Strawberry-Project

Strawberry project: analysis on insecticides and pulmonary function problem might causes

Introduction:

This is a report of the strawberry project. The main purpose for this report is trying to explain what we have done on data cleaning and analysis. We will use graphs and charts to show our work. And we will explain what method we used and why we decide to do so. Also, combining the data from professor's side and our exploration, we discovered some relationship between the exposure of a certain kind of insecticide, pyrethrin, might reduce the pulmonary function.

For this project, we are required to analysis the insecticides, both organic and conventional insecticides. Thus, in this report, other than strawberries data, we should also need some datasets about insecticides. Our assumption of the project is that insecticides might cause pulmonary disease. Thus, we should also include some disease data in our project.

The data are collected from governmental and official websites. The first data was from USDA. We obtained our strawberries and pesticide data from USDA. Also, we used the data from NHANES for the laboratory and experimental result for lung test, in the time period from 2007 to 2012. We also referred to and replicated parts of a published study, *"Urinary 3-phenoxybenzoic acid (3-PBA) concentration and pulmonary function in children: A National Health and Nutrition Examination Survey (NHANES) 2007–2012 analysis"* We tried to redo the experiment, with the data set mentioned in the paper. We used the same data from the paper to show whether our main finding is agreeing or against the paper.

In all of those datasets mentioned above, we have some variables contains NAs, which are missing values. Handling those problems are one of the major works we did for the project. Also, for more convenient for analyzing, we also separated the data set into several different cvs files. For example, like the Census and survey data. What's more, combining data into one dataset is also important. We merged several datasets into one cvs file as well, like time period and different test results. Further, we worked on data visualization for illustrating our findings. In the next body part, we will present our work more specifically.

Body part:

Setup

```
library(knitr)
library(kableExtra)
library(tidyverse)
library(stringr)
library(ggplot2)
library(rstanarm)
library(tidyr)
library(readr)
library(purrr)
library(dplyr)
library(foreign)
```

Read the data and take a first look

```
strawberry <- read_csv("strawberry_10oct25.csv", col_names = TRUE)
cat("There are 2,710 rows and 21 columns", "\n")
```

```
There are 2,710 rows and 21 columns
```

Examine the data. How is it organized?

```
## Is every line associated with a state?

state_all <- strawberry |> distinct(State)

state_all1 <- strawberry |> group_by(State) |> count()

## every row is associated with a state

if(sum(state_all1$n) == dim(strawberry)[1]){print("Yes every row in the data is associated wi
```

```
[1] "Yes every row in the data is associated with a state."
```

```
## rm(state_all, state_all1)
```

Remove columns with a single value in all rows

```
# Define a function to drop columns with only one unique value
drop_one_value_col <- function(df) {
  df |> select(where(~ n_distinct(.) > 1))
}

# Apply the function to your dataset
strawberry_removed <- drop_one_value_col(strawberry)
write.csv(strawberry_removed, "strawberry_removed.csv", row.names = FALSE)
# Check remaining columns
cat("Remaining columns:", ncol(strawberry), "\n")
```

```
Remaining columns: 21
```

```
strawberry_removed <- drop_one_value_col(strawberry)

# 1) clean "STRAWBERRIES ..." prefix
strawberry_removed <- strawberry_removed %>%
  mutate(
    `Data Item` = str_remove(
      `Data Item`,
      regex("^\\s*STRAWBERRIES\\s*[--,]*\\s*", ignore_case = TRUE)
    ),
    `Data Item` = str_squish(`Data Item`)
  )

# 2) how many comma-separated pieces exist (we will keep the first in Data Item, others beco
n_labels <- strawberry_removed %>%
  transmute(n_commas = str_count(`Data Item`, ",")) %>%
  pull(n_commas) %>%
  max(na.rm = TRUE)

# if there are no commas anywhere, still save and exit cleanly
if (is.finite(n_labels) && n_labels > 0) {
  label_names <- paste0("Data label ", seq_len(n_labels))
```

```
  # 3) split ONLY on commas; first piece overwrites Data Item, rest go to label columns
  strawberry_removed <- strawberry_removed %>%
    separate(
      col   = `Data Item`,
      into  = c("Data Item", label_names),
      sep   = "\\s*,\\s*",
      fill  = "right",
      extra = "merge",
      remove = TRUE
    ) %>%
    mutate(
      across(c("Data Item", all_of(label_names)),
             ~ ifelse(is.na(.x), NA_character_, str_squish(.x)))
    )
}

# 4) save + quick check
write.csv(strawberry_removed, "strawberry_removed.csv", row.names = FALSE)
cat("Remaining columns:", ncol(strawberry_removed), "\n")
```

```
Remaining columns: 11
```

```
if (exists("label_names")) cat("New label columns:", paste(label_names, collapse = ", "), "\n
```

```
New label columns: Data label 1, Data label 2
```

Split Census and Survey

According to the instructions from teaching assistant, we decide to split the dataset into two, one for Census data, and one for Survey data.

```
norm_prog <- function(x) str_trim(str_to_lower(x))

# 1) Split into two datasets
strawberry_census <- strawberry_removed %>%
  filter(norm_prog(Program) == "census")

strawberry_survey <- strawberry_removed %>%
  filter(norm_prog(Program) == "survey")
```

```r
# 2) Sanity checks
cat("Total rows in cleaned data: ", nrow(strawberry_removed), "\n")
```

```
Total rows in cleaned data:  2710
```

```r
cat("Rows in CENSUS: ", nrow(strawberry_census), "\n")
```

```
Rows in CENSUS:  168
```

```r
cat("Rows in SURVEY: ", nrow(strawberry_survey), "\n")
```

```
Rows in SURVEY:  2542
```

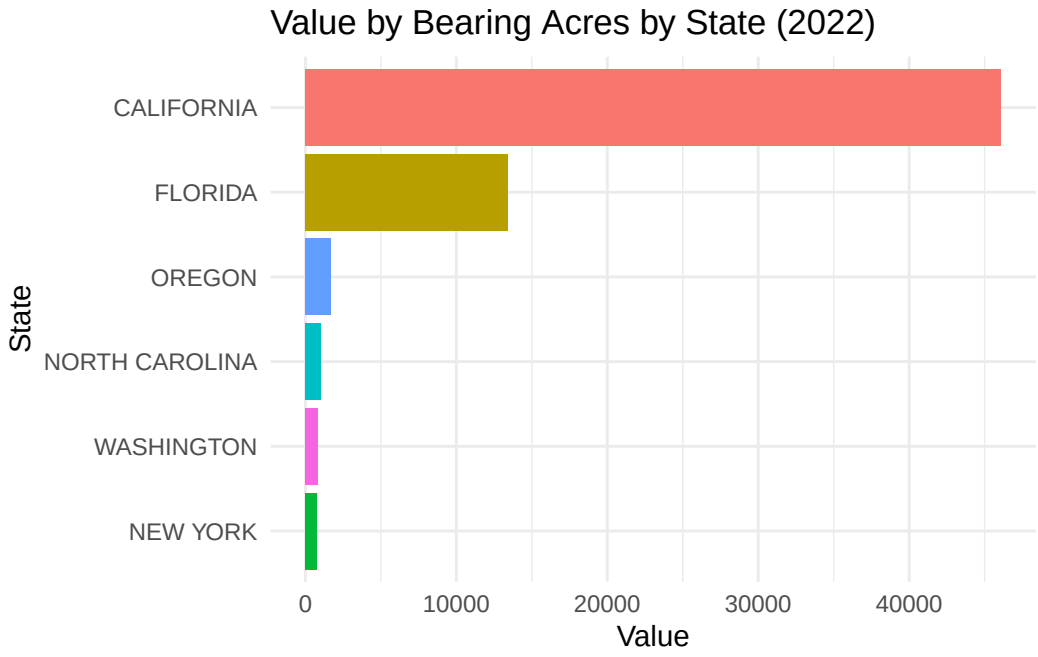
```r
# If there are unexpected/other values in Program, list them:
other_programs <- strawberry_removed %>%
  filter(!norm_prog(Program) %in% c("census", "survey") | is.na(Program)) %>%
  distinct(Program)

if (nrow(other_programs) > 0) {
  cat("Found other Program values (not 'Census'/'Survey'):\n")
  print(other_programs)
}

# 3) Save to disk
write.csv(strawberry_census, "strawberry_census.csv", row.names = FALSE)
write.csv(strawberry_survey, "strawberry_survey.csv", row.names = FALSE)
```

Examine the relationship between Year-State-Variable "Acres Bearing"-Value-CV% in Census

In this section, we want to look at the differences between states, in Census data, on the variable: Acres Bearing. This is referring to the the number of acres of land where strawberries are produced. And here we are looking at two different years, which is 2022 and 2017.

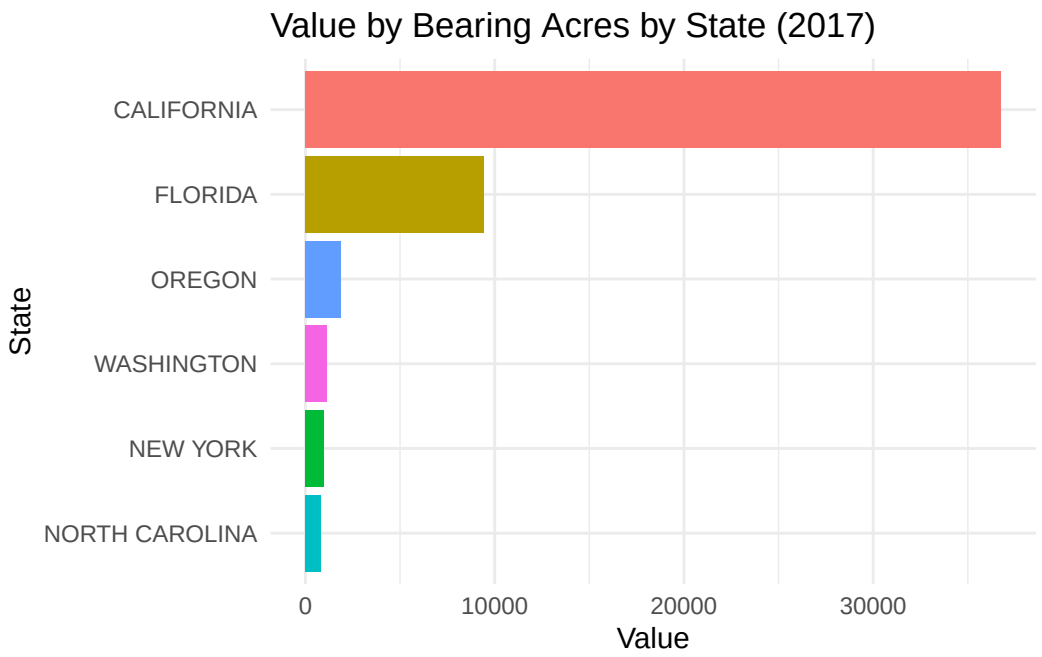
```r
df_ST <- read.csv("strawberry_census.csv")
y2022 <- df_ST %>%
  filter(Year == 2022, Data.Item == "ACRES BEARING") %>%
  mutate(Value_num = parse_number(Value)) %>%
```

```r
  filter(!is.na(Value_num))

# Summarize total (or mean) value by state
state_summary2022 <- y2022 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary2022, aes(x = reorder(State, total_acres), y = total_acres, fill = State)
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Bearing Acres by State (2022)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```




```r
#Year 2017 vs Acres Bearing
y2017 <- df_ST %>%
  filter(Year == 2017, Data.Item == "ACRES BEARING") %>%
```

```r
  mutate(Value_num = parse_number(Value)) %>%
  filter(!is.na(Value_num))

# Summarize total (or mean) value by state
state_summary2017 <- y2017 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary2017, aes(x = reorder(State, total_acres), y = total_acres, fill = State)
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Bearing Acres by State (2017)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```



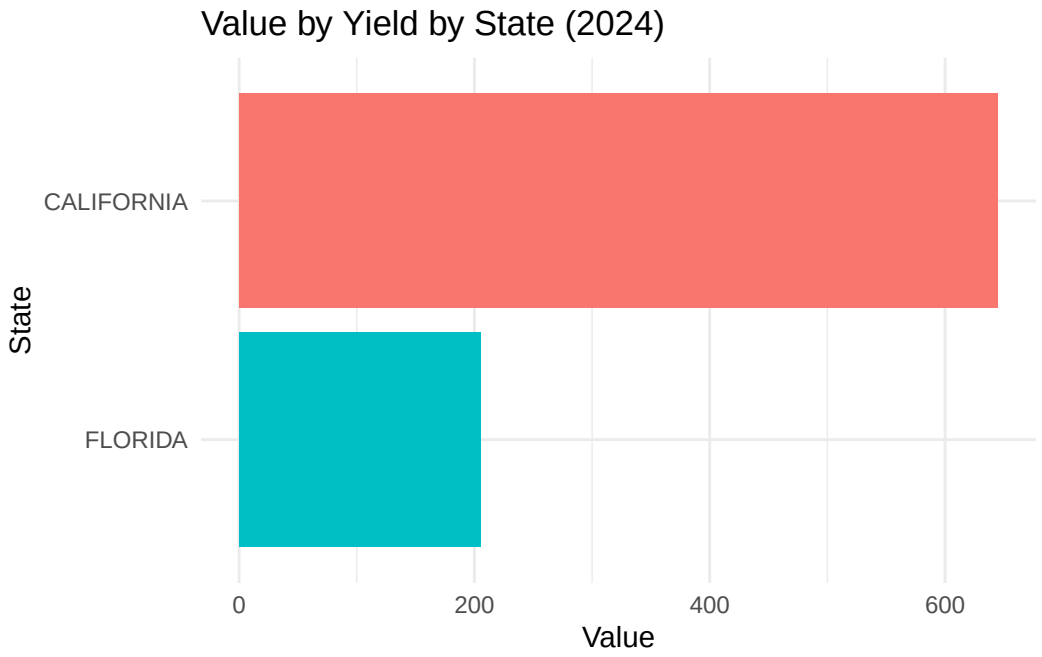
From the plots above, we can easily find out the amount all six states, California is have the most Acres Bearing. Also, if we compare two time periods, we can find that the value of Acres

Bearing had increased during these 5 years (from 2017 to 2022). Then, we can find that North Carolina also increased their value on Acres Bearing from the sixth to the third.

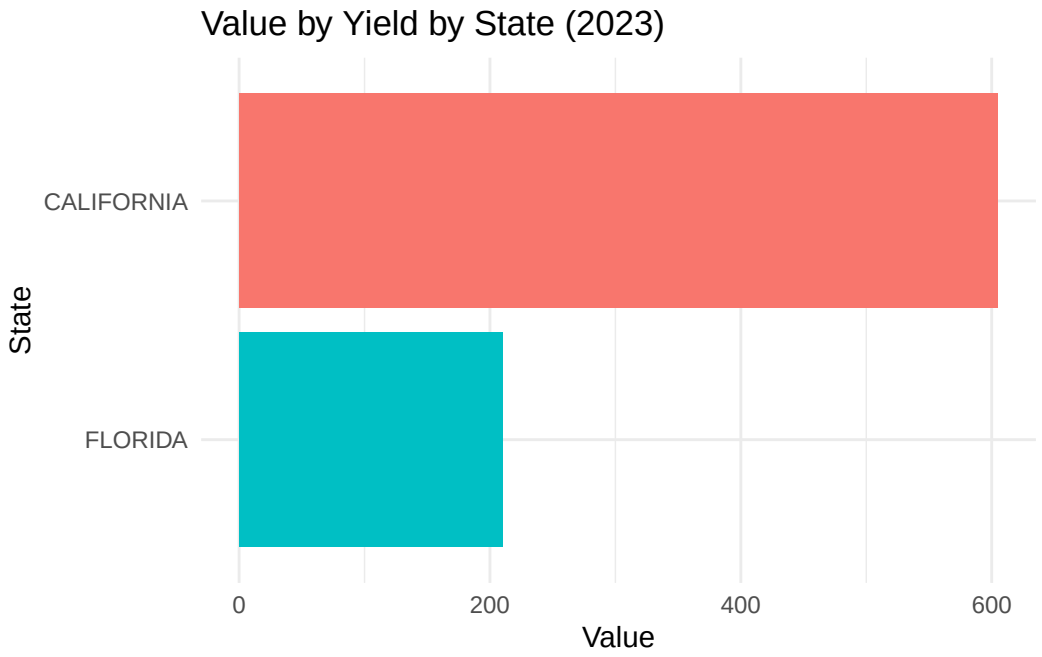
Examine the relationship between Year-State-Variable "Yield"-Value-CV% in Survey

In this part, we go to the Survey data. And we are looking at different value for analysis, which is Yield, this means the amount of strawberries harvested per unit of productive area. We mainly analyzed 5 years. In the years of 2017 and 2018, we analyzed 6 states. While in th years from 2023 to 2025, we only included 2 states, which California and Florida. This is because we find out that these two states are two of the main agriculture states and plants more strawberries.

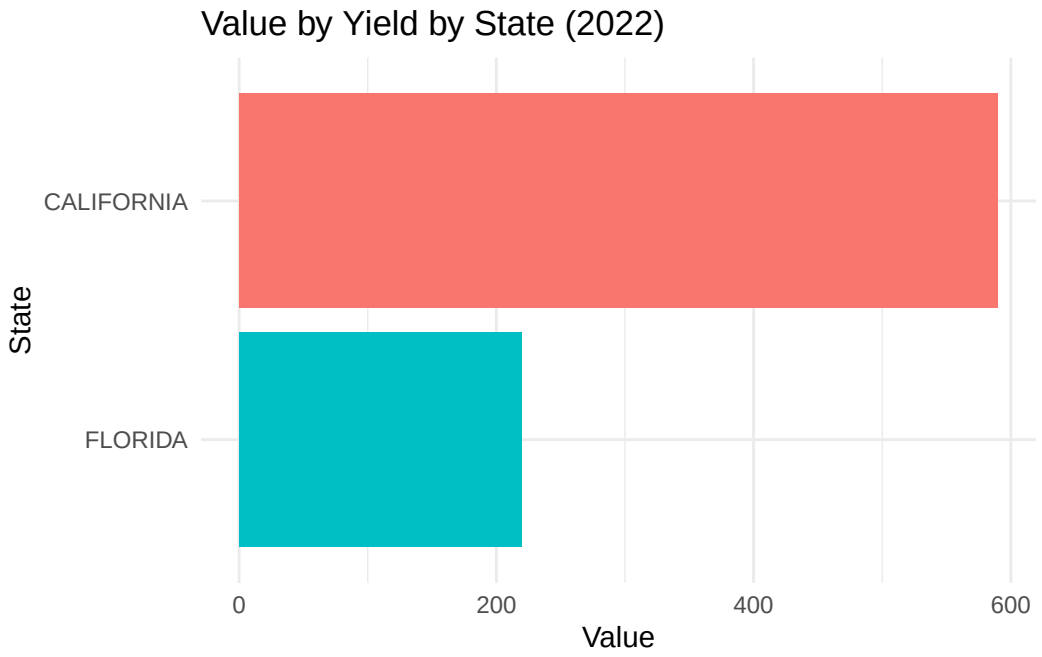
```
df_st <- read.csv("strawberry_survey.csv")

#Year 2024
ySUR_2024 <- df_st %>%
  filter(Year == 2024, Data.Item == "YIELD") %>%
  mutate(Value_num = parse_number(Value)) %>%
  filter(!is.na(Value_num))
# Summarize total (or mean) value by state
state_summary_sur_2024 <- ySUR_2024 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary_sur_2024, aes(x = reorder(State, total_acres), y = total_acres, fill = S
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Yield by State (2024)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```



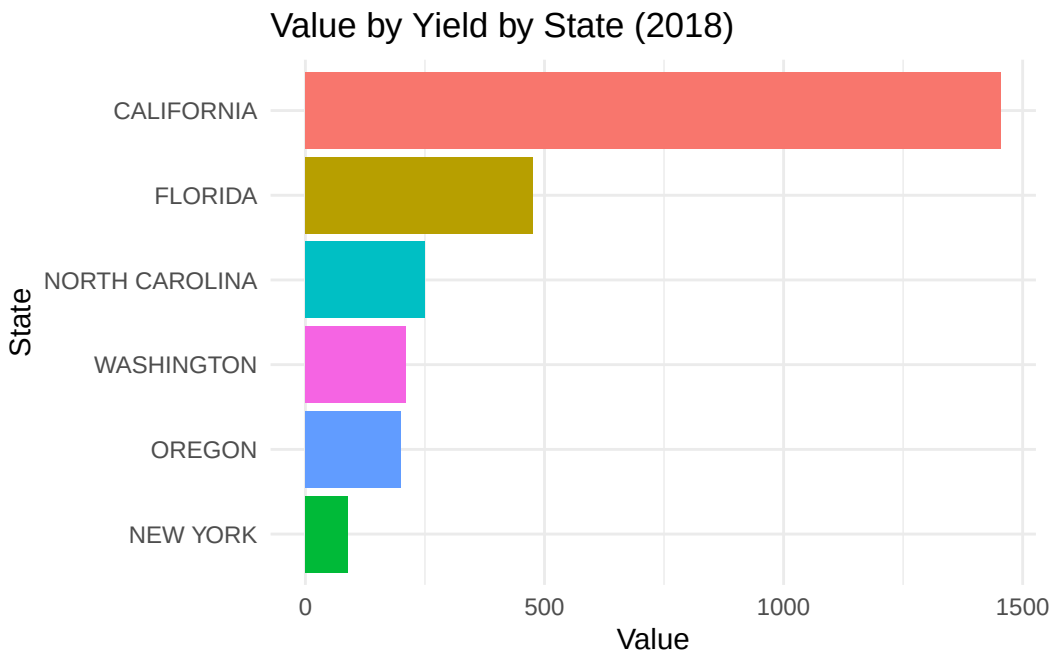

```
#Year 2023
ySUR_2023 <- df_st %>%
  filter(Year == 2023, Data.Item == "YIELD") %>%
  mutate(Value_num = parse_number(Value)) %>%
  filter(!is.na(Value_num))
# Summarize total (or mean) value by state
state_summary_sur_2023 <- ySUR_2023 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary_sur_2023, aes(x = reorder(State, total_acres), y = total_acres, fill = S
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Yield by State (2023)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```



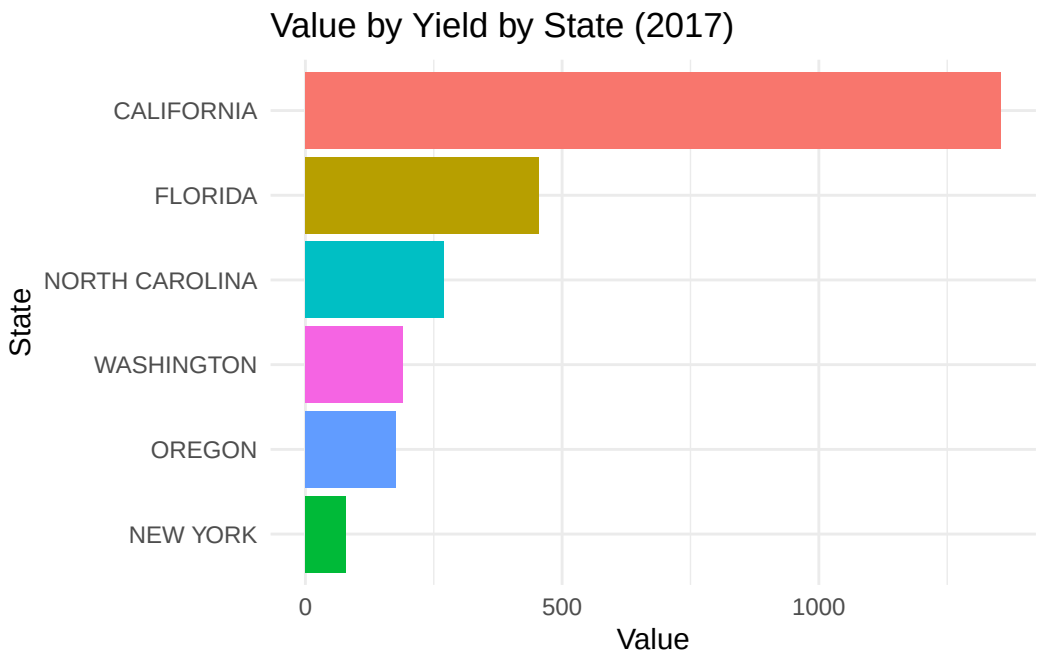

```
#Year 2022
ySUR_2022 <- df_st %>%
  filter(Year == 2022, Data.Item == "YIELD") %>%
  mutate(Value_num = parse_number(Value)) %>%
  filter(!is.na(Value_num))
# Summarize total (or mean) value by state
state_summary_sur_2022 <- ySUR_2022 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary_sur_2022, aes(x = reorder(State, total_acres), y = total_acres, fill = S
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Yield by State (2022)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```




```
#Year 2018
ySUR_2018 <- df_st %>%
  filter(Year == 2018, Data.Item == "YIELD") %>%
  mutate(Value_num = parse_number(Value)) %>%
  filter(!is.na(Value_num))
# Summarize total (or mean) value by state
state_summary_sur_2018 <- ySUR_2018 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary_sur_2018, aes(x = reorder(State, total_acres), y = total_acres, fill = S
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Yield by State (2018)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```




```r
#Year 2017
ySUR_2017 <- df_st %>%
  filter(Year == 2017, Data.Item == "YIELD") %>%
  mutate(Value_num = parse_number(Value)) %>%
  filter(!is.na(Value_num))
# Summarize total (or mean) value by state
state_summary_sur_2017 <- ySUR_2017 %>%
  group_by(State) %>%
  summarize(total_acres = sum(Value_num, na.rm = TRUE)) %>%
  arrange(desc(total_acres))

# Create barplot

ggplot(state_summary_sur_2017, aes(x = reorder(State, total_acres), y = total_acres, fill = S
  geom_col() +
  coord_flip() +  # horizontal bars (optional)
  labs(title = "Value by Yield by State (2017)",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(legend.position = "none")
```



Quick read pesticide file (raw data collected from other sources)

Here we are using our data of pesticide from NSDA. And since insecticides, the topic our group is going to analysis, is a kind of pesticide, we will introduce pesticide data, then select insecticides from them.

```
df_ins <- read.csv("raw-data.csv")
#Remove unnecessary columns, split domain column to simplify dataset
ins_removed <- df_ins |> select(-(6:12)) |> mutate(Domain = str_trim(str_split_fixed(Domain,
```

Group all insecticides in one dataset (raw data to technically correct data)

```
# 1) Filter to INSECTICIDE rows and drop (NA)/(D) in the Value column
#library(dplyr)

insect_dat <- ins_removed %>%
  filter(Domain == "INSECTICIDE") %>%
  filter(!grepl("\\(NA\\)|\\(D\\)",
                `STRAWBERRIES..BEARING...APPLICATIONS..MEASURED.IN.LB.....b.VALUE..b.`)) %>%
  # keep only text after the colon
```

```
  mutate(Domain.Category = trimws(sub("^[^:]+:\\s*", "", Domain.Category))) %>%
  # drop unwanted columns
  select(-c(4, 7, 8, 13, 15, 17, 19, 21))
```

Insecticides used by California and Florida

	A	B	C
1	Traditional Insecticides	**California**	**Florida**
2		Spinetoram	Spinetoram
3		Cyantraniliprole	Cyantraniliprole
4		Flupyradifurone	
5		Imidacloprid	
6		Dinotefuran	
7		Bifenthrin (3)	
8		Fenpropathrin	
9		Novaluron (1)	
10		Flonicamid (2)	
11	Biological and Biopesticide	Azadirachtin (4)	Azadirachtin + Pyrethrin (5)
12		Neem oil	Capsicum oleoresin, garlic oil, canola oil
13		Beauveria bassiana	Beauveria bassiana strain GHA
14		Isaria fumosorosea (Cordyceps javanica)	Cordyceps javanica (formerly Isaria fumosorosea)
15		Isaria fumosorosea strain Apopka	
16		Metarhizium anisopliae strain F52	

References:

1. https://www.nass.usda.gov/Statistics_by_State/California/Publications/Crop_Releases/Noncitrus_Fruits_and_Nuts/2024/CA_Noncitrus_2024.pdf
2. https://pmc.ncbi.nlm.nih.gov/articles/PMC9147324/
3. https://pmc.ncbi.nlm.nih.gov/articles/PMC11682952/
4. https://quickstats.nass.usda.gov/results/84BB2177-5CF7-3AC3-8DBE-2C6F9D62B5EB#FA887518-6961-3384-936D-2A58A843E8D7

Explanation:

Based on the research I conducted using the references below, California applied traditional insecticides (shown in the photo above) to protect strawberries from pests. However, only Bifenthrin, Novaluron, Flonicamid, and bioinsecticides such as Azadirachtin and Pyrethrin had measurable values in pounds, which allowed me to generate the plot. The same pattern

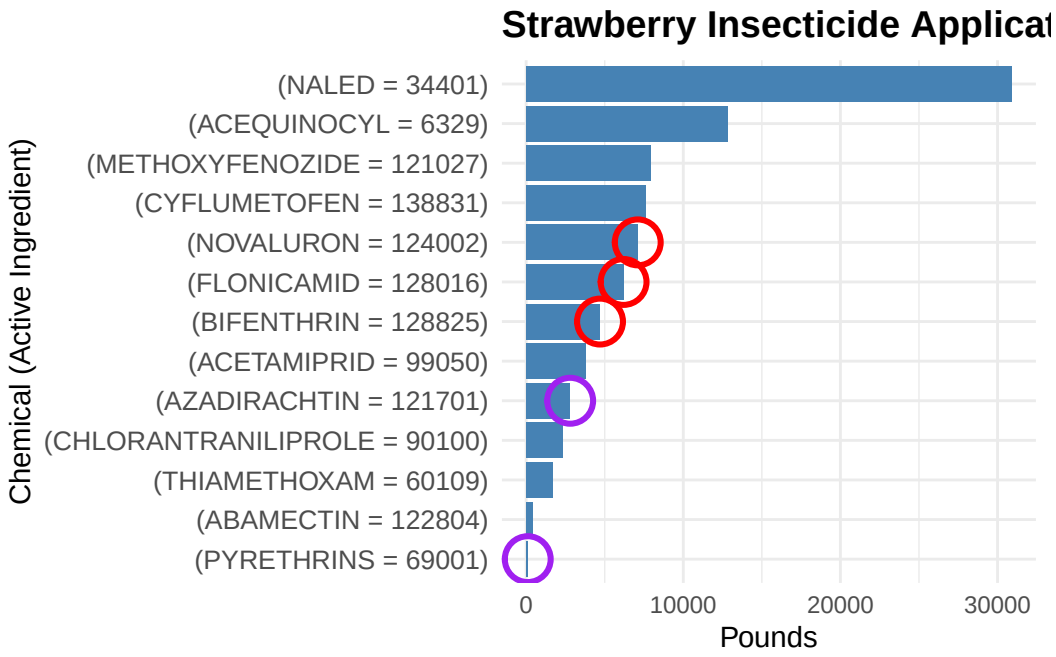
was observed in Florida. The chemicals highlighted in purple represent those with available numeric values for analysis (i.e., not NAs).

Visualization the relationship between insecticides and lbs of strawberries (technically correct data to consistent data)

After analyzing data from strawberries and insecticides, we can now combine them all together to work on the further analysis. We want to find the relationship between insecticides and how many lbs of strawberries each states can produce. Here we use the plots to help us illustrate the relationship.

```
# --- your prep (kept) ---
df_plot <- insect_dat %>%
  rename(
    Chemical = Domain.Category,
    Pounds   = `STRAWBERRIES..BEARING...APPLICATIONS..MEASURED.IN.LB.....b.VALUE..b.`
  ) %>%
  filter(Chemical != "(TOTAL)") %>%
  mutate(Pounds = as.numeric(gsub(",", "", Pounds)))

# --- 1) Define two highlight groups ---
red_targets     <- c("BIFENTHRIN", "NOVALURON", "FLONICAMID")
purple_targets  <- c("AZADIRACHTIN", "PYRETHRINS")

df_plot <- df_plot %>%
  mutate(
    chem_name = toupper(str_extract(Chemical, "[A-Za-z]+")),
    highlight_color = case_when(
      chem_name %in% red_targets ~ "red",
      chem_name %in% purple_targets ~ "purple",
      TRUE ~ NA_character_
    ),
    Chemical_f = reorder(Chemical, Pounds)
  )

# --- 2) Base plot ---
p <- ggplot(df_plot, aes(x = Chemical_f, y = Pounds)) +
  geom_col(fill = "steelblue") +

  # --- red circles ---
  geom_point(
    data = subset(df_plot, highlight_color == "red"),
```

```
    aes(x = Chemical_f, y = Pounds),
    shape = 21, size = 7, stroke = 1.6, colour = "red", fill = NA
  ) +

  # --- purple circles ---
  geom_point(
    data = subset(df_plot, highlight_color == "purple"),
    aes(x = Chemical_f, y = Pounds),
    shape = 21, size = 7, stroke = 1.6, colour = "purple", fill = NA
  ) +

  coord_flip() +
  labs(
    title = "Strawberry Insecticide Application (lbs)",
    x = "Chemical (Active Ingredient)",
    y = "Pounds"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(face = "bold", size = 14),
    axis.text.y = element_text(size = 10)
  )

p
```



Interpretation:

The chemicals circled in red represent traditional insecticides, while those circled in purple represent bioinsecticides (organic). From the chart, we can see that traditional chemical insecticides are associated with higher strawberry yields (in pounds), whereas organic insecticides result in lower yields. The photo above shows that California uses both the red (traditional) and purple (organic) insecticides, while Florida uses only the purple ones. Among all these chemicals, we will focus on Pyrethrins.

Import dataset of popular insecticides

Among those organic interstices, we want to find which one will cause more diseases. And one of the most important reason for getting illness from insecticides is the exposure problem. Thus, we will use the datasets which included different organic insecticides with their exposure data. Then, we can find which insecticide might cause health problem.

```
xc <- read.csv("pyrethrin-vs-other-insecticides.csv")
#Check how many uique insecticides there are in the dataset
unique(xc$SubCategory)
```

```
[1] "Carbamates/Organophosphates Exposures"
```

```
[2] "Organochlorines Exposures"
[3] "Pyrethroids/Pyrethrins/Piperonyl Exposures"
[4] "Boric Acid Exposures"
[5] "Other (baits & gels)/Unknown Exposures"
```

```r
top_chem <- xc %>%
  group_by(SubCategory) %>%
  summarise(mean_value = mean(Value, na.rm = TRUE),
            max_value  = max(Value, na.rm = TRUE),
            n_states   = n()) %>%
  arrange(desc(mean_value))

# Show results
print(top_chem)
```

```
# A tibble: 5 x 4
  SubCategory                                mean_value max_value n_states
  <chr>                                           <dbl>     <dbl>    <int>
1 Pyrethroids/Pyrethrins/Piperonyl Exposures      8.43      14.8        51
2 Other (baits & gels)/Unknown Exposures          4.35       8.42       51
3 Boric Acid Exposures                            2.13       7.77       51
4 Carbamates/Organophosphates Exposures           1.29       2.16       51
5 Organochlorines Exposures                       0.103      0.41       51
```

```r
# Identify the single highest row in the dataset
highest_case <- xc %>% filter(Value == max(Value, na.rm = TRUE))
print(highest_case)
```

```
  StateFIPS      State Year Value Data.Comment  X
1        35 New Mexico 2017  14.8              NA
                              SubCategory
1 Pyrethroids/Pyrethrins/Piperonyl Exposures
```

Since Pyrethroids/Pyrethrins/Piperonyl exposures have the highest value compared to other insecticides, this suggests that this group of chemicals has the highest exposure rate among all insecticides.

Based on the research paper "Urinary 3-phenoxybenzoic acid (3-PBA) concentration and pulmonary function in children: A National Health and Nutrition Examination Survey (NHANES) 2007–2012 analysis," Hu et al. (2021) found that pyrethroid exposure was associated with reduced pulmonary function in children, particularly among boys aged 11–17 years. This project aimed to reproduce the conclusions presented in that study.

We collected data of pulmonary function, Pyrethrins, and Patient's demographics from NHANES to replicate the mentioned study.

After working on the insecticides data, we need to introduce our datasets, and build the relationship between pyrethrins and pulmonary function test result to continue with our research.

Import files

```
pyr_0708 <- read.xport("UPHOPM_E_2007-2008.xpt")
write.csv(pyr_0708, file = "pyrethrin0708.csv")
pyr_0910 <- read.xport("UPHOPM_F_2009-2010.xpt")
write.csv(pyr_0910, file = "pyrethrin0910.csv")
pyr_1112 <- read.xport("UPHOPM_G_2011-2012.xpt")
write.csv(pyr_1112, file = "pyrethrin1112.csv")
lung_0708 <- read.xport("SPX_E_2007-2008.xpt")
write.csv(lung_0708, file = "lung0708.csv")
lung_0910 <- read.xport("SPX_F_2009-2010.xpt")
write.csv(lung_0910, file = "lung0910.csv")
lung_1112 <- read.xport("SPX_G_2011-2012.xpt")
write.csv(lung_1112, file = "lung1112.csv")
demo_0708 <- read.xport("DEMO_E_2007-2008.xpt")
write.csv(demo_0708, file = "demo0708.csv")
demo_0910 <- read.xport("DEMO_F_2009-2010.xpt")
write.csv(demo_0910, file = "demo0910.csv")
demo_1112 <- read.xport("DEMO_G_2011-2012.xpt")
write.csv(demo_1112, file = "demo1112.csv")
```

Setup

```
py78 <- read.csv("pyrethrin0708.csv")
lu78 <- read.csv("lung0708.csv")
de78 <- read.csv("demo0708.csv")
py78_clean <- py78 |> select(SEQN, URXOPM)
lu78_clean <- lu78 |> select(SEQN, SPXNFEV1, SPXNFVC, SPXNPEF)
de78_clean <- de78 |> select(SEQN, RIAGENDR, RIDAGEEX)
# --- 1) Make sure SEQN is the same type everywhere ---
py78_clean <- py78_clean %>% mutate(SEQN = as.integer(SEQN))
lu78_clean <- lu78_clean %>% mutate(SEQN = as.integer(SEQN))
```

```r
de78_clean <- de78_clean %>% mutate(SEQN = as.integer(SEQN))

# --- 2) Check duplicates in each file ---
dups_py <- py78_clean %>% count(SEQN) %>% filter(n > 1)
dups_lu <- lu78_clean %>% count(SEQN) %>% filter(n > 1)
dups_de <- de78_clean %>% count(SEQN) %>% filter(n > 1)
message(sprintf("Duplicates: py=%d, lu=%d, de=%d",
                nrow(dups_py), nrow(dups_lu), nrow(dups_de)))

# --- 3) Do they have the same SEQN (set) and the same order? ---
same_set_py_lu <- setequal(py78_clean$SEQN, lu78_clean$SEQN)
same_set_py_de <- setequal(py78_clean$SEQN, de78_clean$SEQN)
same_order_py_lu <- identical(py78_clean$SEQN, lu78_clean$SEQN)
same_order_py_de <- identical(py78_clean$SEQN, de78_clean$SEQN)

message(sprintf("Same SET of SEQN?  py vs lu: %s; py vs de: %s",
                same_set_py_lu, same_set_py_de))
message(sprintf("Same ORDER of SEQN? py vs lu: %s; py vs de: %s",
                same_order_py_lu, same_order_py_de))
```

Merge on the common SEQN

According to NHANES dataset's definition, SEQN is Respondent sequence number. This is a unique ID number for each patients to verify their own lab and examination result. Thus, we merge the data set by the variable SEQN.

```r
merged_78 <- py78_clean %>%
  inner_join(lu78_clean, by = "SEQN") %>%
  inner_join(de78_clean, by = "SEQN") %>%
  arrange(SEQN) %>%
  select(SEQN, URXOPM, SPXNFEV1, SPXNFVC, SPXNPEF,
         RIAGENDR, RIDAGEEX)

message(sprintf("Merged rows (in all three) = %d; columns = %d",
                nrow(merged_78), ncol(merged_78)))

write_csv(merged_78, "nhanes_0708_merged.csv")
```

Calculate means and impute them to NAs in all the column of the dataset

In our dataset, there are some variables contains NAs in the columns. And since they are numerical, we decide to calculate the means of each columns to impute and replace the NAs in the columns.

```
#Calculate the mean of all columns
means <- sapply(merged_78[, c("URXOPM", "SPXNFEV1", "SPXNFVC", "SPXNPEF", "RIDAGEEX")], mean
#Impute means to NAs in each column
for (col in names(means)) {
  merged_78[[col]][is.na(merged_78[[col]])] <- means[col]
}
write.csv(merged_78, "nhanes_0708_noNA.csv", row.names = FALSE)
```

#Create 3 exposure groups based on URXOPM (3-PBA) #Low exposure: values < LOD (below detection) #Medium exposure: detectable but < median #High exposure: detectable median

In this part, we will mainly focus on the variable URXOPM, and this is representing 3-phenoxybenzoic acid (3-PBA in short) result. This result test reflects how possible someone will have pulmonary function problem. Thus, we want to look at the exposure of 3-PBA and the possibility of getting reduces pulmonary function. Also, since we are redoing the method mentioned in the paper, it will be better for us to explain each variable in the paper. FEV1 in our code representing the forced expiratory volume in 1s; FVC means forced vital capacity; and PEF is peak expiratory flow. All of those variables mentioned above is the way we used to test someones' pulmonary function.

```
cat("Here we replicate the statistical analysis used in the research paper.", "\n")
```

```
Here we replicate the statistical analysis used in the research paper.
```

```
df <- read.csv("nhanes_0708_noNA.csv")
lod <- quantile(df$URXOPM, 0.25, na.rm = TRUE)
median_val <- median(df$URXOPM, na.rm = TRUE)
# Create 3-PBA exposure categories
df <- df |> mutate(
  URXOPM_cat = case_when(
    URXOPM < lod ~ "Low",
    URXOPM >= lod & URXOPM < median_val ~ "Medium",
    URXOPM >= median_val ~ "High"
  )
```

```
)
table(df$URXOPM_cat)
```

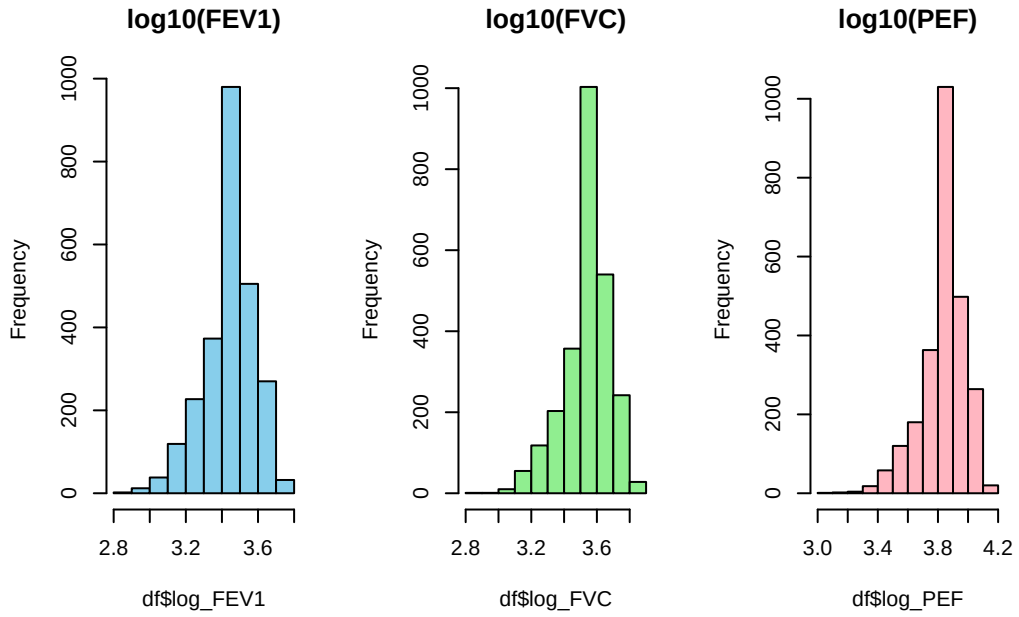
```
  High Medium
  1289   1269
```

```
if (!"URXOPM_cat" %in% names(df)) {
  # (backup: create it if it wasn't created for some reason)
  lod        <- quantile(df$URXOPM, 0.25, na.rm = TRUE)
  median_val <- median(df$URXOPM, na.rm = TRUE)
  df <- df |>
    mutate(
      URXOPM_cat = case_when(
        URXOPM >= lod & URXOPM < median_val ~ "Medium",
        URXOPM >= median_val ~ "High"
      )
    )
}

# Coerce to factor; keep a consistent ordering for plots/models
df$URXOPM_cat <- factor(df$URXOPM_cat, levels = c("Medium", "High"))

# (Optional) choose a reference for modeling; change to "Low" or "Medium" if you prefer
# df$URXOPM_cat <- relevel(df$URXOPM_cat, ref = "Medium")

# Quick diagnostic
print(table(df$URXOPM_cat, useNA = "ifany"))
```

```
Medium   High
  1269   1289
```

```
#Log10-transfor lung function variables
#Since the paper states: "FEV , FVC, and PEF were log -transformed before analysis."
df <- df |> mutate (
  log_FEV1 = log10(SPXNFEV1),
  log_FVC = log10(SPXNFVC),
  log_PEF = log10(SPXNPEF)
)
```

```r
#Check distribution
par(mfrow = c(1, 3))
hist(df$log_FEV1, main = "log10(FEV1)", col = "skyblue")
hist(df$log_FVC, main = "log10(FVC)", col = "lightgreen")
hist(df$log_PEF, main = "log10(PEF)", col = "lightpink")
```



By using the the log function, the plot here is closer to normal distribution.

```r
# --- 1) Recode predictors ---
df <- df %>%
  mutate(
    # Gender: 1 = Male, 2 = Female
    RIAGENDR_f = factor(RIAGENDR, levels = c(1, 2), labels = c("Male", "Female")),

    # Convert RIDAGEEX (months) → years
    age_years = RIDAGEEX / 12,

    # Create 3 age categories
    age_cat = case_when(
      age_years >= 6  & age_years < 11 ~ "6-11",
      age_years >= 11 & age_years <= 17 ~ "11-17",
      TRUE                              ~ "Other"
    ),
```

```
    age_cat = factor(age_cat, levels = c("6-11", "11-17", "Other"))
  )

# --- 2) Set reference levels ---
# baseline = Male and age_cat = "Other"
df$RIAGENDR_m <- relevel(df$RIAGENDR_f, ref = "Female")
df$age_cat    <- relevel(df$age_cat, ref = "Other")
```

Fit models with Male x Age interaction

```
# --- 3) Fit models with Male × Age interaction ---
model_fev1_int <- lm(log_FEV1 ~ URXOPM_cat + RIAGENDR_m * age_cat, data = df)
summary(model_fev1_int)
```

```
Call:
lm(formula = log_FEV1 ~ URXOPM_cat + RIAGENDR_m * age_cat, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56373 -0.06586  0.01107  0.07179  0.30864

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   3.406653   0.004252 801.228  < 2e-16 ***
URXOPM_catHigh                0.012573   0.004470   2.812 0.004954 **
RIAGENDR_mMale                0.110358   0.005185  21.283  < 2e-16 ***
age_cat6-11                  -0.167682   0.009519 -17.616  < 2e-16 ***
age_cat11-17                  0.034396   0.009723   3.538 0.000411 ***
RIAGENDR_mMale:age_cat6-11   -0.063018   0.013487  -4.672 3.13e-06 ***
RIAGENDR_mMale:age_cat11-17 -0.062662   0.013503  -4.641 3.65e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1129 on 2551 degrees of freedom
Multiple R-squared:  0.3532,    Adjusted R-squared:  0.3517
F-statistic: 232.2 on 6 and 2551 DF,  p-value: < 2.2e-16
```

```r
model_fvc_int <- lm(log_FVC ~ URXOPM_cat + RIAGENDR_m * age_cat, data = df)
summary(model_fvc_int)
```

```
Call:
lm(formula = log_FVC ~ URXOPM_cat + RIAGENDR_m * age_cat, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.42783 -0.06842  0.01070  0.06008  0.28345

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   3.509895   0.003904 899.104  < 2e-16 ***
URXOPM_catHigh                0.011268   0.004104   2.745  0.00609 **
RIAGENDR_mMale                0.120429   0.004761  25.296  < 2e-16 ***
age_cat6-11                  -0.199534   0.008739 -22.831  < 2e-16 ***
age_cat11-17                 -0.001806   0.008927  -0.202  0.83968
RIAGENDR_mMale:age_cat6-11   -0.066960   0.012383  -5.407 6.99e-08 ***
RIAGENDR_mMale:age_cat11-17 -0.065557   0.012397  -5.288 1.34e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1037 on 2551 degrees of freedom
Multiple R-squared:  0.4531,    Adjusted R-squared:  0.4518
F-statistic: 352.2 on 6 and 2551 DF,  p-value: < 2.2e-16
```

```r
model_pef_int <- lm(log_PEF ~ URXOPM_cat + RIAGENDR_m * age_cat, data = df)
summary(model_pef_int)
```

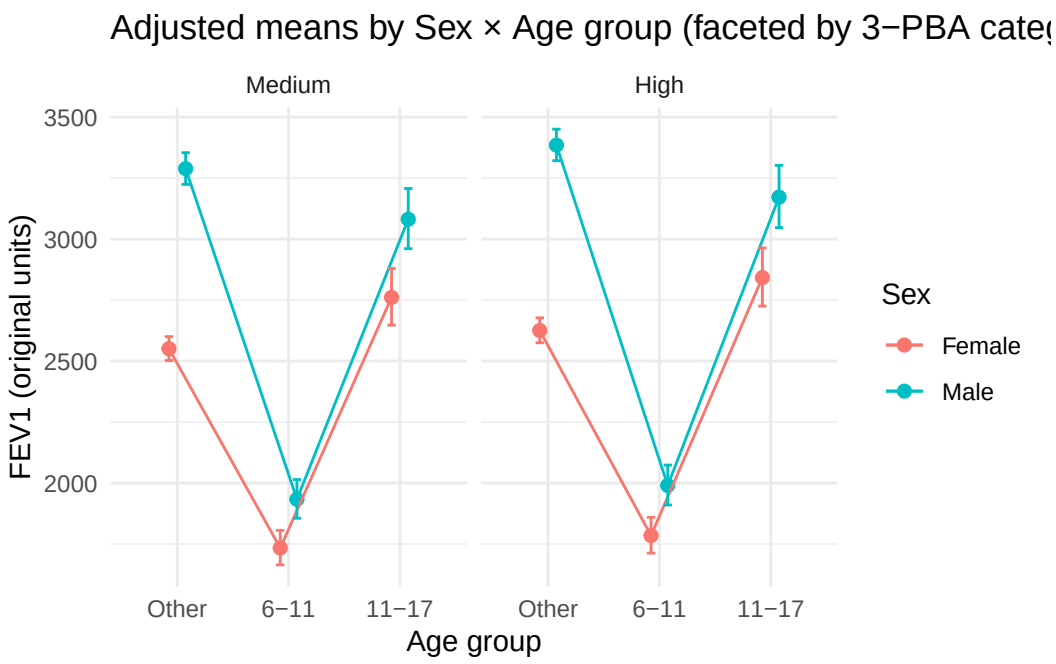
```
Call:
lm(formula = log_PEF ~ URXOPM_cat + RIAGENDR_m * age_cat, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.77863 -0.06695  0.01895  0.06725  0.26047

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   3.817252   0.004267 894.622  < 2e-16 ***
```

```
URXOPM_catHigh                0.004694   0.004486   1.046    0.296
RIAGENDR_mMale                0.110025   0.005204  21.144  < 2e-16 ***
age_cat6-11                  -0.194547   0.009552 -20.366  < 2e-16 ***
age_cat11-17                 -0.013151   0.009757  -1.348    0.178
RIAGENDR_mMale:age_cat6-11   -0.075646   0.013535  -5.589 2.53e-08 ***
RIAGENDR_mMale:age_cat11-17 -0.067989   0.013550  -5.017 5.60e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1133 on 2551 degrees of freedom
Multiple R-squared:  0.3925,    Adjusted R-squared:  0.391
F-statistic: 274.7 on 6 and 2551 DF,  p-value: < 2.2e-16
```

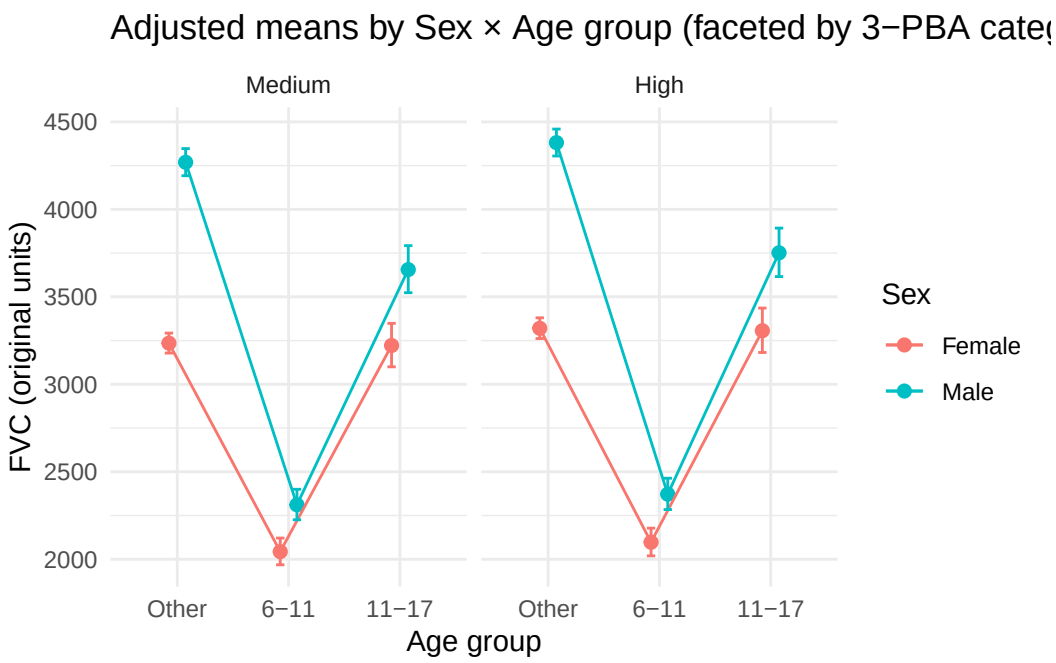
Plot

```r
# model_obj: one of model_fev1_int / model_fvc_int / model_pef_int
# y_lab: nice y-axis label (original units)
plot_interaction <- function(model_obj, df, y_lab = "Outcome (original units)") {

  # Build prediction grid using levels observed in your data
  pred_grid <- expand.grid(
    URXOPM_cat = levels(df$URXOPM_cat),
    RIAGENDR_m = levels(df$RIAGENDR_m),
    age_cat    = levels(df$age_cat)
  )

  # Predict on the log10 scale, then back-transform to original units
  pred <- predict(model_obj, newdata = pred_grid, se.fit = TRUE)
  pred_df <- cbind(pred_grid,
                   fit_log = pred$fit,
                   se      = pred$se.fit) %>%
    mutate(
      # 95% CI on log10 scale
      lwr_log = fit_log - 1.96 * se,
      upr_log = fit_log + 1.96 * se,
      # back-transform to original units
      fit_bt  = 10^fit_log,
      lwr_bt  = 10^lwr_log,
      upr_bt  = 10^upr_log
    )
```

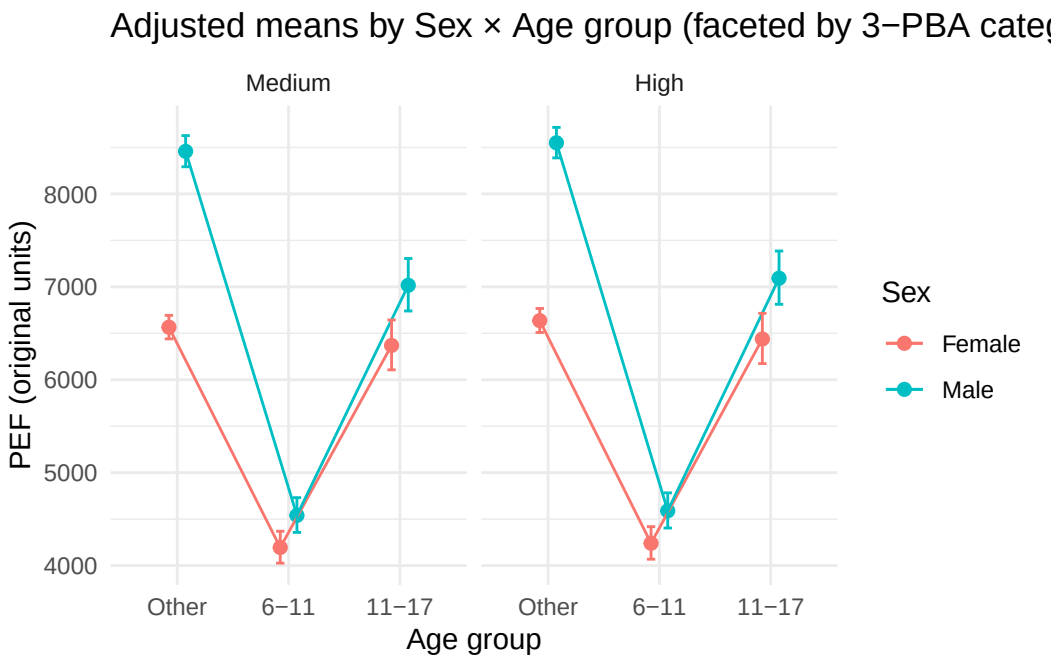
```r
  # Plot: x = age group, color = sex, facet = exposure category
  ggplot(pred_df, aes(x = age_cat, y = fit_bt, color = RIAGENDR_m, group = RIAGENDR_m)) +
    geom_point(position = position_dodge(width = 0.3), size = 2) +
    geom_line(position = position_dodge(width = 0.3)) +
    geom_errorbar(aes(ymin = lwr_bt, ymax = upr_bt),
                  width = 0.15, position = position_dodge(width = 0.3)) +
    facet_wrap(~ URXOPM_cat) +
    labs(
      x = "Age group",
      y = y_lab,
      color = "Sex",
      title = "Adjusted means by Sex × Age group (faceted by 3-PBA category)"
    ) +
    theme_minimal()
}

#Make the three plots
# FEV1 (original units assumed mL or L depending on your variable)
p_fev1 <- plot_interaction(model_fev1_int, df, y_lab = "FEV1 (original units)")
p_fev1
```



```
# FVC
p_fvc  <- plot_interaction(model_fvc_int,  df, y_lab = "FVC (original units)")
p_fvc
```




```
# PEF
p_pef  <- plot_interaction(model_pef_int,  df, y_lab = "PEF (original units)")
p_pef
```



Interpretation

Since FEV1 is Forced expiratory volume in 1 second, it reflects the amount of air a person can forcefully exhale during the firsr second of the total amount of air a person can forcefully exhale. Therefore, low FEV1 suggests airflow obstruction.

Since the coefficient of male x age is -0.062662 and it is statistically significant, it means males (11-17 years old) have smaller-than-expected lung capacity relative to adult males or females of the same age, possibly due to being exposed to 3-PBA.

Since FVC is Forced vital capacity, it reflects the total amount of air a person can forcefully exhale after taking a deepest breath possible. Therefore, low FVC suggests restricted lung.

Since the coefficient of male x age is -0.065557 and it is statistically significant, it means for individuals aged 11–17, the effect of being male on FVC is 0.0656 lower (in log units) than it is among adults.

Since PEF is Peak Expiratory Flow, it reflects the highest speed a person can blow air out during a forced exhalation. Therefore, a low PEF reflects weak respiratory muscles and airways.

Since the coefficient of male x age is -0.067989 and it is statistically significant, it means among adolescents aged 11–17 years, males have a significantly lower PEF (15% lower) than expected based on adult sex differences and age effects.

Overall, males aged 11–17 years exhibited lower lung function, which could be partly attributed to higher exposure to 3-PBA.

Check if the evidence is suggestive, not conclusive (Gender x Age x Pyrethrins concentration in urine)

Due to the above analysis, we only can say higher exposure to 3-PBA might be the reason for male (11-17 years old) to have lower lung function. We have to fit a model to check the interaction between age x gender x 3_PBA.

```
summary(lm(log_FEV1 ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df))
```

```
Call:
lm(formula = log_FEV1 ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56804 -0.06206  0.01111  0.07189  0.30622

Coefficients:
                                            Estimate Std. Error t value
(Intercept)                                 3.404335   0.005097 667.964
RIAGENDR_mMale                              0.116982   0.007405  15.797
age_cat6-11                                -0.176296   0.012999 -13.562
age_cat11-17                                0.035903   0.013830   2.596
URXOPM_catHigh                              0.017314   0.007288   2.376
RIAGENDR_mMale:age_cat6-11                 -0.072940   0.018740  -3.892
RIAGENDR_mMale:age_cat11-17                -0.052138   0.019056  -2.736
RIAGENDR_mMale:URXOPM_catHigh              -0.012844   0.010365  -1.239
age_cat6-11:URXOPM_catHigh                  0.018995   0.019068   0.996
age_cat11-17:URXOPM_catHigh                -0.003148   0.019430  -0.162
RIAGENDR_mMale:age_cat6-11:URXOPM_catHigh   0.017920   0.026989   0.664
RIAGENDR_mMale:age_cat11-17:URXOPM_catHigh -0.022602   0.026985  -0.838
                                           Pr(>|t|)
(Intercept)                                 < 2e-16 ***
RIAGENDR_mMale                              < 2e-16 ***
age_cat6-11                                 < 2e-16 ***
age_cat11-17                               0.009486 **
URXOPM_catHigh                             0.017592 *
RIAGENDR_mMale:age_cat6-11                 0.000102 ***
```

```
RIAGENDR_mMale:age_cat11-17                0.006262 **
RIAGENDR_mMale:URXOPM_catHigh              0.215396
age_cat6-11:URXOPM_catHigh                 0.319271
age_cat11-17:URXOPM_catHigh                0.871320
RIAGENDR_mMale:age_cat6-11:URXOPM_catHigh  0.506773
RIAGENDR_mMale:age_cat11-17:URXOPM_catHigh 0.402342
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1128 on 2546 degrees of freedom
Multiple R-squared:  0.3557,    Adjusted R-squared:  0.3529
F-statistic: 127.8 on 11 and 2546 DF,  p-value: < 2.2e-16
```

```r
summary(lm(log_FVC ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df))
```

```
Call:
lm(formula = log_FVC ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.42733 -0.07217  0.01081  0.05949  0.28294

Coefficients:
                                           Estimate Std. Error t value
(Intercept)                                3.509401   0.004677 750.367
RIAGENDR_mMale                             0.124681   0.006795  18.348
age_cat6-11                               -0.211432   0.011929 -17.724
age_cat11-17                               0.003343   0.012691   0.263
URXOPM_catHigh                             0.012279   0.006688   1.836
RIAGENDR_mMale:age_cat6-11                -0.078964   0.017197  -4.592
RIAGENDR_mMale:age_cat11-17               -0.063206   0.017487  -3.614
RIAGENDR_mMale:URXOPM_catHigh             -0.008082   0.009511  -0.850
age_cat6-11:URXOPM_catHigh                 0.025894   0.017498   1.480
age_cat11-17:URXOPM_catHigh               -0.010145   0.017830  -0.569
RIAGENDR_mMale:age_cat6-11:URXOPM_catHigh  0.021965   0.024766   0.887
RIAGENDR_mMale:age_cat11-17:URXOPM_catHigh -0.006112   0.024763  -0.247
                                           Pr(>|t|)
(Intercept)                                 < 2e-16 ***
RIAGENDR_mMale                              < 2e-16 ***
age_cat6-11                                 < 2e-16 ***
age_cat11-17                               0.792272
```

```
URXOPM_catHigh                                0.066464 .
RIAGENDR_mMale:age_cat6-11                    4.61e-06 ***
RIAGENDR_mMale:age_cat11-17                   0.000307 ***
RIAGENDR_mMale:URXOPM_catHigh                 0.395564
age_cat6-11:URXOPM_catHigh                    0.139049
age_cat11-17:URXOPM_catHigh                   0.569394
RIAGENDR_mMale:age_cat6-11:URXOPM_catHigh     0.375218
RIAGENDR_mMale:age_cat11-17:URXOPM_catHigh    0.805072
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1035 on 2546 degrees of freedom
Multiple R-squared:  0.4558,    Adjusted R-squared:  0.4534
F-statistic: 193.9 on 11 and 2546 DF,  p-value: < 2.2e-16
```

```r
summary(lm(log_PEF ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df))
```

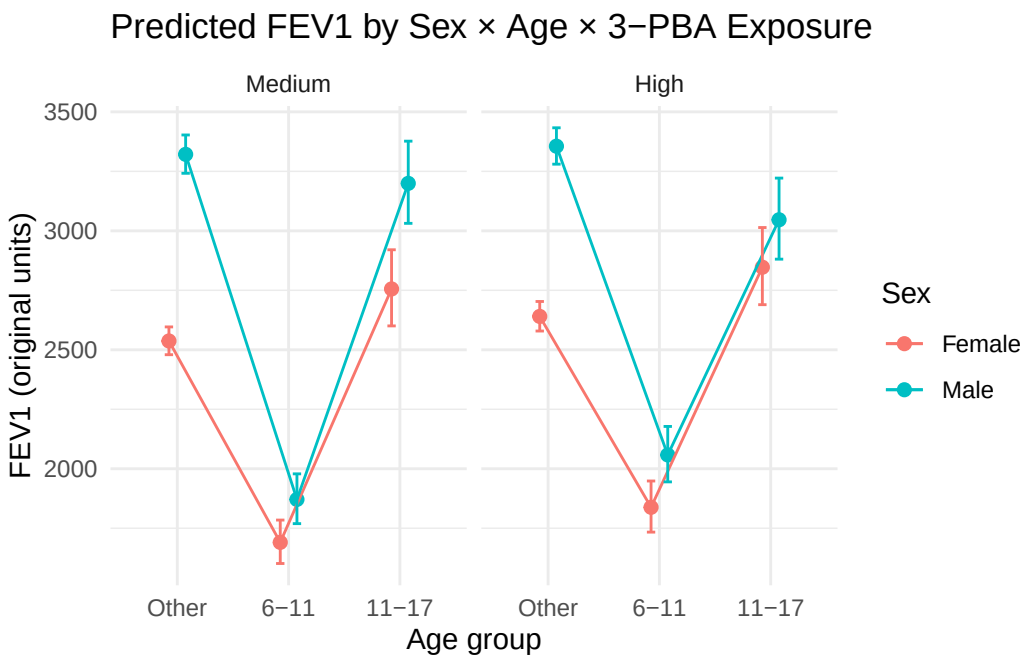
```
Call:
lm(formula = log_PEF ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.77493 -0.06552  0.02039  0.06733  0.25445

Coefficients:
                                              Estimate Std. Error t value
(Intercept)                                   3.813545   0.005111 746.128
RIAGENDR_mMale                                0.120668   0.007426  16.249
age_cat6-11                                  -0.202325   0.013036 -15.520
age_cat11-17                                 -0.012367   0.013870  -0.892
URXOPM_catHigh                                0.012274   0.007309   1.679
RIAGENDR_mMale:age_cat6-11                   -0.091626   0.018794  -4.875
RIAGENDR_mMale:age_cat11-17                  -0.059960   0.019111  -3.138
RIAGENDR_mMale:URXOPM_catHigh                -0.020632   0.010394  -1.985
age_cat6-11:URXOPM_catHigh                    0.017355   0.019122   0.908
age_cat11-17:URXOPM_catHigh                  -0.001844   0.019485  -0.095
RIAGENDR_mMale:age_cat6-11:URXOPM_catHigh     0.029758   0.027066   1.099
RIAGENDR_mMale:age_cat11-17:URXOPM_catHigh   -0.017739   0.027062  -0.656
                                             Pr(>|t|)
(Intercept)                                   < 2e-16 ***
RIAGENDR_mMale                                < 2e-16 ***
```

```
age_cat6-11                                      < 2e-16 ***
age_cat11-17                                     0.37265
URXOPM_catHigh                                   0.09321 .
RIAGENDR_mMale:age_cat6-11                      1.15e-06 ***
RIAGENDR_mMale:age_cat11-17                      0.00172 **
RIAGENDR_mMale:URXOPM_catHigh                    0.04726 *
age_cat6-11:URXOPM_catHigh                       0.36419
age_cat11-17:URXOPM_catHigh                      0.92462
RIAGENDR_mMale:age_cat6-11:URXOPM_catHigh        0.27166
RIAGENDR_mMale:age_cat11-17:URXOPM_catHigh       0.51220
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1131 on 2546 degrees of freedom
Multiple R-squared:  0.3957,    Adjusted R-squared:  0.3931
F-statistic: 151.6 on 11 and 2546 DF,  p-value: < 2.2e-16
```

Plot

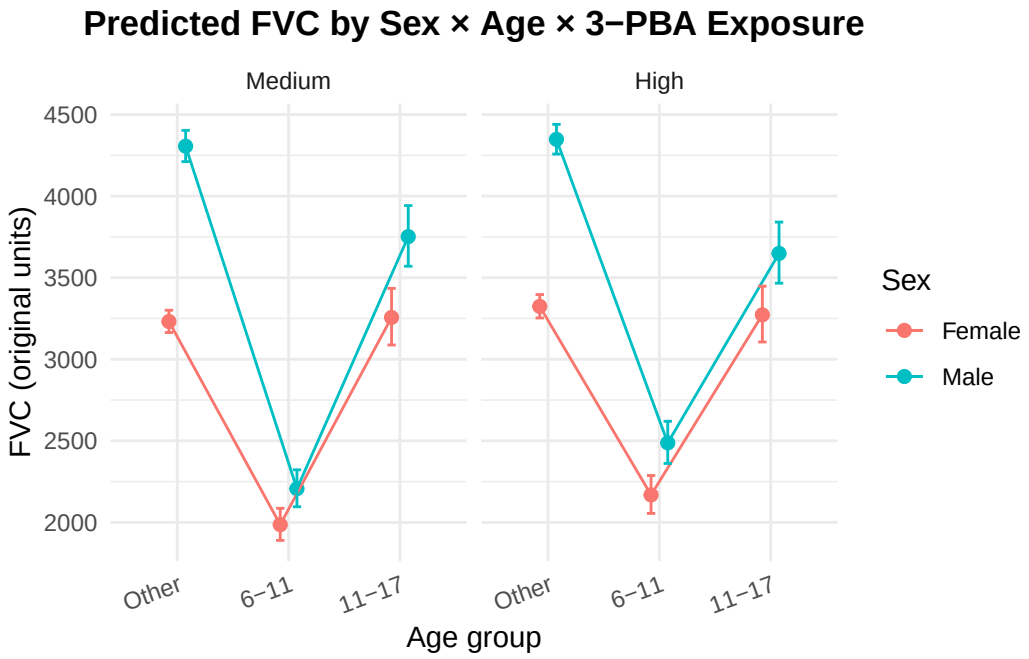
```r
#Fit the model
model_3way <- lm(log_FEV1 ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df)

#Create a grid of predictor combinations
pred_grid1 <- expand.grid(
  RIAGENDR_m = levels(df$RIAGENDR_m),
  age_cat    = levels(df$age_cat),
  URXOPM_cat = levels(df$URXOPM_cat)
)

#Get predicted log10(FEV1) and standard errors
pred1 <- predict(model_3way, newdata = pred_grid1, se.fit = TRUE)

#Combine and back-transform predictions to original FEV1 units
pred_df1 <- cbind(pred_grid1,
                  fit_log = pred1$fit,
                  se = pred1$se.fit) %>%
  mutate(
    lwr_log = fit_log - 1.96 * se,
    upr_log = fit_log + 1.96 * se,
    fit_bt  = 10^fit_log,
    lwr_bt  = 10^lwr_log,
```

```
    upr_bt  = 10^upr_log
  )

#Plot predicted FEV1 by age, colored by sex, faceted by exposure level
ggplot(pred_df1,
       aes(x = age_cat, y = fit_bt,
           color = RIAGENDR_m, group = RIAGENDR_m)) +
  geom_point(position = position_dodge(width = 0.3), size = 2) +
  geom_line(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = lwr_bt, ymax = upr_bt),
                width = 0.15,
                position = position_dodge(width = 0.3)) +
  facet_wrap(~ URXOPM_cat) +
  labs(
    title = "Predicted FEV1 by Sex × Age × 3-PBA Exposure",
    x = "Age group",
    y = "FEV1 (original units)",
    color = "Sex"
  ) +
  theme_minimal()
```



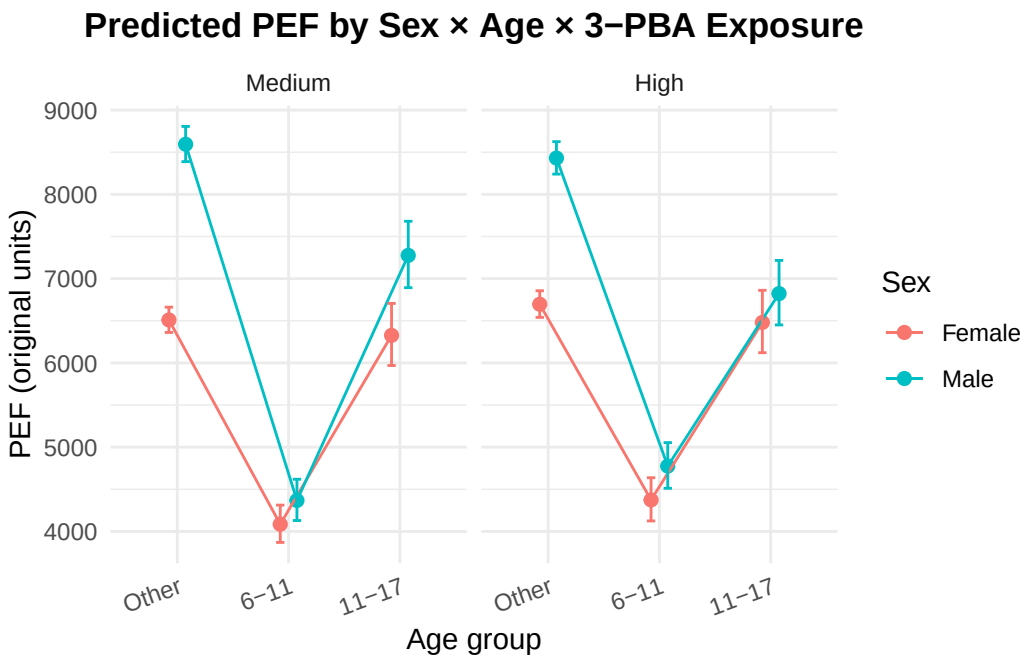
```r
#Fit the model
model_3way_fvc <- lm(log_FVC ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df)

#Create prediction grid using all combinations of predictors
pred_grid2 <- expand.grid(
  RIAGENDR_m = levels(df$RIAGENDR_m),
  age_cat    = levels(df$age_cat),
  URXOPM_cat = levels(df$URXOPM_cat)
)

#Predict FVC on log10 scale + get standard errors
pred2 <- predict(model_3way_fvc, newdata = pred_grid2, se.fit = TRUE)

#Combine predictions and back-transform to original FVC units
pred_df2 <- cbind(pred_grid2,
                  fit_log = pred2$fit,
                  se      = pred2$se.fit) %>%
  mutate(
    lwr_log = fit_log - 1.96 * se,
    upr_log = fit_log + 1.96 * se,
    fit_bt  = 10^fit_log,
    lwr_bt  = 10^lwr_log,
    upr_bt  = 10^upr_log
  )

#Plot: x = age group, color = sex, facets = 3-PBA exposure
ggplot(pred_df2, aes(x = age_cat,
                     y = fit_bt,
                     color = RIAGENDR_m,
                     group = RIAGENDR_m)) +
  geom_point(position = position_dodge(width = 0.3), size = 2) +
  geom_line(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = lwr_bt, ymax = upr_bt),
                width = 0.15,
                position = position_dodge(width = 0.3)) +
  facet_wrap(~ URXOPM_cat) +
  labs(
    title = "Predicted FVC by Sex × Age × 3-PBA Exposure",
    x = "Age group",
    y = "FVC (original units)",
    color = "Sex"
  ) +
```

```
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.text.x = element_text(angle = 20, hjust = 1)
  )
```




```
# 1) Fit the model
model_3way_pef <- lm(log_PEF ~ RIAGENDR_m * age_cat * URXOPM_cat, data = df)

# 2) Prediction grid (use levels present in your data)
pred_grid3 <- expand.grid(
  RIAGENDR_m = levels(df$RIAGENDR_m),
  age_cat    = levels(df$age_cat),
  URXOPM_cat = levels(df$URXOPM_cat)
)

# 3) Predict on log10 scale with SEs
pred3 <- predict(model_3way_pef, newdata = pred_grid3, se.fit = TRUE)

# 4) Combine + back-transform to original units
pred_df3 <- cbind(pred_grid3,
                  fit_log = pred3$fit,
```

```
                  se      = pred3$se.fit) %>%
  mutate(
    lwr_log = fit_log - 1.96 * se,
    upr_log = fit_log + 1.96 * se,
    fit_bt  = 10^fit_log,
    lwr_bt  = 10^lwr_log,
    upr_bt  = 10^upr_log
  )

# 5) Plot: x = age group, color = sex, facets = 3-PBA exposure
ggplot(pred_df3,
       aes(x = age_cat, y = fit_bt, color = RIAGENDR_m, group = RIAGENDR_m)) +
  geom_point(position = position_dodge(width = 0.3), size = 2) +
  geom_line(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = lwr_bt, ymax = upr_bt),
                width = 0.15,
                position = position_dodge(width = 0.3)) +
  facet_wrap(~ URXOPM_cat) +
  labs(
    title = "Predicted PEF by Sex × Age × 3-PBA Exposure",
    x = "Age group",
    y = "PEF (original units)",
    color = "Sex"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.text.x = element_text(angle = 20, hjust = 1)
  )
```



Interpretation

Across all three measures of lung function, the three-way interaction between sex, age, and exposure (Male × 11–17 yrs × High 3-PBA) was negative but not statistically significant. This suggests that, although 11-17 year-old males tend to have lower lung function overall, the additional effect of high 3-PBA exposure on this group is not distinguishable from the general effects already observed in the population.

In fact, all the graphs showed that females aged 6–11 years had the lowest lung function. These results and graphs do not support the claim made in the research paper that males aged 11–17 years have lower lung function due to exposure to 3-PBA. Instead, these findings may simply indicate that the younger age group (6–11 years) has lower lung function than the older group (11–17 years), independent of 3-PBA exposure.

Conclusion:

In this project, we analyzed multiple datasets on strawberry production, pesticide usage, and health outcomes to investigate potential relationships between insecticide exposure and pulmonary function.

Our findings suggest that pyrethrin exposure is negatively associated with pulmonary function among children aged 6–11 years. However, we were unable to replicate the conclusion presented

in the referenced study, which claimed that males aged 11–17 years exhibit lower lung function due to exposure to 3-PBA. Instead, our results indicate that the younger age group (6–11 years) generally has lower lung function than the older group (11–17 years), independent of 3-PBA exposure—a pattern consistent with the study's actual observations rather than its interpretation.

Given the widespread global use of pyrethrin, future research employing prospective study designs is warranted. Expanding such studies to include adult populations would also provide valuable insights into the broader health effects of pyrethrin exposure.