Checkpoint # 2

Group # 3

2024-11-24

Research Question

Considering climate factors and chemical factors in agriculture, construction of which continent and what crop type would bring more economic impact?

Hypothesis

Null Hypothesis : There is no significant difference in economic impact across different continents and crop types based on climate factors (average temperature, precipitation, extreme weather events), chemical factors (fertilizer use, pesticide use) and crop yield.

Alternative Hypothesis : Economic impact significantly differs across continents and crop types due to variations in climate factors and chemical inputs.

Libraries

```
library(dplyr)
```

```
##
##          : 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(modelr)
library(boot)
library(randomForest)
```

```
## Warning:    'randomForest'  R   4.4.2
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
##         : 'randomForest'
##
## The following object is masked from 'package:ggplot2':
##
##     margin
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(agricolae)
```

Import Dataset

```r
data <- read.csv("climate_change_impact_on_agriculture_2024.csv")
```

Preparation and cleaning the data (Juhyun Lee)

```r
# mutate continent columns

country_to_continent <- data.frame(
  Country = c("Argentina", "Australia", "Brazil","Canada","China","France",
              "India","Nigeria", "Russia","USA"),
  Continent = c("South America", "Oceania", "South America",
                "North America", "East Asia", "Europe",
                "South Asia", "Africa","Eurasia","North America")
)

data_with_continent <- data %>%
  left_join(country_to_continent, by = "Country")

data <- data_with_continent %>%
  select(Year, Country, Continent, Region, everything())
```

```r
aggregated_data <- data %>%
  group_by(Year, Continent, Crop_Type) %>%
  summarize(
    avg_crop_yield =
      mean(Crop_Yield_MT_per_HA, na.rm = TRUE),
    avg_extreme_weather_events =
      mean(Extreme_Weather_Events, na.rm= TRUE),
    avg_temp_c =
      mean(Average_Temperature_C, na.rm = TRUE),
    avg_total_precipitation_mm =
      mean(Total_Precipitation_mm, na.rm = TRUE),
    avg_co2_emissions_mt =
      mean(CO2_Emissions_MT, na.rm =TRUE),
    avg_pesticide_use_kg_per_ha =
      mean(Pesticide_Use_KG_per_HA, na.rm=TRUE),
    avg_fertilizer_use_kg_per_ha =
      mean(Fertilizer_Use_KG_per_HA, na.rm =TRUE),
    avg_soil_health_index =
      mean(Soil_Health_Index, na.rm=TRUE),
    avg_economic_impact_million_usd =
      mean(Economic_Impact_Million_USD, na.rm = TRUE)
  ) %>%
  ungroup()
```

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

```r
data <- data %>%
  left_join(aggregated_data, by = c("Year", "Continent"))
```

```
## Warning in left_join(., aggregated_data, by = c("Year", "Continent")): Detected an unexpected many-t
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 592 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

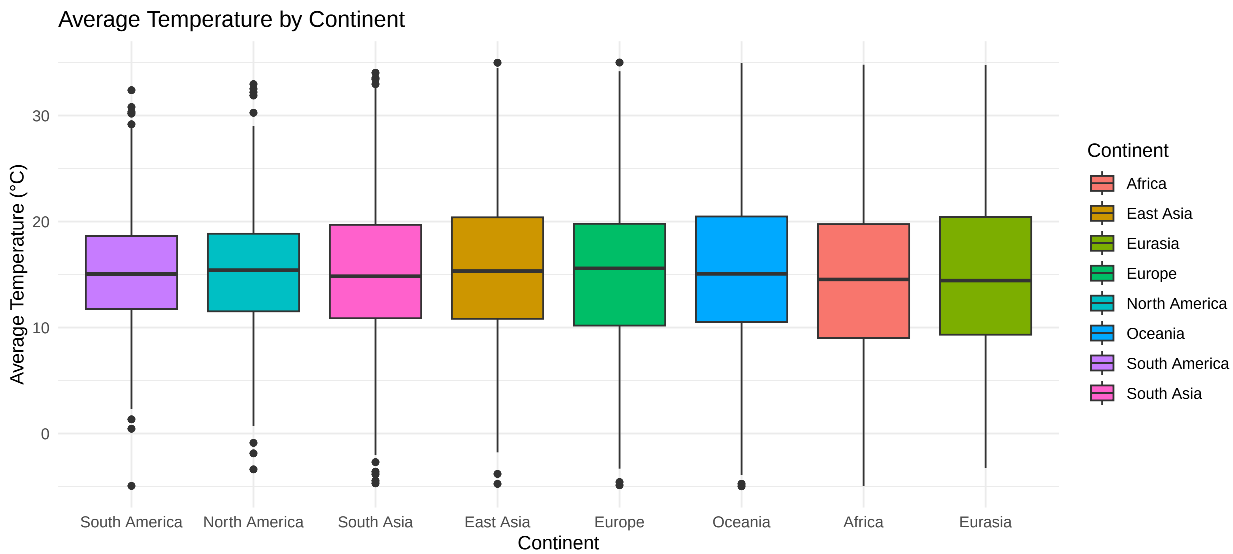
```r
data_constracted <- data %>%
  select(-c(6:9, 12:14, 16))
```

EDA by continent (Daehee Cho, Donghyun Park)

Average_Temperature by Year

```r
ggplot(aggregated_data) +
  geom_boxplot(mapping = aes(x = reorder(Continent, avg_temp_c, FUN = IQR),
                             y = avg_temp_c, fill = Continent)) +
  labs(
    title = "Average Temperature by Continent",
    x = "Continent",
    y = "Average Temperature (°C)"
```

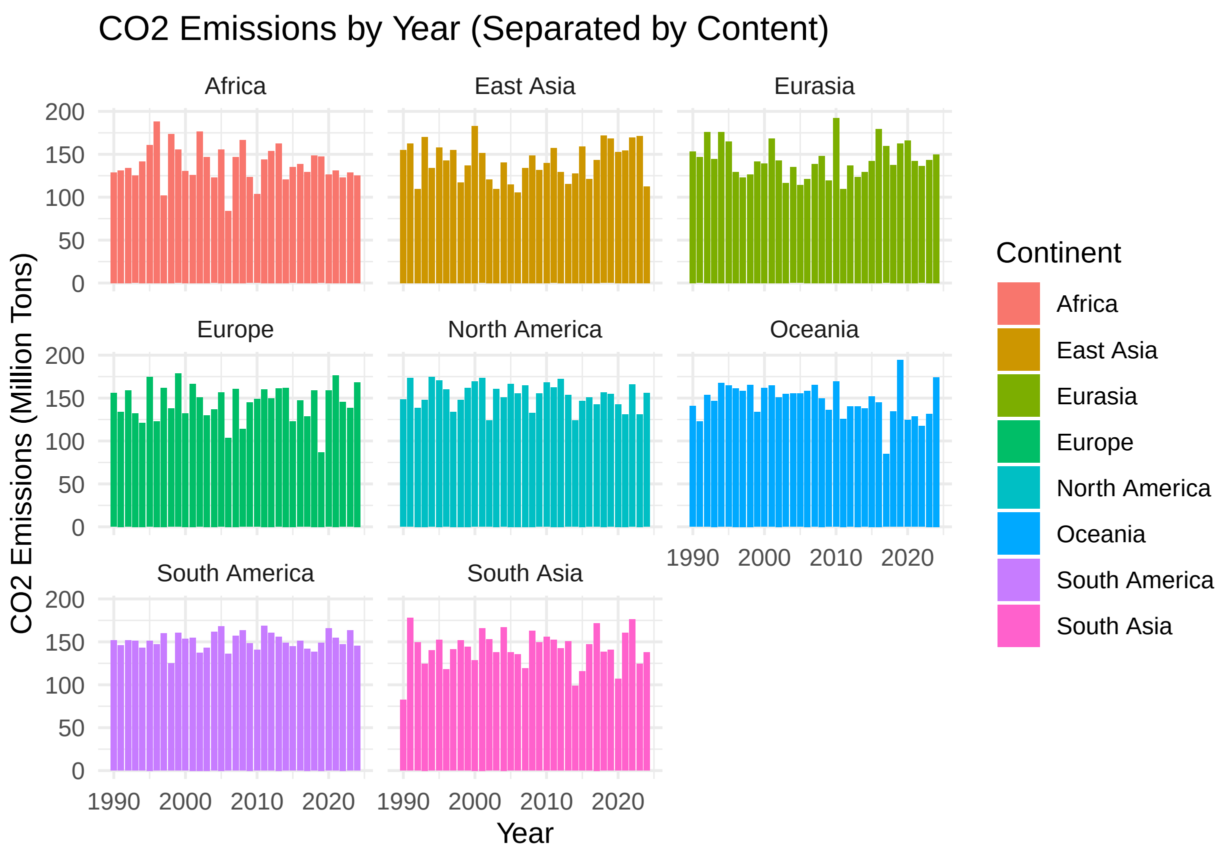
```
) +
theme_minimal()
```



CO2 Emissions by Year

```
ggplot(aggregated_data, aes(x = Year, y =  avg_co2_emissions_mt,
                            fill = Continent)) +
  geom_histogram(stat = "identity") +
  facet_wrap(~ Continent) +
  labs(title = "CO  Emissions by Year (Separated by Content)",
       x = "Year",
       y = "CO  Emissions (Million Tons)",
       fill = "Continent"
  ) +
  theme_minimal()
```

```
## Warning in geom_histogram(stat = "identity"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```

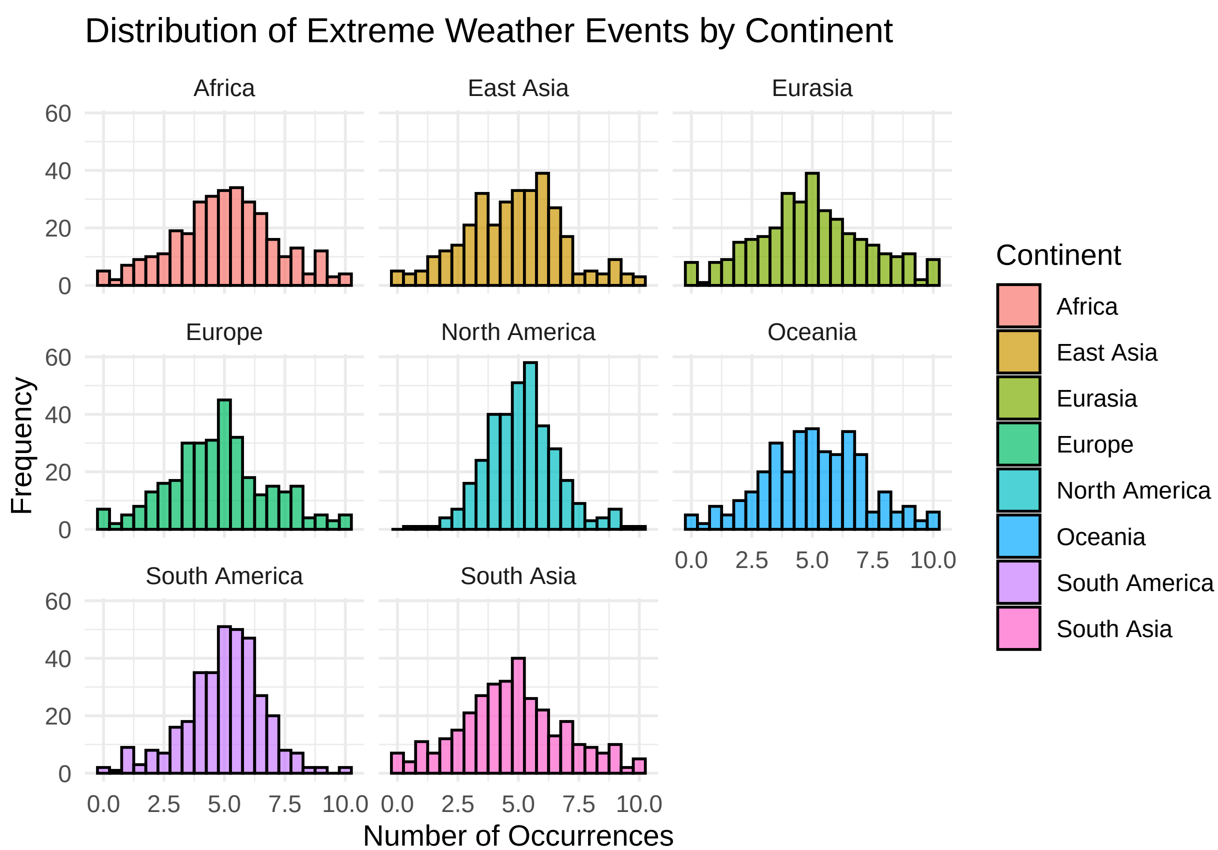



```
co2_stability <- aggregated_data %>%
  group_by(Continent) %>%
  summarize(
    variance = var(avg_co2_emissions_mt, na.rm = TRUE),
    std_dev1 = sd(avg_co2_emissions_mt, na.rm = TRUE),
    mean_co2 = mean(avg_co2_emissions_mt, na.rm = TRUE),
    cv = std_dev1 / mean_co2
  )

print(co2_stability)
```

```
## # A tibble: 8 x 5
##   Continent     variance std_dev1 mean_co2    cv
##   <chr>            <dbl>    <dbl>    <dbl> <dbl>
## 1 Africa            32.1     5.67     14.9 0.380
## 2 East Asia         32.8     5.73     15.0 0.381
## 3 Eurasia           39.7     6.30     15.1 0.418
## 4 Europe            32.1     5.67     15.6 0.363
## 5 North America     18.2     4.27     15.4 0.277
## 6 Oceania           33.4     5.78     15.3 0.376
## 7 South America     14.7     3.83     15.1 0.254
## 8 South Asia        34.7     5.89     15.1 0.390
```

Extreme weather by year

```r
ggplot(aggregated_data, aes(x = avg_extreme_weather_events, fill = Continent))+
  geom_histogram(binwidth = 0.5, color = "black",
                 alpha = 0.7, position = "identity") +
  facet_wrap(~ Continent) +
  labs(
    title = "Distribution of Extreme Weather Events by Continent",
    x = "Number of Occurrences",
    y = "Frequency",
    fill = "Continent"
  ) +
  theme_minimal()
```



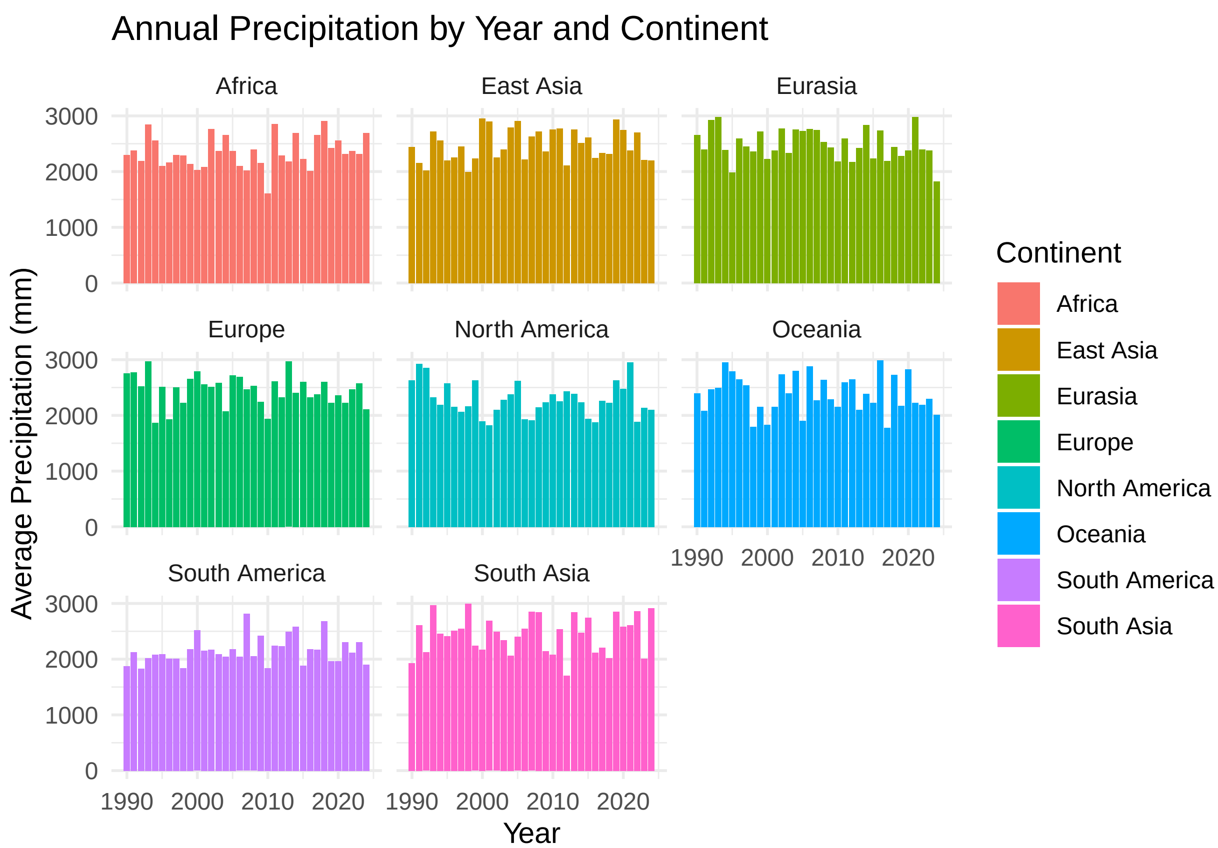

```r
variance_std <- aggregated_data %>%
  group_by(Continent) %>%
  summarize(
    variance = var(avg_extreme_weather_events, na.rm = TRUE),
    std_dev = sd(avg_extreme_weather_events, na.rm = TRUE)
  )

# Print the results
print(variance_std)
```

```
## # A tibble: 8 x 3
```

```
##   Continent     variance std_dev
##   <chr>            <dbl>   <dbl>
## 1 Africa            4.44    2.11
## 2 East Asia         4.15    2.04
## 3 Eurasia           5.18    2.28
## 4 Europe            4.34    2.08
## 5 North America     2.15    1.47
## 6 Oceania           4.36    2.09
## 7 South America     2.65    1.63
## 8 South Asia        4.73    2.17
```

Precipitation vs Year

```r
ggplot(aggregated_data, aes(x = Year,
                            y = avg_total_precipitation_mm,
                            fill = Continent)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ Continent) +
   labs(
      title = "Annual Precipitation by Year and Continent",
      x = "Year",
      y = "Average Precipitation (mm)",
      fill = "Continent"
   ) +
 theme_minimal()
```




```r
precipitation_stability <- aggregated_data %>%
  group_by(Continent) %>%
  summarize(
    variance = var(avg_total_precipitation_mm, na.rm = TRUE),
    std_dev = sd(avg_total_precipitation_mm, na.rm = TRUE),
    mean_precipitation = mean(avg_total_precipitation_mm, na.rm = TRUE),
    cv = std_dev / mean_precipitation
  )

print(precipitation_stability)
```

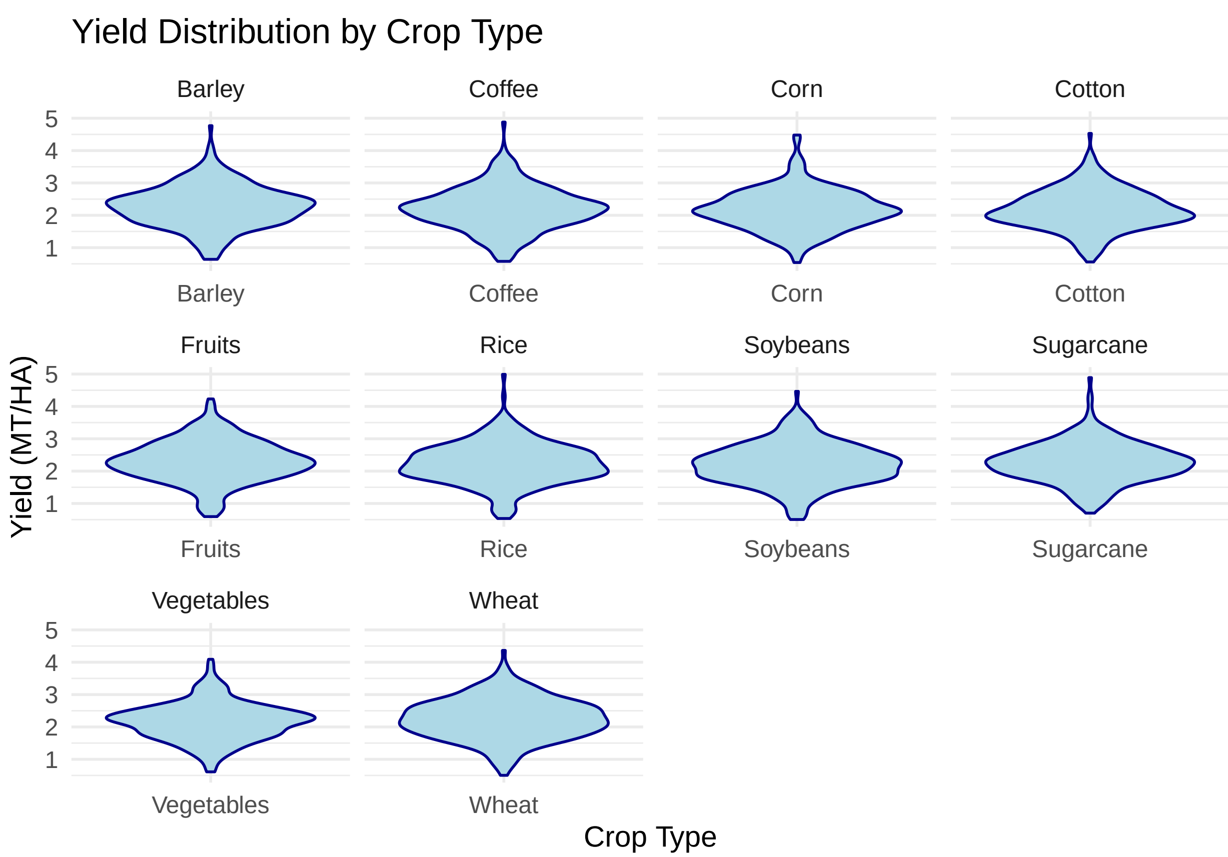
```
## # A tibble: 8 x 5
##   Continent     variance std_dev mean_precipitation    cv
##   <chr>            <dbl>   <dbl>              <dbl> <dbl>
## 1 Africa         256601.    507.              1577. 0.321
## 2 East Asia      290171.    539.              1669. 0.323
## 3 Eurasia        301941.    549.              1583. 0.347
## 4 Europe         318203.    564.              1646. 0.343
## 5 North America  152401.    390.              1645. 0.237
## 6 Oceania        287098.    536.              1601. 0.335
## 7 South America  135801.    369.              1552. 0.237
## 8 South Asia     302928.    550.              1648. 0.334
```

EDA by Crop type (Sumin Chun, Janghee Cho)

```
crop_types <- unique(aggregated_data$Crop_Type)
```

Crop type vs yield

```
ggplot(aggregated_data, aes(x = Crop_Type, y = avg_crop_yield)) +
  geom_violin(fill = "lightblue", color = "darkblue") +
  facet_wrap(~ Crop_Type, scales = "free_x") +
  labs(
    title = "Yield Distribution by Crop Type",
    x = "Crop Type",
    y = "Yield (MT/HA)"
  ) +
  theme_minimal()
```



1. Hypothesis H0: All the mean of average crop yield from different crop types are the same. Ha: At least one of them is different.

2. alpha = 0.05
3. Test = ANOVA Test

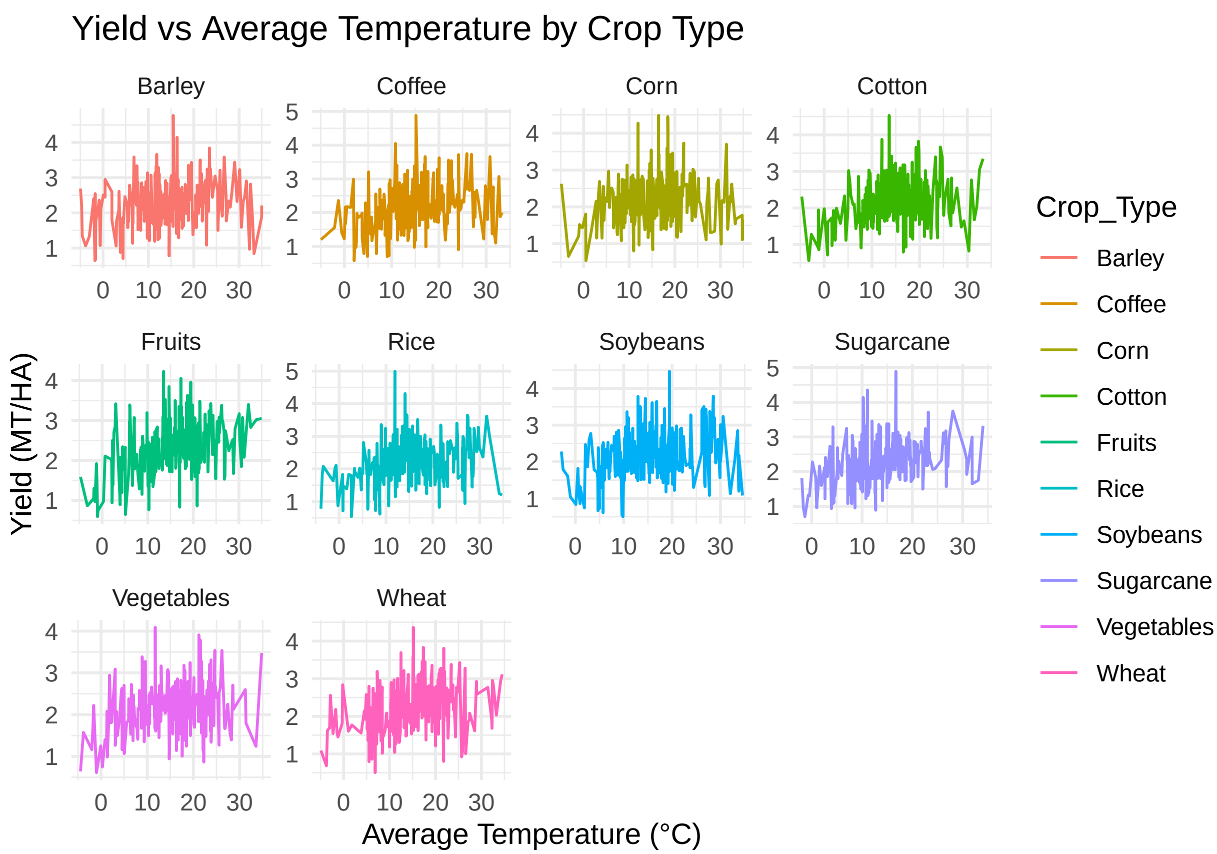
```r
anova_result <- aov(avg_crop_yield ~ Crop_Type, data = aggregated_data)


summary(anova_result)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Crop_Type       9    5.4  0.6041   1.543  0.127
## Residuals    2670 1045.4  0.3915
```

4. Critical Region p-value >= alpha : Do not reject H0. p-value < alpha : Reject H0.
5. Conclusion Since the p-value is larger than alpha, we do not reject the hypothesis. So this plot does not have significant difference.

Average Temperature vs Yield

```r
ggplot(aggregated_data, aes(x = avg_temp_c,
                            y = avg_crop_yield, color = Crop_Type)) +
  geom_line() +
  facet_wrap(~ Crop_Type, scales = "free") +
  labs(
    title = "Yield vs Average Temperature by Crop Type",
    x = "Average Temperature (°C)",
    y = "Yield (MT/HA)"
  ) +
  theme_minimal()
```




```r
slopes_resilience <- aggregated_data %>%
  group_by(Crop_Type) %>%
  summarize(
    slope = coef(lm(avg_crop_yield ~ avg_temp_c, data = cur_data()))[2],
    intercept = coef(lm(avg_crop_yield ~ avg_temp_c, data = cur_data()))[1],
  ) %>%
  arrange(abs(slope))
```

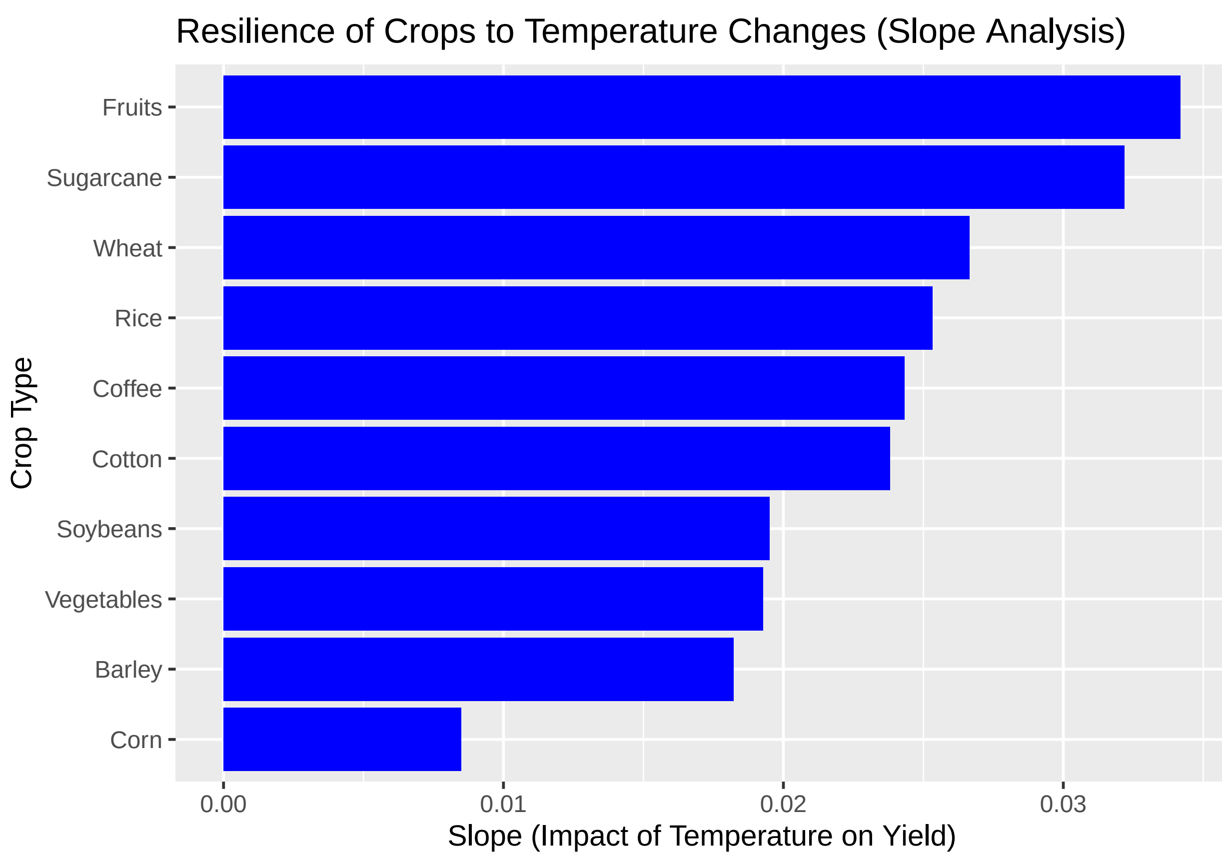
```
## Warning: There was 1 warning in `summarize()`.
## i In argument: `slope = coef(lm(avg_crop_yield ~ avg_temp_c, data =
##   cur_data()))[2]`.
## i In group 1: `Crop_Type = "Barley"`.
## Caused by warning:
## ! `cur_data()` was deprecated in dplyr 1.1.0.
## i Please use `pick()` instead.
```

```r
# Print
print(slopes_resilience)
```

```
## # A tibble: 10 x 3
##    Crop_Type     slope intercept
##    <chr>         <dbl>     <dbl>
##  1 Corn        0.00850      2.04
##  2 Barley      0.0182       2.02
##  3 Vegetables 0.0193       1.88
```

```
## 4 Soybeans   0.0195       1.90
## 5 Cotton     0.0238       1.82
## 6 Coffee     0.0243       1.84
## 7 Rice       0.0253       1.85
## 8 Wheat      0.0266       1.84
## 9 Sugarcane  0.0322       1.80
## 10 Fruits    0.0342       1.77
```

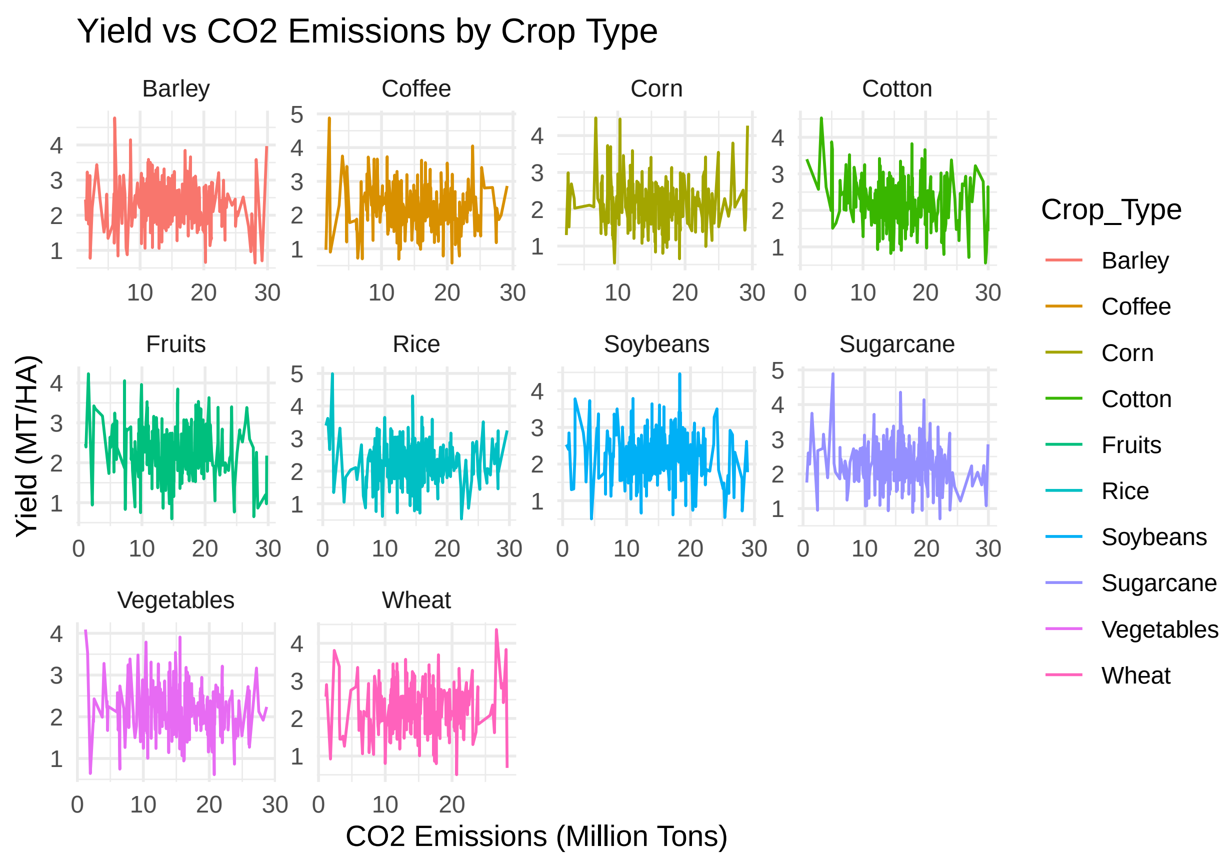
```r
ggplot(slopes_resilience, aes(x = reorder(Crop_Type, abs(slope)),
                              y = slope, fill = slope > 0)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  scale_fill_manual(values = c("blue")) +
  labs(
    title = "Resilience of Crops to Temperature Changes (Slope Analysis)",
    x = "Crop Type",
    y = "Slope (Impact of Temperature on Yield)"
  )
```



CO2 Emissions vs Yield

```r
ggplot(aggregated_data, aes(x = avg_co2_emissions_mt,
                            y = avg_crop_yield, color = Crop_Type)) +
  geom_line() +
```

```
  facet_wrap(~ Crop_Type, scales = "free") +
  labs(
    title = "Yield vs CO  Emissions by Crop Type",
    x = "CO  Emissions (Million Tons)",
    y = "Yield (MT/HA)"
  ) +
  theme_minimal()
```

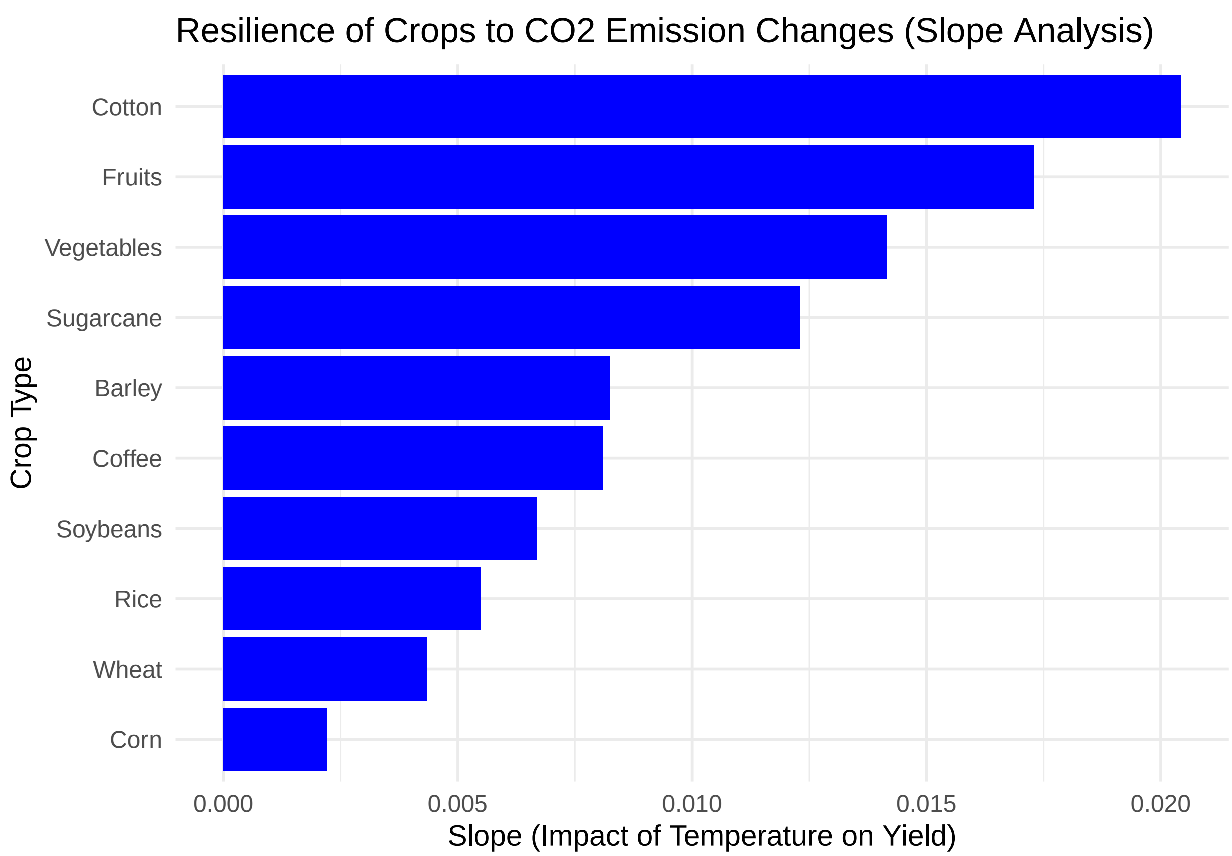



```
slopes_resilience1 <- aggregated_data %>%
  group_by(Crop_Type) %>%
  summarize(
    slope = abs(coef(lm(avg_crop_yield ~ avg_co2_emissions_mt,
                        data = cur_data()))[2]),
    intercept = coef(lm(avg_crop_yield ~ avg_co2_emissions_mt,
                        data = cur_data()))[1],
  ) %>%
  arrange(desc(slope))

# Print
print(slopes_resilience1)
```

```
## # A tibble: 10 x 3
##    Crop_Type    slope intercept
##    <chr>        <dbl>     <dbl>
##  1 Cotton      0.0204      2.50
```

```
##   2 Fruits     0.0173        2.57
##   3 Vegetables 0.0142        2.39
##   4 Sugarcane  0.0123        2.45
##   5 Barley     0.00825       2.40
##   6 Coffee     0.00810       2.33
##   7 Soybeans   0.00670       2.30
##   8 Rice       0.00549       2.32
##   9 Wheat      0.00433       2.19
## 10 Corn       0.00221       2.21
```

```r
ggplot(slopes_resilience1, aes(x = reorder(Crop_Type, abs(slope)),
                               y = slope, fill = slope > 0)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  scale_fill_manual(values = c("blue", "red")) +
  labs(
    title = "Resilience of Crops to CO2 Emission Changes (Slope Analysis)",
    x = "Crop Type",
    y = "Slope (Impact of Temperature on Yield)"
  ) +
  theme_minimal()
```

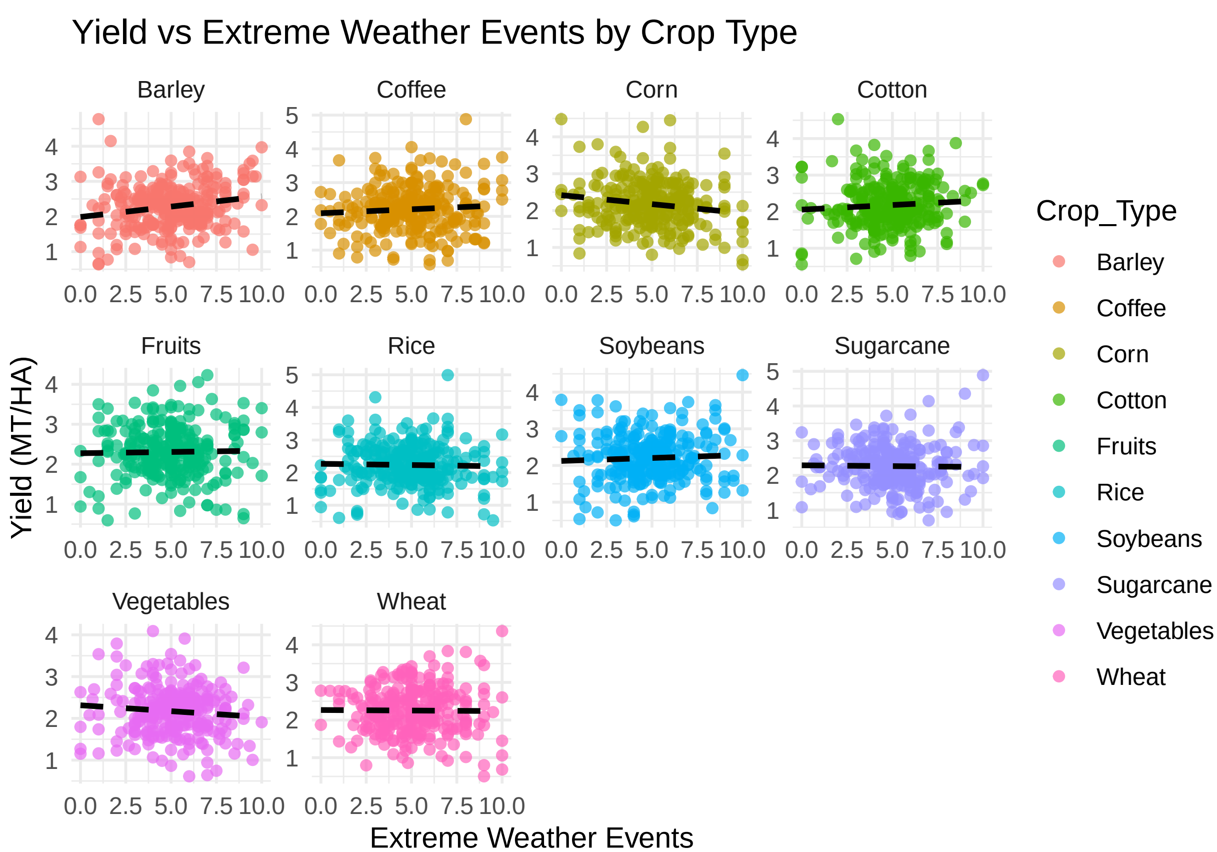


Extreme Weather events vs Yield

```r
ggplot(aggregated_data, aes(x = avg_extreme_weather_events, y = avg_crop_yield,
                            color = Crop_Type)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed",
              color = "black", size = 1) +
  facet_wrap(~ Crop_Type, scales = "free") +
  labs(
    title = "Yield vs Extreme Weather Events by Crop Type",
    x = "Extreme Weather Events",
    y = "Yield (MT/HA)"
  ) +
  theme_minimal()
```

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

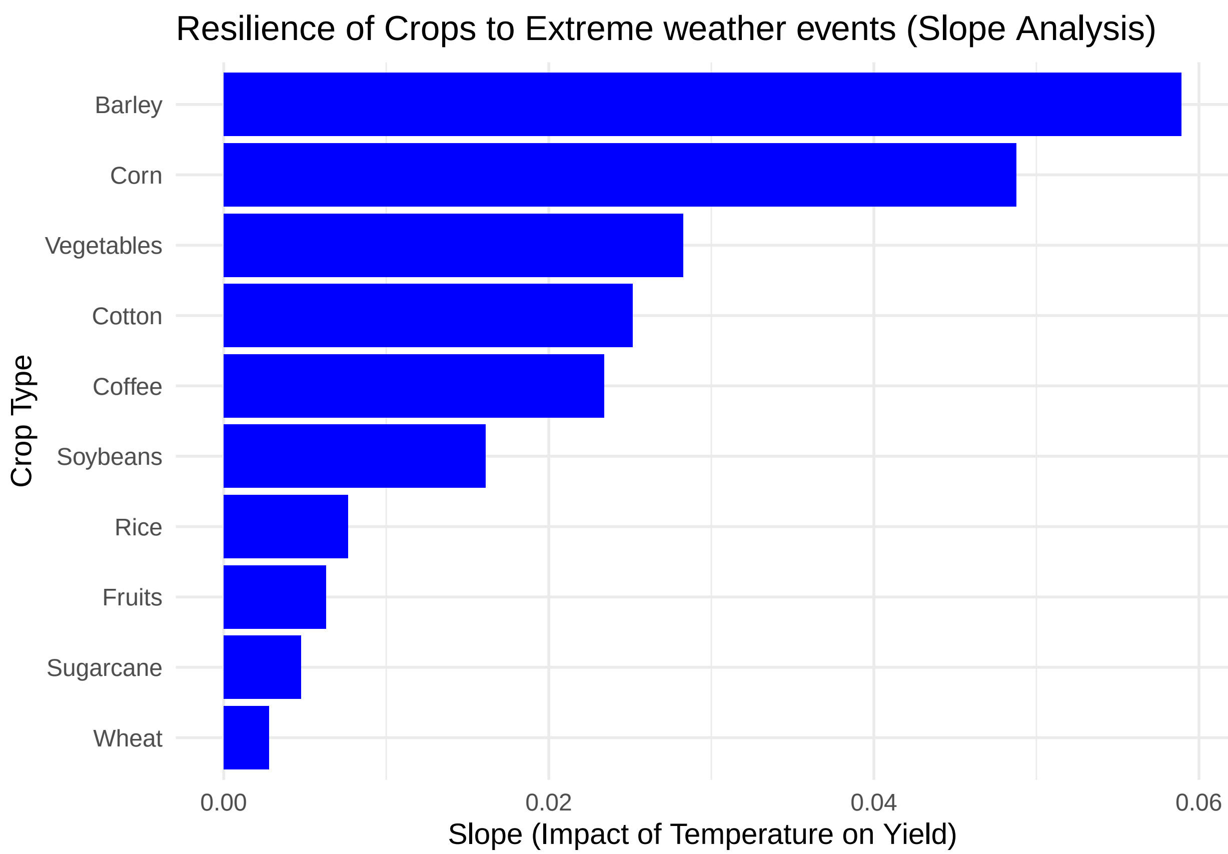
```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
slopes_resilience2 <- aggregated_data %>%
  group_by(Crop_Type) %>%
  summarize(
    slope = abs(coef(lm(avg_crop_yield ~ avg_extreme_weather_events, data = cur_data()))[2]),
    intercept = coef(lm(avg_crop_yield ~ avg_extreme_weather_events, data = cur_data()))[1],
  ) %>%
  arrange(slope)

# Print
print(slopes_resilience2)
```

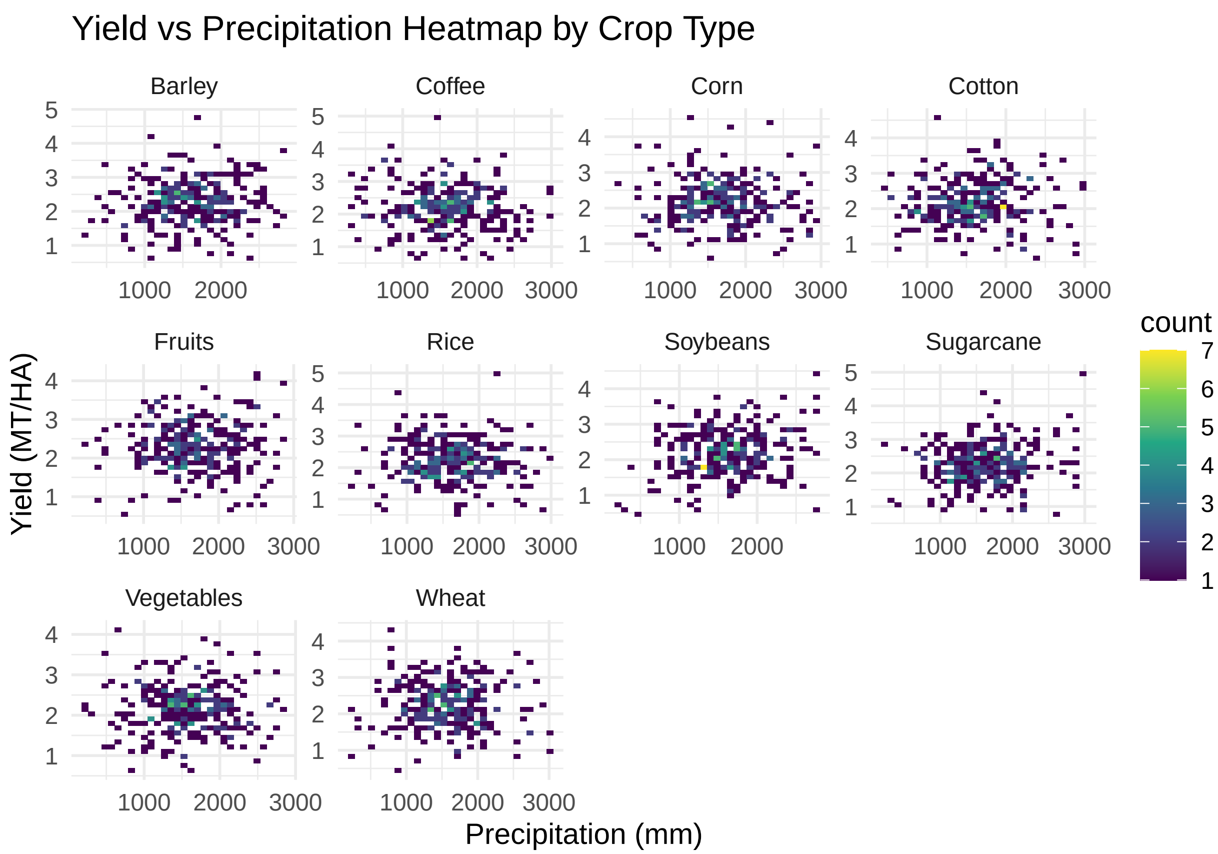
```
## # A tibble: 10 x 3
##    Crop_Type    slope intercept
##    <chr>        <dbl>     <dbl>
##  1 Wheat      0.00278      2.27
##  2 Sugarcane  0.00477      2.29
##  3 Fruits     0.00629      2.27
##  4 Rice       0.00765      2.27
##  5 Soybeans   0.0161       2.12
##  6 Coffee     0.0234       2.09
##  7 Cotton     0.0251       2.06
##  8 Vegetables 0.0283       2.31
##  9 Corn       0.0488       2.42
## 10 Barley     0.0589       1.99
```

```r
ggplot(slopes_resilience2, aes(x = reorder(Crop_Type, abs(slope)),
                               y = slope, fill = slope > 0)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  scale_fill_manual(values = c("blue", "red")) +
  labs(
    title = "Resilience of Crops to Extreme weather events (Slope Analysis)",
    x = "Crop Type",
    y = "Slope (Impact of Temperature on Yield)"
  ) +
  theme_minimal()
```

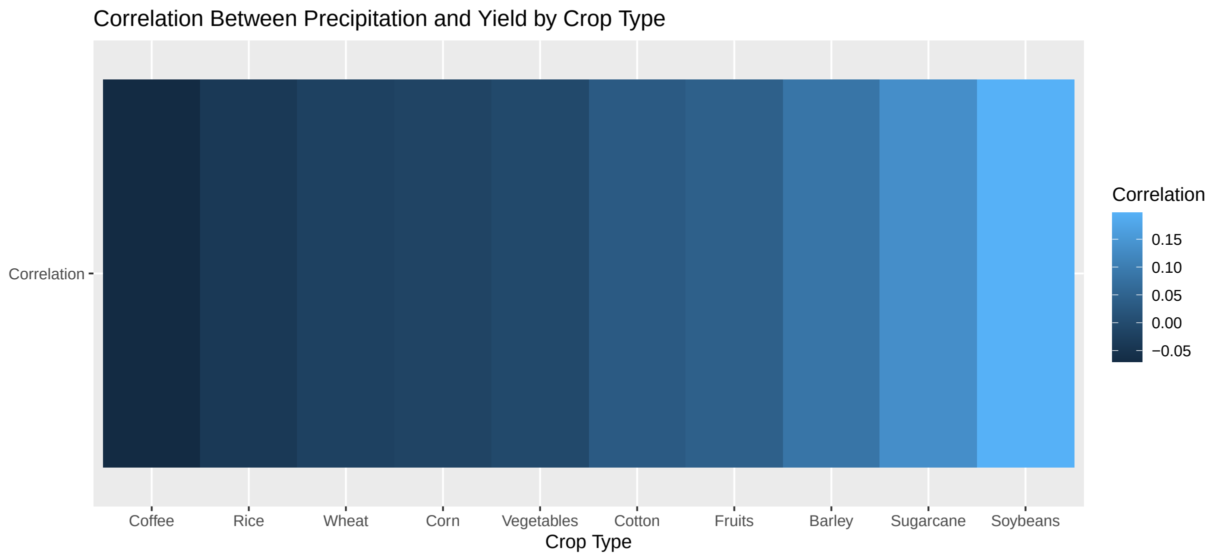


Precipitation vs Yield

```
ggplot(aggregated_data,
       aes(x = avg_total_precipitation_mm, y = avg_crop_yield)) +
  geom_bin2d() +
  facet_wrap(~ Crop_Type, scales = "free") +
  scale_fill_viridis_c() +
  labs(
    title = "Yield vs Precipitation Heatmap by Crop Type",
    x = "Precipitation (mm)",
    y = "Yield (MT/HA)"
  ) +
  theme_minimal()
```




```r
correlation_matrix <- aggregated_data %>%
  group_by(Crop_Type) %>%
  summarize(correlation = cor(avg_total_precipitation_mm,
                              avg_crop_yield))
correlation_matrix <- correlation_matrix %>%
  mutate(Crop_Type = reorder(Crop_Type, correlation))


ggplot(correlation_matrix, aes(x = Crop_Type, y = "Correlation",
                               fill = correlation)) +
  geom_tile() +
  labs(
    title = "Correlation Between Precipitation and Yield by Crop Type",
    x = "Crop Type",
    y = "",
    fill = "Correlation"
  )
```



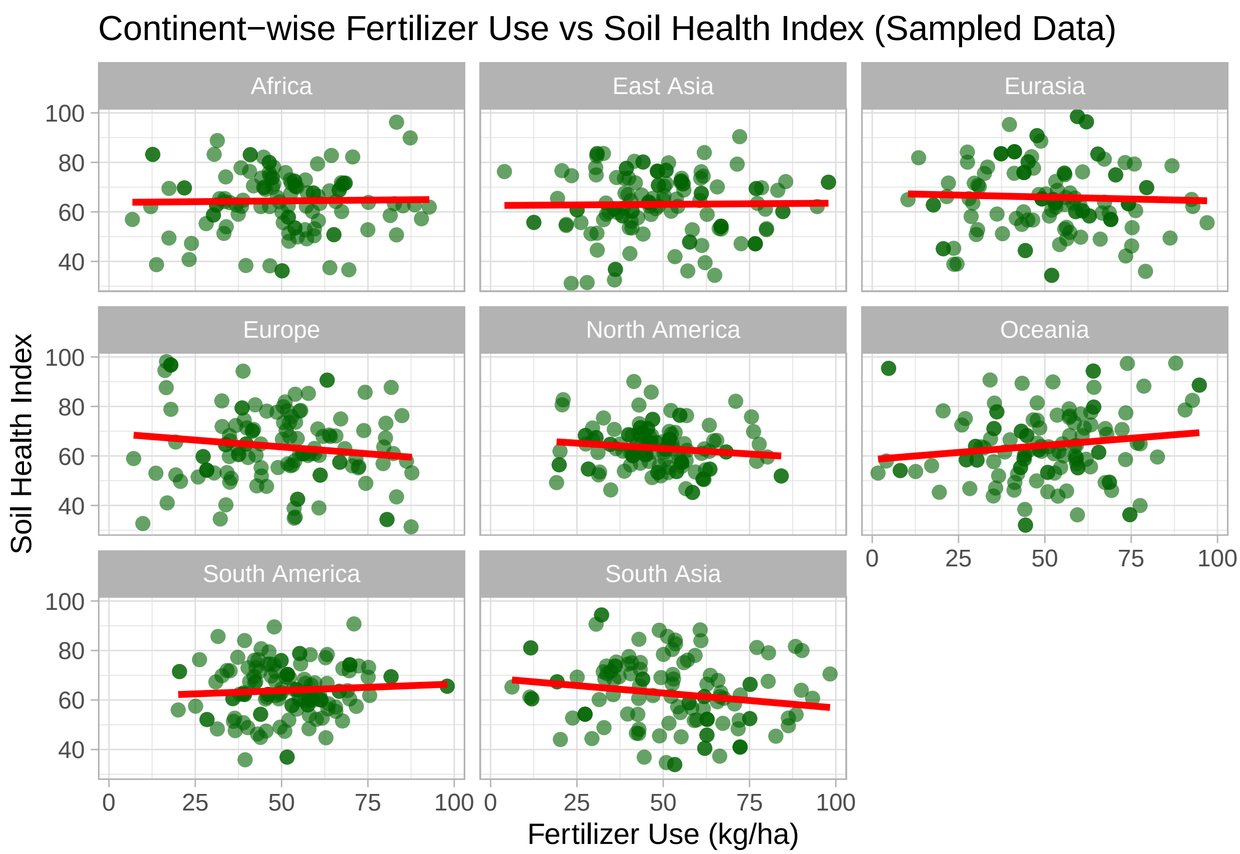
Modeling (Joonsoo Choi)

Fertilizer and Soil Health Index

- continent

```r
set.seed(10000)

sampled_data <- aggregated_data %>% sample_n(1000, replace = TRUE)

ggplot(sampled_data, aes(x = avg_fertilizer_use_kg_per_ha,
                         y = avg_soil_health_index)) +
  geom_point(color = "darkgreen", size = 2, alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE, linewidth = 1.2) +
  facet_wrap(~ Continent) +
  labs(
    title = "Continent-wise Fertilizer Use vs Soil Health Index (Sampled Data)",
    x = "Fertilizer Use (kg/ha)",
    y = "Soil Health Index"
  ) +
  theme_light()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



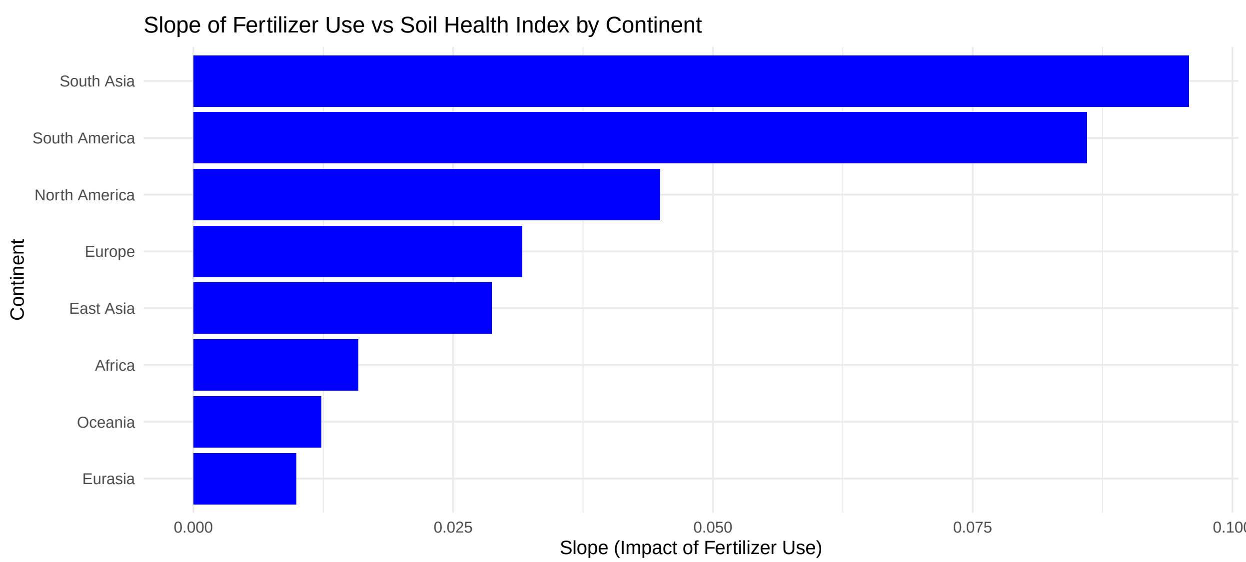
- Slope 1

```r
slopes <- aggregated_data %>%
  group_by(Continent) %>%
  summarise(
    slope = abs(coef(lm(avg_soil_health_index ~
                          avg_fertilizer_use_kg_per_ha))[2]),
    p_value = summary(lm(avg_soil_health_index ~
                           avg_fertilizer_use_kg_per_ha))$coefficients[2, 4]
  ) %>%
  arrange(slope)

print(slopes)
```

```
## # A tibble: 8 x 3
##   Continent       slope p_value
##   <chr>           <dbl>   <dbl>
## 1 Eurasia       0.00990  0.813
## 2 Oceania       0.0123   0.756
## 3 Africa        0.0159   0.692
## 4 East Asia     0.0287   0.459
## 5 Europe        0.0316   0.432
## 6 North America 0.0449   0.247
## 7 South America 0.0860   0.0299
## 8 South Asia    0.0958   0.0108
```

- Slope

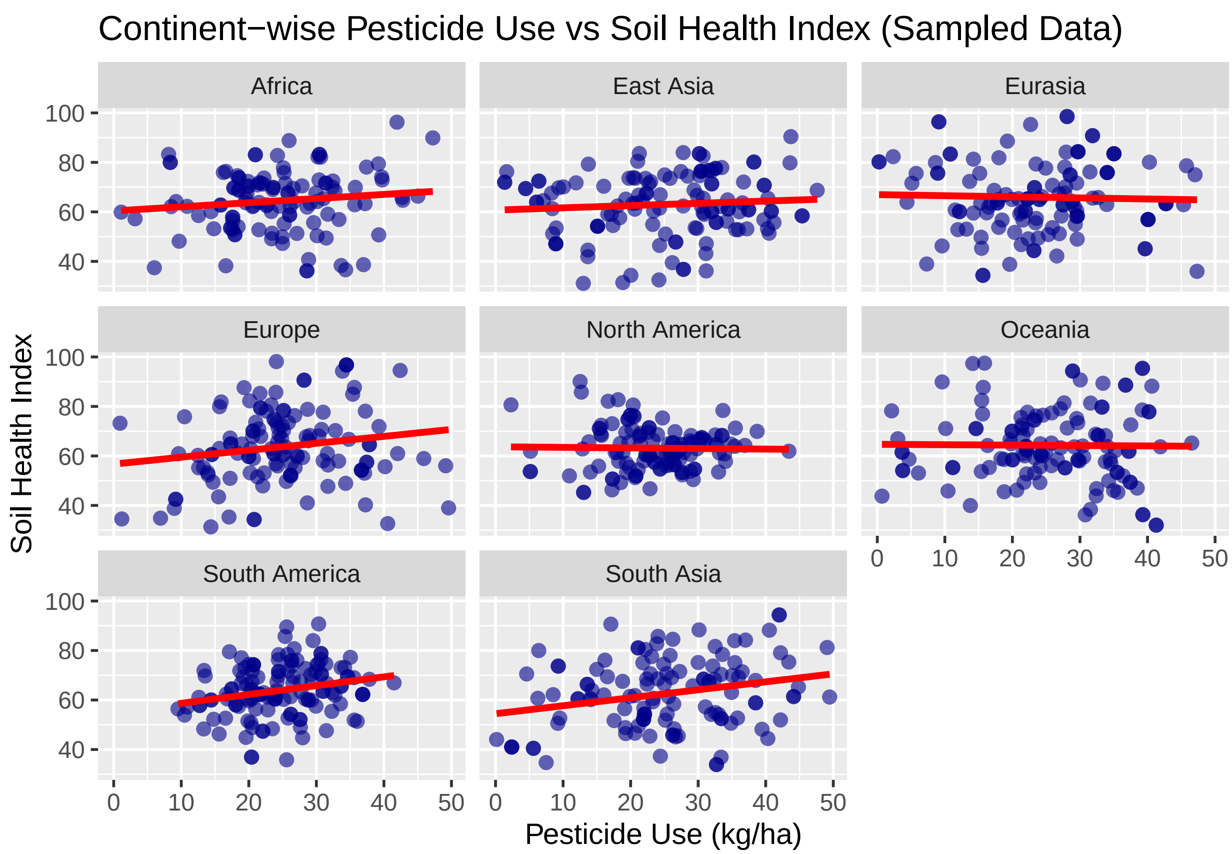
```r
ggplot(slopes, aes(x = reorder(Continent, slope),
                   y = slope, fill = slope > 0))+
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  scale_fill_manual(values = c("blue", "red")) +
  labs(
    title = "Slope of Fertilizer Use vs Soil Health Index by Continent",
    x = "Continent",
    y = "Slope (Impact of Fertilizer Use)"
  ) +
  theme_minimal()
```



Pesticide Use and Soil health index

```r
ggplot(sampled_data, aes(x = avg_pesticide_use_kg_per_ha,
                         y = avg_soil_health_index)) +
  geom_point(color = "darkblue", size = 2, alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE, linewidth = 1.2) +
  facet_wrap(~ Continent) +
  labs(
    title = "Continent-wise Pesticide Use vs Soil Health Index (Sampled Data)",
    x = "Pesticide Use (kg/ha)",
    y = "Soil Health Index"
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



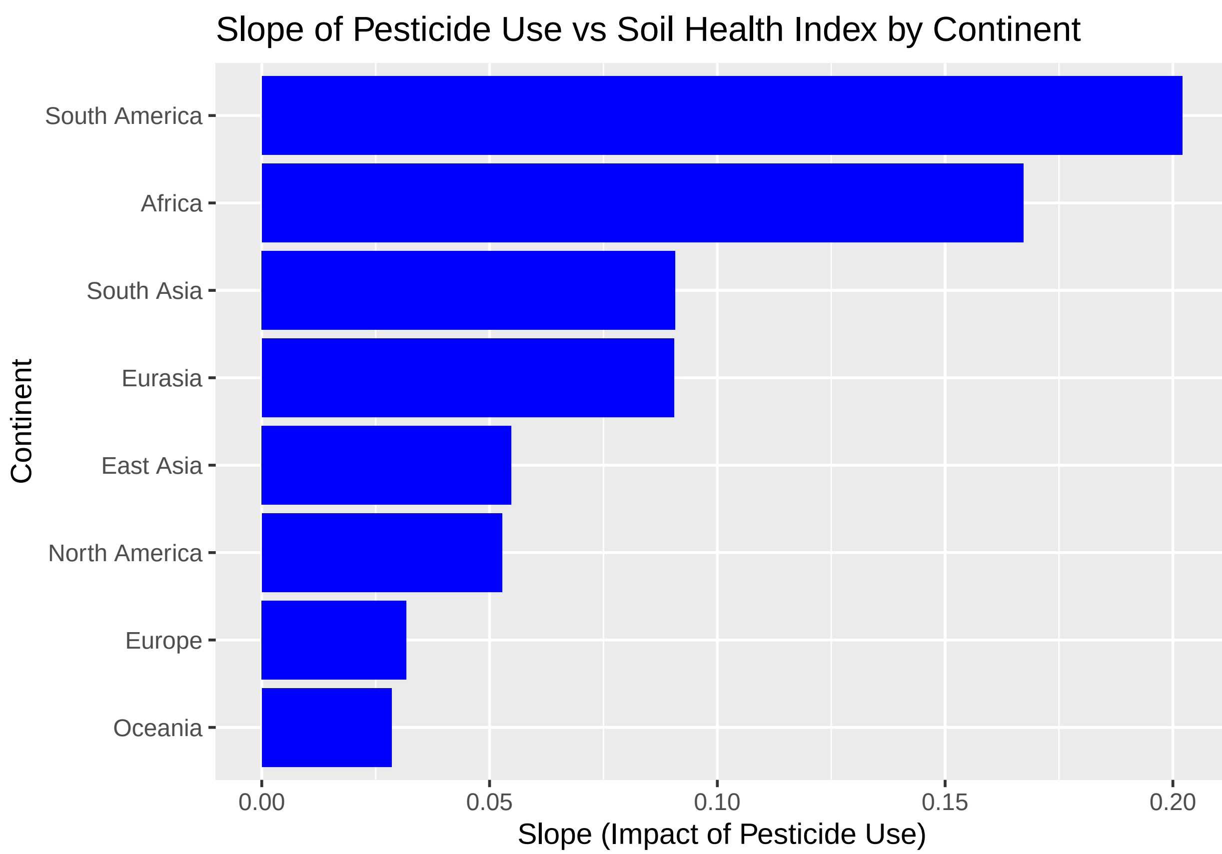
- Slope 2

```r
slopes_pesticide <- aggregated_data %>%
  group_by(Continent) %>%
  summarise(
    slope = abs(coef(lm(avg_soil_health_index ~
                          avg_pesticide_use_kg_per_ha))[2]),
    p_value = summary(lm(avg_soil_health_index ~
                           avg_pesticide_use_kg_per_ha))$coefficients[2, 4]
  )%>%
  arrange(slope)

print(slopes_pesticide)
```

```
## # A tibble: 8 x 3
##   Continent      slope p_value
##   <chr>          <dbl>   <dbl>
## 1 Oceania       0.0285  0.716
## 2 Europe        0.0318  0.695
## 3 North America 0.0528  0.451
## 4 East Asia     0.0548  0.461
## 5 Eurasia       0.0905  0.279
## 6 South Asia    0.0908  0.210
## 7 Africa        0.167   0.0214
## 8 South America 0.202   0.0109
```

- Slope visualization

```
ggplot(slopes_pesticide, aes(x = reorder(Continent, slope),
                             y = slope, fill = slope > 0)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  scale_fill_manual(values = c("blue", "red")) +
  labs(
    title = "Slope of Pesticide Use vs Soil Health Index by Continent",
    x = "Continent",
    y = "Slope (Impact of Pesticide Use)"
  )
```



Yield ~ Soil Health Index (Seunghoon Oh)

- Model 3

```
# Check if the required variables are available in the dataset
if (!all(c("avg_crop_yield", "avg_soil_health_index")
         %in% colnames(aggregated_data))) {
  stop("Variables 'yield' and 'soil_health_index' are missing in the dataset.")
}

# Model: Yield as a function of Soil Health Index
```

```r
model_Crop_Yield_MT_per_HA_Soil_Health_Index <-
  lm(avg_crop_yield ~ avg_soil_health_index, data = aggregated_data)

# Summary of the model
summary(model_Crop_Yield_MT_per_HA_Soil_Health_Index)
```

```
##
## Call:
## lm(formula = avg_crop_yield ~ avg_soil_health_index, data = aggregated_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71725 -0.40170 -0.01418  0.38650  2.77571
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.2653045  0.0625331   36.23   <2e-16 ***
## avg_soil_health_index -0.0005859  0.0009449   -0.62    0.535
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6264 on 2678 degrees of freedom
## Multiple R-squared:  0.0001435,  Adjusted R-squared:  -0.0002298
## F-statistic: 0.3845 on 1 and 2678 DF,  p-value: 0.5353
```

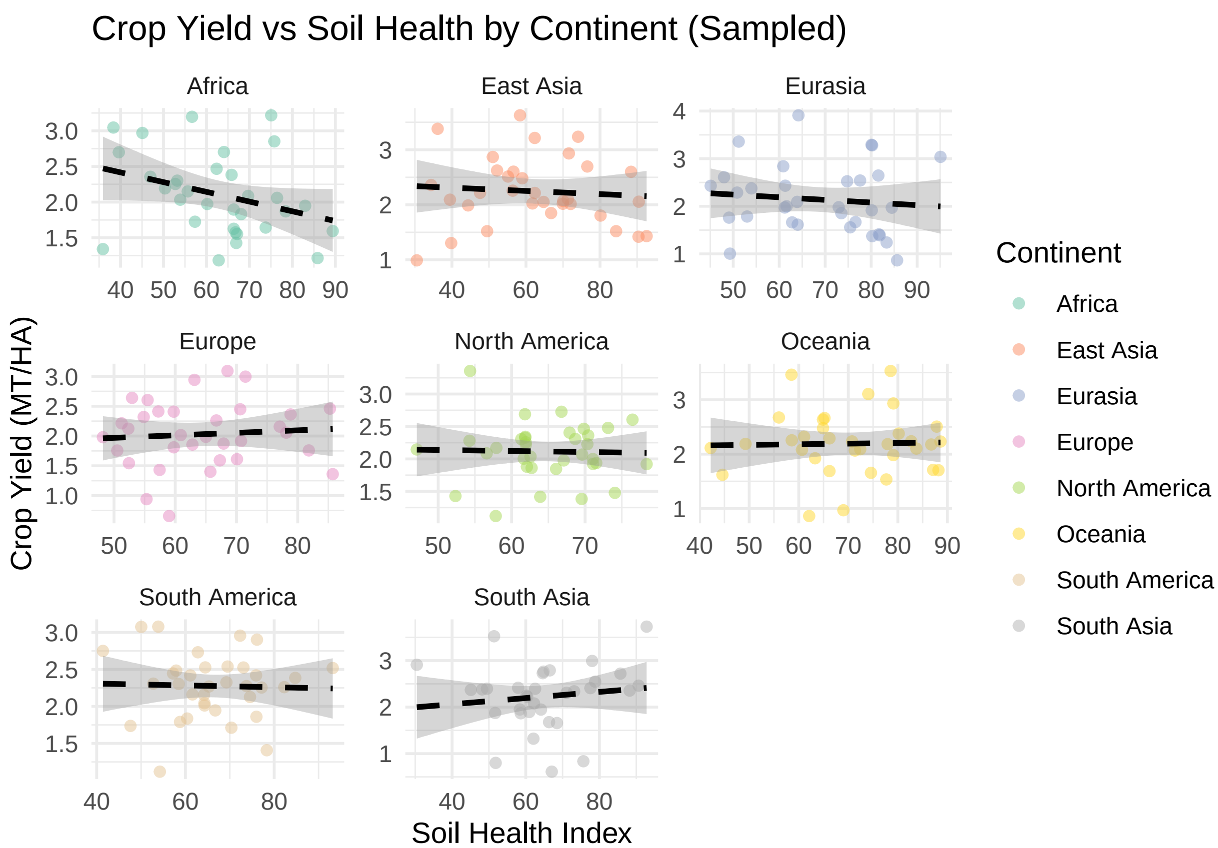
- Very low R-squared value.

```r
# Rename 'avg_soil_health_index' to 'Soil_Health'
data1 <- aggregated_data %>%
  rename(Soil_Health = avg_soil_health_index)
```

- Continent

```r
sampled_data_by_continent <- data1 %>%
  group_by(Continent) %>%
  sample_frac(0.1) %>%
  ungroup()

ggplot(sampled_data_by_continent, aes(x = Soil_Health, y = avg_crop_yield)) +
  geom_point(size = 1.5, alpha = 0.5, aes(color = Continent)) +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  labs(
    title = "Crop Yield vs Soil Health by Continent (Sampled)",
    x = "Soil Health Index",
    y = "Crop Yield (MT/HA)"
  ) +
  scale_color_brewer(palette = "Set2") +
  theme_minimal() +
  facet_wrap(~ Continent, scales = "free") # Facet by Country
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



- Slope

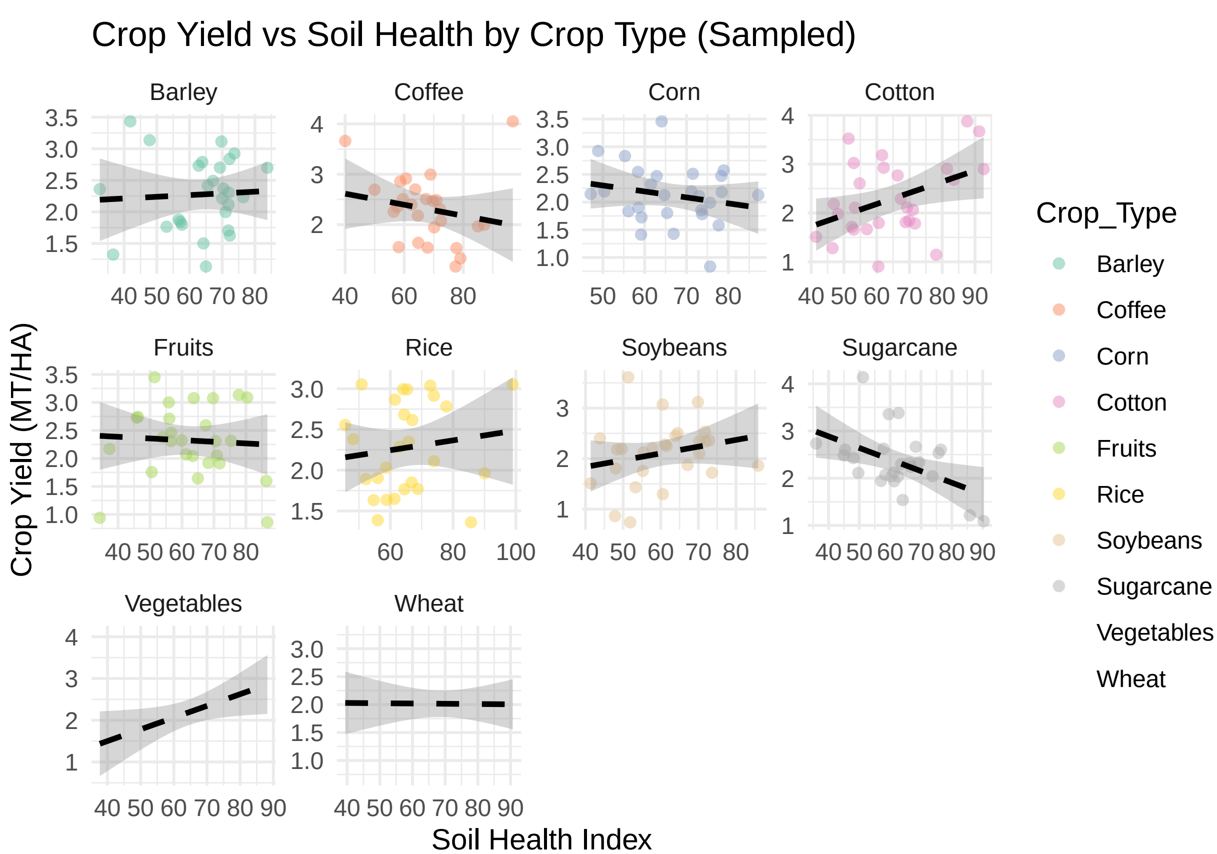
```
continent_slopes <- data1 %>%
  group_by(Continent) %>%
  summarize(
    slope = coef(lm(avg_crop_yield ~ Soil_Health, data = cur_data()))[2],
    intercept = coef(lm(avg_crop_yield ~ Soil_Health, data = cur_data()))[1]
  ) %>%
  arrange(desc(slope))  # Sort by slope in descending order

# Print the results
print(continent_slopes)
```

```
## # A tibble: 8 x 3
##   Continent        slope intercept
##   <chr>            <dbl>     <dbl>
## 1 Europe         0.00246      2.01
## 2 South Asia     0.00103      2.20
## 3 Eurasia        0.000860     2.15
## 4 Africa         0.000210     2.24
## 5 East Asia     -0.00143      2.34
## 6 North America -0.00212      2.35
## 7 Oceania       -0.00381      2.48
## 8 South America -0.00387      2.50
```

- Crop_Type

```r
sampled_data_by_continent <- data1 %>%
  group_by(Crop_Type) %>%
  sample_frac(0.1) %>%
  ungroup()

ggplot(sampled_data_by_continent, aes(x = Soil_Health, y = avg_crop_yield)) +
  geom_point(size = 1.5, alpha = 0.5, aes(color = Crop_Type)) +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  labs(
    title = "Crop Yield vs Soil Health by Crop Type (Sampled)",
    x = "Soil Health Index",
    y = "Crop Yield (MT/HA)"
  ) +
  scale_color_brewer(palette = "Set2") +
  theme_minimal() +
  facet_wrap(~ Crop_Type, scales = "free") # Facet by Country
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors

## Warning: Removed 53 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



* Slope

```r
crop_type_slopes <- data1 %>%
  group_by(Crop_Type) %>%
  summarize(
    slope = abs(coef(lm(avg_crop_yield ~ Soil_Health, data = cur_data()))[2]),
    intercept = coef(lm(avg_crop_yield ~ Soil_Health, data = cur_data()))[1]
  ) %>%
  arrange(slope)

# Print the results
print(crop_type_slopes)
```

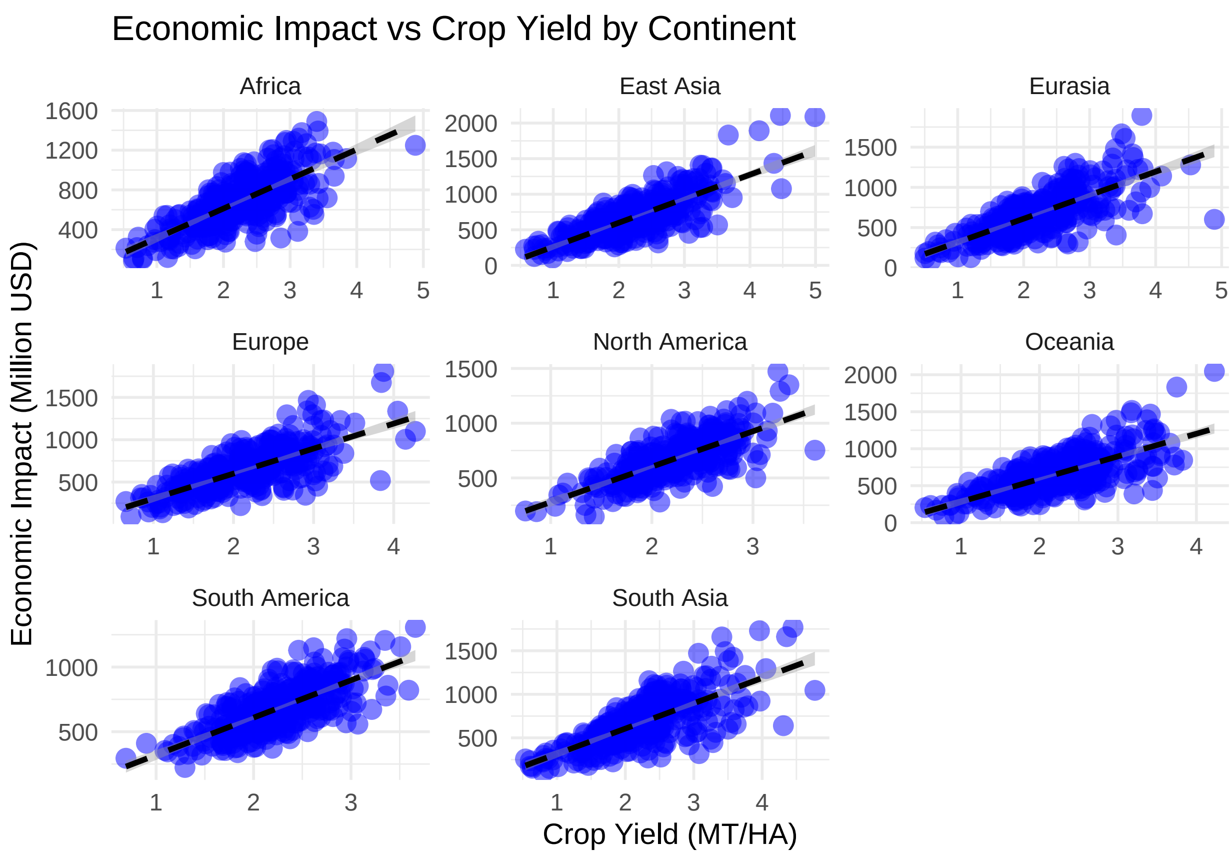
```
## # A tibble: 10 x 3
##    Crop_Type      slope intercept
##    <chr>          <dbl>     <dbl>
##  1 Cotton     0.0000348      2.18
##  2 Fruits     0.000260       2.32
##  3 Soybeans   0.000670       2.25
##  4 Rice       0.00109        2.16
##  5 Barley     0.00121        2.20
##  6 Wheat      0.00139        2.16
##  7 Vegetables 0.00173        2.06
##  8 Coffee     0.00180        2.33
##  9 Corn       0.00490        2.49
## 10 Sugarcane  0.00523        2.61
```

Visualization

- Continent

```r
ggplot(data1, aes(x = avg_crop_yield, y = avg_economic_impact_million_usd)) +
  geom_point(size = 3, alpha = 0.5, color = "blue") +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  labs(
    title = "Economic Impact vs Crop Yield by Continent",
    x = "Crop Yield (MT/HA)",
    y = "Economic Impact (Million USD)"
  ) +
  theme_minimal() +
  facet_wrap(~ Continent, scales = "free")
```

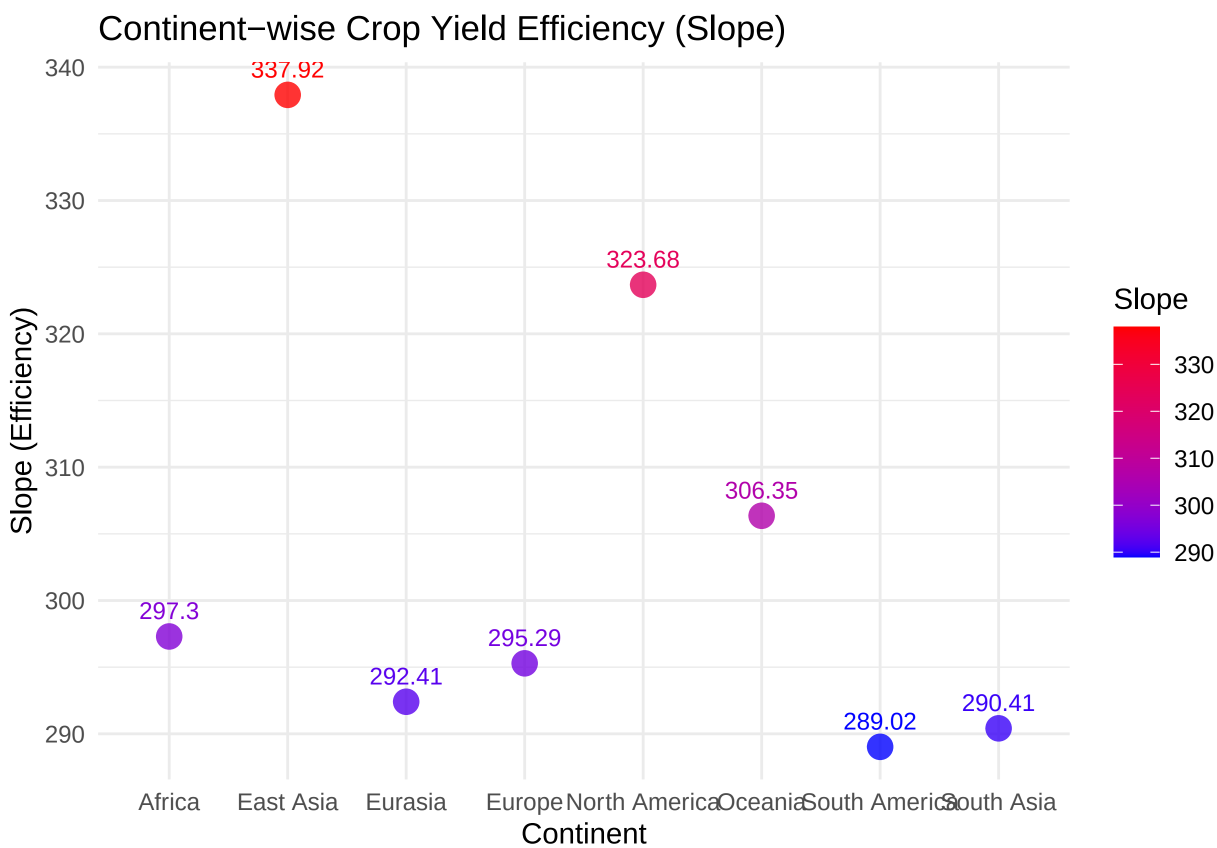
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
continent_slopes <- data1 %>%
  group_by(Continent) %>%
  summarize(
    Slope = coef(lm(avg_economic_impact_million_usd
                    ~ avg_crop_yield, data = cur_data()))[2],
    Intercept = coef(lm(avg_economic_impact_million_usd
                        ~ avg_crop_yield, data = cur_data()))[1]
  ) %>%
  arrange(desc(Slope))

print(continent_slopes)
```

```
## # A tibble: 8 x 3
##   Continent     Slope Intercept
##   <chr>         <dbl>     <dbl>
## 1 East Asia      338.    -76.7
## 2 North America  324.    -44.9
## 3 Oceania        306.    -22.6
## 4 Africa         297.     17.4
## 5 Europe         295.      9.30
## 6 Eurasia        292.     25.5
## 7 South Asia     290.     25.3
## 8 South America  289.     30.8
```

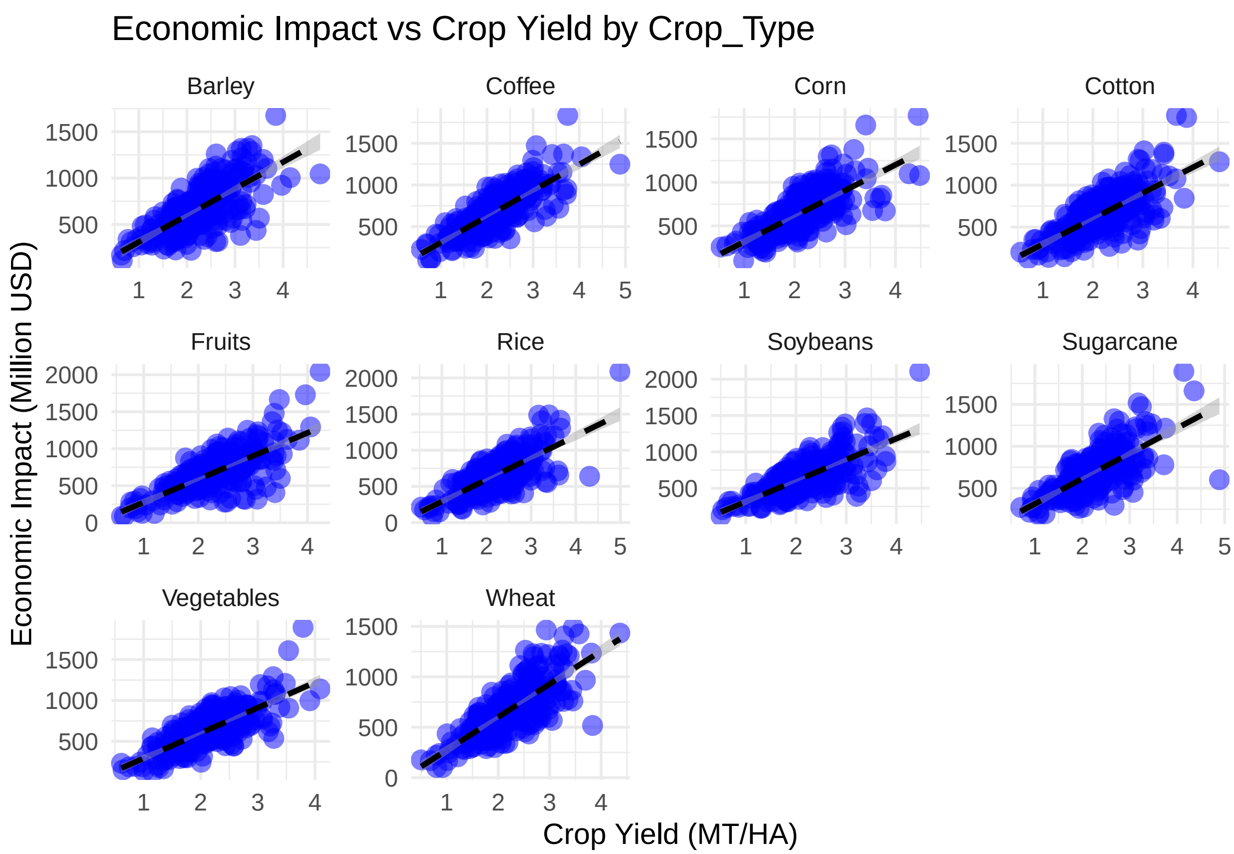
```r
ggplot(continent_slopes, aes(x = Continent, y = Slope, color = Slope)) +
  geom_point(size = 4, alpha = 0.8) +
  geom_text(aes(label = round(Slope, 2)), vjust = -1, size = 3) +
  labs(
    title = "Continent-wise Crop Yield Efficiency (Slope)",
    x = "Continent",
    y = "Slope (Efficiency)"
  ) +
  scale_color_gradient(low = "blue", high = "red") +
  theme_minimal()
```



- Crop_Type

```r
ggplot(data1, aes(x = avg_crop_yield, y = avg_economic_impact_million_usd)) +
  geom_point(size = 3, alpha = 0.5, color = "blue") +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  labs(
    title = "Economic Impact vs Crop Yield by Crop_Type",
    x = "Crop Yield (MT/HA)",
    y = "Economic Impact (Million USD)"
  ) +
  theme_minimal() +
  facet_wrap(~ Crop_Type, scales = "free")
```

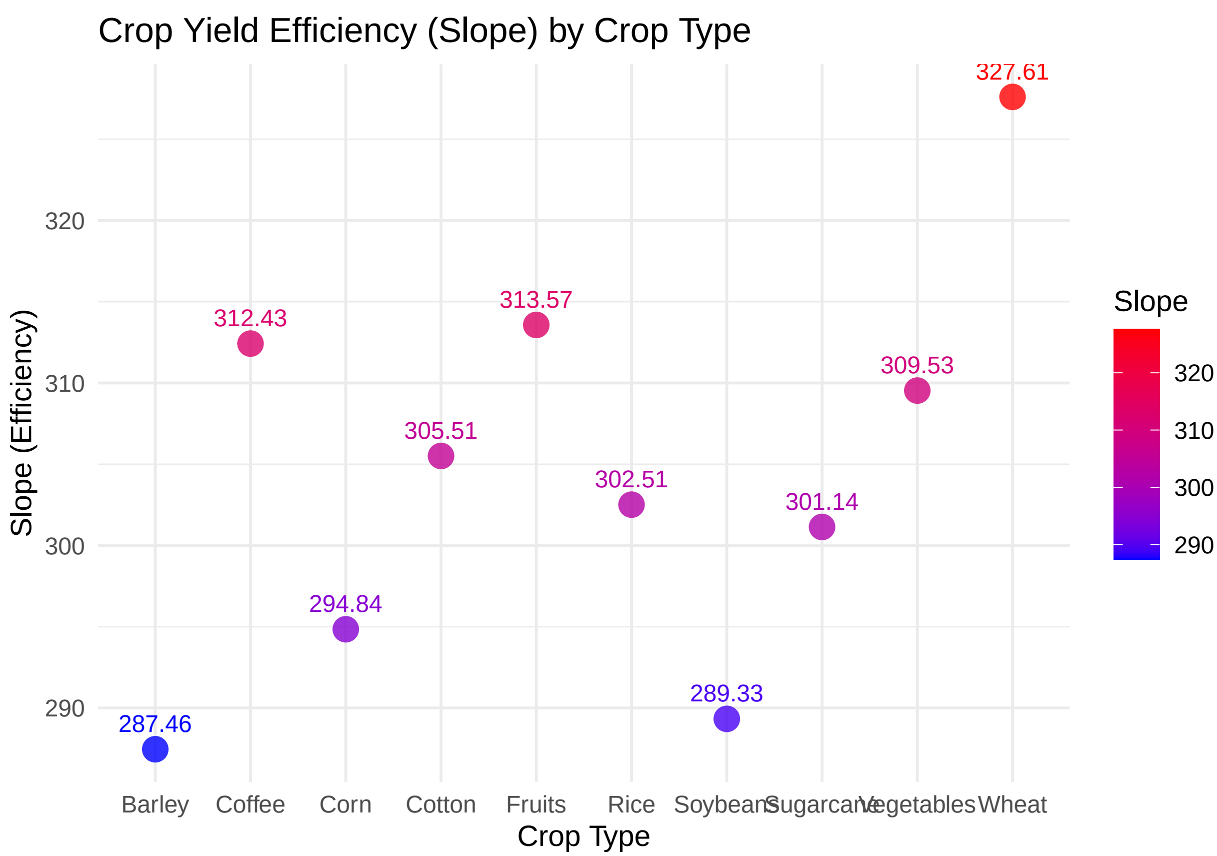
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
crop_type_slopes <- data1 %>%
  group_by(Crop_Type) %>%
  summarize(
    Slope = coef(lm(avg_economic_impact_million_usd ~
                    avg_crop_yield, data = cur_data()))[2],
    Intercept = coef(lm(avg_economic_impact_million_usd ~
                        avg_crop_yield, data = cur_data()))[1]
  ) %>%
  arrange(desc(Slope))

print(crop_type_slopes)
```

```
## # A tibble: 10 x 3
##    Crop_Type  Slope Intercept
##    <chr>      <dbl>     <dbl>
##  1 Wheat       328.    -54.2
##  2 Fruits      314.    -38.8
##  3 Coffee      312.     -4.16
##  4 Vegetables  310.    -17.1
##  5 Cotton      306.     -9.65
##  6 Rice        303.    -11.6
##  7 Sugarcane   301.     12.1
##  8 Corn        295.     22.4
##  9 Soybeans    289.     25.3
## 10 Barley      287.     23.3
```

```r
ggplot(crop_type_slopes, aes(x = Crop_Type, y = Slope, color = Slope)) +
  geom_point(size = 4, alpha = 0.8) +
  geom_text(aes(label = round(Slope, 2)), vjust = -1, size = 3) +
  labs(
    title = "Crop Yield Efficiency (Slope) by Crop Type",
    x = "Crop Type",
    y = "Slope (Efficiency)"
  ) +
  scale_color_gradient(low = "blue", high = "red") +
  theme_minimal()
```



Prediction Analytics (Juhyun Lee)

```r
set.seed(10000)

train_df <- aggregated_data %>% sample_frac(0.7)

test_df <- anti_join(aggregated_data, train_df)
```

```
## Joining with `by = join_by(Year, Continent, Crop_Type, avg_crop_yield,
## avg_extreme_weather_events, avg_temp_c, avg_total_precipitation_mm,
## avg_co2_emissions_mt, avg_pesticide_use_kg_per_ha,
## avg_fertilizer_use_kg_per_ha, avg_soil_health_index,
## avg_economic_impact_million_usd)`
```

```r
train_df %>%
  summarize(
    total = n(),
    missing = sum(is.na(avg_crop_yield)),
    fraction_missing = missing / total
  )
```

```
## # A tibble: 1 x 3
##   total missing fraction_missing
##   <int>   <int>            <dbl>
## 1  1876       0                0
```

```r
train_df <- train_df %>%
  mutate(avg_crop_yield = if_else(is.na(avg_crop_yield),
                                  mean(avg_crop_yield, na.rm = TRUE),
                                  avg_crop_yield))
```

crop_yield

```r
rf_model <- randomForest(
  avg_crop_yield ~ avg_temp_c + avg_extreme_weather_events +
    avg_total_precipitation_mm + Continent + Crop_Type,
  data = train_df,
  ntree = 100,
  mtry = 2,
  importance = TRUE
)

print(rf_model)
```

```
##
## Call:
##  randomForest(formula = avg_crop_yield ~ avg_temp_c + avg_extreme_weather_events +      avg_total_pre
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.3826241
##                     % Var explained: 2.4
```

```r
test_df <- test_df %>%
  mutate(
    predicted_yield = predict(rf_model, newdata = test_df)
  )


mae <- mean(abs(test_df$predicted_yield - test_df$avg_crop_yield))
print(paste("Mean Absolute Error:", mae))
```

```
## [1] "Mean Absolute Error: 0.470680760607547"
```

```r
head(test_df %>%
  select(Continent, Crop_Type, avg_crop_yield, predicted_yield) %>%
  arrange(predicted_yield))
```

```
## # A tibble: 6 x 4
##   Continent  Crop_Type avg_crop_yield predicted_yield
##   <chr>      <chr>              <dbl>           <dbl>
## 1 Europe     Cotton             0.855            1.28
## 2 East Asia  Fruits             1.30             1.37
## 3 South Asia Rice               1.84             1.39
## 4 Europe     Sugarcane          1.82             1.41
## 5 Oceania    Sugarcane          1.84             1.44
## 6 Africa     Soybeans           0.72             1.46
```

economic_impact

```r
rf_model1 <- randomForest(
  avg_economic_impact_million_usd ~ avg_temp_c + avg_extreme_weather_events +
    avg_total_precipitation_mm + Continent + Crop_Type + avg_crop_yield,
  data = train_df,
  ntree = 100,
  mtry = 2,
  importance = TRUE
)


print(rf_model1)
```

```
##
## Call:
##  randomForest(formula = avg_economic_impact_million_usd ~ avg_temp_c +      avg_extreme_weather_even
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 34589.74
##                     % Var explained: 48.76
```

```r
test_df <- test_df %>%
  mutate(
    predicted_economic_impact = predict(rf_model1, newdata = test_df)
  )

mae <- mean(abs(test_df$predicted_economic_impact -
                  test_df$avg_economic_impact_million_usd))
print(paste("Mean Absolute Error:", mae))
```

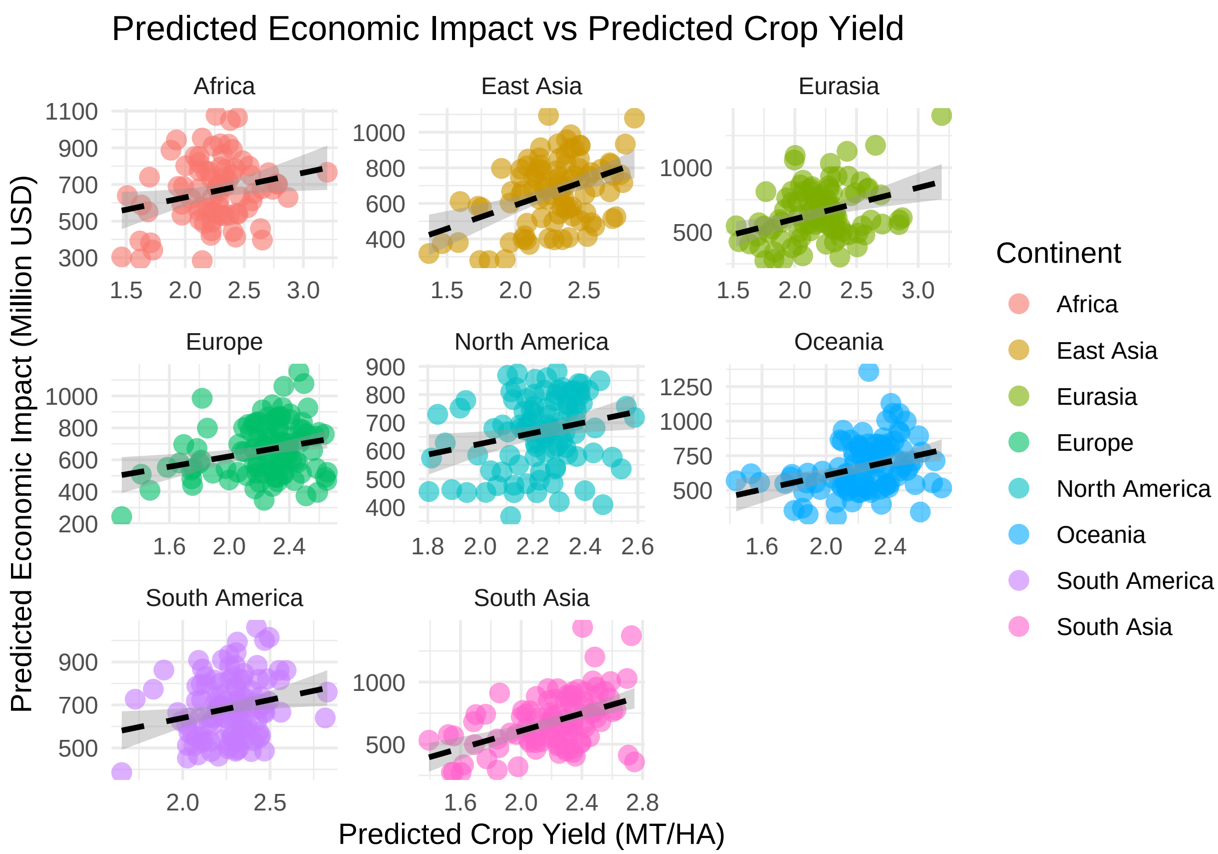
```
## [1] "Mean Absolute Error: 139.108089454045"
```

```r
head(test_df %>%
  select(Year,Continent, Crop_Type, avg_economic_impact_million_usd,
         predicted_economic_impact) %>%
  arrange(predicted_economic_impact))
```

```
## # A tibble: 6 x 5
##    Year Continent  Crop_Type  avg_economic_impact_milli~1 predicted_economic_i~2
##   <int> <chr>      <chr>                            <dbl>                  <dbl>
## 1  1992 Europe     Cotton                            353.                   244.
## 2  2009 South Asia Corn                              258.                   273.
## 3  1992 South Asia Sugarcane                         358.                   274.
## 4  2002 Eurasia    Vegetables                        188.                   275.
## 5  2023 South Asia Vegetables                        151.                   276.
## 6  2002 East Asia  Coffee                            226.                   278.
## # i abbreviated names: 1: avg_economic_impact_million_usd,
## #   2: predicted_economic_impact
```

```r
ggplot(test_df, aes(x = predicted_yield, y = predicted_economic_impact,
                    color = Continent)) +
  geom_point(size = 3, alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  labs(
    title = "Predicted Economic Impact vs Predicted Crop Yield",
    x = "Predicted Crop Yield (MT/HA)",
    y = "Predicted Economic Impact (Million USD)",
    color = "Continent"
  ) +

  theme_minimal()+
  facet_wrap(~ Continent, scales = "free")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

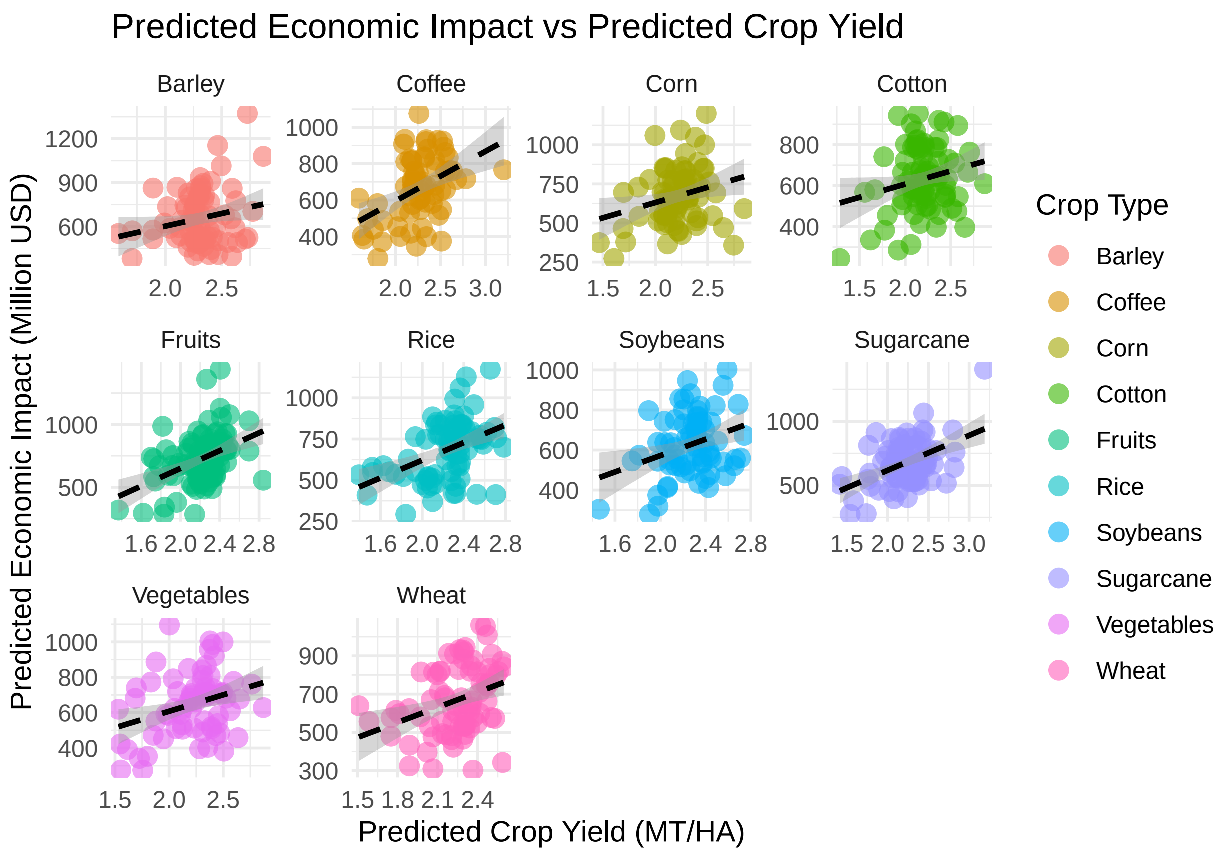



```r
slope_intercept <- test_df %>%
  group_by(Continent) %>%
  summarize(
    slope = coef(lm(predicted_economic_impact ~ predicted_yield,
                    data = cur_data()))[2],
    intercept = coef(lm(predicted_economic_impact ~ predicted_yield, data =
                          cur_data()))[1]
  )


print(slope_intercept)
```

```
## # A tibble: 8 x 3
##   Continent     slope intercept
##   <chr>         <dbl>     <dbl>
## 1 Africa         134.     361.
## 2 East Asia      266.      59.7
## 3 Eurasia        244.     112.
## 4 Europe         163.     295.
## 5 North America  191.     243.
## 6 Oceania        255.      98.3
## 7 South America  169.     302.
## 8 South Asia     350.     -92.6
```

```r
ggplot(test_df, aes(x = predicted_yield, y = predicted_economic_impact,
                    color = Crop_Type)) +
  geom_point(size = 3, alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, color = "black", linetype = "dashed") +
  labs(
    title = "Predicted Economic Impact vs Predicted Crop Yield",
    x = "Predicted Crop Yield (MT/HA)",
    y = "Predicted Economic Impact (Million USD)",
    color = "Crop Type"
  ) +

  theme_minimal()+
  facet_wrap(~ Crop_Type, scales = "free")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
slope_intercept_crop <- test_df %>%
  group_by(Crop_Type) %>%
  summarize(
    slope = coef(lm(predicted_economic_impact ~ predicted_yield,
                    data = cur_data()))[2],
    intercept = coef(lm(predicted_economic_impact ~ predicted_yield,
                        data = cur_data()))[1]
  ) %>% arrange(desc(intercept))
```

```r
print(slope_intercept_crop)
```

```
## # A tibble: 10 x 3
##    Crop_Type  slope intercept
##    <chr>      <dbl>     <dbl>
##  1 Cotton      128.     351.
##  2 Barley      173.     256.
##  3 Corn        196.     240.
##  4 Vegetables  186.     236.
##  5 Soybeans    204.     164.
##  6 Rice        268.      84.4
##  7 Wheat       263.      79.4
##  8 Sugarcane   272.      72.9
##  9 Coffee      273.      47.9
## 10 Fruits      353.     -56.3
```