw() +
theme(legend.title = element_blank(),
      legend.direction = "horizontal",
      legend.position = c(0.3, 0.1),
      legend.key = element_rect(fill = "white", size = 1.5),
      legend.key.width = unit(0.4, "lines"),
      legend.key.height = unit(0.5, "lines"),
      legend.background = element_blank(),
      legend.text = element_text(colour = 'black', size = 18),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.title.x = element_text(size = 24, colour = 'black', face = 'bold'),
      axis.title.y = element_text(size = 20, colour = 'black', face = 'bold'),
      axis.text.y = element_text(colour = 'black', size = 22),
      axis.text.x = element_text(colour = 'black', size = 22, hjust = 0.5, vjust = 0.5))
p2 # Display the second plot

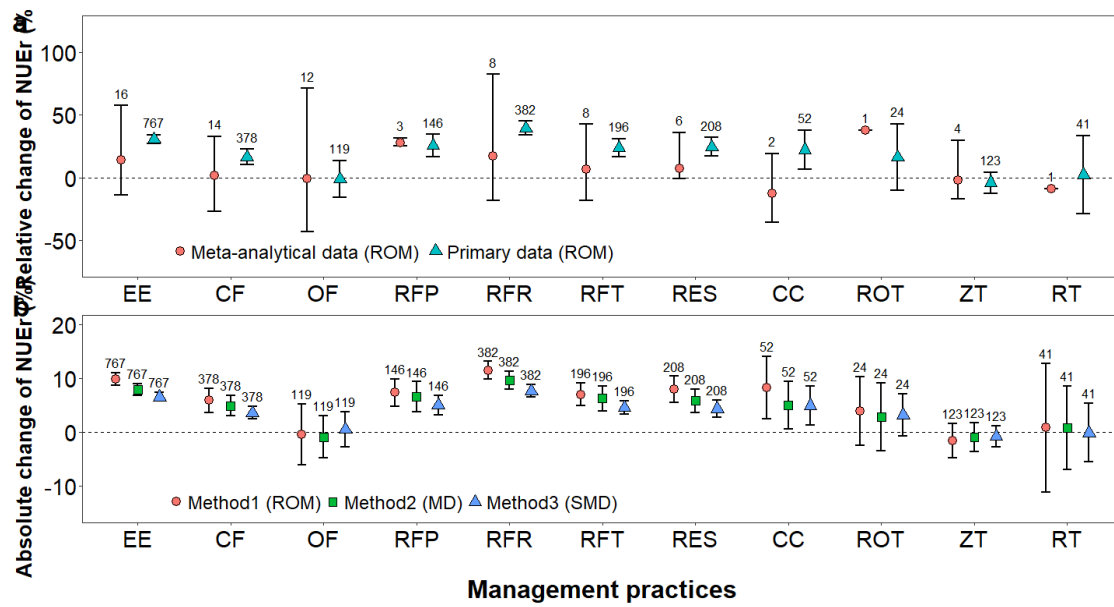
```



```

# Combine both plots (p1 and p2) into a single plot (p) with vertical alignment
p <- ggarrange(p1, p2, ncol = 1, nrow = 2, align = "v", # Arrange plots vertically
               labels = c("a", "b"), label.x = 0, label.y = c(1, 1.05),
               font.label = list(size = 28), hjust = -0.2, vjust = 1)
p # Display the combined plot

```



*# Save the combined plot as a PNG file with specified dimensions*

`ggsave(plot = p, file = "F:/研究生/研究生课程/数据驱动与可重复性研究/小组作业/picture/Figure 2.png", width = 410, height = 270, units = "mm")`