

pesticide_risk_plots

Joe Collins

2025-03-05

Pesticide Risk Plots

Set the working directory and load the necessary packages

Data

Read the combined data set

```
#combined_dat <- read.csv(file = "data/pesticide_data/pesticide_properties_data.csv")  
  
# source(file = "~/Documents/GitHub/pesticide_info_web_scraper/pesticide_risk_summary_stats.R")  
  
dat <- read.csv(file = "sym_link_pesticide_data/data/pesticide_data/danish_pli_total_pli_data.csv")  
dat$year <- factor(dat$year, levels = c(2022, 2023, 2024))  
dat$treatment <- factor(dat$treatment, levels = c("Conservation", "Conventional"))  
  
#_____ # Plots by year
```

~ Total PLI x rate

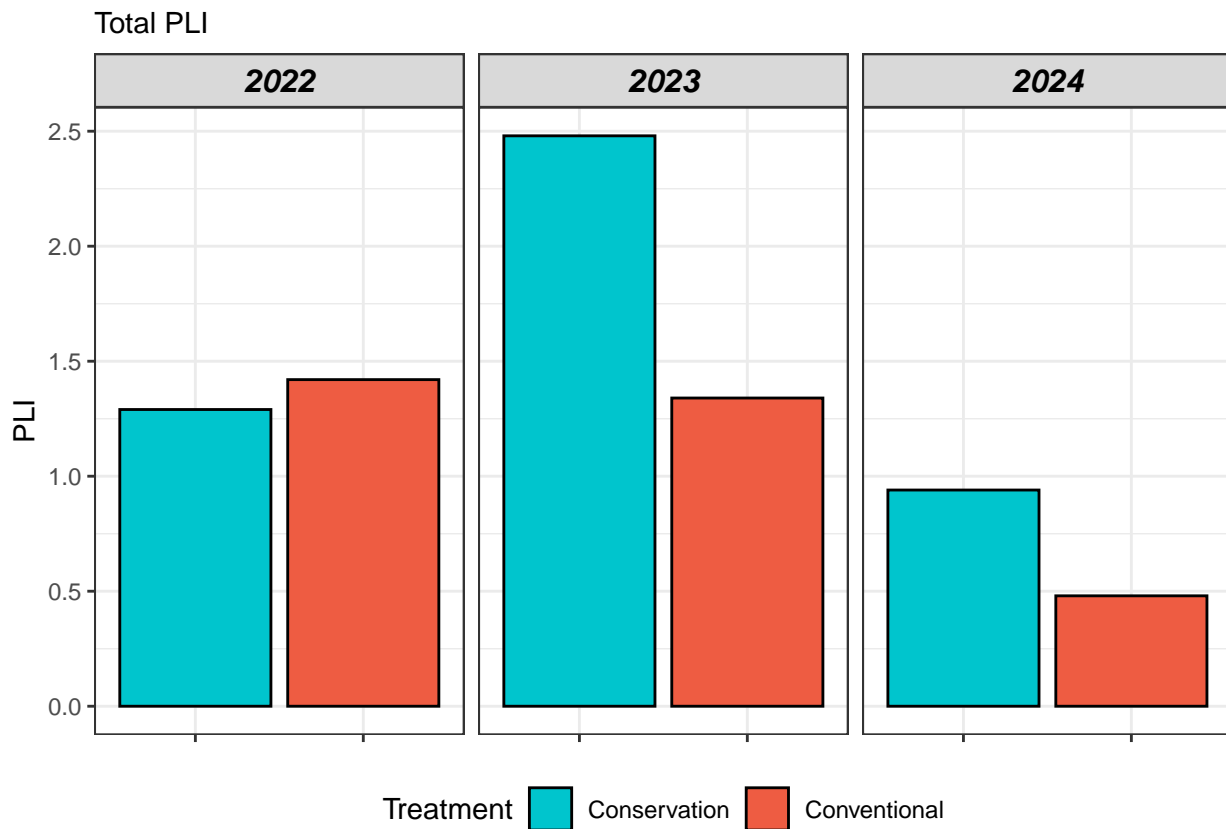
```
sum <-  
  dat %>%  
  group_by(year, treatment) %>%  
  summarise(  
    n = n(),  
    sum = round(x = sum(total_pli_x_rate, na.rm = TRUE), digits = 2)  
  )  
  
## `summarise()` has grouped output by 'year'. You can override using the  
## `.groups` argument.  
  
p1 <-  
  ggplot(data = sum,  
    aes(x = treatment,  
      y = sum,  
      fill = treatment)) +  
  geom_bar(stat = "identity",  
    color = "black",  
    position = "dodge") +
```

```

labs(
  subtitle = "Total PLI",
  x = "Treatment",
  y = "PLI") +
theme_bw() +
scale_fill_manual(values=c("turquoise3","tomato2"),
  name = "Treatment") +
theme(strip.text.x = element_text(size = 12,
  color = "black",
  face = "bold.italic"),
  legend.position = "bottom",
  axis.text.x = element_blank(),
  axis.title.x = element_blank()) +
facet_wrap(~ year,
  ncol = 4,
  scales = 'free_x')

```

p1



~ Ecotoxicity PLI x rate

```
names(dat)
```

```

## [1] "X"                "year"
## [3] "treatment"        "crop"
## [5] "ai_name"          "category"

```

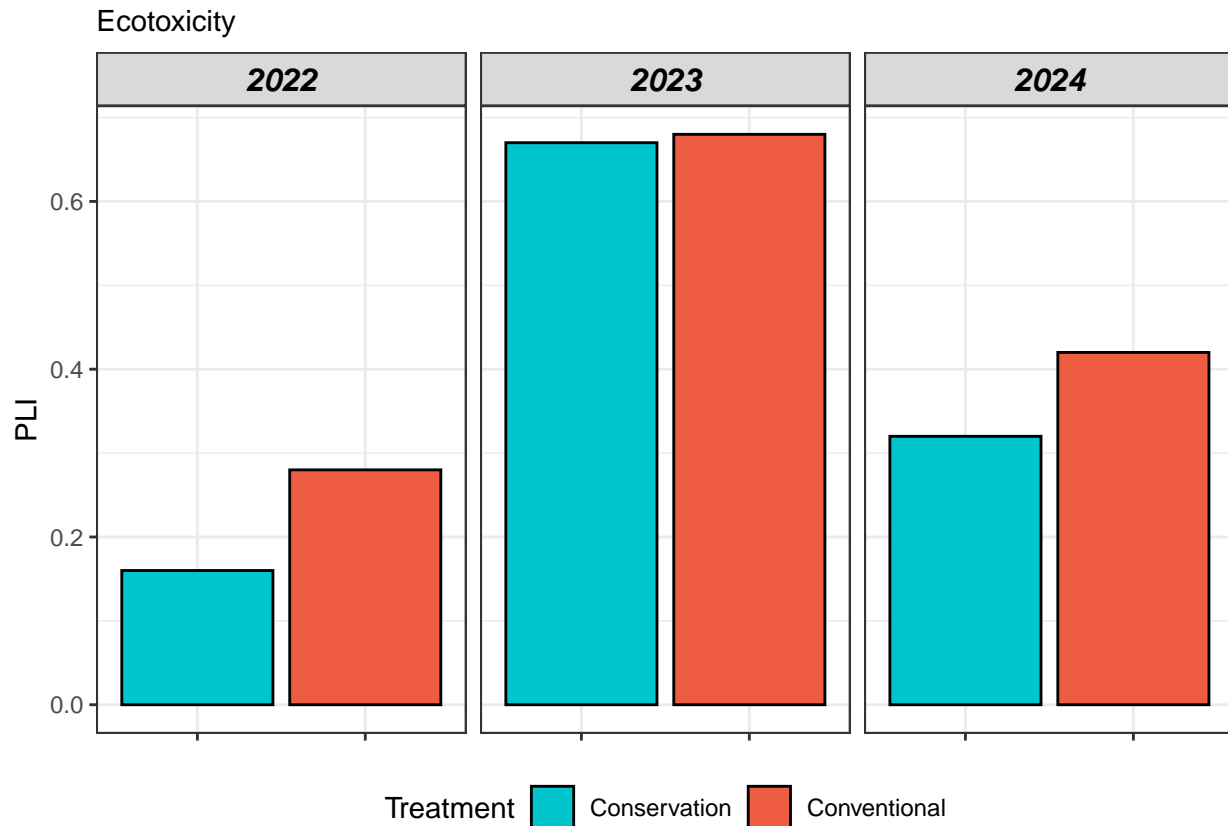
```
## [7] "avg_normalized_rate_kg_ha" "DT50_norm"
## [9] "SCI_GROW_norm"            "BCF_norm"
## [11] "Env_Fate_Indicator"       "fish_weighted"
## [13] "daphnia_weighted"         "worm_weighted"
## [15] "birds_weighted"           "mammals_weighted"
## [17] "algae_weighted"           "aq_plants_weighted"
## [19] "bees_weighted"            "EcoTox_Indicator"
## [21] "h_codes"                  "PL_HH"
## [23] "Env_Fate_Scaled"          "EcoTox_Scaled"
## [25] "PL_HH_Scaled"             "total_pli"
## [27] "env_fate_indic_x_rate"    "ecotox_indic_x_rate"
## [29] "human_health_indic_x_rate" "total_pli_x_rate"
```

```
sum <-
  dat %>%
  group_by(year, treatment) %>%
  summarise(
    n = n(),
    sum = round(x = sum(ecotox_indic_x_rate, na.rm = TRUE), digits = 2)
  )
```

`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.

```
p2 <-
ggplot(data = sum,
       aes(x = treatment,
           y = sum,
           fill = treatment)) +
geom_bar(stat = "identity",
         color = "black",
         position = "dodge") +
labs(
  subtitle = "Ecotoxicity",
  x = "Treatment",
  y = "PLI") +
theme_bw() +
scale_fill_manual(values=c("turquoise3", "tomato2"),
                  name = "Treatment") +
theme(strip.text.x = element_text(size = 12,
                                   color = "black",
                                   face = "bold.italic"),
      legend.position = "bottom",
      axis.text.x = element_blank(),
      axis.title.x = element_blank()) +
facet_wrap(~ year,
           ncol = 4,
           scales = 'free_x')
```

p2



~ Environmental fate PLI x rate

```
names(dat)
```

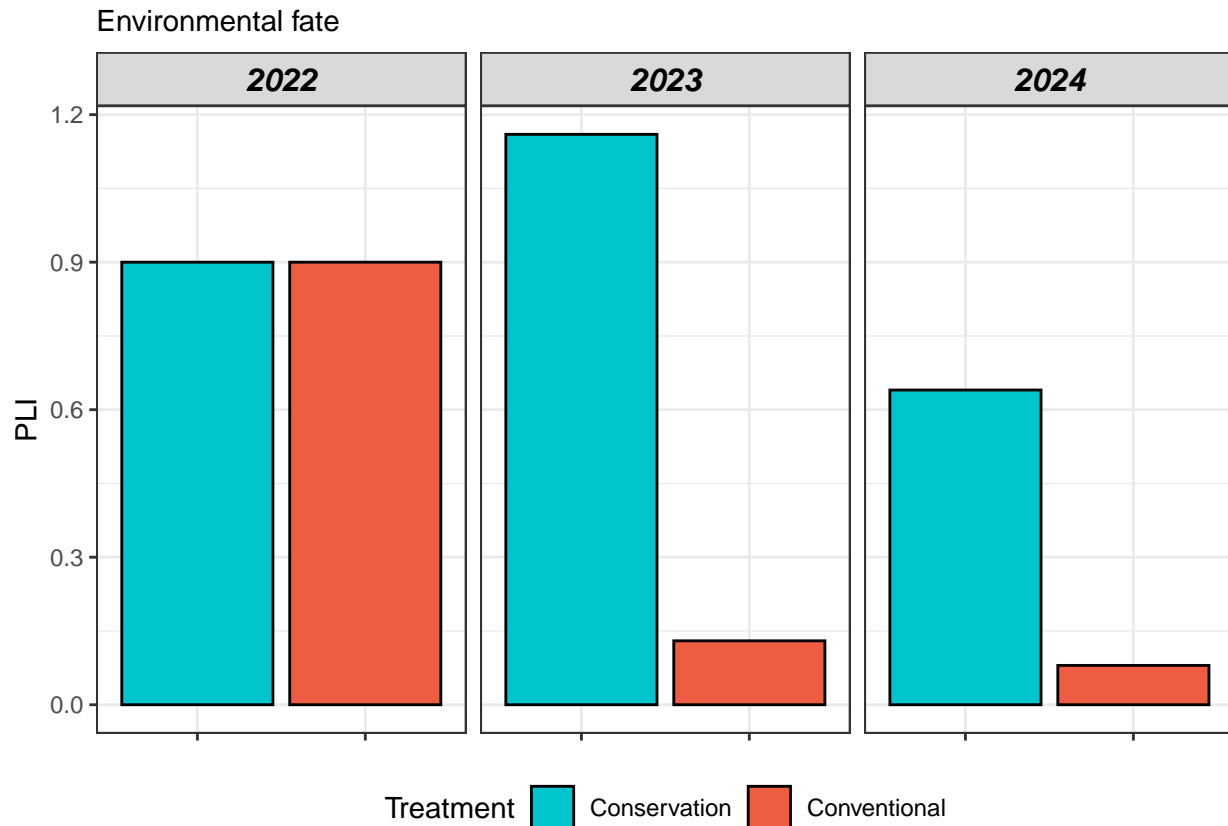
```
## [1] "X" "year"
## [3] "treatment" "crop"
## [5] "ai_name" "category"
## [7] "avg_normalized_rate_kg_ha" "DT50_norm"
## [9] "SCI_GROW_norm" "BCF_norm"
## [11] "Env_Fate_Indicator" "fish_weighted"
## [13] "daphnia_weighted" "worm_weighted"
## [15] "birds_weighted" "mammals_weighted"
## [17] "algae_weighted" "aq_plants_weighted"
## [19] "bees_weighted" "EcoTox_Indicator"
## [21] "h_codes" "PL_HH"
## [23] "Env_Fate_Scaled" "EcoTox_Scaled"
## [25] "PL_HH_Scaled" "total_pli"
## [27] "env_fate_indic_x_rate" "ecotox_indic_x_rate"
## [29] "human_health_indic_x_rate" "total_pli_x_rate"
```

```
sum <-
  dat %>%
  group_by(year, treatment) %>%
  summarise(
    n = n(),
    sum = round(x = sum(env_fate_indic_x_rate, na.rm = TRUE), digits = 2)
  )
```

`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.

```
p3 <-  
ggplot(data = sum,  
       aes(x = treatment,  
           y = sum,  
           fill = treatment)) +  
geom_bar(stat = "identity",  
        color = "black",  
        position = "dodge") +  
labs(  
  subtitle = "Environmental fate",  
  x = "Treatment",  
  y = "PLI") +  
theme_bw() +  
scale_fill_manual(values=c("turquoise3","tomato2"),  
                  name = "Treatment") +  
theme(strip.text.x = element_text(size = 12,  
                                   color = "black",  
                                   face = "bold.italic"),  
      legend.position = "bottom",  
      axis.text.x = element_blank(),  
      axis.title.x = element_blank()) +  
facet_wrap(~ year,  
          ncol = 4,  
          scales = 'free_x')
```

p3



~ Human health PLI x rate

```
names(dat)
```

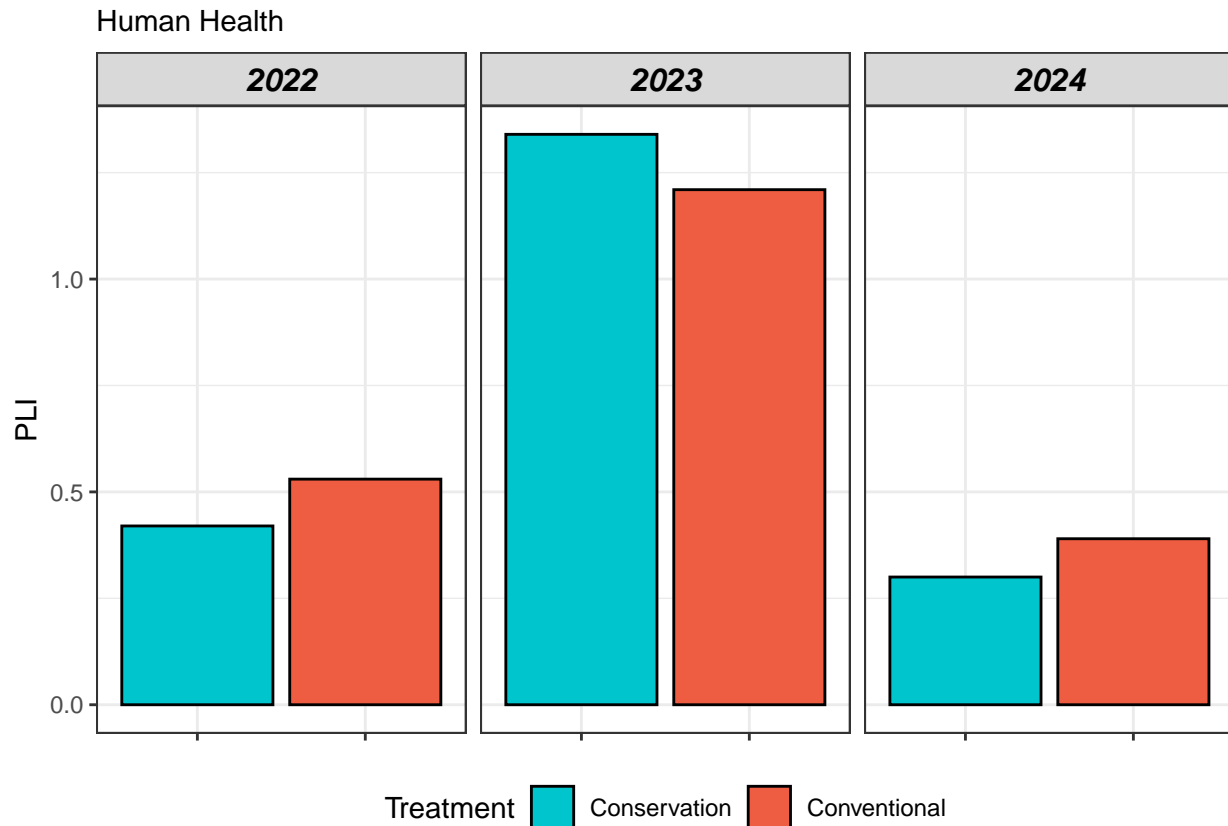
```
## [1] "X" "year"
## [3] "treatment" "crop"
## [5] "ai_name" "category"
## [7] "avg_normalized_rate_kg_ha" "DT50_norm"
## [9] "SCI_GROW_norm" "BCF_norm"
## [11] "Env_Fate_Indicator" "fish_weighted"
## [13] "daphnia_weighted" "worm_weighted"
## [15] "birds_weighted" "mammals_weighted"
## [17] "algae_weighted" "aq_plants_weighted"
## [19] "bees_weighted" "EcoTox_Indicator"
## [21] "h_codes" "PL_HH"
## [23] "Env_Fate_Scaled" "EcoTox_Scaled"
## [25] "PL_HH_Scaled" "total_pli"
## [27] "env_fate_indic_x_rate" "ecotox_indic_x_rate"
## [29] "human_health_indic_x_rate" "total_pli_x_rate"
```

```
sum <-
  dat %>%
  group_by(year, treatment) %>%
  summarise(
    n = n(),
    sum = round(x = sum(human_health_indic_x_rate, na.rm = TRUE), digits = 2)
  )
```

`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.

```
p4 <-  
ggplot(data = sum,  
       aes(x = treatment,  
           y = sum,  
           fill = treatment)) +  
geom_bar(stat = "identity",  
        color = "black",  
        position = "dodge") +  
labs(  
  subtitle = "Human Health",  
  x = "Treatment",  
  y = "PLI") +  
theme_bw() +  
scale_fill_manual(values=c("turquoise3","tomato2"),  
                  name = "Treatment") +  
theme(strip.text.x = element_text(size = 12,  
                                   color = "black",  
                                   face = "bold.italic"),  
      legend.position = "bottom",  
      axis.text.x = element_blank(),  
      axis.title.x = element_blank()) +  
facet_wrap(~ year,  
          ncol = 4,  
          scales = 'free_x')
```

p4



~ Total mass PLI x rate

```
names(dat)
```

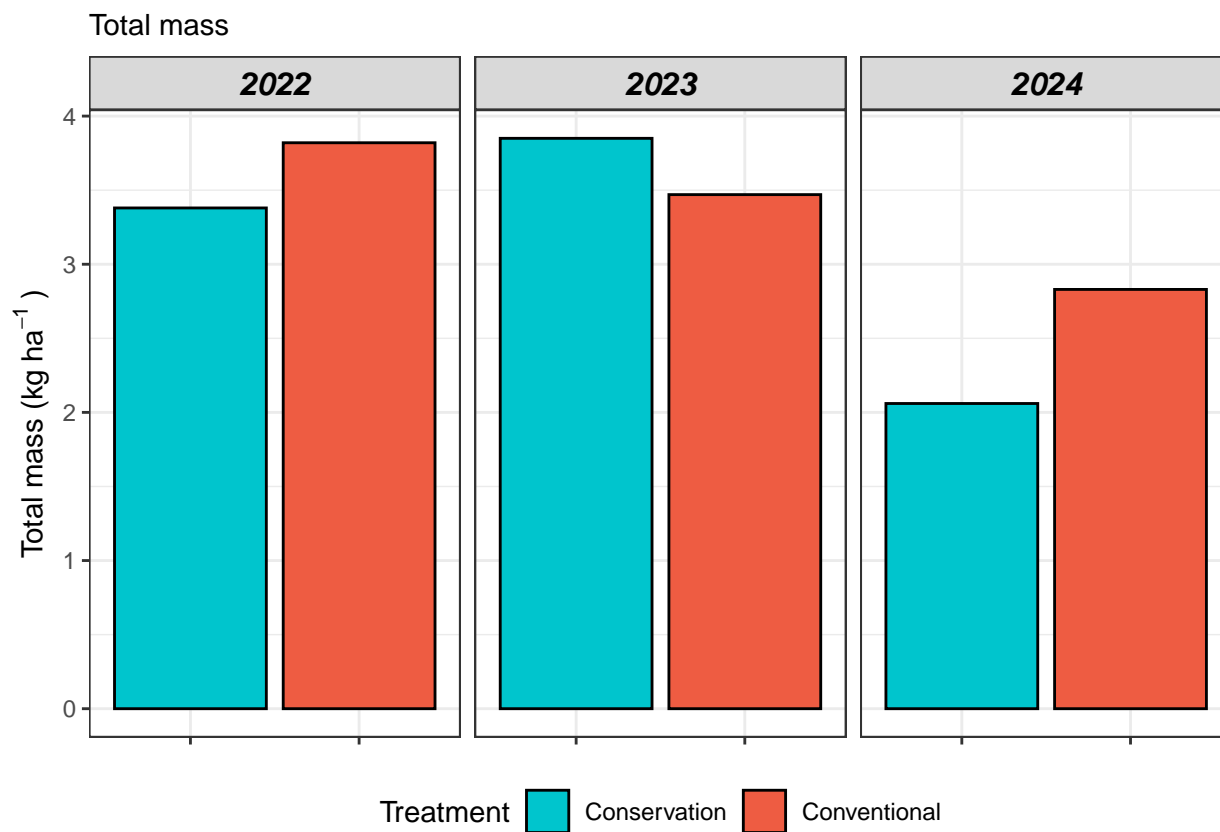
```
## [1] "X" "year"
## [3] "treatment" "crop"
## [5] "ai_name" "category"
## [7] "avg_normalized_rate_kg_ha" "DT50_norm"
## [9] "SCI_GROW_norm" "BCF_norm"
## [11] "Env_Fate_Indicator" "fish_weighted"
## [13] "daphnia_weighted" "worm_weighted"
## [15] "birds_weighted" "mammals_weighted"
## [17] "algae_weighted" "aq_plants_weighted"
## [19] "bees_weighted" "EcoTox_Indicator"
## [21] "h_codes" "PL_HH"
## [23] "Env_Fate_Scaled" "EcoTox_Scaled"
## [25] "PL_HH_Scaled" "total_pli"
## [27] "env_fate_indic_x_rate" "ecotox_indic_x_rate"
## [29] "human_health_indic_x_rate" "total_pli_x_rate"
```

```
sum <-
  dat %>%
  group_by(year, treatment) %>%
  summarise(
    n = n(),
    sum = round(x = sum(avg_normalized_rate_kg_ha, na.rm = TRUE), digits = 2)
  )
```


`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.

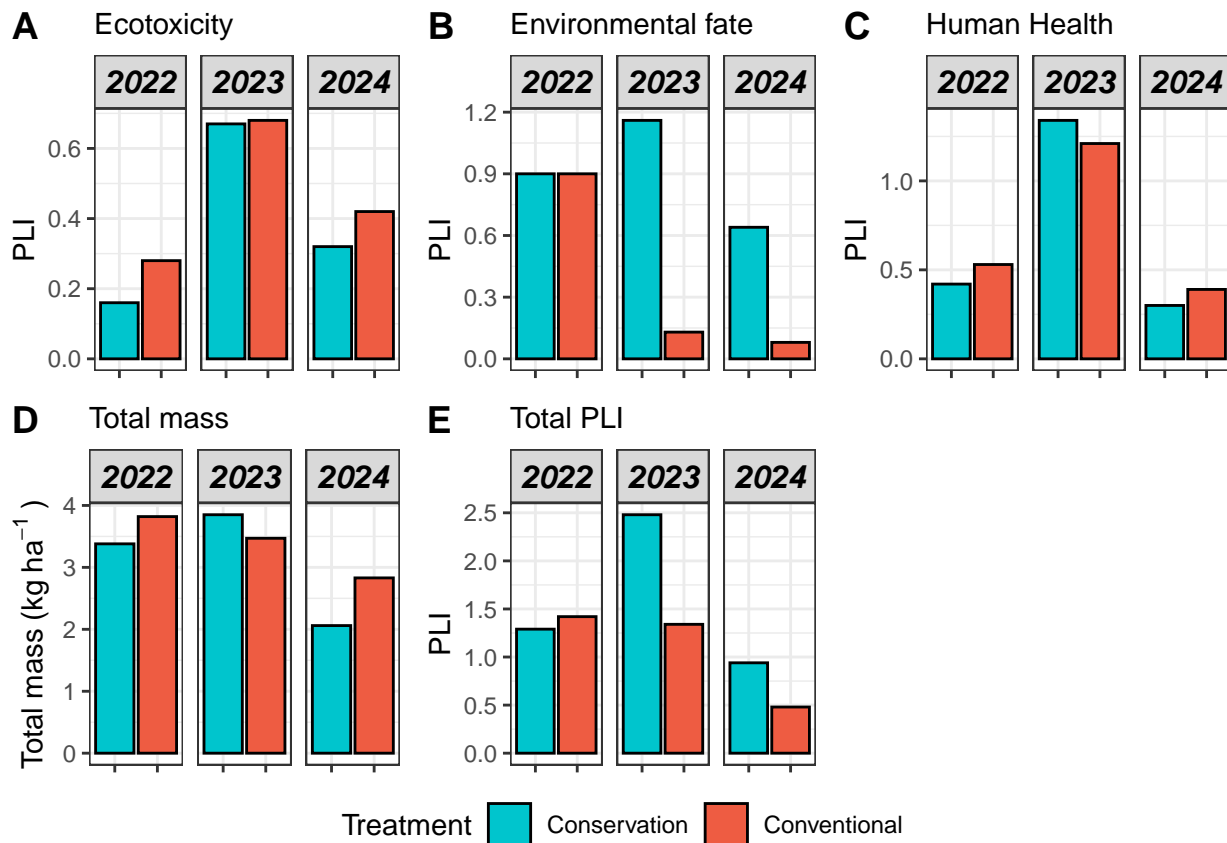
```
p5 <-  
ggplot(data = sum,  
       aes(x = treatment,  
           y = sum,  
           fill = treatment)) +  
geom_bar(stat = "identity",  
        color = "black",  
        position = "dodge") +  
labs(  
  y = expression("Total mass (kg)"~"ha"^{~-1}~"),  
  x = "Treatment",  
  subtitle = "Total mass"  
  ) +  
theme_bw() +  
scale_fill_manual(values=c("turquoise3","tomato2"),  
                  name = "Treatment") +  
theme(strip.text.x = element_text(size = 12,  
                                   color = "black",  
                                   face = "bold.italic"),  
      legend.position = "bottom",  
      axis.text.x = element_blank(),  
      axis.title.x = element_blank()) +  
facet_wrap(~ year,  
          ncol = 4,  
          scales = 'free_x')
```

p5



joint plot

```
ggarrange(p2,p3,p4,p5,p1,
  ncol = 3,
  nrow = 2,
  labels = c("A","B","C","D","E"),
  common.legend = TRUE, legend = "bottom")
```



```
ggsave(filename = "sym_link_pesticide_data/plots/pesticide_risk_plots/danish_pli_join_plot.png",
        width = 10, height = 6)
```

#_____ # Plots by treatment

~ Total PLI x rate

```
sum <-
  dat %>%
  group_by(treatment, year) %>%
  summarise(
    n = n(),
    sum = round(x = sum(total_pli_x_rate, na.rm = TRUE), digits = 2)
  ) %>%
  group_by(treatment) %>%
  summarise(
    n = n(),
    # sum = round(sum(total_pli_x_rate, na.rm = TRUE), 2),
    mean = round(mean(sum, na.rm = TRUE), 2),
    sd = round(sd(sum, na.rm = TRUE), 2),
    se = round(sd(sum, na.rm = TRUE) / sqrt(n), 2) # Standard Error
  )
```

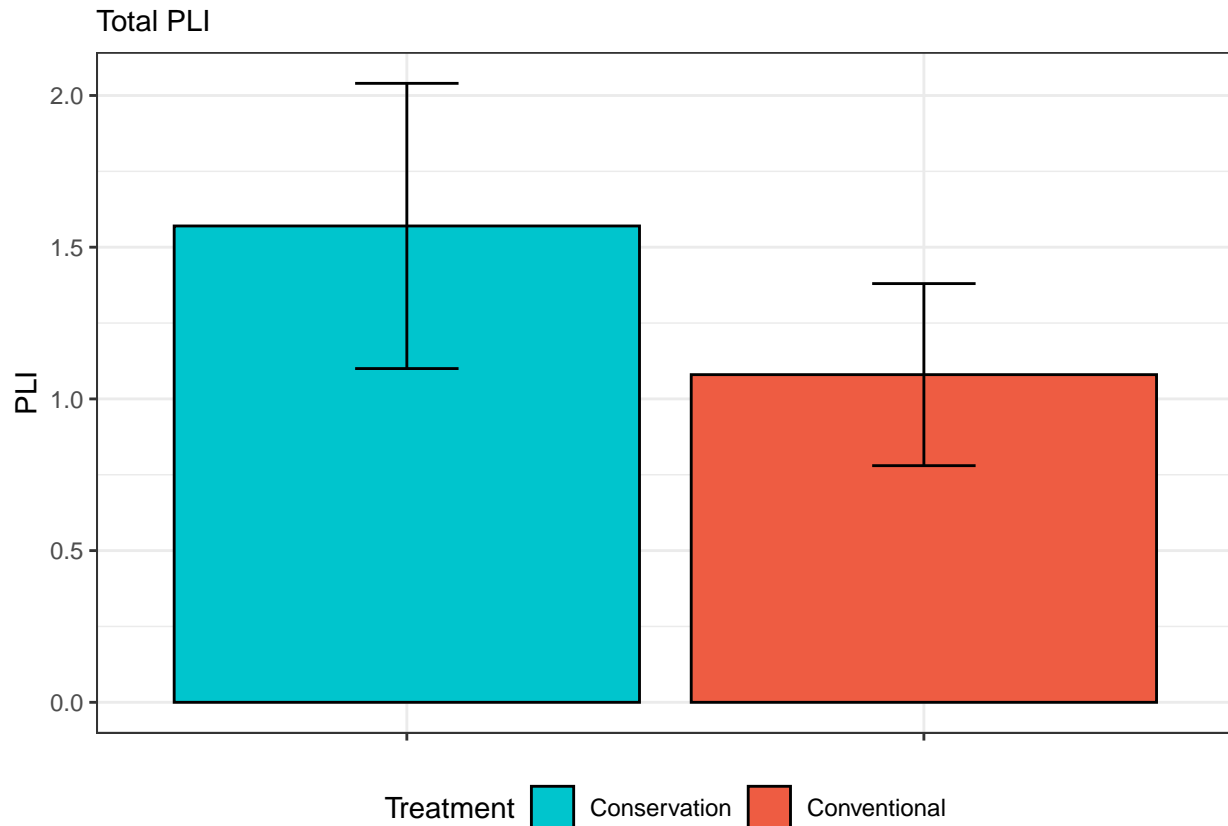
```
## `summarise()` has grouped output by 'treatment'. You can override using the
## `.groups` argument.
```

```
sum
```

```
## # A tibble: 2 x 5
##   treatment      n mean   sd   se
##   <fct>      <int> <dbl> <dbl> <dbl>
## 1 Conservation     3  1.57  0.81  0.47
## 2 Conventional     3  1.08  0.52  0.3
```

```
p1 <-
ggplot(data = sum,
       aes(x = treatment,
           y = mean,
           fill = treatment)) +
geom_bar(stat = "identity",
       color = "black",
       position = "dodge") +
labs(
  subtitle = "Total PLI",
  x = "Treatment",
  y = "PLI") +
theme_bw() +
scale_fill_manual(values=c("turquoise3", "tomato2"),
                  name = "Treatment") +
  geom_errorbar(aes(ymin=mean-se,
                    ymax=mean+se),
                width=.2,                      # Width of the error bars
                position=position_dodge(.9)) +
theme(strip.text.x = element_text(size = 12,
                                   color = "black",
                                   face = "bold.italic"),
      legend.position = "bottom",
      axis.text.x = element_blank(),
      axis.title.x = element_blank())
```

```
p1
```



~ Ecotoxicity PLI x rate

```
sum <-
  dat %>%
  group_by(treatment, year) %>%
  summarise(
    n = n(),
    sum = round(x = sum(ecotox_indic_x_rate, na.rm = TRUE), digits = 2)
  ) %>%
  group_by(treatment) %>%
  summarise(
    n = n(),
    # sum = round(sum(total_pli_x_rate, na.rm = TRUE), 2),
    mean = round(mean(sum, na.rm = TRUE), 2),
    sd = round(sd(sum, na.rm = TRUE), 2),
    se = round(sd(sum, na.rm = TRUE) / sqrt(n), 2) # Standard Error
  )
```

`summarise()` has grouped output by 'treatment'. You can override using the
`.groups` argument.

```
sum
```

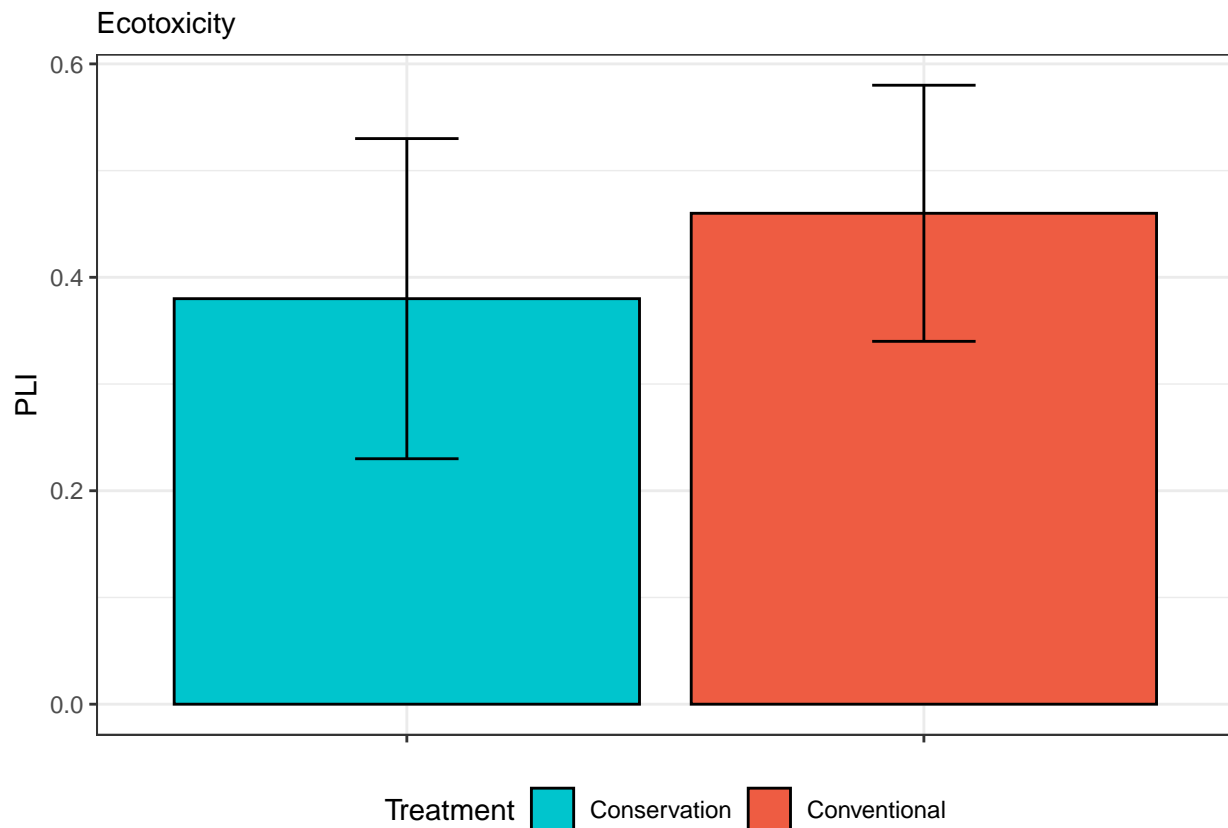
```
## # A tibble: 2 x 5
##   treatment      n mean   sd   se
##   <fct>      <int> <dbl> <dbl> <dbl>
## 1 Conservation    3  0.38  0.26  0.15
## 2 Conventional    3  0.46  0.2  0.12
```

```

p2 <-
ggplot(data = sum,
      aes(x = treatment,
          y = mean,
          fill = treatment)) +
geom_bar(stat = "identity",
      color = "black",
      position = "dodge") +
labs(
  subtitle = "Ecotoxicity",
  x = "Treatment",
  y = "PLI") +
theme_bw() +
scale_fill_manual(values=c("turquoise3","tomato2"),
  name = "Treatment") +
  geom_errorbar(aes(ymin=mean-se,
                    ymax=mean+se),
                width=.2,                      # Width of the error bars
                position=position_dodge(.9)) +
theme(strip.text.x = element_text(size = 12,
                                    color = "black",
                                    face = "bold.italic"),
      legend.position = "bottom",
      axis.text.x = element_blank(),
      axis.title.x = element_blank())

```

p2



~ Environmental fate PLI x rate

```
sum <-
  dat %>%
  group_by(treatment, year) %>%
  summarise(
    n = n(),
    sum = round(x = sum(env_fate_indic_x_rate, na.rm = TRUE), digits = 2)
  ) %>%
  group_by(treatment) %>%
  summarise(
    n = n(),
    # sum = round(sum(total_pli_x_rate, na.rm = TRUE), 2),
    mean = round(mean(sum, na.rm = TRUE), 2),
    sd = round(sd(sum, na.rm = TRUE), 2),
    se = round(sd(sum, na.rm = TRUE) / sqrt(n), 2) # Standard Error
  )
```

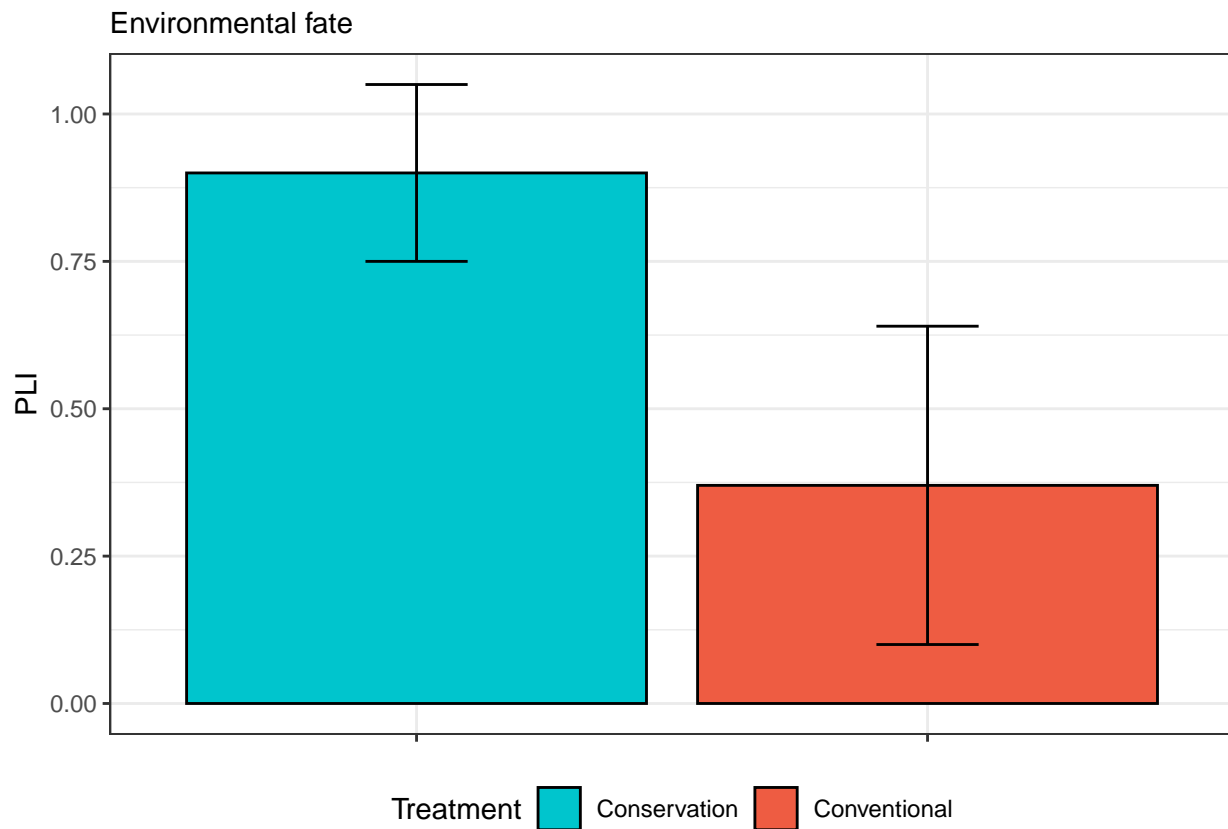
`summarise()` has grouped output by 'treatment'. You can override using the
`.groups` argument.

```
sum
```

```
## # A tibble: 2 x 5
##   treatment      n mean   sd   se
##   <fct>      <int> <dbl> <dbl> <dbl>
## 1 Conservation     3  0.9  0.26  0.15
## 2 Conventional     3  0.37  0.46  0.27
```

```
p3 <-
ggplot(data = sum,
       aes(x = treatment,
           y = mean,
           fill = treatment)) +
geom_bar(stat = "identity",
       color = "black",
       position = "dodge") +
labs(
  subtitle = "Environmental fate",
  x = "Treatment",
  y = "PLI") +
theme_bw() +
scale_fill_manual(values=c("turquoise3", "tomato2"),
  name = "Treatment") +
geom_errorbar(aes(ymin=mean-se,
  ymax=mean+se),
  width=.2, # Width of the error bars
  position=position_dodge(.9)) +
theme(strip.text.x = element_text(size = 12,
  color = "black",
  face = "bold.italic"),
  legend.position = "bottom",
  axis.text.x = element_blank(),
  axis.title.x = element_blank())
```

p3



~ Human health PLI x rate

```
names(dat)
```

```
## [1] "X"                "year"
## [3] "treatment"        "crop"
## [5] "ai_name"          "category"
## [7] "avg_normalized_rate_kg_ha" "DT50_norm"
## [9] "SCI_GROW_norm"    "BCF_norm"
## [11] "Env_Fate_Indicator" "fish_weighted"
## [13] "daphnia_weighted"  "worm_weighted"
## [15] "birds_weighted"    "mammals_weighted"
## [17] "algae_weighted"    "aq_plants_weighted"
## [19] "bees_weighted"     "EcoTox_Indicator"
## [21] "h_codes"           "PL_HH"
## [23] "Env_Fate_Scaled"   "EcoTox_Scaled"
## [25] "PL_HH_Scaled"      "total_pli"
## [27] "env_fate_indic_x_rate" "ecotox_indic_x_rate"
## [29] "human_health_indic_x_rate" "total_pli_x_rate"
```

```
# sum <-
#   dat %>%
#   group_by(treatment) %>%
#   summarise(
#     n = n(),
```



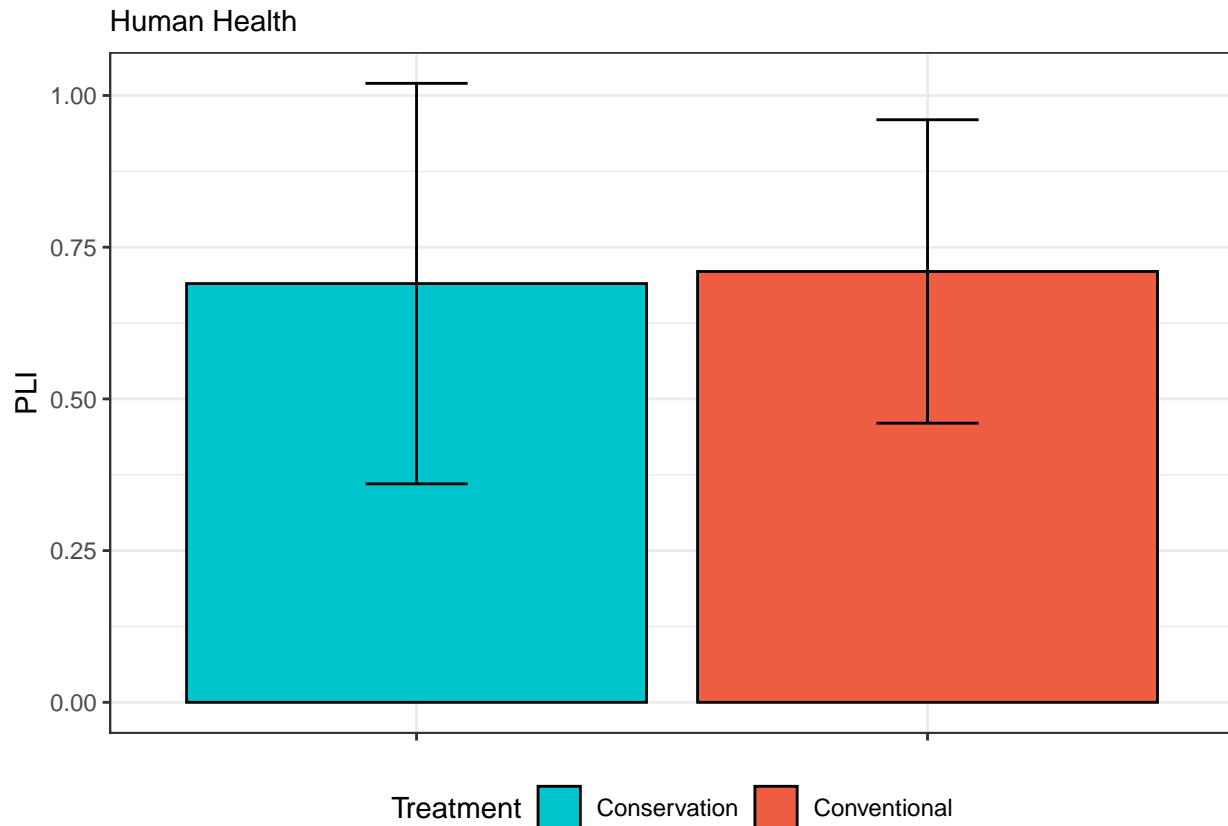
```
#      sum = round(x = sum(human_health_indic_x_rate, na.rm = TRUE), digits = 2)
#    )

sum <-
  dat %>%
  group_by(treatment, year) %>%
  summarise(
    n = n(),
    sum = round(x = sum(human_health_indic_x_rate, na.rm = TRUE), digits = 2)
  ) %>%
  group_by(treatment) %>%
  summarise(
    n = n(),
    # sum = round(sum(total_pli_x_rate, na.rm = TRUE), 2),
    mean = round(mean(sum, na.rm = TRUE), 2),
    sd = round(sd(sum, na.rm = TRUE), 2),
    se = round(sd(sum, na.rm = TRUE) / sqrt(n), 2) # Standard Error
  )
```

`summarise()` has grouped output by 'treatment'. You can override using the
`.groups` argument.

```
p4 <-
ggplot(data = sum,
       aes(x = treatment,
           y = mean,
           fill = treatment)) +
geom_bar(stat = "identity",
        color = "black",
        position = "dodge") +
labs(
  subtitle = "Human Health",
  x = "Treatment",
  y = "PLI") +
theme_bw() +
scale_fill_manual(values=c("turquoise3", "tomato2"),
                  name = "Treatment") +
geom_errorbar(aes(ymin=mean-se,
                  ymax=mean+se),
              width=.2, # Width of the error bars
              position=position_dodge(.9)) +
theme(strip.text.x = element_text(size = 12,
                                   color = "black",
                                   face = "bold.italic"),
      legend.position = "bottom",
      axis.text.x = element_blank(),
      axis.title.x = element_blank())
```

p4



~ Total mass PLI x rate

```
names(dat)
```

```
## [1] "X" "year"
## [3] "treatment" "crop"
## [5] "ai_name" "category"
## [7] "avg_normalized_rate_kg_ha" "DT50_norm"
## [9] "SCI_GROW_norm" "BCF_norm"
## [11] "Env_Fate_Indicator" "fish_weighted"
## [13] "daphnia_weighted" "worm_weighted"
## [15] "birds_weighted" "mammals_weighted"
## [17] "algae_weighted" "aq_plants_weighted"
## [19] "bees_weighted" "EcoTox_Indicator"
## [21] "h_codes" "PL_HH"
## [23] "Env_Fate_Scaled" "EcoTox_Scaled"
## [25] "PL_HH_Scaled" "total_pli"
## [27] "env_fate_indic_x_rate" "ecotox_indic_x_rate"
## [29] "human_health_indic_x_rate" "total_pli_x_rate"
```

```
# sum <-
#   dat %>%
#   group_by(treatment) %>%
#   summarise(
#     n = n(),
#     sum = round(x = sum(avg_normalized_rate_kg_ha, na.rm = TRUE), digits = 2)
#   )
```

```

sum <-
  dat %>%
  group_by(treatment, year) %>%
  summarise(
    n = n(),
    sum = round(x = sum(avg_normalized_rate_kg_ha, na.rm = TRUE), digits = 2)
  ) %>%
  group_by(treatment) %>%
  summarise(
    n = n(),
    # sum = round(sum(total_pli_x_rate, na.rm = TRUE), 2),
    mean = round(mean(sum, na.rm = TRUE), 2),
    sd = round(sd(sum, na.rm = TRUE), 2),
    se = round(sd(sum, na.rm = TRUE) / sqrt(n), 2) # Standard Error
  )

```

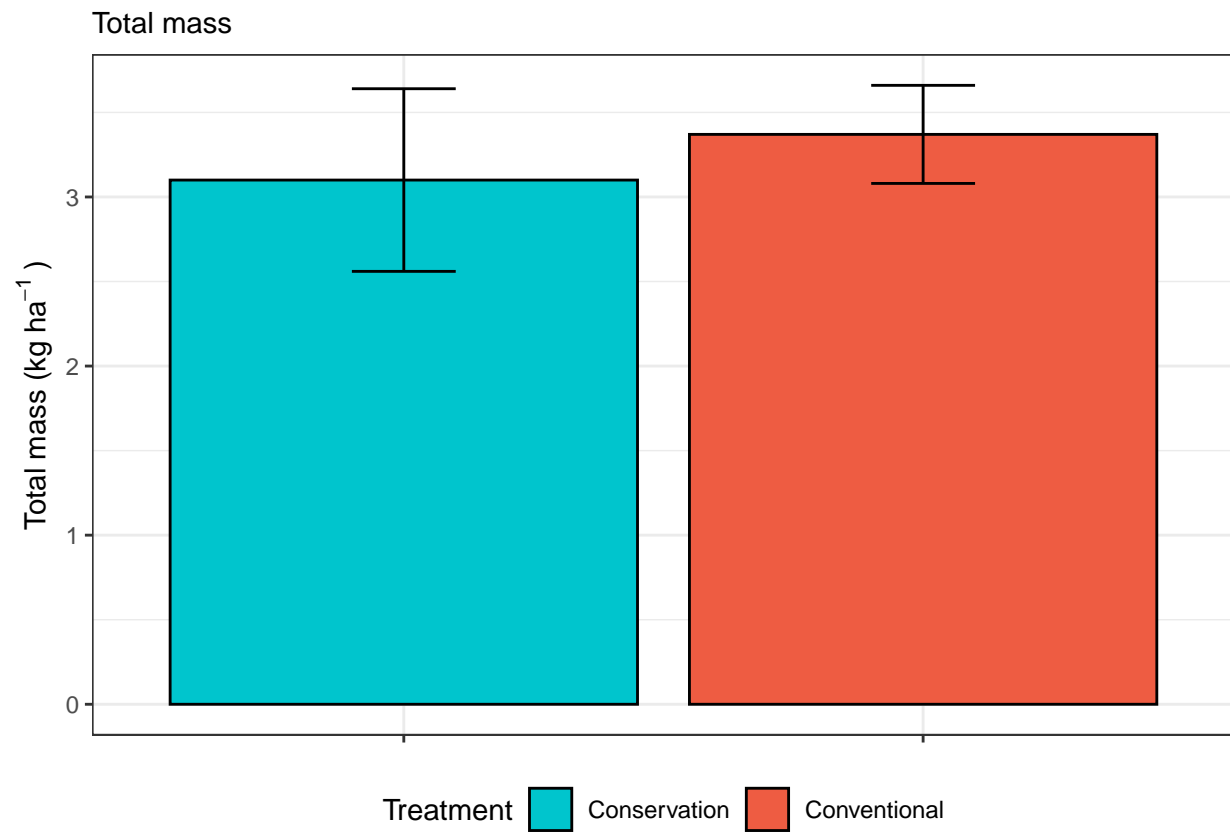
`summarise()` has grouped output by 'treatment'. You can override using the
 ## `.groups` argument.

```

p5 <-
ggplot(data = sum,
       aes(x = treatment,
           y = mean,
           fill = treatment)) +
geom_bar(stat = "identity",
        color = "black",
        position = "dodge") +
labs(
  subtitle = "Total mass",
  x = "Treatment",
  y = expression("Total mass (kg" ~ "ha" ^ {-1} ~ ")")) +
theme_bw() +
scale_fill_manual(values=c("turquoise3", "tomato2"),
                  name = "Treatment") +
geom_errorbar(aes(ymin=mean-se,
                  ymax=mean+se),
              width=.2, # Width of the error bars
              position=position_dodge(.9)) +
theme(strip.text.x = element_text(size = 12,
                                   color = "black",
                                   face = "bold.italic"),
      legend.position = "bottom",
      axis.text.x = element_blank(),
      axis.title.x = element_blank())

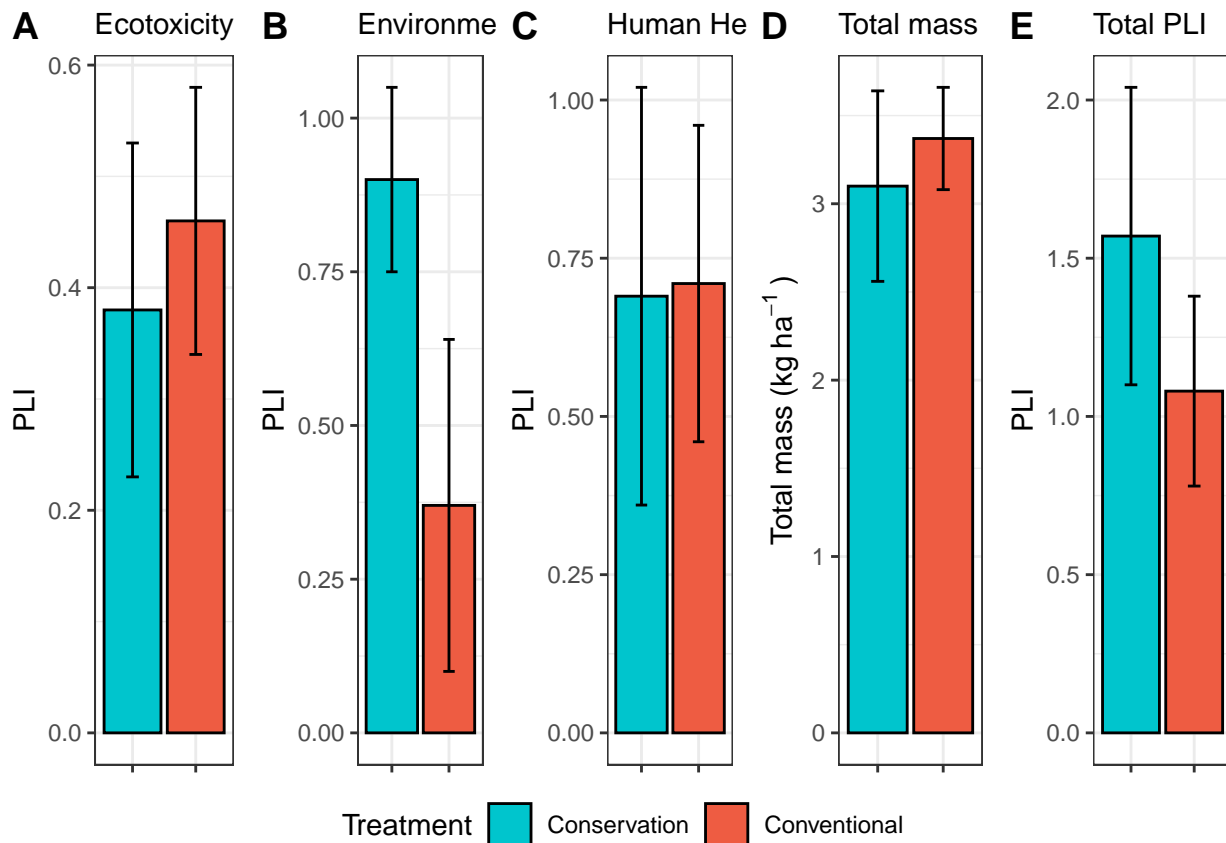
```

p5



joint plot

```
ggarrange(p2,p3,p4,p5,p1,  
  ncol = 5,  
  nrow = 1,  
  labels = c("A","B","C","D","E"),  
  common.legend = TRUE, legend = "bottom")
```



```
ggsave(filename = "sym_link_pesticide_data/plots/pesticide_risk_plots/danish_pli_treatment_joint_plot.pdf",
        width = 10, height = 3.5)
```

```
# _____ # Proportions
```

~ application rates

```
# Summarize total usage by category and treatment
category_summary <- dat %>%
  group_by(treatment, category) %>% # Group by both treatment and category
  summarise(total_usage = sum(avg_normalized_rate_kg_ha, na.rm = TRUE), .groups = "drop") %>%
  group_by(treatment) %>% # Group again by treatment to calculate proportions correctly
  mutate(percentage = (total_usage / sum(total_usage)) * 100) %>%
  ungroup()
```

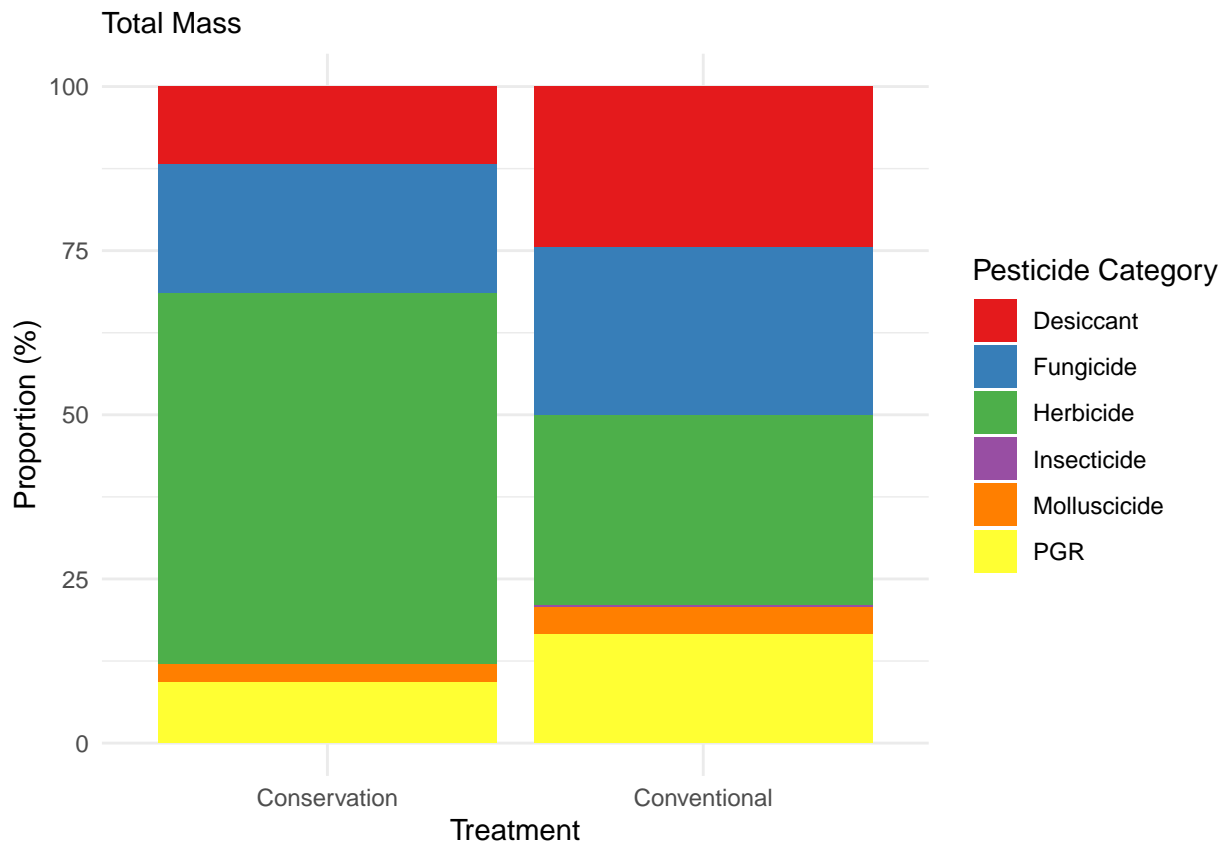
```
category_summary
```

```
## # A tibble: 11 x 4
##   treatment category total_usage percentage
##   <fct>      <chr>      <dbl>      <dbl>
## 1 Conservation Desiccant      1.09      11.8
## 2 Conservation Fungicide      1.83      19.7
## 3 Conservation Herbicide      5.25     56.5
## 4 Conservation Molluscicide    0.253      2.72
## 5 Conservation PGR             0.860      9.26
## 6 Conventional Desiccant      2.48     24.5
```

```
## 7 Conventional Fungicide      2.59      25.6
## 8 Conventional Herbicide      2.93      28.9
## 9 Conventional Insecticide    0.035     0.346
## 10 Conventional Molluscicide  0.424     4.18
## 11 Conventional PGR           1.67     16.5
```

```
a <-
ggplot(category_summary, aes(x = treatment, y = percentage, fill = category)) +
  geom_col(position = "stack") + # Stacked bars
  scale_fill_brewer(palette = "Set1") + # Change to any Brewer palette
  labs(subtitle = "Total Mass",
       x = "Treatment",
       y = "Proportion (%)",
       fill = "Pesticide Category") +
  theme_minimal()
```

a



~ total PLI

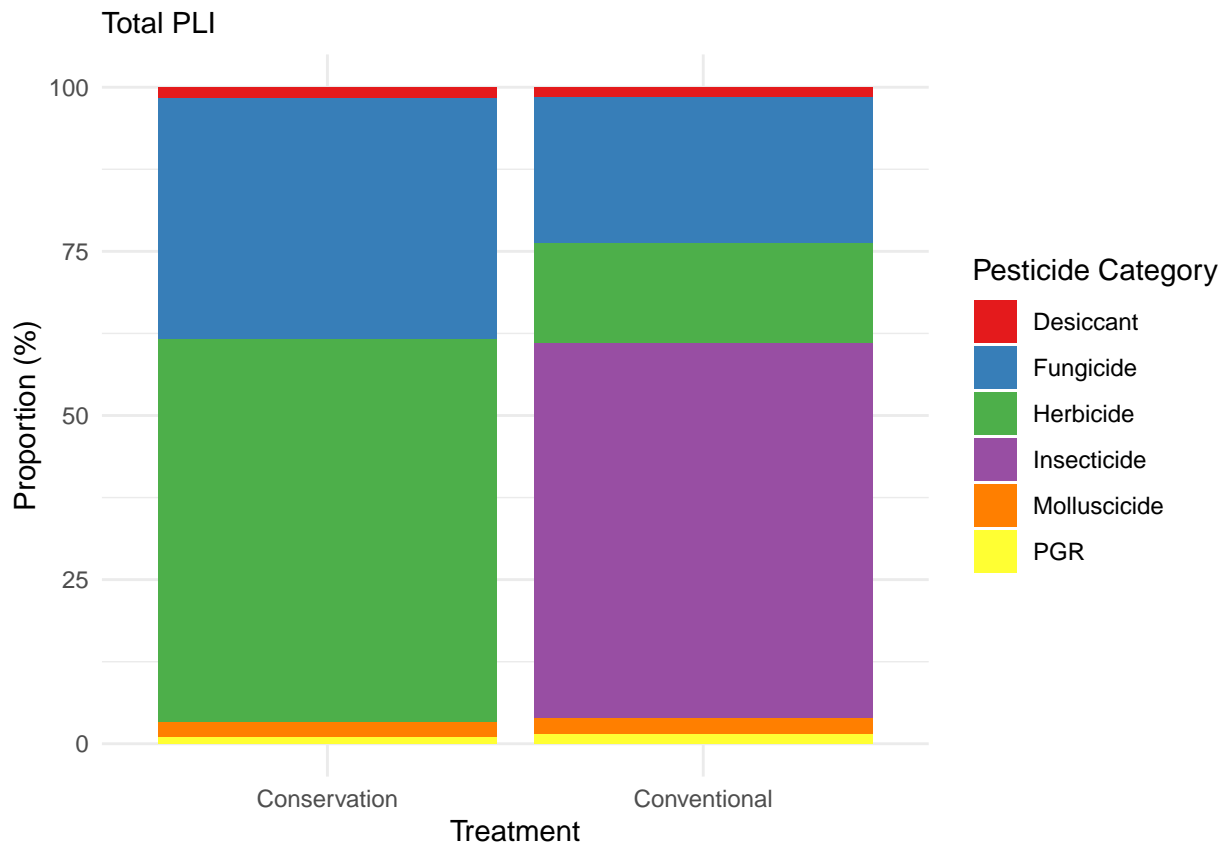
```
# Summarize total usage by category and treatment
category_summary <- dat %>%
  group_by(treatment, category) %>% # Group by both treatment and category
  summarise(total_x = sum(total_pli, na.rm = TRUE), .groups = "drop") %>%
  group_by(treatment) %>% # Group again by treatment to calculate proportions correctly
  mutate(percentage = (total_x / sum(total_x)) * 100) %>%
  ungroup()
```

```
category_summary
```

```
## # A tibble: 11 x 4
##   treatment    category    total_x percentage
##   <fct>        <chr>        <dbl>      <dbl>
## 1 Conservation Desiccant    0.164      1.74
## 2 Conservation Fungicide    3.45      36.7
## 3 Conservation Herbicide    5.47      58.2
## 4 Conservation Molluscicide 0.218      2.33
## 5 Conservation PGR          0.0934     0.994
## 6 Conventional Desiccant    0.328      1.51
## 7 Conventional Fungicide    4.84      22.3
## 8 Conventional Herbicide    3.33      15.3
## 9 Conventional Insecticide  12.4      57.0
## 10 Conventional Molluscicide 0.546      2.51
## 11 Conventional PGR         0.306      1.41
```

```
b <-
ggplot(category_summary,
  aes(x = treatment,
      y = percentage,
      fill = category)) +
  geom_col(position = "stack") + # Stacked bars
  scale_fill_brewer(palette = "Set1") + # Change to any Brewer palette
  labs(subtitle = "Total PLI",
      x = "Treatment",
      y = "Proportion (%)",
      fill = "Pesticide Category") +
  theme_minimal()
```

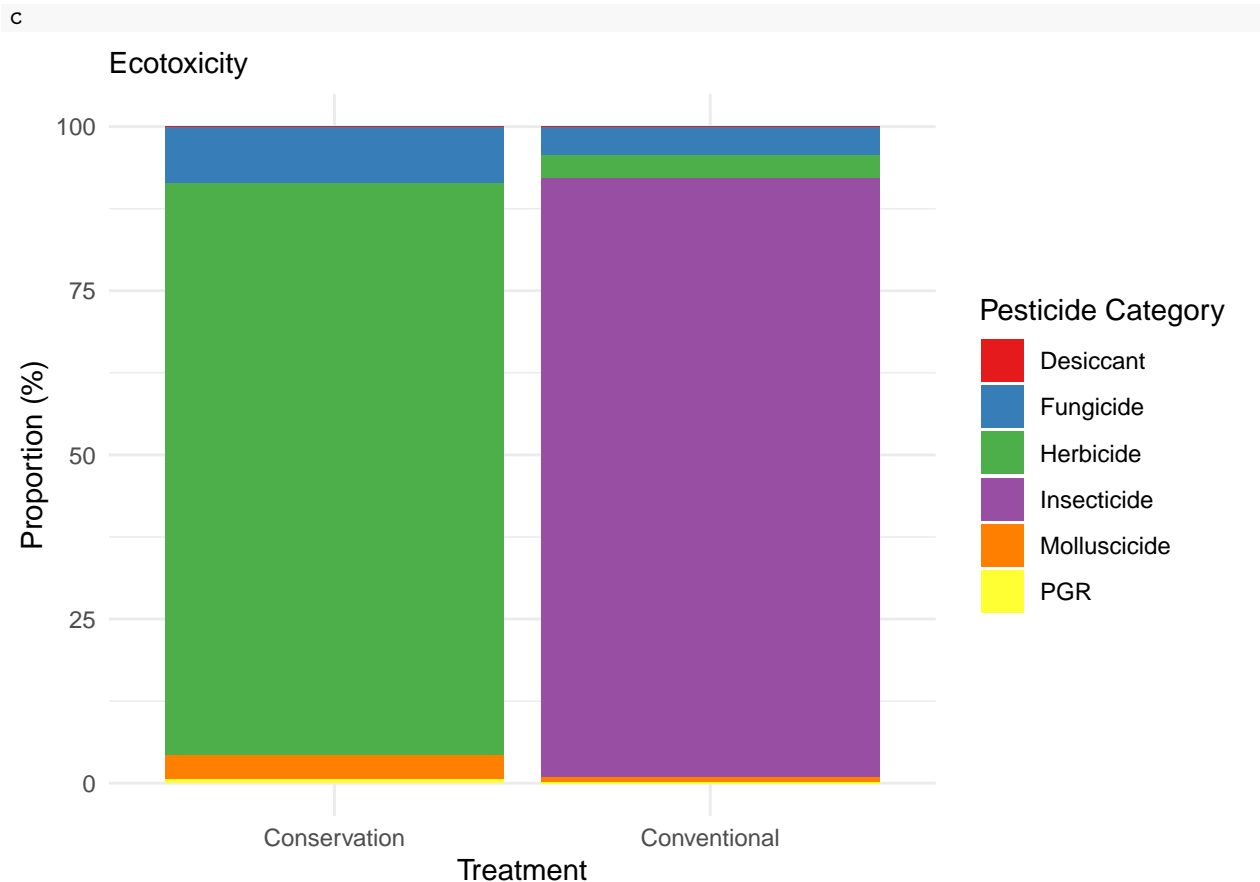
b



~ ecotoxicity PLI

```
# Summarize total usage by category and treatment
category_summary <- dat %>%
  group_by(treatment, category) %>% # Group by both treatment and category
  summarise(total_x = sum(EcoTox_Indicator, na.rm = TRUE), .groups = "drop") %>%
  group_by(treatment) %>% # Group again by treatment to calculate proportions correctly
  mutate(percentage = (total_x / sum(total_x)) * 100) %>%
  ungroup()

c <-
ggplot(category_summary,
  aes(x = treatment,
    y = percentage,
    fill = category)) +
  geom_col(position = "stack") + # Stacked bars
  scale_fill_brewer(palette = "Set1") + # Change to any Brewer palette
  labs(subtitle = "Ecotoxicity",
    x = "Treatment",
    y = "Proportion (%)",
    fill = "Pesticide Category") +
  theme_minimal()
```

~ environmental fate PLI

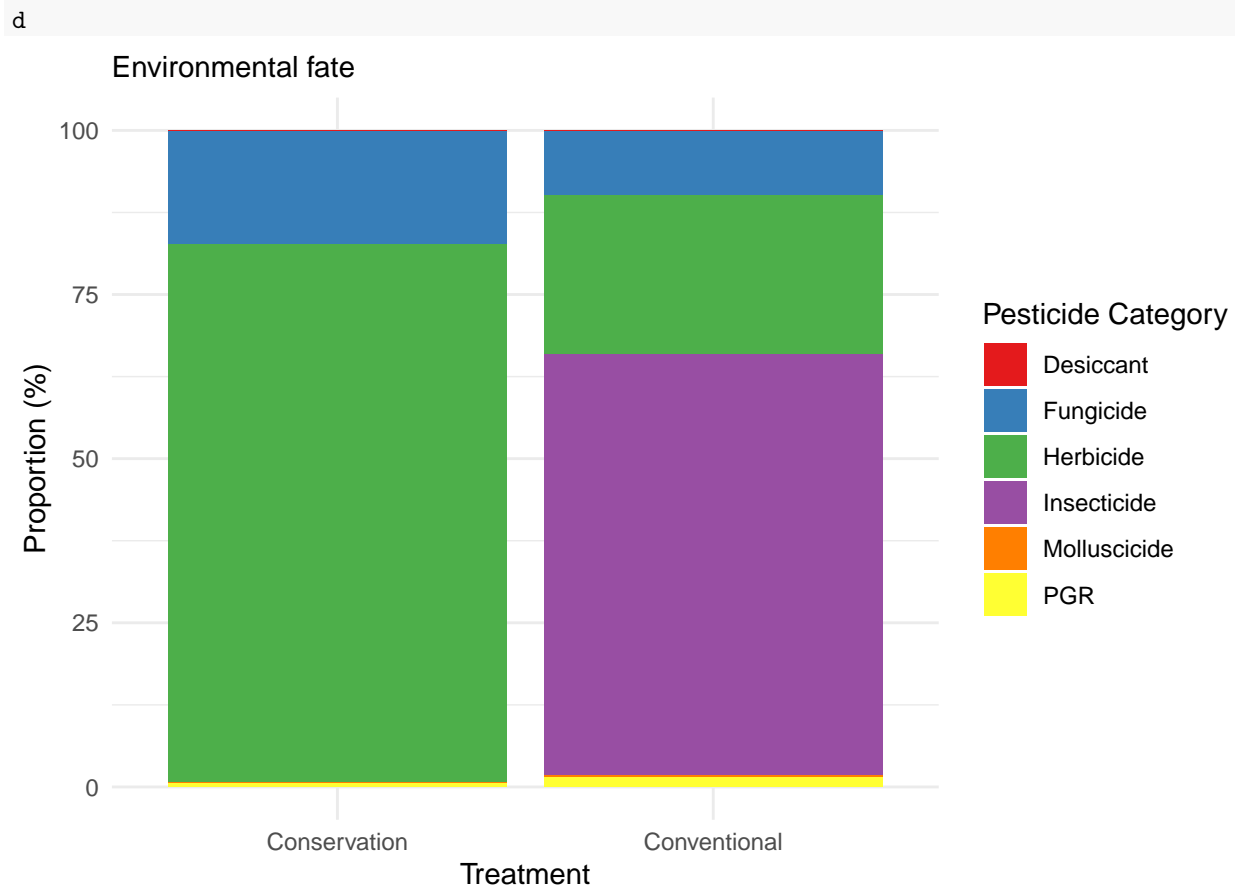
```
# Summarize total usage by category and treatment
category_summary <- dat %>%
  group_by(treatment, category) %>% # Group by both treatment and category
  summarise(total_x = sum(Env_Fate_Indicator, na.rm = TRUE), .groups = "drop") %>%
  group_by(treatment) %>% # Group again by treatment to calculate proportions correctly
  mutate(percentage = (total_x / sum(total_x)) * 100) %>%
  ungroup()
```

category_summary

```
## # A tibble: 11 x 4
##   treatment    category total_x percentage
##   <fct>        <chr>      <dbl>      <dbl>
## 1 Conservation Desiccant    0.00235    0.0499
## 2 Conservation Fungicide    0.819     17.3
## 3 Conservation Herbicide    3.86      81.8
## 4 Conservation Molluscicide 0.00966    0.205
## 5 Conservation PGR          0.0271    0.574
## 6 Conventional Desiccant    0.00471    0.0615
## 7 Conventional Fungicide    0.753     9.84
## 8 Conventional Herbicide    1.85      24.2
## 9 Conventional Insecticide  4.91      64.2
```

```
## 10 Conventional Molluscicide 0.0242    0.316
## 11 Conventional PGR           0.106    1.38
```

```
d <-
ggplot(category_summary,
  aes(x = treatment,
      y = percentage,
      fill = category)) +
  geom_col(position = "stack") + # Stacked bars
  scale_fill_brewer(palette = "Set1") + # Change to any Brewer palette
  labs(subtitle = "Environmental fate",
       x = "Treatment",
       y = "Proportion (%)",
       fill = "Pesticide Category") +
  theme_minimal()
```



~ Human health PLI

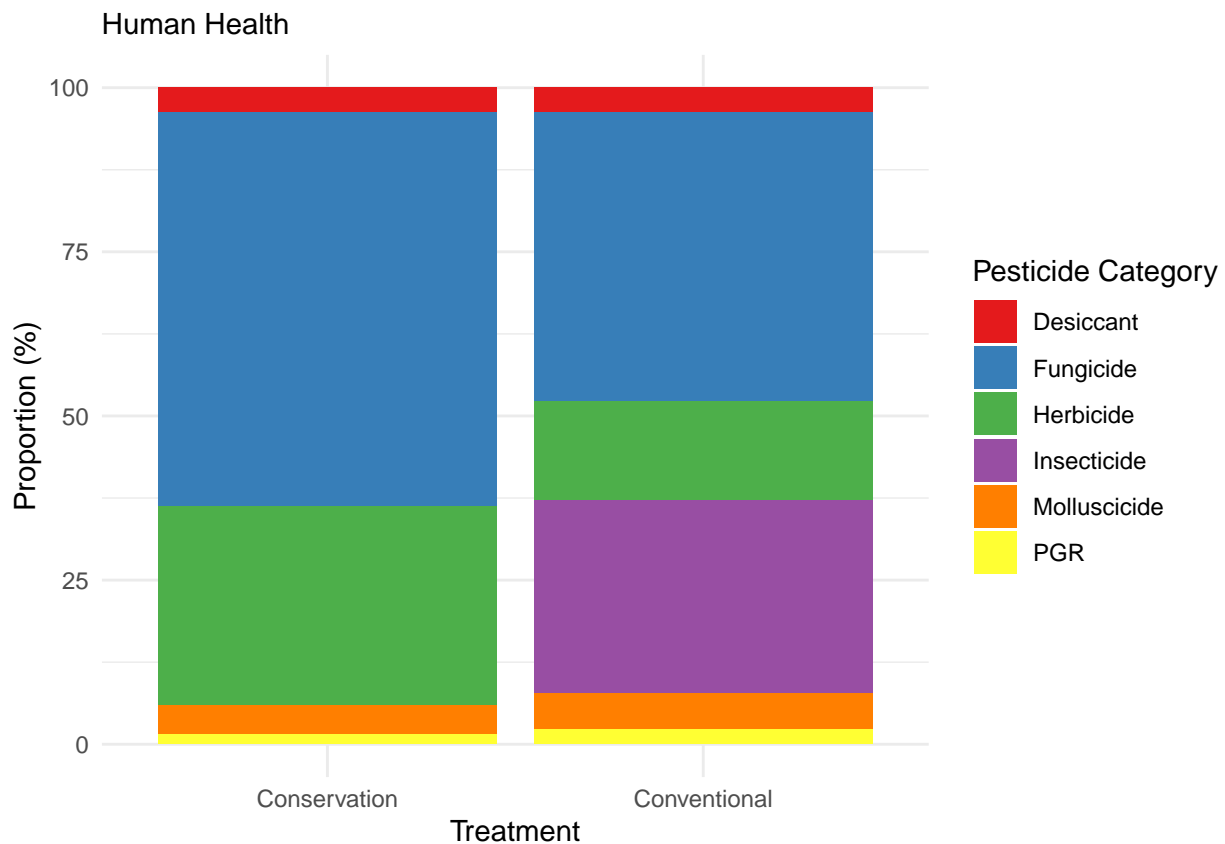
```
# Summarize total usage by category and treatment
category_summary <- dat %>%
  group_by(treatment, category) %>% # Group by both treatment and category
  summarise(total_x = sum(PL_HH, na.rm = TRUE), .groups = "drop") %>%
  group_by(treatment) %>% # Group again by treatment to calculate proportions correctly
  mutate(percentage = (total_x / sum(total_x)) * 100) %>%
```

```

ungroup()

e <-
ggplot(category_summary,
  aes(x = treatment,
      y = percentage,
      fill = category)) +
  geom_col(position = "stack") + # Stacked bars
  scale_fill_brewer(palette = "Set1") + # Change to any Brewer palette
  labs(subtitle = "Human Health",
      x = "Treatment",
      y = "Proportion (%)",
      fill = "Pesticide Category") +
  theme_minimal()
e

```



joint plot

```

library(ggpubr)

# Modify each plot to rotate x-axis labels
a <- a + theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.title = element_blank())

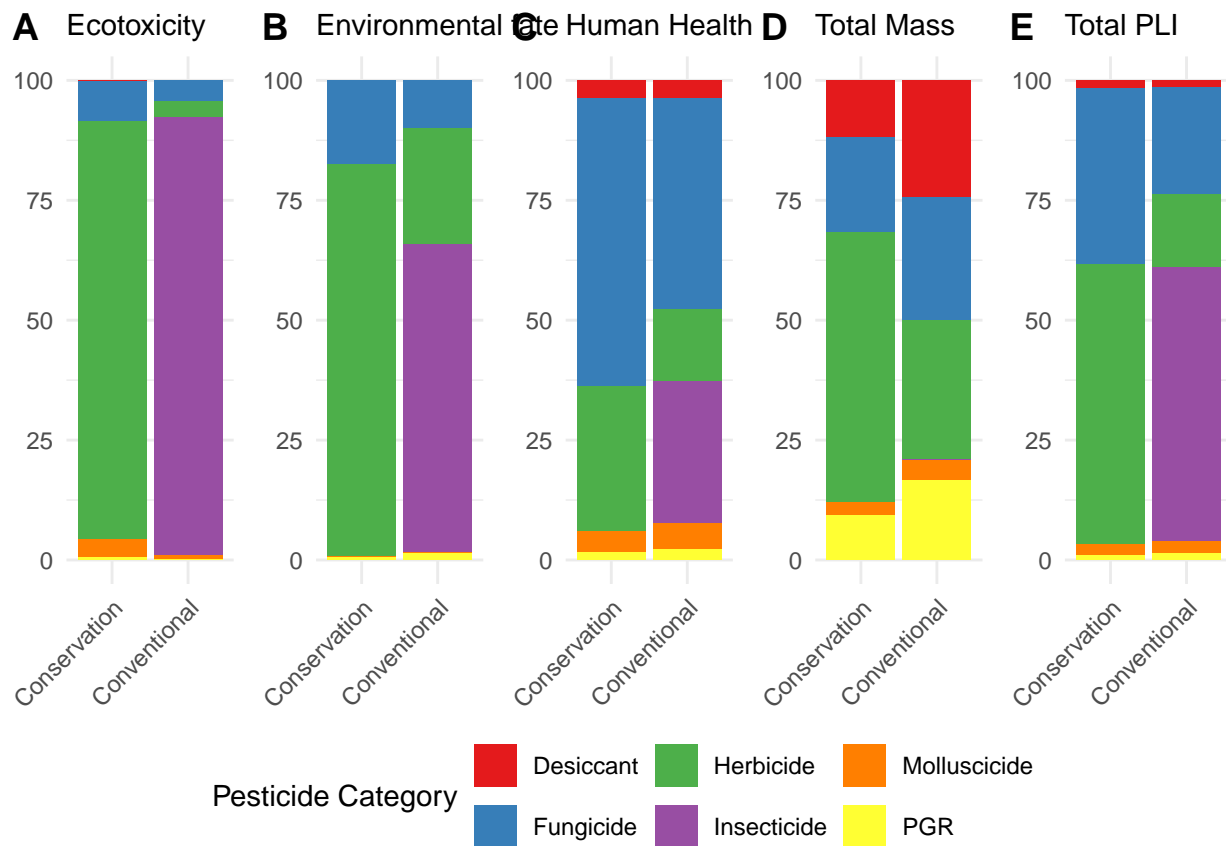
```

```

b <- b + theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.title = element_blank())
c <- c + theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.title = element_blank())
d <- d + theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.title = element_blank())
e <- e + theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.title = element_blank())

# Arrange the plots with ggarrange
ggarrange(c, d, e, a, b,
          ncol = 5,
          nrow = 1,
          labels = c("A", "B", "C", "D", "E"),
          common.legend = TRUE, legend = "bottom")

```



```

ggsave(filename = "sym_link_pesticide_data/plots/pesticide_risk_plots/danish_pli_proportion_joint_plot.",
        width = 9, height = 4.5)

```