Supplementary Information

Multilevel Meta-Analysis and Probabilistic Population-Wide Health Risk Assessment of Dietary Pesticide Exposure in Ethiopia

Asefa et al.

July 2025

Contents

Section 0: Introduction

This Supplementary Information (SI) file provides additional details on our methodology, along with step-by-step implementation in R. Source data, including extracted datasets and R code scripts are available at https://github.com/aelsai/Dietary-Pesticide-Exposure-in-Ethiopia.git.

Overall, this file is structured into two main sections. The first section offers additional details and justifications for key methodological components (e.g., handling of non-detects and missing data imputation, multilevel meta-analysis, and how these methods align with our study objectives). The second section covers R implementation and preliminary results not included in the main text. Accordingly, this file is organized into five main outlines: SI Method (background and literature search), dataset preparation and exploration, missing data handling (non-detects, standard deviation, sample size), multilevel meta-analysis, and probabilistic risk assessment.

1   Section 1: SI Methods

1.1   Literature Search and Dataset Preparation

The studies included in this analysis were part of a more comprehensive literature database compiled to support a national-scale, multi-pathway, multi-pesticide exposure and cumulative risk assessment in Ethiopia (see the Supplementary file for the unregistered systematic review protocol).

Briefly, a systematic search was conducted across 13 international and local databases and repositories using validated search terms to identify studies reporting pesticide levels in multiple environmental media (e.g., air, water, soil, food) in Ethiopia. Search terms were categorized into pesticide-related terms (e.g., agrochemical, insecticide, fungicide, herbicide), occurrence- or exposure-related terms (e.g., pollution, exposure, monitoring, residue, contamination, air, soil, water, food), and geographic location (Ethiopia). As recommended by Lagisz et al. (2025; https://doi.org/10.1017/rsm.2024.6), evaluation of these terms using a Scopus search against 25 benchmark articles retrieved a priori revealed 100% sensitivity. The final search retrieved 1,539 records, from which 45 studies specifically addressing pesticide residues in food were included here. The overall process is summarized using the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) flow diagram, as shown in SI Figure 1.

```
library(here)
knitr::include_graphics(here("Figure/PRISMA.png"))
```

In addition to the main eligibility criteria presented in our protocol, we assessed whether primary, non-duplicate food residue data were presented. Accordingly, five studies were excluded because sufficient quantitative data were not provided. From the remaining 40 studies, we extracted study characteristics (first author, title, DOI, publication year), sample characteristics (food source type, analytical instrument, sample size, sample year), and pesticide data (name, summary statistics). We also extracted and summarized all screened pesticides, regardless of detection status, along with reported limits of detection and quantification (LOD/LOQ). Whenever available, raw residue concentrations were prioritized (i.e., unique sample/location measurements) over summary statistics (see Source Data).
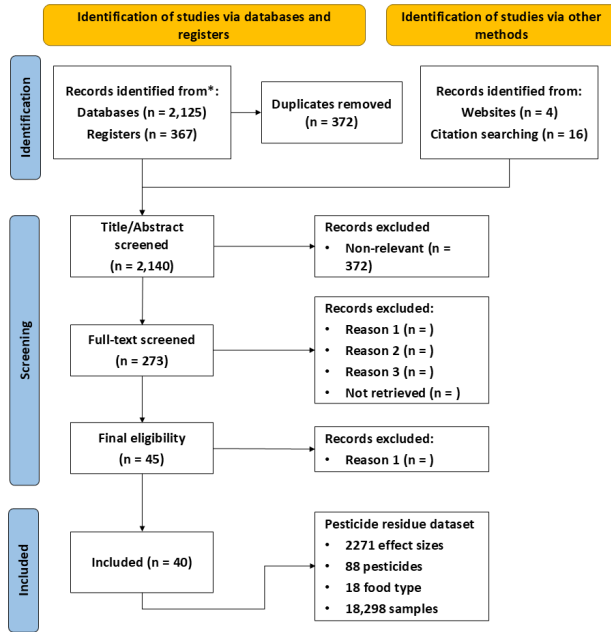
Figure 1: PRISMA flow diagram for study identification, screening and inclusion

1.2 Regulatory Risk Assessment in Ethiopia

Ethiopia established a legal framework for pesticide regulation with the enactment of Proclamation No. 674/2010, and subsequently introduced a scientifically based pesticide registration system in 2014 to ensure that pesticide use does not pose unacceptable risks to human health, animal health, or the environment (Deneer et al., 2014). Despite this regulatory advancement, the current risk assessment process primarily emphasizes environmental exposure, particularly surface and groundwater, while largely neglecting food safety. According to the pesticide registration and control regulation (Schedule II, Article 1.1.4), the evaluation of dietary exposure in both humans and animals under local conditions is a mandatory component of the registration process. However, in practice, this aspect of risk assessment remains underdeveloped. Ethiopia currently adopts Codex Alimentarius (CXLs) maximum residue limits (MRLs) by default. Yet, safety assessments are rarely conducted for domestic and exported foods. In theory, a deterministic approach is employed for chronic dietary risk assessment using the WHO-GEMS/Food International Estimated Daily Intake (IEDI) model, which has been adapted by excluding non-relevant commodities. Nonetheless, when Codex MRLs or international risk assessment scenarios do not align with local pesticide use patterns, Ethiopia lacks the legislative framework to define its own national MRLs or establish context-specific risk values.

1.3 Addressing Data Challenges in Dietary Risk Assessment

In data-limited Global South contexts like Ethiopia, where pesticide use is widespread but monitoring and regulatory frameworks are weak, synthesizing sparse residue data via meta-analysis is crucial for quantifying health risks. However, conventional methods struggle with hierarchical clustering (e.g., dependencies across studies, regions, or food groups), left-censoring (non-detects below detection limits), missing values, variabilities (e.g., in consumption or residues), and uncertainties (e.g., measurement errors). Our compiled pesticide food residue dataset exemplifies these issues, with 41% of effect sizes being non-detects (NDs), 64% missing standard deviations (SDs), 12.5% missing sample sizes (SS), and 6.5% missing detection limits (DLs). If left unaddressed, these challenges can severely bias estimates, leading to incomplete or incorrect conclusions. To overcome these, our novel MMA-MC-PRA framework incorporates advanced imputation

techniques, enabling robust national-scale estimates and probabilistic risk distributions that test hypotheses on residue variations, moderators, and population risks.

1.3.1 Imputation of Non-Detects in Left-Censored Residue Data

Like most environmental datasets, our dataset is also highly left-censored, with 41% non-detects (NDs). Simple substitution methods, such as replacing NDs with zero, DL, or DL/2, are commonly recommended by regulatory bodies (e.g., US EPA, EFSA) for their simplicity but can introduce substantial bias, particularly when NDs exceed 30% of the dataset, leading to under- or overestimation of summary statistics and risk assessments. To mitigate this, we employed four advanced imputation techniques suitable for environmental and food residue data: zero-inflated lognormal (ZILN), maximum likelihood estimation (MLE) with censored lognormal fitting, regression on order statistics (ROS), and the non-parametric Kaplan–Meier (KM) method.

Background and justifications for each method are presented below; for details, readers are referred to EFSA (2010; https://doi.org/10.2903/j.efsa.2010.1557), Szarka et al. (2018; https://doi.org/10.1021/acs.jafc.8b00863), Shoari and Dubé (2018; https://doi.org/10.1002/etc.4046), Chen et al. (2015; https://doi.org/10.1007/s13253-015-0196-3), and Sang et al. (2024; https://doi.org/10.1016/j.jenvman.2023.119813).

Zero-Inflated Lognormal (ZILN) Imputation: The ZILN method models left-censored data as a mixture of true zeros and a lognormal distribution, accounting for zero-inflation common in pesticide residues where NDs may represent either absence or concentrations below the DL. This approach is particularly suitable for datasets with high ND proportions and bimodal patterns, as it reduces bias by separately estimating the probability of true zeros (e.g., via a binomial component) and imputing non-zero NDs from a truncated lognormal distribution.

Maximum Likelihood Estimation (MLE): MLE imputes NDs by fitting a lognormal distribution to the censored data and drawing random values from the truncated distribution below the DL. We found that lognormal distributions best fit the residue data, and MLE provides efficient parameter estimates that maximize the likelihood of observed data, including censored points. This parametric method outperforms simple substitutions by incorporating uncertainty in NDs through random sampling from the fitted distribution, making it robust for risk assessment when data fit lognormality assumptions.

Regression on Order Statistics (ROS): ROS is a semi-parametric method that imputes NDs by regressing ordered detected values against their normal quantiles (probability plotting positions), assuming an underlying lognormal distribution, and extrapolating to censored points. It handles left-censoring effectively without requiring full parametric assumptions, reducing bias in mean and variance estimates compared to substitution methods, especially in moderately censored datasets.

Kaplan-Meier (KM) Method: The KM method is a non-parametric survival analysis technique adapted for left-censored data, estimating the empirical cumulative distribution function (ECDF) by treating NDs as left-censored survival times and imputing via inverse transform sampling from the survival curve. As a distribution-free approach, KM avoids parametric assumptions (e.g., lognormality), making it ideal for datasets where the underlying distribution is unknown or non-lognormal, and it provides unbiased estimates of percentiles and means in highly censored data.

These methods were selected based on their robustness for handling left-censored data, as demonstrated in prior studies (see references above). Imputation was performed using the `EnvStats`, `NADA`, and `survival` packages.

1.3.2 Imputation of Missing Standard Deviations and Sample Sizes

As noted earlier, SDs and sample sizes were missing in approximately 64% and 12% of our residue dataset, respectively, posing challenges for downstream meta-analytic calculations that require complete variance estimates. While deterministic formulas (e.g., Hozo et al., 2005) can estimate SD from available statistics like means and ranges, they are inadequate for high missingness levels, as they ignore uncertainty and

inter-variable dependencies, potentially introducing bias. Instead, we employed Multivariate Imputation by Chained Equations (MICE) with predictive mean matching (pmm) to handle this missingness.

MICE is an iterative, multivariate imputation technique that models each incomplete variable conditionally on others via chained regression equations, producing multiple plausible datasets that account for imputation uncertainty. Unlike single imputation or complete-case analysis, MICE reduces bias and variance underestimation in datasets with substantial missingness by incorporating relationships among variables, making it suitable for environmental meta-analyses where data are often incomplete and hierarchically structured. The pmm method non-parametrically matches missing values to observed donor values with similar predicted means from regression models, preserving the original data distribution and handling non-normality effectively.

For detailed methodological backgrounds, readers are referred to Azur et al. (2011; https://doi.org/10.1002/mpr.329), Ian et al. (2011; https://doi.org/10.1002/sim.4067), and Kambach et al. (2020; https://doi.org/10.1002/ece3.6806).

MICE was implemented using the `mice` package, generating 50 imputed datasets over 20 iterations, with visual diagnostics (density plots) confirming that imputations preserved the underlying data distributions. We assumed data were missing at random (MAR) and that SD and SS were strongly dependent on mean residue values, despite the dataset's hierarchical structure (e.g., nested by study, pesticide, food group). To select a single representative dataset for analysis, we calculated the total percentage absolute error between imputed and observed means for SD and SS across imputations, choosing the one with the minimal error to minimize bias. This approach ensures the chosen dataset closely aligns with observed central tendencies and serves as a pragmatic alternative to pooling all imputations when a single dataset is needed for complex meta-analysis, while retaining computational efficiency.

1.4   Multilevel Meta-Analysis Modelling

Our residue data exhibit hierarchical dependencies, with multiple effect sizes nested within studies (i.e., more than one effect size from a single study). Such statistical non-independence can arise from clustering in sampling methods, locations, and analytical techniques within the same study. Traditional meta-analyses may inadequately capture this complexity, leading to biased estimates and underestimated heterogeneity. Multilevel meta-analysis (MMA) addresses these issues by modeling variance at multiple levels, providing more accurate pooled estimates and insights into heterogeneity sources for environmental policy and risk assessment. For detailed backgrounds on MMA, readers are referred to Assink and Wibbelink (2016; https://doi.org/10.20982/tqmp.12.3.p154), Nakagawa et al. (2023; https://doi.org/10.1186/s13750-023-00301-6), Van den Noortgate et al. (2013; https://doi.org/10.3758/s13428-012-0261-6), and Harrer et al. (2021; https://bookdown.org/MathiasHarrer/Doing_Meta_Analysis_in_R/multilevel-ma.html).

We implemented three-level MMA models using the `rma.mv` function from the `metafor` package, incorporating robust variance estimation (RVE) via `clubSandwich` to correct for small-sample bias and study clustering. Models accounted for: *Level 1 (sampling variance), Level 2 (within-study variance among effect sizes), and Level 3 (between-study variance).* A tiered strategy was applied to test our hypotheses.

The intercept-only model (Model 1) estimated the overall pooled log-transformed mean residue concentration without moderators, using restricted maximum likelihood (REML) for unbiased variance components. This baseline model quantifies the grand mean while partitioning heterogeneity into within- and between-study components, essential for datasets with nested structures to avoid inflated Type I errors. Subgroup models (Models 2.1 and 2.2) incorporated fixed effects for pesticide use type and food group, respectively, to identify categorical sources of variability, such as differential residue accumulation by pesticide type or food category, which is critical for targeted risk assessments in agricultural contexts. RVE was applied post-fitting to ensure robust standard errors, and multilevel I^2 confirmed substantial heterogeneity, validating further moderator exploration. Uni-moderator models (Models 3.1–3.6) were fitted using maximum likelihood (ML) estimation to evaluate individual predictors (instrument, food origin, region, zone, use type, food group) against a null model. ML facilitates nested model comparisons through likelihood ratio tests (LRTs), enabling identification of significant moderators *(p < 0.05)* that explain heterogeneity in multilevel frameworks.

Significant moderators from Model 3 were refitted using REML (Models 4.1–4.4) with RVE for final effect estimates, I^2 (variance distribution across levels), and R^2 (proportion explained by moderators) reporting.

For granular estimates, we fitted 251 independent two-level random-effects models using `rma.uni` for unique pesticide-food pairs with at least two effect sizes (k 2), stratified by a combined pesticide–food identifier. This classical random-effects approach avoids overparameterization in sparse subgroups where multilevel random effects (e.g., in `rma.mv`) may be inestimable due to low k, providing precise, pair-specific pooled estimates (tau^2 for between-study variance) while accounting for within-pair heterogeneity. REML was used for each model, yielding detailed insights into specific combinations unattainable in a single multilevel model.

1.5 Monte Carlo-Based Probabilistic Risk Assessment

Preparing comprehensive general population exposure data for dietary risk assessments is a highly complex and tedious undertaking, often fraught with challenges related to data availability and quality. The same is true for compiling hazard data, where high uncertainty is tied to contaminants' toxicological information. In this regard, leveraging the outputs of our MMA (Meta-Analysis Modeling) framework, we integrated Monte Carlo-based probabilistic risk assessment (MC-PRA) to arrive at scientifically rigorous and realistic population-wide risk assessments in Ethiopia. Unlike deterministic approaches that use conservative point estimates and may overestimate hazards, MC-PRA accounts for input uncertainties (e.g., residue variability, consumption heterogeneity) via stochastic simulations, providing percentile-based outputs and exceedance probabilities that better inform risk management, especially in data-limited settings like Ethiopia.

For details on probabilistic risk assessment, see Flinders et al. (2025; https://doi.org/10.1093/inteam/vjaf016), Nielsen et al. (2023; https://doi.org/10.1186/s12940-022-00918-z), and EFSA (2012; https://doi.org/10.2903/j.efsa.2012.2839).

Our MC-PRA modelling was based on the well-established WHO/FAO framework (https://www.who.int/publications/i/item/9789241572408) used globally, as well as U.S. EPA (2001; https://semspub.epa.gov/work/HQ/134487.pdf) and EFSA (2012; https://doi.org/10.2903/j.efsa.2012.2839) guidelines. Briefly, we considered chronic non-cancer and cancer risks of dietary pesticide intake among the general adult population in Ethiopia. We used the estimated daily intake (EDI) calculated by combining meta-analytic residue levels (251 pesticides across 18 foods, including drinking water) with Ethiopian-specific food consumption rates, adjusted for adult body weight. The food consumption data was derived from the 2018/19 socioeconomic survey panel dataset (https://doi.org/10.48529/k739-c548), where a total of 20,932 unique survey results from 7,527 households were used to calculate national mean and standard deviation food consumption rates. We also used assumed body weight with a mean of 60 kg and SD of 6 kg, bounded between 30 and 120 kg.

Estimated Daily Intake (EDI):

$$\text{EDI}_{i,j} = \frac{C_{i,j} \times CR_{i,j}}{BW}$$

where:

- $C_{i,j}$: Concentration of pesticide j in food item i (mg/kg)
- $CR_{i,j}$: Consumption rate of food item i (kg/day)

- BW: Body weight of the individual (kg)

As part of risk characterization, we used hazard quotient (HQ) for non-cancer risks to compare the EDI with a chronic Toxicological Reference Value (TRVs; 76 pesticides) derived from the acceptable daily intake (ADI) or reference dose (RfD) retrieved from EFSA (https://food.ec.europa.eu/plants/pesticides/eu-pesticides-database_en) or IRIS (https://www.epa.gov/iris) databases, respectively, with the lowest value selected conservatively. Missing TRVs (4 pesticides) were retrieved from Pesticide Properties DataBase (https://

sitem.herts.ac.uk/aeru/ppdb/en/index.htm). For cancer risks, we used the lifetime cancer risk (LCR) by multiplying the EDI by the oral slope factor (OSF; 23 pesticides) derived from IRIS database.

Hazard Quotient (HQ):

$$HQ = \frac{\text{EDI}_{i,j}}{TRV}$$

Lifetime Cancer Risk (LCR):

$$LCR_i = \text{EDI}_{i,j} \times OSF$$

where:

- TRV: Chronic oral toxicity reference value for non-cancer effects (mg/kg/day)
- OSF: Oral slope factor for cancer potency (mg/kg/day ^{1})

A Monte Carlo simulation (n=10,000 iterations) combined with Latin Hypercube Sampling (LHS) was employed to robustly account for the inherent variability and uncertainty in the input parameters using `mc2d` and `lhs` packages. In each simulation iteration, values were sampled from the defined probability distributions for pesticide concentration, food consumption, and body weight to compute EDI, HQ, and LCR based on log-normal parameters derived as below. From the resulting simulated distributions, risk percentiles (5th, 50th, 95th) were extracted and exceedance rates were summarized using cumulative distribution functions (CDFs) plots. Furthermore, exceedance probabilities were calculated to quantify the likelihood that estimated risks surpass unacceptable thresholds. HQ values greater than 1 indicate a potential for non-cancer health effects, while LCR values greater than (10^{-4}) (1 in 10,000) are considered a potential public health concern per U.S. EPA risk benchmarks.

$$\mu_{\log} = \log\left(\frac{\mu^2}{\sqrt{\sigma^2 + \mu^2}}\right), \quad \sigma_{\log} = \sqrt{\log\left(1 + \frac{\sigma^2}{\mu^2}\right)}$$

where:

- $\mu_{\log}$: The mean of the natural logarithm of the distribution (log-mean).
- $\sigma_{\log}$: The standard deviation of the natural logarithm of the distribution (log-standard deviation).
- μ: The arithmetic mean of the original (untransformed) data.
- σ: The arithmetic standard deviation of the original (untransformed) data.

Additionally, we estimated the relative food contribution to total EDI to identify major dietary sources of pesticide exposure as below.

$$P_{i,j} = \frac{\text{EDI}_{i,j}}{\text{EDI}_j}$$

where: - $P_{i,j}$: Relative contribution of food item i to total EDI of pesticide j
- $\text{EDI}_{i,j}$: EDI from food item i
- EDI_j: Total EDI from all food items for pesticide j

We also computed hazard index (HI), the sum of individual HQs to assess the potential cumulative non-cancer risk from pesticide mixture exposures.

Hazard Index (HI):

$$HI = \sum_{j=1}^{n} HQ_{i,j}$$

where: - n: Number of pesticides considered
- $HQ_{i,j}$: Hazard quotient of pesticide j in food item i

Maximum Residue Limits (MRLs) exceedance analysis

To evaluate the regulatory compliance of foodborne pesticide residues in Ethiopia, we assessed the extent to which the pooled pesticide-food mean concentrations exceeded established MRLs. The pooled mean concentrations (g/kg) from our meta-analysis were compared against MRL values (n=208 pesticide-food pairs) from regulatory sources (EFSA, USDA). Derivation these MRLs is provided in `R_Script2`. We summarized the overall proportion of exceedances, as well as the distribution of exceedances across pesticide compounds and food commodities using the equation below. Cumulative distribution functions of the log-transformed MRL ratios were visualized to highlight subgroup-specific differences. A ratio >1 indicates exceedance!

$$\text{MRL Exceedance Ratio} = \frac{C_i}{\text{MRL}_i}$$

where:

- C_i is the pooled residue concentration for pesticide–food combination i

- MRL_i is the corresponding MRL.

2 SI Section 2: Dataset Processing and Exploration

Pesticide residue concentration data extracted from included was loaded and processed as presented below (the main dataset). Ancillary/Supplemental datasets (i.e., maximum residue level, toxicity reference value and food consumption rate datasets) were loaded and processed as presented in `R_Script2`.

Preliminary Step

Before proceeding with the next sections, let's install, update and/or load required packages

```
# Uncomment to install/update packages
# install.packages(c('here', 'readxl', 'readr',
# 'synthesisr', 'dplyr', 'tidyverse', 'purrr', 'stringr',
# 'tibble', 'ggplot2', 'ggridges', 'ggthemes', 'ggsci',
# 'ggsankey', 'ggalluvial', 'tidygraph', 'EnvStats',
# 'NADA', 'NADA2', 'survival', 'mice', 'VIM', 'lattice',
# 'metafor', 'clubSandwich', 'lme4', 'orchaRd', 'emmeans',
# 'patchwork', 'moments', 'flextable', 'mc2d', 'lhs',
# 'truncnorm', 'fitdistrplus'))

# update.packages(ask = FALSE, checkBuilt = TRUE)

# load packages
library(here)
library(readxl)
library(readr)
library(synthesisr)
```

```r
library(dplyr)
library(tidyverse)
library(purrr)
library(stringr)
library(tibble)
library(ggplot2)
library(ggridges)
library(ggthemes)
library(ggsci)
library(ggsankey)
library(ggalluvial)
library(tidygraph)
library(EnvStats)
library(NADA)
library(NADA2)
library(survival)
library(mice)
library(VIM)
library(lattice)
library(metafor)
library(clubSandwich)
library(lme4)
library(orchaRd)
library(emmeans)
library(patchwork)
library(moments)
library(flextable)
library(mc2d)
library(lhs)
library(truncnorm)
library(fitdistrplus)
```

2.1 Pesticide Residue Data Processing

The compiled residue dataset encompass pesticide concentration values in Ethiopian foods, and metadata related to sample (i.e., region, zone, food type) and analytical method (i.e., instrument, limit of detection/quantification).

```r
#load and clean residue dataset
pest_raw <- read_excel(here("Data/source_data.xlsx"), sheet = "pest_raw") %>%
  #select relevant variables
  dplyr::select(study_id, region, zone, pest_met, pest_parent, type, class,
                food, food_group, food_origin, instrument, LOD, LOQ, n_samp,
                Mean, SD, detected, DL_present) %>%
  #drop pesticide types other than insecticides, herbicides & fungicides
  filter(type != "Other") %>%
  #add unique effect size id and detection limit
  mutate(es_id = sprintf("es_%02d", 1:n()), DL = coalesce(LOD, LOQ)) %>%
  #load categorical variables as factor
  mutate_if(is.character, as.factor) %>%
  #load numerical variables as number
  mutate(Mean = as.numeric(ifelse(
```

```
    as.character(Mean) == "ND", NA, as.character(Mean))), #convert ND to NA
    SD = as.numeric(SD), n_samp = as.numeric(n_samp), LOD = as.numeric(LOD),
    LOQ = as.numeric(LOQ), DL = as.numeric(DL)) %>%
  droplevels()

#rename unique pesticide/metabolite residues
pest_raw$pesticide<- pest_raw$pest_met
```

2.2 Pesticide Residue Data Exploration

First glance at the dataset

```
# head(pest_raw)
length(levels(pest_raw$study_id))  #No. of study = 40
length(levels(pest_raw$es_id))  #No. of effect size = 2271
length(levels(pest_raw$pest_met))  #87 pesticides/metabolities
length(levels(pest_raw$pest_parent))  #63 pesticide parents
length(levels(pest_raw$type))  #3 major use types
length(levels(pest_raw$food))  #18 food subgroups
length(levels(pest_raw$food_group))  #6 main food groups
length(levels(pest_raw$zone))  #15 main sampling zones
length(levels(pest_raw$region))  #8 regions in Ethiopia
```

Sankey Diagram

The Sankey diagram (Main text, Figure 1) summarizes the distribution of key study attributes, including food group, region, pesticide use type, detection status, and analytical instrumentation.

```
sankey.dat <- pest_raw %>%
    dplyr::select(detected, DL_present, instrument, type, food_group,
        region)

sankey_data <- make_long(sankey.dat, region, type, food_group,
    detected, instrument, DL_present)

node_props <- sankey_data %>%
    group_by(x, node) %>%
    summarise(n = n(), .groups = "drop") %>%
    group_by(x) %>%
    mutate(total = sum(n), percent = round(100 * n/total, 1),
        node_label = paste0(node, " (", percent, "%)")) %>%
    ungroup()

sankey_data <- sankey_data %>%
    left_join(node_props, by = c("x", "node"))

sp <- ggplot(sankey_data, aes(x = x, next_x = next_x, node = node,
    next_node = next_node, fill = node, label = node_label)) +
    geom_sankey(flow.alpha = 1, node.color = "transparent") +
    geom_sankey_label(size = 2.3, color = "black", fill = NA,
        fontface = "bold", na.rm = TRUE, label.size = 0) + theme_sankey(base_size = 10) +
    labs(x = NULL) + theme(legend.position = "none", axis.text.x = element_text(color = "black",
```

```
     size = 9, face = "bold")) + scale_x_discrete(labels = c("Region",
     "Use Type", "Food Group", "Detected?", "Instrument", "DL Provided?"),
     position = "top")

# sp

# ggsave('Figure/Figure1_sankey_diagram.png', plot = sp,
# width = 6.5, height = 3.5, units = 'in', dpi = 800)
```

Analyzed Vs. Detected pesticides

In total, 225 unique pesticide compounds and metabolites were screened, among which 88 (39%) were detected in at least one sample.

```
pest_screened <- read_excel(here("Data/source_data.xlsx"), sheet = "all_screen") %>%
  group_by(pesticide) %>%
  summarise(n_study = n_distinct(study_id),
            n_detect = sum(detected == "Y"), #detected at least in one sample!
            .groups = "drop")

pest_detected <- pest_raw %>% group_by(pest_met) %>%
  summarise(n_ES = n(), n_sample = sum(n_samp, na.rm = TRUE),
            .groups = "drop") %>%
  mutate(pest_met = str_to_lower(str_squish(pest_met)))

pest_all <- pest_screened %>%
  left_join(pest_detected, by = c("pesticide" = "pest_met"))

#sum(pest_all$n_detect == 0)          #137 not detected
#sum((pest_all$n_sample), na.rm = TRUE) #18298 samples
```

Most commonly studied pesticide residues

```
pest_studied <- pest_raw %>%
    group_by(pest_met, type) %>%
    summarise(n_study = n_distinct(study_id), n_ES = n(), n_sample = sum(n_samp,
        na.rm = TRUE), .groups = "drop") %>%
    arrange(desc(n_study), desc(n_ES), desc(n_sample))

head(pest_studied, 10)
```

```
## # A tibble: 10 x 5
##    pest_met           type        n_study n_ES n_sample
##    <fct>              <fct>         <int> <int>    <dbl>
##  1 p,p'-DDT           Insecticide      27   235     1598
##  2 p,p'-DDE           Insecticide      26   216     1858
##  3 p,p'-DDD           Insecticide      16    68      875
##  4 aldrin             Insecticide      14    94      508
##  5 alpha-endosulfan   Insecticide      12   107      854
##  6 alpha-HCH          Insecticide      12    43      410
##  7 o,p'-DDT           Insecticide      10   104      754
##  8 heptachlor         Insecticide      10    41      456
##  9 endosulfan-sulfate Insecticide       9    62      463
## 10 trans-chlordane    Insecticide       9    61      504
```

Most commonly studied food groups

```r
food_studied <- pest_raw %>%
    group_by(food, food_group, food_origin) %>%
    summarise(n_study = n_distinct(study_id), n_ES = n(), n_sample = sum(n_samp,
        na.rm = TRUE), .groups = "drop") %>%
    arrange(desc(n_study), desc(n_ES), desc(n_sample))

head(food_studied, 10)
```

```
## # A tibble: 10 x 6
##    food          food_group food_origin  n_study  n_ES n_sample
##    <fct>         <fct>      <fct>          <int> <int>    <dbl>
##  1 Fish          Meat       Animal Orgin      13   213     4636
##  2 Khat          Others     Others             8   298     2021
##  3 Tomato        Vegetables Plant Origin       6   254     2175
##  4 Honey         Others     Others             4   368     1872
##  5 Onion         Vegetables Plant Origin       4   232     1428
##  6 Milk          Milk       Animal Orgin       3   202     1055
##  7 Cabbage       Vegetables Plant Origin       3    86     2332
##  8 Wheat         Cereals    Plant Origin       2   100      396
##  9 Potato        Vegetables Plant Origin       2    95      700
## 10 Drinking water Others    Others             2    79      180
```

2.3   Pesticide Residue Data Summary

See Main text for detailed dataset characteristics reporting. Our residue data encompass 2,271 unique measurements (effect sizes) derived from 18,298 food samples collected across Ethiopia. Legacy organochlorine pesticides, particularly DDT and its metabolites *(p,p -DDT, p,p -DDE, p,p -DDD)*, as well as hexachlorocyclohexanes *(e.g., alpha-HCH)* and endosulfans *(e.g., alpha-endosulfan, endosulfan-sulfate)*, were the most frequently detected residues. Overall, the complied pesticide residue dataset underscore both the *diversity* and *complexity* of pesticide contamination monitoring in Ethiopia's food supply and provide a robust foundation for the multilevel meta-analyses and safety assessments.

3   SI Section 3: Addressing Residue Data Challenges

3.1   Overview of Missing Data

Understanding the pattern and extent of missing data is crucial for selecting appropriate imputation strategies. Let's explore missingness across the key effect size variables.

Missingness Overview

```r
sum(is.na(pest_raw$Mean))   # ~938 non-detects (NDs)
sum(is.na(pest_raw$SD))     # ~1451 missing SDs
sum(is.na(pest_raw$DL))     # ~149 missing detection limits
sum(is.na(pest_raw$n_samp)) # ~283 missing sample sizes
```

Visualize missing data patterns

As it can be seen from Figure 2, a substantial proportion of our dataset contains missing values *(41% ~ mean, 64% ~ standard deviation, 6.5% ~ detection limit, 12.5% ~ sample size)*, indicating the need for robust methods of missing data handling.

Note that missing Mean values primarily attribute to non-detects (NDs)!

```r
pest_es <- pest_raw %>%
    dplyr::select(study_id, pesticide, food, DL, Mean, SD, n_samp)

aggr(pest_es, col = c("navyblue", "red"), numbers = TRUE, sortVars = FALSE,
    prop = FALSE, labels = names(pest_es), cex.axis = 0.7, ylab = c("Missing Data",
        "Pattern"))
```
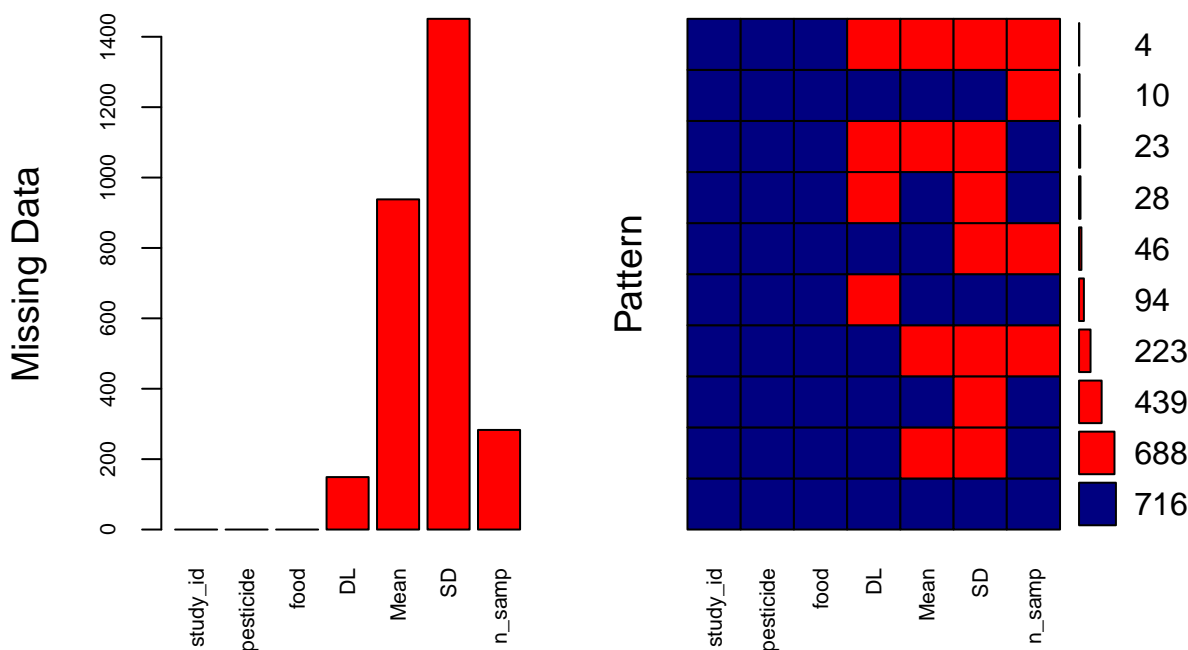

Figure 2: Missing data pattern visualization

3.2 Residue Data Distribution

Before, proceeding with missing data handling, let's explore and process our residue data characteristics. In this section, we identified and fitted the distribution of our data, and removed potential extreme outliers. We also checked the censoring structure in our data, assessing the possibility of Zero Inflation.

Pesticide residue concentration distribution

As shown in the following code chunk and Figure 3, the distribution is strongly *right-skewed and leptokurtic* with numerous *extreme outliers.* Shapiro-Wilk test further confirms *non-normality.*

```r
# summary statistics
pest_raw$Mean <- as.numeric(pest_raw$Mean)
pest_dist <- pest_raw %>%
    filter(!is.na(Mean))
dist.test <- na.omit(pest_dist$Mean)
```

```
summary(dist.test)   #Min:0, Median:13, Mean:756.5, Max:365100
skewness(dist.test)  #28.64699
kurtosis(dist.test)  #901.495
shapiro.test(dist.test)  #p-value < 2.2e-16


# visual inspection
p1 <- ggplot(pest_dist, aes(y = Mean)) + geom_boxplot()
p2 <- ggplot(pest_dist, aes(x = Mean)) + geom_histogram()
p3 <- ggplot(pest_dist, aes(sample = Mean)) + stat_qq() + stat_qq_line()


p1 | p2 | p3
```
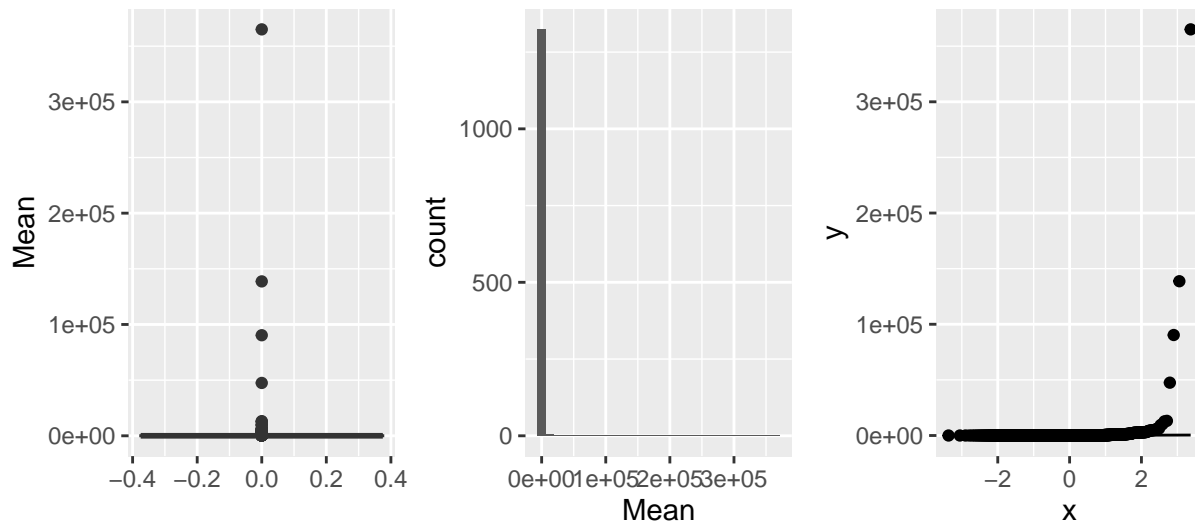

Figure 3: Boxplot, Histogram and QQ-plot showing pesticide residue concentration data

Fit distribution to observed data

The lognormal distribution provides the best fit based on AIC and other fit statistics.

```
# observed data
pest_fit <- pest_raw %>%
    filter(!is.na(Mean))
fit.test <- na.omit(pest_fit$Mean)

fit_ln <- fitdist(fit.test, "lnorm")   #best Fit
fit_gamma <- fitdist(fit.test, "gamma")
fit_weibull <- fitdist(fit.test, "weibull")
fit_exp <- fitdist(fit.test, "exp")

# goodness-of-fit statistics
fit_stats <- gofstat(list(fit_ln, fit_gamma, fit_weibull, fit_exp),
    fitnames = c("Lognormal", "Gamma", "Weibull", "Exponential"))
fit_stats


## Goodness-of-fit statistics
##                                 Lognormal      Gamma    Weibull Exponential
```

```
## Kolmogorov-Smirnov statistic 0.04791581   0.2059193   0.089792    0.6131046
## Cramer-von Mises statistic    0.67656065  22.4447938  3.164661  208.5088160
## Anderson-Darling statistic    3.83135707         Inf 19.582632         Inf
##
## Goodness-of-fit criteria
##                                 Lognormal    Gamma  Weibull Exponential
## Akaike's Information Criterion  14049.05  15085.85 14289.25    20340.07
## Bayesian Information Criterion  14059.44  15096.24 14299.64    20345.26
```

Extreme outliers removal

Our residue data was reduced to 2,099 effect sizes after extreme outliers removal, with improved stability for further analysis.

```
# removal using 3×IQR rule
Q1 <- quantile(pest_raw$Mean, 0.25, na.rm = TRUE)
Q3 <- quantile(pest_raw$Mean, 0.75, na.rm = TRUE)
IQR_val <- Q3 - Q1
lower_bound <- Q1 - 3 * IQR_val
upper_bound <- Q3 + 3 * IQR_val


pest_iqr <- pest_raw %>%
    filter((Mean >= lower_bound & Mean <= upper_bound) | is.na(Mean))
```

3.3 Impute Missing Detection Limit

The imputation of non-detects is highly dependent on the availability of detection limits (DLs). From the total missing DLs of 149, only 27 cases had both DL and Mean missing. We used simple substitution with median value to impute missing DLs, with expected minimal distortion.

```
median_dl <- median(pest_iqr$DL, na.rm = TRUE)
pest_iqr <- pest_iqr %>%
    mutate(DL = ifelse(is.na(DL), median_dl, DL))


# final dataset post-processing
pest_df <- pest_iqr %>%
    mutate(Mean = as.numeric(Mean), detected = !is.na(Mean))
```

Assess Zero Inflation and Censoring Structure

One main concern in handling non-detects *(left-censored data)* is potential zero-expansion or zero-inflation, where a high proportion of NDs might signify true zero concentrations rather than just concentrations below detection limits. We assessed if there may be zero-inflation in our residue data as below.

```
# left-censored variable use DL for ND & Mean for detects
pest_df_lc <- pest_df %>%
    mutate(lc = ifelse(detected, Mean, DL), loglc = log(lc))


# plot distribution of log-concentrations
ggplot(pest_df_lc, aes(x = loglc, fill = detected)) + geom_histogram(binwidth = 0.5,
    alpha = 0.7, position = "identity") + labs(title = "Detected vs. Non-Detected Concentrations",
    x = "Log(Concentration)", y = "") + scale_fill_manual(values = c(`TRUE` = "steelblue",
    `FALSE` = "firebrick"), labels = c(`TRUE` = "Detected", `FALSE` = "ND (~DL)")) +
    theme_minimal()
```
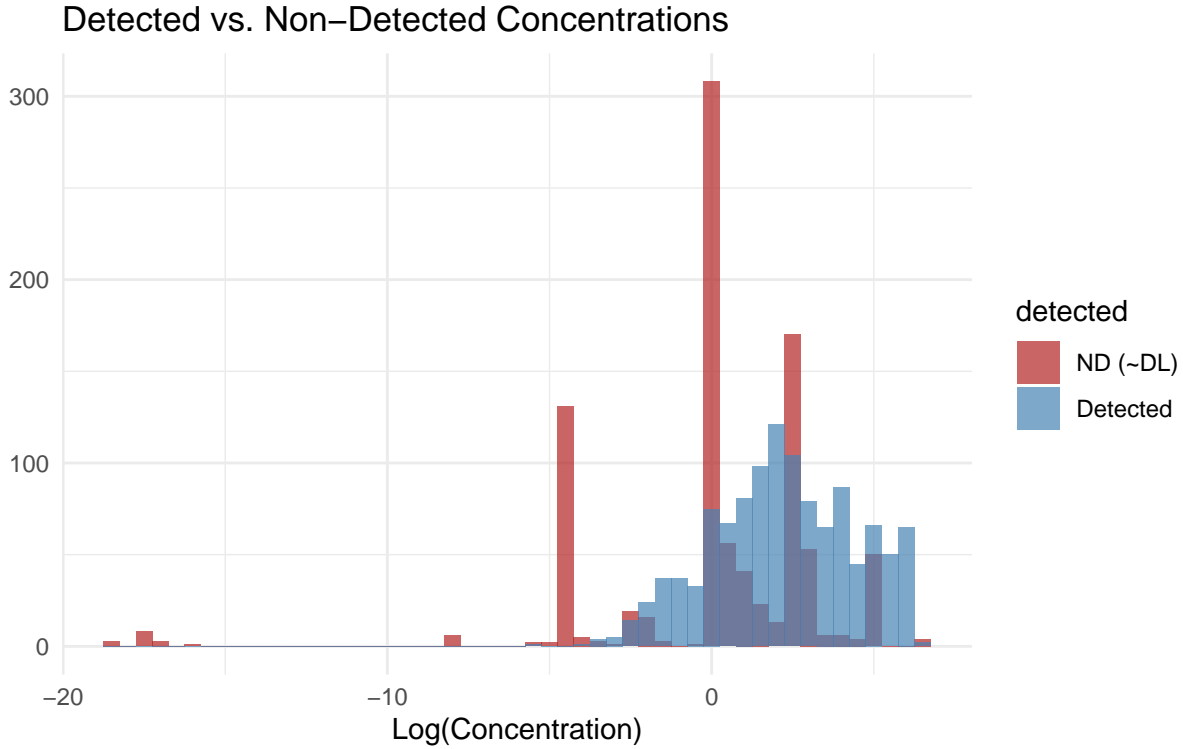
Figure 4: Visual representation of existance of zero expansion/zero-inflation

As shown above, the resulting bimodal pattern indicates a large cluster of NDs at low values and a skewed distribution of detected residues, suggesting a mixture of true zeros and censored positive concentrations. Therefore, appropriate handling of zero-inflation in our data is needed.

3.4 Imputation of Non-Detects (Left-Censored Residue Data)

For ZILN, a 50% zero-inflation rate was applied, and parameters were estimated using censored maximum likelihood via `elnormCensored`, followed by random draws for imputation. MLE was implemented using `elnormCensored` for fitting, followed by uniform random draws transformed via the inverse cumulative distribution function. For ROS, the `cenros` function from the `NADA` package was used to compute modeled values for NDs. KM imputation involved creating a survival object with `survfit`, approximating the survival function, and generating uniform random draws for NDs below the DL using the survival package.

See **SI Methods** section for detailed background and justifications.

Run ND imputation models

```
## 1.ZILN Imputation

# set parameters
ziln_fit_dat <- pest_df$Mean
ziln_fit_dat[!pest_df$detected] <- pest_df$DL[!pest_df$detected]

ziln_fit <- elnormCensored(x = ziln_fit_dat, censored = !pest_df$detected,
    censoring.side = "left", ci = FALSE)

# extract estimated log-normal parameters
```

```r
meanlog_ziln <- ziln_fit$parameters["meanlog"]
sdlog_ziln <- ziln_fit$parameters["sdlog"]

# apply zero-inflated imputation
pest_df$ZILN <- pest_df$Mean
nd_indices <- which(!pest_df$detected)

set.seed(123)
num_true_zeros <- round(length(nd_indices) * 0.5)
shuffled_nd_indices <- sample(nd_indices)

# assign zeros
indices_true_zeros <- shuffled_nd_indices[1:num_true_zeros]
pest_df$ZILN[indices_true_zeros] <- 0

# assign random lognormal values truncated at DL
indices_truncated <- shuffled_nd_indices[(num_true_zeros + 1):length(shuffled_nd_indices)]
if (length(indices_truncated) > 0) {
    nd_dls_draws <- pest_df$DL[indices_truncated]
    p_at_nd_dls_ziln <- plnorm(q = nd_dls_draws, meanlog = meanlog_ziln,
        sdlog = sdlog_ziln)
    rand_ps_truncated_ziln <- runif(n = length(indices_truncated),
        min = 0, max = p_at_nd_dls_ziln)
    random_ziln <- qlnorm(p = rand_ps_truncated_ziln, meanlog = meanlog_ziln,
        sdlog = sdlog_ziln)
    pest_df$ZILN[indices_truncated] <- random_ziln
}

## 2.MLE Imputation (truncated lognormal random draws)
MLE_fit_dat <- pest_df$Mean
MLE_fit_dat[!pest_df$detected] <- pest_df$DL[!pest_df$detected]

MLE_fit <- elnormCensored(x = MLE_fit_dat, censored = !pest_df$detected,
    censoring.side = "left", ci = FALSE)
meanlog_mle <- MLE_fit$parameters["meanlog"]
sdlog_mle <- MLE_fit$parameters["sdlog"]

nd_dls <- pest_df$DL[nd_indices]
p_at_nd_dls <- plnorm(q = nd_dls, meanlog = meanlog_mle, sdlog = sdlog_mle)
rand_ps_truncated <- runif(n = length(nd_indices), min = 0, max = p_at_nd_dls)
MLE_random <- qlnorm(p = rand_ps_truncated, meanlog = meanlog_mle,
    sdlog = sdlog_mle)
pest_df$MLE <- ifelse(pest_df$detected, pest_df$Mean, MLE_random)

## 3.ROS (Regression on Order Statistics)
obs_ros <- pest_df$Mean
cens_ros <- !pest_df$detected
cenros_out <- cenros(obs_ros, cens_ros)
pest_df$ROS <- ifelse(pest_df$detected, pest_df$Mean, cenros_out$modeled)

## 4.KM (Kaplan-Meier Imputation)
obs_km <- ifelse(pest_df$detected, pest_df$Mean, pest_df$DL)
cens_km <- !pest_df$detected
```

```r
surv_obj <- survfit(Surv(obs_km, !cens_km, type = "left") ~ 1)

S_hat <- function(t) {
    approx(x = surv_obj$time, y = surv_obj$surv, xout = t, method = "constant",
        f = 0, rule = 2)$y
}

nd_inds_km <- which(!pest_df$detected)
dl_vals_km <- obs_km[nd_inds_km]
p_upper_km <- S_hat(dl_vals_km)

set.seed(123)
u_km <- runif(n = length(nd_inds_km), min = 0, max = p_upper_km)
imps_km <- approx(x = surv_obj$surv, y = surv_obj$time, xout = u_km,
    method = "constant", f = 1, rule = 2)$y
pest_df$KM <- obs_km
pest_df$KM[nd_inds_km] <- imps_km
```

Comparison of ND imputation models

```r
## summary statistics
methods <- c("Mean", "ZILN", "MLE", "ROS", "KM")
summary_stats <- map_dfr(methods, ~{
    data.frame(Method = .x, Min = min(pest_df[[.x]], na.rm = TRUE),
        Q1 = quantile(pest_df[[.x]], 0.25, na.rm = TRUE), Median = median(pest_df[[.x]],
            na.rm = TRUE), Mean = mean(pest_df[[.x]], na.rm = TRUE),
        Q3 = quantile(pest_df[[.x]], 0.75, na.rm = TRUE), Max = max(pest_df[[.x]],
            na.rm = TRUE))
})

print(summary_stats)  #Mean = detected-only data
```

```
##          Method          Min           Q1   Median     Mean       Q3     Max
## 25%...1    Mean 5.000000e-03 2.0000000000  8.530000 56.61612 50.02000 527.00
## 25%...2    ZILN 0.000000e+00 0.0003482245  0.900000 31.56213 11.58725 527.00
## 25%...3     MLE 3.308730e-09 0.0687664732  1.175527 31.72647 12.00000 527.00
## 25%...4     ROS 4.214597e-03 0.7639752437  3.650000 35.04080 17.30000 527.00
## 25%...5      KM 1.000000e-08 2.9900000000 13.500000 72.49263 72.00000 575.25
```

```r
## visual inspection
pest_long <- pest_df %>%
    pivot_longer(cols = c(Mean, ZILN, MLE, ROS, KM), names_to = "Method",
        values_to = "Concentration") %>%
    filter(!is.na(Concentration))

ggplot(pest_long, aes(x = Concentration, color = Method)) + geom_density(linewidth = 1) +
    scale_x_log10() + scale_color_npg() + theme_bw() + labs(title = "Imputation Method Comparison",
    x = "Residue Concentration (log scale)", y = "", color = "Method") +
    theme(plot.title = element_text(size = 12, face = "bold",
        hjust = 0.5))
```

As shown in Figure 5, *ZILN and MLE* skew the distribution leftward due to conservative substitution. *ROS* provides moderate recovery of central values but still deviates from original mode. *KM* closely matches
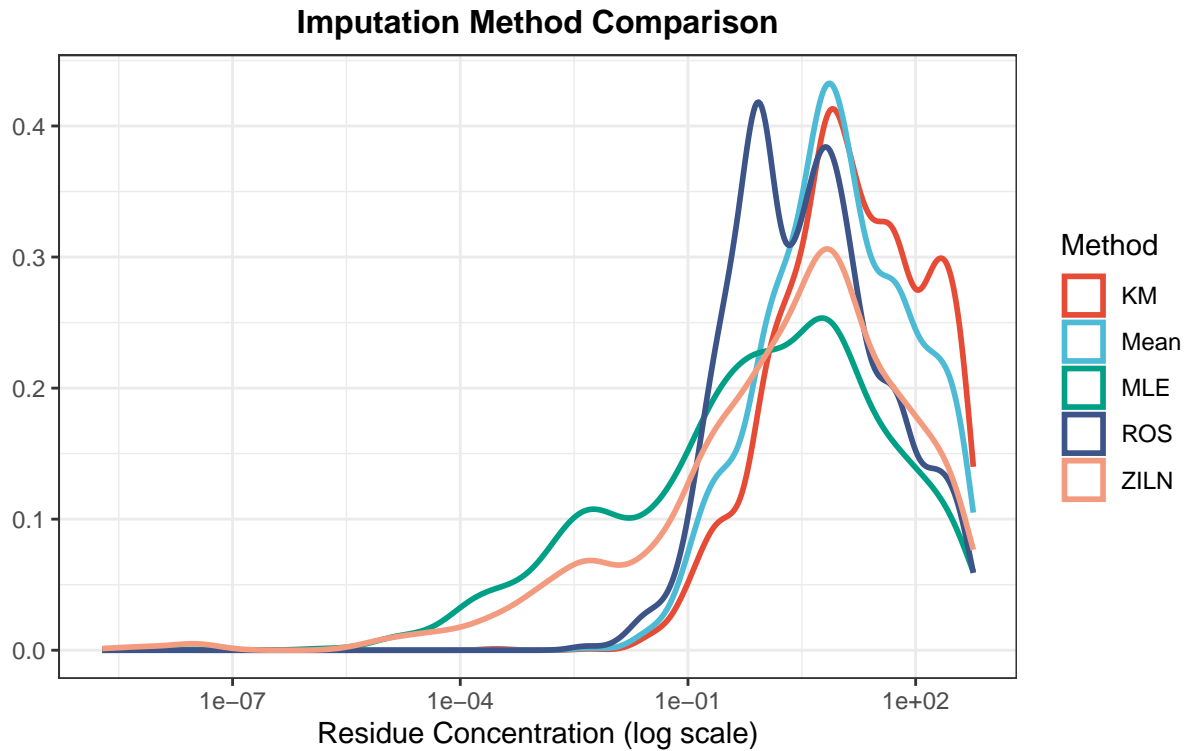
Figure 5: Kernel density curves comparing ND imputation methods (log scale)

the distribution of detected values, with higher mean and median, making it a more conservative and distribution-preserving method.

Overall, we selected and used the Kaplan–Meier (KM) imputation results given its accuracy and robustness in handling left-censored pesticide residue data.

```
pest_imp1 <- pest_df %>%
    mutate(Mean = KM) %>%
    dplyr::select(-ZILN, -MLE, -ROS, -KM)
```

3.5 Imputation of Missing SD and Sample Size

See **SI Methods** section for detailed background and justifications.

Run MICE Imputation Model

```
# select relevant variables
pest_imp_var <- pest_imp1 %>%
    dplyr::select(study_id, pesticide, food, n_samp, Mean, SD)

# initialize MICE
ini <- mice(pest_imp_var, maxit = 0, printFlag = FALSE)
meth <- ini$method
pred <- ini$predictorMatrix

# exclude non-numeric grouping variables
exclude_vars <- c("study_id", "pesticide", "food")
```

```
meth[exclude_vars] <- ""
pred[exclude_vars, ] <- 0
pred[, exclude_vars] <- 0

# define predictive model for imputation
meth["SD"] <- "pmm"
pred["SD", c("n_samp", "Mean")] <- 1
meth["n_samp"] <- "pmm"
pred["n_samp", c("SD", "Mean")] <- 1

# run MICE with 50 imputations
pest_imp <- mice(pest_imp_var, m = 50, maxit = 20, method = meth,
    predictorMatrix = pred, seed = 999, printFlag = FALSE)


## visual inspection of MICE imputation

# plot(pest_imp)
densityplot(pest_imp)
```
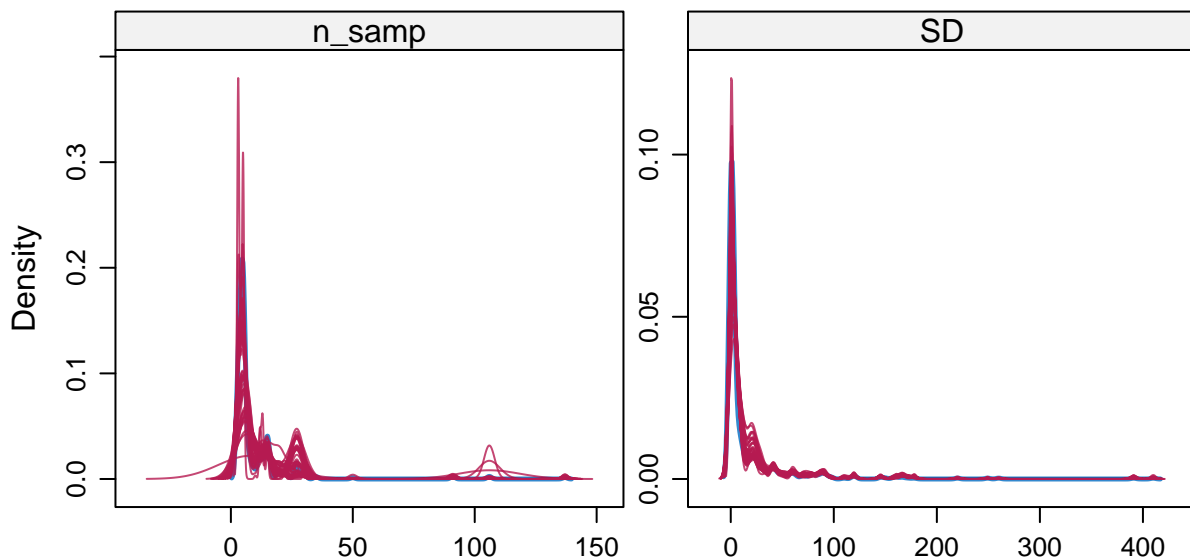

Figure 6: Density plots of MICE imputation result, indicating the imputation process did not distort the underlying data distribution

Select Best-Imputed Dataset

To retain only one representative dataset among the 50 MICE iterations, we calculated the total percentage absolute error between imputed and observed means of SD and n_samp, and selected the dataset with the smallest error (i.e., error minimization approach).

```
## Best imputation based on error minimization

# identify missing cases
SD_na <- which(is.na(pest_imp1$SD))
n_samp_na <- which(is.na(pest_imp1$n_samp))

# reference means
```

```
mean_SD <- mean(pest_imp1$SD, na.rm = TRUE)
mean_n_samp <- mean(pest_imp1$n_samp, na.rm = TRUE)

# initialize error dataframe
error_df <- data.frame(imputation = 1:pest_imp$m, SD_error = NA_real_,
    n_samp_error = NA_real_, total_error = NA_real_)

# loop through imputations
for (i in 1:pest_imp$m) {
    df_i <- complete(pest_imp, i)

    imputed_SD_vals <- df_i$SD[SD_na]
    imputed_n_vals <- df_i$n_samp[n_samp_na]

    # compute % absolute error relative to observed means
    sd_pct_error <- mean(abs(imputed_SD_vals - mean_SD)/abs(mean_SD),
        na.rm = TRUE)
    n_pct_error <- mean(abs(imputed_n_vals - mean_n_samp)/abs(mean_n_samp),
        na.rm = TRUE)

    error_df$SD_error[i] <- sd_pct_error
    error_df$n_samp_error[i] <- n_pct_error
    error_df$total_error[i] <- sd_pct_error + n_pct_error
}

# sort by total error and identify best dataset
error_df <- error_df %>%
    arrange(total_error)
head(error_df, 5)
```

```
##   imputation SD_error n_samp_error total_error
## 1         39 1.443528    0.5882705    2.031799
## 2         42 1.406540    0.6406663    2.047207
## 3          2 1.461167    0.5929966    2.054164
## 4         31 1.427469    0.6466172    2.074086
## 5         32 1.517326    0.5597208    2.077047
```

It has been shown that imputation number 39 had the smallest total error (~2%), closely matching the central tendencies of the observed data.

Attention!!!

The result above is based on the current imputation only. If another imputation were performed, the best dataset could be different. However, the fundamental concept remains the same (i.e., identify and select the imputation with the smallest total error).

Process the Final Complete Dataset

Having identified the optimal imputed dataset, we now integrate these imputed SD and n_samp values into our primary dataset and also perform final data transformations required for meta-analysis, including calculating yi (log-transformed KM-imputed Mean) and vi (its corresponding sampling variance).

```
# extract the best imputed dataset
pest_imp_best <- complete(pest_imp, action = 39)
# this could be differ next time!
```

```
# replace missing SD and n_samp in original dataset
pest_imp2 <- pest_imp1
pest_imp2$SD[SD_na] <- pest_imp_best$SD[SD_na]
pest_imp2$n_samp[n_samp_na] <- pest_imp_best$n_samp[n_samp_na]

# handle zero SD values (17 cases) to avoid log(0) errors
epsilon <- min(pest_imp2$SD[pest_imp2$SD > 0], na.rm = TRUE)
pest_imp2$SD[pest_imp2$SD == 0] <- epsilon

# apply log transformation
pest_imp2 <- pest_imp2 %>%
    mutate(log_Mean = ifelse(!is.na(Mean), log(Mean), NA), log_SD = ifelse(!is.na(SD),
        log(SD), NA), log_DL = ifelse(!is.na(DL), log(DL), NA),
        log_n_samp = ifelse(!is.na(n_samp), log(n_samp), NA))

# prepare final analysis dataset with effect sizes
pest_df_comp <- pest_imp2 %>%
    mutate(study_id = as.factor(study_id), es_id = as.factor(es_id),
        pesticide = as.factor(pesticide), food = as.factor(food),
        food_group = as.factor(food_group), yi = log_Mean, vi = (log_SD/sqrt(log_n_samp))^2)

# replace zero variances (84 cases) with smallest non-zero
# variance
epsilon2 <- min(pest_df_comp$vi[pest_df_comp$vi > 0], na.rm = TRUE)
pest_df_comp$vi[pest_df_comp$vi == 0] <- epsilon2

# final formatting: nested structure and NA removal
pest_df_comp <- pest_df_comp %>%
    mutate(es_id = interaction(study_id, es_id, drop = TRUE)) %>%
    drop_na(yi, vi, study_id, es_id, type, food_group, pesticide,
        food)
```

4 SI Section 4: Multilevel Meta-analysis Framework

This section presents a comprehensive meta-analytic framework to quantify pooled pesticide residue concentrations in foods sampled across Ethiopia and to explore heterogeneity using subgroup and moderator analyses. It also describes a specialized meta-analysis approach used to estimate pesticide residue concentrations for very specific combinations of pesticides and food types.

See **SI Methods** section for detailed background and justifications.

In summary, our three-level MMA model accounts for:

- *Level 1 (Sampling Variance):* The inherent precision (or lack thereof) of each individual data point (effect size), provided as vi.

- *Level 2 (Within-Study Variance):* Variability among pesticide data points (effect sizes) collected within the same study.

- *Level 3 (Between-Study Variance):* Variability among the true effects of different studies.

4.1 Overall and Subgroup Meta-analysis

The overall pooled mean pesticide concentration was estimated using a three-level meta-analytic model with no moderators (`Model1`). Subgroup models (`Model2.1` and `Model2.2`) introduced fixed effects for pesticide use type and food group, respectively. All models used Restricted Maximum Likelihood (REML) for variance estimation.

```r
# Model1: Overall effect (intercept only)
Model1 <- rma.mv(yi = yi, V = vi, random = list(~1 | study_id,
    ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

# Model2: Pesticide use type and food group subgroups
Model2.1 <- rma.mv(yi = yi, V = vi, mods = ~type - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

Model2.2 <- rma.mv(yi = yi, V = vi, mods = ~food_group - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

# apply RVE
Model1_rve <- robust(Model1, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)
Model2.1_rve <- robust(Model2.1, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)
Model2.2_rve <- robust(Model2.2, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)

# calculate I²
Model1_i2 <- i2_ml(Model1_rve)
Model2.1_i2 <- i2_ml(Model2.1_rve)
Model2.2_i2 <- i2_ml(Model2.2_rve)

# view summary
summary(Model1_rve)
```

```
##
## Multivariate Meta-Analysis Model (k = 2099; method: REML)
##
##      logLik    Deviance         AIC         BIC        AICc
## -4350.3054   8700.6107   8706.6107   8723.5569   8706.6222
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed     factor
## sigma^2.1  2.0311  1.4252     40     no   study_id
## sigma^2.2  1.8111  1.3458   2099     no      es_id
##
## Test for Heterogeneity:
## Q(df = 2098) = 18791062.2603, p-val < .0001
##
## Number of estimates:   2099
## Number of clusters:    40
## Estimates per cluster: 2-296 (mean: 52.48, median: 26)
##
## Model Results:
```

23

```
## 
## estimate       se¹      tval¹       df¹      pval¹     ci.lb¹     ci.ub¹
##   1.9835    0.2382    8.3285    38.27    <.0001    1.5015    2.4656    ***
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 1) results based on cluster-robust inference (var-cov estimator: CR2,
##    approx t-test and confidence interval, df: Satterthwaite approx)
```

summary(Model2.1_rve)

```
## 
## Multivariate Meta-Analysis Model (k = 2099; method: REML)
## 
##      logLik    Deviance         AIC         BIC        AICc
## -4335.7817   8671.5635   8681.5635   8709.8024   8681.5922
## 
## Variance Components:
## 
##              estim     sqrt   nlvls   fixed     factor
## sigma^2.1   2.0614   1.4358      40      no   study_id
## sigma^2.2   1.7865   1.3366    2099      no      es_id
## 
## Test for Residual Heterogeneity:
## QE(df = 2096) = 17974308.2394, p-val < .0001
## 
## Number of estimates:   2099
## Number of clusters:    40
## Estimates per cluster: 2-296 (mean: 52.48, median: 26)
## 
## Test of Moderators (coefficients 1:3):¹
## F(df1 = 3, df2 = 2.97) = 14.0945, p-val = 0.0291
## 
## Model Results:
## 
##                   estimate       se¹      tval¹       df¹      pval¹     ci.lb¹     ci.ub¹
## typeFungicide       2.7365    0.8330    3.2852      8.45    0.0103    0.8334    4.6395
## typeHerbicide       2.8033    0.9145    3.0654      7.36    0.0171    0.6622    4.9444
## typeInsecticide     1.9441    0.2442    7.9603     38.28    <.0001    1.4498    2.4384
## 
## typeFungicide        *
## typeHerbicide        *
## typeInsecticide    ***
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 1) results based on cluster-robust inference (var-cov estimator: CR2,
##    approx t/F-tests and confidence intervals, df: Satterthwaite approx)
```

summary(Model2.2_rve)
```

```
## 
## Multivariate Meta-Analysis Model (k = 2099; method: REML)
## 
##      logLik    Deviance         AIC         BIC        AICc
## -4328.1977   8656.3954   8672.3954   8717.5663   8672.4645
## 
## Variance Components:
## 
##              estim    sqrt  nlvls  fixed    factor
## sigma^2.1   1.6070  1.2677     40     no  study_id
## sigma^2.2   1.8065  1.3441   2099     no     es_id
## 
## Test for Residual Heterogeneity:
## QE(df = 2093) = 11891062.3356, p-val < .0001
## 
## Number of estimates:   2099
## Number of clusters:    40
## Estimates per cluster: 2-296 (mean: 52.48, median: 26)
## 
## Test of Moderators (coefficients 1:6):¹
## F(df1 = 6, df2 = 1.51) = 4.6259, p-val = 0.2463
## 
## Model Results:
## 
##                      estimate      se¹    tval¹      df¹    pval¹    ci.lb¹
## food_groupCereals      2.3561   0.2907   8.1058     2.96   0.0041    1.4242
## food_groupFruits       1.1829   0.3970   2.9798     5.76   0.0259    0.2017
## food_groupMeat         1.3433   0.2130   6.3076    11.6   <.0001    0.8775
## food_groupMilk         2.8009   0.7741   3.6182     2      0.0689   -0.5375
## food_groupOthers       2.6837   0.3166   8.4754    14.89  <.0001    2.0084
## food_groupVegetables   1.4126   0.3573   3.9535     6.11   0.0072    0.5420
##                        ci.ub¹
## food_groupCereals      3.2880     **
## food_groupFruits       2.1641      *
## food_groupMeat         1.8091    ***
## food_groupMilk         6.1394      .
## food_groupOthers       3.3591    ***
## food_groupVegetables   2.2832     **
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 1) results based on cluster-robust inference (var-cov estimator: CR2,
##    approx t/F-tests and confidence intervals, df: Satterthwaite approx)
```

As shown in the summary above, the overall pooled log-concentration was *1.98 µg/kg (1.49, 2.46)*. Heterogeneity was substantial *(52% between-study and 48% with-in study)* and significant variability was also observed across *use type and food group*. The results confirm our hypothesis *H1!* and also highlights the need for further meta-regression analysis to identify specific predictor moderators.

4.2 Uni-Moderator Meta-Regression Analysis (Model Comparison via ML)

To identify significant moderators of heterogeneity, we used `Maximum Likelihood (ML)` estimation in Model 3. This choice was driven by the need to compare nested models via *Likelihood Ratio Tests (LRTs)*. Each unimoderator model *(i.e., instrument, food origin, region, zone, use type, food group)* was compared to the null model. A significant LRT ($p < 0.05$) indicates that the moderator explains a significant portion of the heterogeneity.

```r
# null model
Model_null <- rma.mv(yi, vi, random = list(~1 | study_id, ~1 |
    es_id), data = pest_df_comp, test = "t", method = "ML")

# Model3: ML moderator models
Model3.1 <- rma.mv(yi, vi, mods = ~instrument - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "ML")

Model3.2 <- rma.mv(yi, vi, mods = ~food_origin - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "ML")

Model3.3 <- rma.mv(yi, vi, mods = ~region - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "ML")

Model3.4 <- rma.mv(yi, vi, mods = ~zone - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "ML")

Model3.5 <- rma.mv(yi, vi, mods = ~type - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "ML")

Model3.6 <- rma.mv(yi, vi, mods = ~food_group - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "ML")

# model comparison (Residuals/ANOVA)
model_comp <- list(instrument = anova(Model3.1, Model_null),
    food_origin = anova(Model3.2, Model_null), region = anova(Model3.3,
        Model_null), zone = anova(Model3.4, Model_null), type = anova(Model3.5,
        Model_null), food_group = anova(Model3.6, Model_null))
print(model_comp)
```

```
## $instrument
##
##          df       AIC       BIC      AICc    logLik    LRT    pval                QE
## Full      7 8718.0970 8757.6415 8718.1506 -4352.0485                      15837190.2976
## Reduced   3 8713.2145 8730.1622 8713.2260 -4353.6073 3.1175 0.5384 18791062.2603
##
##
## $food_origin
##
##          df       AIC       BIC      AICc    logLik     LRT    pval
## Full      5 8705.8455 8734.0915 8705.8741 -4347.9227
## Reduced   3 8713.2145 8730.1622 8713.2260 -4353.6073 11.3690 0.0034
##                    QE
## Full      14414778.1936
## Reduced   18791062.2603
##
```

```
## 
## $region
## 
##         df      AIC      BIC     AICc    logLik     LRT    pval
## Full    10 8712.8214 8769.3136 8712.9268 -4346.4107
## Reduced  3 8713.2145 8730.1622 8713.2260 -4353.6073 14.3931 0.0446
##                   QE
## Full    13019589.7715
## Reduced 18791062.2603
## 
## 
## $zone
## 
##         df      AIC      BIC     AICc    logLik     LRT    pval
## Full    17 8697.0027 8793.0393 8697.2967 -4331.5013
## Reduced  3 8713.2145 8730.1622 8713.2260 -4353.6073 44.2119 <.0001
##                   QE
## Full    11184930.2350
## Reduced 18791062.2603
## 
## 
## $type
## 
##         df      AIC      BIC     AICc    logLik     LRT    pval
## Full     5 8694.5837 8722.8298 8694.6124 -4342.2919
## Reduced  3 8713.2145 8730.1622 8713.2260 -4353.6073 22.6308 <.0001
##                   QE
## Full    17974308.2394
## Reduced 18791062.2603
## 
## 
## $food_group
## 
##         df      AIC      BIC     AICc    logLik     LRT    pval
## Full     8 8704.2775 8749.4712 8704.3464 -4344.1387
## Reduced  3 8713.2145 8730.1622 8713.2260 -4353.6073 18.9370 0.0020
##                   QE
## Full    11891062.3356
## Reduced 18791062.2603
```

Uni-Moderator ML fitted meta-regression models showed that *food origin, sampling zone, pesticide use type, and food group* are the significant predictors/moderators of pesticide residue concentrations in Ethiopian foods. The result provide strong support for our hypothesis *H2!*

4.3   Uni-Moderator Meta-Regression Analysis (REML-Fitted Models)

The significant moderators identified from Model 3 were re-fitted using REML (Model 4) to generate final estimates of moderator effects. RVE was again applied to ensure robust inference. For each model, we estimated I^2 values to quantify heterogeneity at the study and effect size levels and R^2 values to indicate the proportion of between-study heterogeneity explained by the moderator.

```
# Model4: REML moderator models
Model4.1 <- rma.mv(yi, vi, mods = ~instrument - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

Model4.2 <- rma.mv(yi, vi, mods = ~food_origin - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

Model4.3 <- rma.mv(yi, vi, mods = ~region - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

Model4.4 <- rma.mv(yi, vi, mods = ~zone - 1, random = list(~1 |
    study_id, ~1 | es_id), data = pest_df_comp, test = "t", method = "REML")

# apply RVE
Model4.1_rve <- robust(Model4.1, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)
Model4.2_rve <- robust(Model4.2, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)
Model4.3_rve <- robust(Model4.3, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)
Model4.4_rve <- robust(Model4.4, cluster = pest_df_comp$study_id,
    clubSandwich = TRUE, adjust = TRUE)

# calculate I² and R²
Model4.1_i2 <- i2_ml(Model4.1_rve)
Model4.2_i2 <- i2_ml(Model4.2_rve)
Model4.3_i2 <- i2_ml(Model4.3_rve)
Model4.4_i2 <- i2_ml(Model4.4_rve)

Model4.1_r2 <- r2_ml(Model4.1_rve)
Model4.2_r2 <- r2_ml(Model4.2_rve)
Model4.3_r2 <- r2_ml(Model4.3_rve)
Model4.4_r2 <- r2_ml(Model4.4_rve)
Model4.5_r2 <- r2_ml(Model2.1_rve)
Model4.6_r2 <- r2_ml(Model2.2_rve)

# view summary
summary(Model4.1_rve)
summary(Model4.2_rve)
summary(Model4.3_rve)
summary(Model4.4_rve)
```

Uni-Moderator REML fitted meta-regression models showed that *Food Origin* explained a substantial 16.3% of the observed heterogeneity (R^2), followed by *food group* (7.9%), *zone* (5.6%) and *pesticide type* (1.9%). The higher concentrations were found in foods under 'Others' category (e.g., honey, khat), and in regions such as South and Southwest. Furthermore, some zones like Bench-Sheko, Gurage, and Hararge exhibited particularly elevated levels, whereas Ilu Aba Bora showed a very low concentration.

4.4 Stratified Meta-Analysis of Pesticide–Food Combinations

Instead of fitting one large, complex multilevel model for all data, we fitted *a two-level random-effects models (also known as classical random-effects models)*, many smaller, independent meta-analysis models *(n= 251 models)*, each for a unique pesticide–food pair with effect sizes greater than 2 to get a very granular

estimations. The approach uses the `rma.uni()` function from the `metafor` package in a stratified manner. This was preferable to `MMA, rma.mv()` due to small effect sizes k, where higher-level random effects may not be estimable.

```r
# create a column indicating the pesticide-food combination
pest_df_comp <- pest_df_comp %>%
    mutate(pest_food_combo = paste(pest_met, food, sep = "_"))

# split data by pesticide-food combination
combo_list <- split(pest_df_comp, pest_df_comp$pest_food_combo)

# fit rma.uni for each group
Model.uni <- map(combo_list, function(df) {
    tryCatch({
        if (nrow(df) >= 2) {
            rma.uni(yi = yi, vi = vi, data = df, test = "t",
                method = "REML")
        } else {
            NA
        }
    }, error = function(e) {
        NA
    })
})

# extract summary info from each model
summary_table <- map_df(names(Model.uni), function(name) {
    model <- Model.uni[[name]]
    if (inherits(model, "rma")) {
        tibble(pest_food_combo = name, k = model$k, estimate = model$b[1],
            se = model$se[1], ci_lb = model$ci.lb, ci_ub = model$ci.ub,
            pval = model$pval[1], tau2 = model$tau2)
    } else {
        tibble(pest_food_combo = name, k = NA, estimate = NA,
            se = NA, ci_lb = NA, ci_ub = NA, pval = NA, tau2 = NA)
    }
})
```

4.5   Extract Model Summaries for Reporting

We extracted coefficients, standard errors, p-values, and heterogeneity statistics (I^2) from the fitted RVE models. Estimates are back-transformed from the log scale to the original concentration (µg/kg) for easier interpretation, along with their 95% confidence intervals. Additionally, the number of effect sizes (k) and unique studies (N) are gathered for each subgroup.

List and label the models

```r
model_list <- list(Model1.1 = Model1_rve, Model2.1 = Model2.1_rve,
    Model2.2 = Model2.2_rve, Model4.1 = Model4.1_rve, Model4.2 = Model4.2_rve,
    Model4.3 = Model4.3_rve, Model4.4 = Model4.4_rve)

i2_list <- list(Model1.1 = Model1_i2, Model2.1 = Model2.1_i2,
    Model2.2 = Model2.2_i2, Model4.1 = Model4.1_i2, Model4.2 = Model4.2_i2,
```

```
        Model4.3 = Model4.3_i2, Model4.4 = Model4.4_i2)

r2_list <- list(Model2.1 = Model4.5_r2, Model2.2 = Model4.6_r2,
    Model4.1 = Model4.1_r2, Model4.2 = Model4.2_r2, Model4.3 = Model4.3_r2,
    Model4.4 = Model4.4_r2)

model_labels <- names(model_list)
```

Extract Estimates

```
# function to extract i2
extract_i2 <- function(model_rve, model_i2, model_label) {
    est <- broom::tidy(model_rve)
    het <- as.data.frame(model_i2)

    # repeat I² values to match number of terms
    het_full <- data.frame(i2.lvl2 = rep(het[2, 1], nrow(est)),
        i2.lvl3 = rep(het[3, 1], nrow(est)), i2.tot = rep(het[1,
            1], nrow(est)))

    bind_cols(est, het_full) %>%
        mutate(Model = model_label)
}

# extract summary list as dataframe (back transform log)
summary_list <- purrr::pmap(list(model_list, i2_list, model_labels),
    extract_i2)

summary_df <- bind_rows(summary_list) %>%
    mutate(Mean = exp(estimate), SE = std.error, LL_95 = exp(estimate -
        1.96 * std.error), UL_95 = exp(estimate + 1.96 * std.error),
        pval = round(p.value, 4), i2.lvl2 = round(i2.lvl2, 2),
        i2.lvl3 = round(i2.lvl3, 2), i2.tot = round(i2.tot, 2),
        Mean = round(Mean, 2), LL_95 = round(LL_95, 2), UL_95 = round(UL_95,
            2))
```

Process Estimates

```
# study-Level k and N per subgroup
get_k_N <- function(data, moderator_col = NULL) {
    if (is.null(moderator_col)) {
        tibble(term = "Overall", k = nrow(data), N = n_distinct(data$study_id))
    } else {
        data %>%
            group_by(.data[[moderator_col]]) %>%
            summarise(k = n(), N = n_distinct(study_id), .groups = "drop") %>%
            rename(term = .data[[moderator_col]])
    }
}

# collect k and N
kN_list <- list(Model1.1 = get_k_N(pest_df_comp), Model2.1 = get_k_N(pest_df_comp,
    "type"), Model2.2 = get_k_N(pest_df_comp, "food_group"),
```

```r
    Model4.1 = get_k_N(pest_df_comp, "instrument"), Model4.2 = get_k_N(pest_df_comp,
        "food_origin"), Model4.3 = get_k_N(pest_df_comp, "region"),
    Model4.4 = get_k_N(pest_df_comp, "zone"))

kN_combined <- bind_rows(purrr::imap(kN_list, ~mutate(.x, Model = .y))) %>%
    rename(Subgroup = term)

Table1 <- summary_df %>%
    mutate(term = as.character(term), term = ifelse(term == "intrcpt",
        "Overall", term), Model_Type = case_when(str_detect(Model,
        "^Model1") ~ "Overall Estimation", str_detect(Model,
        "^Model2") ~ "Subgroup Analysis", str_detect(Model, "^Model4") ~
        "Moderator Meta-Regression"), Moderator = case_when(Model ==
        "Model1.1" ~ "All Pesticides", Model == "Model2.1" ~
        "Use Type", Model == "Model2.2" ~ "Food Group", Model ==
        "Model4.1" ~ "Instrument", Model == "Model4.2" ~ "Food Origin",
        Model == "Model4.3" ~ "Region", Model == "Model4.4" ~
            "Zone", TRUE ~ "Unknown"), Subgroup = str_remove(term,
        "^(type|food_group|instrument|region|zone|food_origin)"),
        Subgroup = ifelse(Moderator == "All Pesticides", "Overall",
            Subgroup)) %>%
    left_join(kN_combined, by = c("Model", "Subgroup")) %>%
    dplyr::select(Model_Type, Model, Moderator, Subgroup, N,
        k, Mean, LL_95, UL_95, pval, i2.lvl2, i2.lvl3, i2.tot)

# write.csv(Table1, 'Table/Table1_Estimates_Summary.csv',
# row.names = FALSE)
```

Process Model comparison

```r
# extract model comparison results
r2_values <- sapply(r2_list, function(x) {
    if (inherits(x, "numeric"))
        round(x[1], 3) else NA
})

Table2 <- tibble::tibble(Moderator = names(model_comp), LRT = sapply(model_comp,
    function(x) round(x$LRT, 2)), pval = sapply(model_comp, function(x) round(x$pval,
    4)), R2 = sapply(r2_values, function(x) if (is.numeric(x)) round(x[1],
    3) else NA))

# write.csv(Table2, 'Table/Table2_Model_Comparison.csv',
# row.names = FALSE)
```

Process Stratified Estimates

```r
# pooled pesticide-food dataset
pest_food_df <- summary_table %>%
    separate(col = pest_food_combo, into = c("pesticide", "food"),
        sep = "_", remove = FALSE) %>%
    mutate(Mean = round(exp(estimate), 2), SE = round(exp(se),
        2), LL_95 = round(exp(ci_lb), 2), UL_95 = round(exp(ci_ub),
        2))
```

```r
# calculate total sample sizes
pest_food_df_n <- pest_df_comp %>%
    mutate(n_samp = n_samp) %>%
    group_by(pest_food_combo) %>%
    summarise(n_samp = sum(n_samp, na.rm = TRUE), .groups = "drop")

pest_food_df <- pest_food_df %>%
    left_join(pest_food_df_n, by = "pest_food_combo") %>%
    mutate(SD = round(SE * sqrt(n_samp), 2))  #Calculate SD from SE

# add pesticide and food metadata from original dataset
pest_metadata <- pest_df_comp %>%
    dplyr::select(pesticide = pest_met, type, class) %>%
    distinct()

food_metadata <- pest_df_comp %>%
    dplyr::select(food, food_origin, food_group) %>%
    distinct()

Table3 <- pest_food_df %>%
    left_join(pest_metadata, by = "pesticide") %>%
    left_join(food_metadata, by = "food") %>%
    dplyr::select(pesticide, type, class, food, food_origin,
        food_group, k, n_samp, Mean, SD) %>%
    filter(!is.na(k)) %>%
    droplevels()

# write.csv(Table3, 'Table/Table3_pest_food.csv', row.names
# = FALSE)
```

Display Tables

```r
## Table 1 (see Main text!!)

# Table1_disp <- Table1 %>% dplyr::select(-Model_Type,
# -Model, -i2.tot)

# flextable(Table1_disp) %>% set_caption('Pooled Estimates
# ( g/kg) and Heterogeneity Across Models') %>% autofit()
# %>% theme_booktabs() %>% align(align = 'left', part =
# 'all') %>% fontsize(size = 8, part = 'all')

## Table 2 (see Main text!!)

# print(Table2)

## Table 3
table3_disp <- Table3 %>%
    dplyr::select(Pesticide = pesticide, Food = food, K = k,
        Mean, SD) %>%
    arrange(desc(K), desc(Mean))

# flextable(table3_disp) %>% set_caption('Pooled
```

```
# Pesticide-Food Combination Estimates ( g/kg)') %>%
# autofit() %>% theme_booktabs() %>% align(align = 'left',
# part = 'all') %>% fontsize(size = 8, part = 'all')

head(table3_disp)
```

```
## # A tibble: 6 x 5
##   Pesticide       Food      K  Mean    SD
##   <chr>           <chr> <int> <dbl> <dbl>
## 1 p,p'-DDE        Khat     43 14.1   20
## 2 p,p'-DDT        Khat     37 19.8   18.7
## 3 p,p'-DDE        Milk     36 10.4   21.0
## 4 p,p'-DDT        Milk     34 38.8   20.4
## 5 alpha-endosulfan Milk    33  4.14  17.3
## 6 beta-endosulfan Milk     33  3.87  18.5
```

OrchaRd Plots

OrchaRd plots are powerful visualizations for meta-analysis, showing overall and subgroup-specific pooled estimates, their confidence intervals, and the individual effect sizes within each subgroup [https://doi.org/10.32942/X2QC7]. They help in intuitively understanding the magnitude and variability of effects across different moderator categories. We ploted estimates of significant residue concentration predictors.

```
op1 <- orchard_plot(Model4.2_rve, mod = "food_origin", group = "study_id",
    k = TRUE, g = TRUE, alpha = 0.7, xlab = "") + theme(axis.text.y = element_text(angle = 360,
    hjust = 1)) + theme(legend.position = "none", legend.direction = "horizontal",
    legend.title = element_text(size = 8), legend.text = element_text(size = 8))

op2 <- orchard_plot(Model2.2_rve, mod = "food_group", group = "study_id",
    k = TRUE, g = TRUE, alpha = 0.7, xlab = "") + theme(axis.text.y = element_text(angle = 360,
    hjust = 1)) + theme(legend.position = "none", legend.direction = "horizontal",
    legend.title = element_text(size = 8), legend.text = element_text(size = 8))

op3 <- orchard_plot(Model4.4_rve, mod = "zone", group = "study_id",
    k = TRUE, g = TRUE, alpha = 0.7, xlab = "") + theme(axis.text.y = element_text(angle = 360,
    hjust = 1)) + theme(legend.position = "none", legend.direction = "horizontal",
    legend.title = element_text(size = 8), legend.text = element_text(size = 8))

op4 <- orchard_plot(Model2.1_rve, mod = "type", group = "study_id",
    k = TRUE, g = TRUE, alpha = 0.7, xlab = "") + theme(axis.text.y = element_text(angle = 360,
    hjust = 1)) + theme(legend.position = "none", legend.direction = "horizontal",
    legend.title = element_text(size = 8), legend.text = element_text(size = 8))

op <- (op1/op4/op2 | op3) + plot_annotation(tag_levels = "A") &
    theme(plot.title = element_text(size = 12, face = "bold",
        hjust = 0), plot.tag = element_text(size = 10, face = "bold"))

# ggsave('Figure/Residue_Summary.png', plot = op, width =
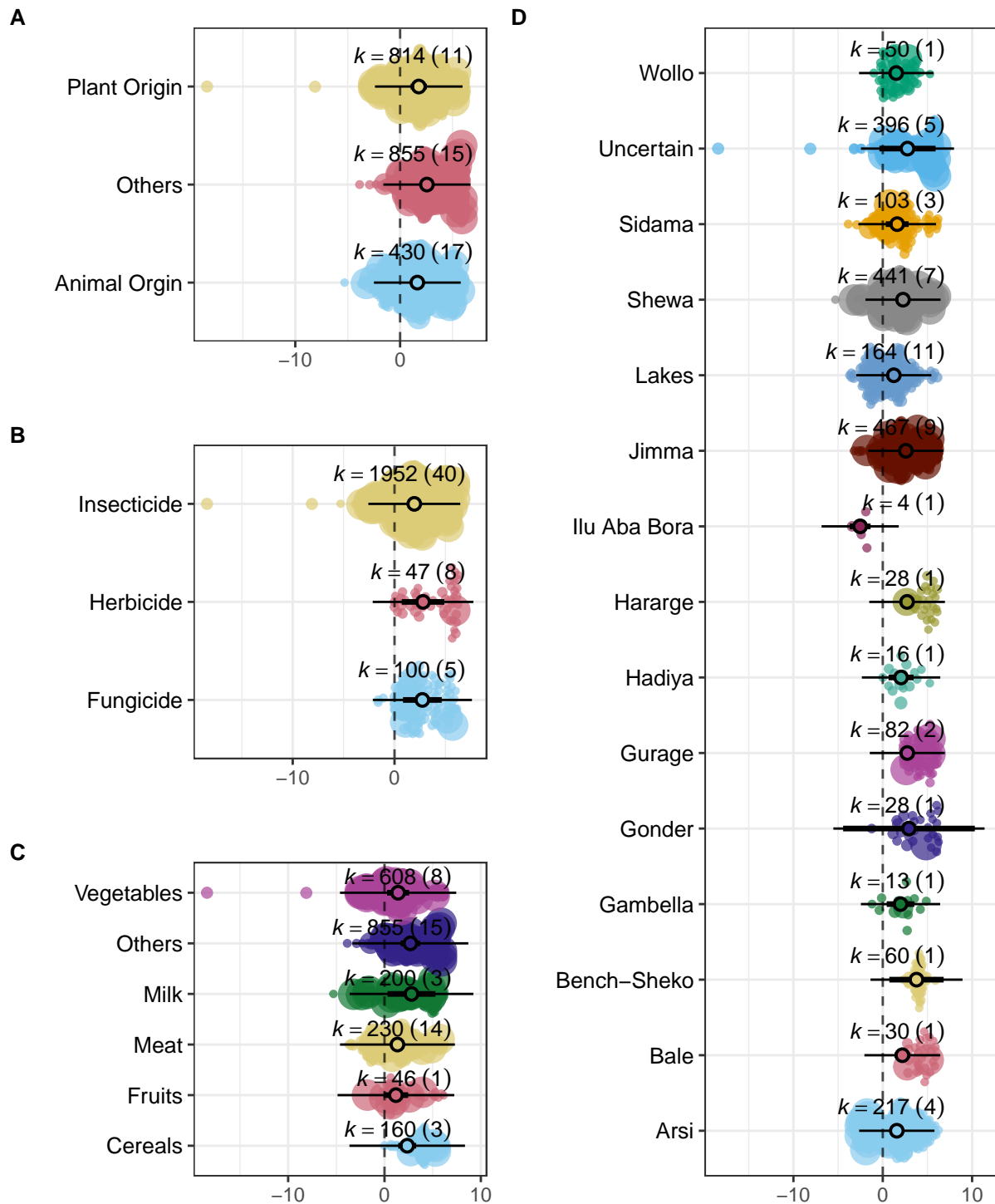# 7, height = 8, units = 'in', dpi = 800)
op
```

Figure 7: Pooled residue summary according to significant predictors (log-mean)

5 SI Section 5: Monte Carlo Probablistic Risk Assessment Framework

This section focuses on implementation of MC-PRA to estimate population-wide dietary pesticide exposure and associated chronic health risks (non-cancer/cancer). But first, the Maximum Residue Limit (MRL) exceedance analysis was conducted to assess the overall non-compliance rate of pesticide residues in Ethiopian foods.

5.1 Data Sources and Preparation

Input data for safety assessments were processed and prepared as provided in `R_Script2`.

```r
# load MRL dataset
pest_mrl <- read.csv(here("Data/inputs/pest_mrl_df.csv"))  #see R_Script2

# add MRL to pooled residue concentration
pest_mrl_df <- Table3 %>%
    mutate(pesticide = str_to_lower(str_squish(pesticide)), food = str_to_lower(str_squish(food))) %>%
    left_join(pest_mrl, by = c("pesticide", "food")) %>%
    mutate(MRL_conv = (MRL * 1000)) %>%
    filter(!is.na(MRL))

# Toxicity Reference Values and Food consumption rate
tox_dat <- read.csv(here("Data/inputs/pest_trv_df.csv"))
fc_dat <- read.csv(here("Data/inputs/fc_processed.csv"))

# prepare Residue Dataset (convert µg/kg to mg/kg)
res_dat <- Table3 %>%
    mutate(across(c(pesticide, food), ~str_to_lower(str_squish(.)))) %>%
    mutate(Ci = Mean/1000, Ci_sd = SD/1000) %>%
    left_join(tox_dat %>%
        dplyr::select(pesticide, TRV, OSF), by = "pesticide") %>%
    left_join(fc_dat %>%
        dplyr::select(food, CRi, CRi_sd), by = "food") %>%
    mutate(pesticide = str_to_title(pesticide), food = str_to_title(food))
```

5.2 Maximum Residue Limit (MRL) Exceedance Analysis

Approximately *26.5%, 55 out of 208 of pesticide-food combinations exceeded allowed MRL.* 28 out of 51 pesticides with MRL were found to be safe (below allowed MRL) and 2 pesticides (i.e., dimetachlor and heptachlor epoxide) had a 100% exceedance rate. On the other hand, all of the food groups exceeded MRL, ranging from 11% to 57%, except for foods in meat group (i.e., animal and fish meat).

```r
# compute MRL Exceedance ratio
pest_mrl_exced <- pest_mrl_df %>%
    mutate(MRL_exced = Mean/MRL_conv, exceed = MRL_exced > 1)

# what is the overall MRL exceedance rate?
mrl_overall <- pest_mrl_exced %>%
    summarise(k = n(), Above = sum(exceed), `%` = round(Above/k *
        100, 2))
print(mrl_overall)
```

```
## # A tibble: 1 x 3
##       k Above   `%`
##   <int> <int> <dbl>
## 1   208    55  26.4
```

```r
# MRL exceedance according to pesticides and food groups?
pest_mrl_stat <- pest_mrl_exced %>%
    group_by(pesticide) %>%
    summarise(k = n(), Above = sum(exceed), `%` = round(Above/k *
        100, 2)) %>%
    ungroup()

# pest_mrl_stat %>% filter(`%` == 0) %>% tally(name = 'N
# pesticides') #28 pest_mrl_stat %>% filter(`%` > 50) %>%
# tally(name = 'N pesticides') #6 pest_mrl_stat %>%
# filter(`%` == 100) %>% tally(name = 'N pesticides') #2

food_mrl_stat <- pest_mrl_exced %>%
    group_by(food_group) %>%
    summarise(k = n(), Above = sum(exceed), `%` = round(Above/k *
        100, 2)) %>%
    ungroup()

# food_mrl_stat %>% filter(`%` == 0) %>% tally(name = 'N
# pesticides') #3 food_mrl_stat %>% filter(`%` > 50) %>%
# tally(name = 'N pesticides') #5 food_mrl_stat %>%
# filter(`%` == 100) %>% tally(name = 'N pesticides') #0
```

The overall MRL excedance ratio is summarized the following Figure (see Main text, Figure 2). The Figure illustrates the overall and subgroup-specific cumulative distribution of the logarithm of the MRL exceedance ratio, with a dashed red line indicating the MRL exceedance threshold (see the main text).

```r
## Figure 2
plot_data <- pest_mrl_exced %>%
    filter(!is.na(MRL_exced), MRL_exced > 0, is.finite(log(MRL_exced)))

mrl1 <- ggplot(plot_data, aes(log(MRL_exced))) + stat_ecdf(geom = "smooth",
    size = 1) + geom_vline(xintercept = log(1), linetype = "dashed",
    color = "red", size = 0.5) + scale_color_npg() + labs(x = "",
    y = "") + theme_bw() + theme(legend.position = "none", axis.text.x = element_text(face = "bold",
    size = 6), axis.text.y = element_text(face = "bold", size = 6),
    plot.title = element_text(face = "bold", size = 9, hjust = 0.5))

mrl2 <- ggplot(plot_data, aes(x = log(MRL_exced), color = food_group,
    shape = food_group)) + stat_ecdf(geom = "point", size = 1.5,
    alpha = 0.7) + geom_vline(xintercept = log(1), linetype = "dashed",
    color = "red", size = 0.5) + scale_color_npg() + labs(x = "",
    y = "") + theme_bw() + theme(plot.title = element_text(face = "bold",
    size = 9, hjust = 0.5), axis.text.x = element_text(face = "bold",
    size = 6), axis.text.y = element_text(face = "bold", size = 6),
    legend.position = c(0.2, 0.83), legend.direction = "vertical",
    legend.title = element_blank(), legend.text = element_text(size = 6.5),
    legend.background = element_rect(fill = alpha("white", 0.5),
        color = NA))
```

```
mrl3 <- ggplot(plot_data, aes(x = log(MRL_exced), color = type,
    shape = type)) + stat_ecdf(geom = "point", size = 1.5, alpha = 0.7) +
    geom_vline(xintercept = log(1), linetype = "dashed", color = "red",
        size = 0.5) + scale_color_npg() + labs(x = "", y = "") +
    theme_bw() + theme(plot.title = element_text(face = "bold",
    size = 9, hjust = 0.5), axis.text.x = element_text(face = "bold",
    size = 6), axis.text.y = element_text(face = "bold", size = 6),
    legend.position = c(0.2, 0.8), legend.direction = "vertical",
    legend.title = element_blank(), legend.text = element_text(size = 6.5),
    legend.background = element_rect(fill = alpha("white", 0.5),
        color = NA))

mrl_p <- (mrl1/mrl3 | mrl2) + plot_annotation(tag_levels = "A") &
    theme(plot.tag = element_text(size = 10, face = "bold"))

# mrl_p #see Main text, Figure 2!

# ggsave('Figure/MRL_exceedance.png', plot = mrl_p, width =
# 6.5, height = 6, units = 'in', dpi = 800)
```

5.3   Non-Cancer and Cancer Risk Assessment

Run Probabilistic Risk Estimations

Input distribution for probabilistic estimations were defined as follows:

- C_i and CR_i were modeled using log-normal distributions, characterized by arithmetic means and standard deviations derived from meta-analytic estimates and food consumption survey data.

- BW was modeled using a truncated normal distribution ($\mu = 60$, $\sigma = 6$, truncated to [30, 120] kg).

```
# set constants
BW_mean <- 60
BW_sd <- 6
n_sim <- 10000

# helper function to get log-normal parameters
get_lognorm_params <- function(mean, sd) {
    if (is.na(mean) || is.na(sd) || mean <= 0 || sd <= 0) {
        return(list(meanlog = NA, sdlog = NA))
    }
    meanlog <- log(mean^2/sqrt(sd^2 + mean^2))
    sdlog <- sqrt(log(1 + (sd/mean)^2))
    list(meanlog = meanlog, sdlog = sdlog)
}

# functions to run risk estimates
set.seed(123)

simulate_pesticide_risk <- function(data) {
    results_list <- vector("list", length = nrow(data))
```

37

```r
    for (i in seq_len(nrow(data))) {
        row <- data[i, ]

        # get log-normal parameters
        ci_params <- get_lognorm_params(row$Ci, row$Ci_sd)
        cri_params <- get_lognorm_params(row$CRi, row$CRi_sd)

        # skip if parameters cannot be computed
        if (any(is.na(c(ci_params$meanlog, ci_params$sdlog, cri_params$meanlog,
            cri_params$sdlog)))) {
            next
        }

        # simulate samples
        Ci_samples <- rlnorm(n_sim, ci_params$meanlog, ci_params$sdlog)
        CRi_samples <- rlnorm(n_sim, cri_params$meanlog, cri_params$sdlog)
        BW_samples <- rtruncnorm(n_sim, a = 30, b = 120, mean = BW_mean,
            sd = BW_sd)

        # calculating simulated values
        EDI_samples <- (Ci_samples * CRi_samples)/BW_samples
        HQ_samples <- EDI_samples/row$TRV
        LCR_samples <- EDI_samples * row$OSF

        df <- tibble(pesticide = row$pesticide, food = row$food,
            food_group = row$food_group, sim_id = 1:n_sim, EDI_sim = EDI_samples,
            HQ_sim = HQ_samples, LCR_sim = LCR_samples)

        results_list[[i]] <- df
    }

    bind_rows(results_list)
}

# run simulation
res_risk_prob <- simulate_pesticide_risk(res_dat)
```

Results Summary

```r
risk_Table <- res_risk_prob %>%
    group_by(pesticide, food) %>%
    summarise(EDI_prob = mean(EDI_sim, na.rm = TRUE), HQ_p5 = quantile(HQ_sim,
        0.05, na.rm = TRUE), HQ_p50 = quantile(HQ_sim, 0.5, na.rm = TRUE),
        HQ_p95 = quantile(HQ_sim, 0.95, na.rm = TRUE), HQ_exceed = mean(HQ_sim >
            1, na.rm = TRUE) * 100, LCR_p5 = quantile(LCR_sim,
            0.05, na.rm = TRUE), LCR_p50 = quantile(LCR_sim,
            0.5, na.rm = TRUE), LCR_p95 = quantile(LCR_sim, 0.95,
            na.rm = TRUE), LCR_exceed = mean(LCR_sim > 1e-04,
            na.rm = TRUE) * 100, .groups = "drop")

# write.csv(risk_Table, 'Table/Table_risk_summary1.csv',
# row.names = FALSE)
```

The cumulative distribution functions for HQ and LCR, along with the percentage of HQ and LCR ex-

ceedance rates aggregated by pesticide and food categories were summarized in the following Figure (see Main text, Figure 3).

```r
# Figure 3
risk_plot1 <- res_risk_prob %>%
    filter(!is.na(HQ_sim))
risk_plot2 <- res_risk_prob %>%
    filter(!is.na(LCR_sim))
risk_pest1 <- risk_plot1 %>%
    group_by(pesticide) %>%
    summarise(HQ_exceed = mean(HQ_sim > 1) * 100) %>%
    filter(HQ_exceed >= 1) %>%
    ungroup()
risk_pest2 <- risk_plot2 %>%
    group_by(pesticide) %>%
    summarise(LCR_exceed = mean(LCR_sim > 1e-04) * 100, .groups = "drop")
risk_food1 <- risk_plot1 %>%
    group_by(food) %>%
    summarise(HQ_exceed = mean(HQ_sim > 1) * 100, .groups = "drop")
risk_food2 <- risk_plot2 %>%
    group_by(food) %>%
    summarise(LCR_exceed = mean(LCR_sim > 1e-04) * 100, .groups = "drop")

pr1 <- ggplot(risk_plot1, aes(x = HQ_sim)) + stat_ecdf(geom = "smooth",
    color = "blue", size = 1) + geom_vline(xintercept = 1, linetype = "dashed",
    color = "red") + scale_x_log10() + labs(x = "", y = "") +
    theme_bw()

pr2 <- ggplot(risk_plot2, aes(x = LCR_sim)) + stat_ecdf(geom = "smooth",
    color = "darkgreen", size = 1) + geom_vline(xintercept = 1e-04,
    linetype = "dashed", color = "red") + scale_x_log10() + labs(x = "",
    y = "") + theme_bw()

p_hq <- ggplot(risk_pest1, aes(x = reorder(pesticide, HQ_exceed),
    y = HQ_exceed)) + geom_col(fill = "steelblue") + coord_flip() +
    labs(x = "", y = "") + theme_bw()

p_lcr <- ggplot(risk_pest2, aes(x = reorder(pesticide, LCR_exceed),
    y = LCR_exceed)) + geom_col(fill = "lightgreen") + coord_flip() +
    labs(x = "", y = "") + theme_bw()

f_hq <- ggplot(risk_food1, aes(x = reorder(food, HQ_exceed),
    y = HQ_exceed)) + geom_col(fill = "steelblue") + coord_flip() +
    labs(x = "", y = "") + theme_bw()

f_lcr <- ggplot(risk_food2, aes(x = reorder(food, LCR_exceed),
    y = LCR_exceed)) + geom_col(fill = "lightgreen") + coord_flip() +
    labs(x = "", y = "") + theme_bw()

pr <- pr1 | pr2
prp <- pr1 | p_hq | f_hq
prf <- pr2 | p_lcr | f_lcr
rplt <- (prp/prf) + plot_annotation(tag_levels = "A") & theme(plot.tag = element_text(size = 10,
    face = "bold"))
```

```
# rplt #see Main Text, Figure 3!

# ggsave('Figure/Risk_summary.png', plot = rplt, width =
# 10, height = 11, units = 'in', dpi = 800)
```

Cumulative risks

Beyond individual pesticide-food combination risks, it is crucial to assess the cumulative health risks from exposure to multiple pesticides and to understand the contributions of different food sources to this overall exposure.

Relative Contributions to EDI

To understand how different food items contribute to overall pesticide exposure, we compute the relative contribution of each food to the total EDI of a pesticide:

$$P_{i,j} = \frac{\text{EDI}_{i,j}}{\text{EDI}_j}$$

where:

- $P_{i,j}$: Relative contribution of food item i to total EDI of pesticide j

- $\text{EDI}_{i,j}$: EDI from food item i

- EDI_j: Total EDI from all food items for pesticide j

```
# calculate mean EDI for each pesticide-food pair
edi_summary <- res_risk_prob %>%
    group_by(pesticide, food_group) %>%
    summarise(EDI_mean = mean(EDI_sim, na.rm = TRUE), .groups = "drop")

# calculate total EDI per pesticide
total_edi <- edi_summary %>%
    group_by(pesticide) %>%
    summarise(EDI_total = sum(EDI_mean, na.rm = TRUE), .groups = "drop")

# join and compute relative contribution
relative_contributions <- edi_summary %>%
    left_join(total_edi, by = "pesticide") %>%
    mutate(P_ij = EDI_mean/EDI_total) %>%
    arrange(desc(P_ij))

# top contributors
print(relative_contributions, 5)
```

```
## # A
## #   tibble:
## #   151
## #   x
## #   5
## # i 141
## #   more
## #   rows
```

```
## # i 5
## #   more
## #   variables:
## #   pesticide <chr>, ...
```

Hazard Index (HI)

The cumulative non-cancer risk is characterized by computing the Hazard Index (HI), which aggregates the Hazard Quotients (HQs) of multiple pesticides consumed through different foods. An HI > 1 may indicate a potential health concern.

$$HI = \sum_{j=1}^{n} HQ_{i,j}$$

where:

- n: Number of pesticides considered

- $HQ_{i,j}$: Hazard quotient of pesticide j in food item i

```
# group by simulation and pesticide
HI_per_sim <- res_risk_prob %>%
    group_by(sim_id, pesticide) %>%
    summarise(HQ_sum_per_pest_sim = sum(HQ_sim, na.rm = TRUE),
        .groups = "drop") %>%
    group_by(sim_id) %>%
    # then group by simulation ID to get total HI for that
    # sim
summarise(HI_sim_total = sum(HQ_sum_per_pest_sim, na.rm = TRUE),
    .groups = "drop")

# summarize the distribution of HI
HI_summary <- HI_per_sim %>%
    summarise(HI_mean = mean(HI_sim_total, na.rm = TRUE), HI_p5 = quantile(HI_sim_total,
        0.05, na.rm = TRUE), HI_p50 = quantile(HI_sim_total,
        0.5, na.rm = TRUE), HI_p95 = quantile(HI_sim_total, 0.95,
        na.rm = TRUE), HI_exceed_1 = mean(HI_sim_total > 1, na.rm = TRUE) *
        100, .groups = "drop")

# print(HI_summary)
```

6   Publication Bias and Sensitivity Analysis

Compare Exposure Estimates conducted with imputed dataset and missing dataset removed.

Complete case » removing NDs Use randomly selected imp dataset and average of imp dataset