Exploring Global Food and Feed Production Trends: A Comprehensive Analysis

Nikhitha Gunta

2023-12-08

Introduction

The provided dataset offers a crucial perspective on global food production, highlighting the balance between food cultivated for human consumption and feed produced for animals. This information is particularly relevant given the projected surge in the world's population from 7.3 billion to 9.7 billion by 2050. Addressing the challenges of feeding this growing population necessitates innovative approaches in agricultural practices and dietary habits. Moreover, these strategies must be developed in the context of an evolving climate, which both impacts and is impacted by agricultural methods. Analyzing this dataset can provide valuable insights into current production trends and help in formulating sustainable solutions to meet future food demands while considering environmental implications

Data Acquisition

I acquired the dataset from Kaggle: https://www.kaggle.com/datasets/dorbicycle/world-foodfeed-production/data

```
# Load necessary libraries
library(tidyverse) # for data manipulation and visualization
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
```

```
library(dplyr)  # for data manipulation
library(ggplot2) # for data visualization

# Read in the dataset from the specified location
agriculture_data <- read.csv("E:/FInal Project/FAO.csv")

# Display the structure of the dataframe
str(agriculture_data)
```

```
## 'data.frame':    21477 obs. of  63 variables:
##  $ Area.Abbreviation: chr  "AFG" "AFG" "AFG" "AFG" ...
##  $ Area.Code        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Area             : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Item.Code        : int  2511 2805 2513 2513 2514 2514 2517 2520 2531 2536 ...
##  $ Item             : chr  "Wheat and products" "Rice (Milled Equivalent)" "Barley and products" "Ba
##  $ Element.Code     : int  5142 5142 5521 5142 5521 5142 5142 5142 5142 5521 ...
##  $ Element          : chr  "Food" "Food" "Feed" "Food" ...
##  $ Unit             : chr  "1000 tonnes" "1000 tonnes" "1000 tonnes" "1000 tonnes" ...
##  $ latitude         : num  33.9 33.9 33.9 33.9 33.9 ...
##  $ longitude        : num  67.7 67.7 67.7 67.7 67.7 ...
##  $ Y1961            : int  1928 183 76 237 210 403 17 0 111 45 ...
##  $ Y1962            : int  1904 183 76 237 210 403 18 0 97 45 ...
##  $ Y1963            : int  1666 182 76 237 214 410 19 0 103 45 ...
##  $ Y1964            : int  1950 220 76 238 216 415 20 0 110 45 ...
##  $ Y1965            : int  2001 220 76 238 216 415 21 0 113 31 ...
##  $ Y1966            : int  1808 195 75 237 216 413 22 0 117 14 ...
##  $ Y1967            : int  2053 231 71 225 235 454 23 0 128 19 ...
##  $ Y1968            : int  2045 235 72 227 232 448 24 0 130 30 ...
##  $ Y1969            : int  2154 238 73 230 236 455 25 0 134 34 ...
##  $ Y1970            : int  1819 213 74 234 200 383 26 0 125 15 ...
##  $ Y1971            : int  1963 205 71 223 201 386 26 0 147 0 ...
##  $ Y1972            : int  2215 233 70 219 216 416 27 0 138 0 ...
##  $ Y1973            : int  2310 246 72 225 228 439 27 0 143 28 ...
##  $ Y1974            : int  2335 246 76 240 231 445 28 0 160 32 ...
##  $ Y1975            : int  2434 255 77 244 234 451 29 0 169 20 ...
##  $ Y1976            : int  2512 263 80 255 240 463 37 0 324 28 ...
##  $ Y1977            : int  2282 235 60 185 228 439 32 0 176 24 ...
##  $ Y1978            : int  2454 254 65 203 234 451 33 0 225 24 ...
##  $ Y1979            : int  2443 270 64 198 228 440 31 0 232 34 ...
##  $ Y1980            : int  2129 259 64 202 226 437 31 0 240 61 ...
##  $ Y1981            : int  2133 248 60 189 210 407 29 0 247 50 ...
##  $ Y1982            : int  2068 217 55 174 199 384 27 0 248 43 ...
##  $ Y1983            : int  1994 217 53 167 192 371 28 0 242 38 ...
##  $ Y1984            : int  1851 197 51 160 182 353 26 0 235 46 ...
##  $ Y1985            : int  1791 186 48 151 173 334 25 0 226 23 ...
##  $ Y1986            : int  1683 200 46 145 170 330 23 0 217 25 ...
##  $ Y1987            : int  2194 193 46 145 154 298 23 0 196 3 ...
##  $ Y1988            : int  1801 202 47 148 148 287 23 0 198 45 ...
##  $ Y1989            : int  1754 191 46 145 137 265 23 0 184 54 ...
##  $ Y1990            : int  1640 199 43 135 144 279 24 0 205 47 ...
##  $ Y1991            : int  1539 197 43 132 126 245 24 0 203 29 ...
##  $ Y1992            : int  1582 249 40 120 90 170 18 0 210 29 ...
##  $ Y1993            : int  1840 218 50 155 141 272 22 0 210 29 ...
##  $ Y1994            : int  1855 260 46 143 150 289 20 0 211 29 ...
##  $ Y1995            : int  1853 319 41 125 159 310 21 0 212 29 ...
##  $ Y1996            : int  2177 254 44 138 108 209 17 0 213 29 ...
##  $ Y1997            : int  2343 326 50 159 90 173 20 0 214 28 ...
##  $ Y1998            : int  2407 347 48 154 99 192 21 0 214 28 ...
##  $ Y1999            : int  2463 270 43 141 72 141 17 0 217 28 ...
##  $ Y2000            : int  2600 372 26 84 35 66 20 0 219 29 ...
##  $ Y2001            : int  2668 411 29 83 48 93 20 0 215 29 ...
##  $ Y2002            : int  2776 448 70 122 89 170 18 0 217 29 ...
##  $ Y2003            : int  3095 460 48 144 63 117 16 1 347 51 ...
```

```
##  $ Y2004            : int  3249 419 58 185 120 231 15 2 276 50 ...
##  $ Y2005            : int  3486 445 236 43 208 67 21 1 294 29 ...
##  $ Y2006            : int  3704 546 262 44 233 82 11 1 294 61 ...
##  $ Y2007            : int  4164 455 263 48 249 67 19 0 260 65 ...
##  $ Y2008            : int  4252 490 230 62 247 69 21 0 242 54 ...
##  $ Y2009            : int  4538 415 379 55 195 71 18 0 250 114 ...
##  $ Y2010            : int  4605 442 315 60 178 82 14 0 192 83 ...
##  $ Y2011            : int  4711 476 203 72 191 73 14 0 169 83 ...
##  $ Y2012            : int  4810 425 367 78 200 77 14 0 196 69 ...
##  $ Y2013            : int  4895 422 360 89 200 76 12 0 230 81 ...
```

```
# Convert from a wide dataset to a long dataset using pivot_longer
agriculture_data_long <- agriculture_data %>%
  pivot_longer(cols = starts_with("Y"),
               names_to = "Year",
               values_to = "Production")

# Display a summary of the long dataset
summary(agriculture_data_long)
```

```
##  Area.Abbreviation    Area.Code         Area             Item.Code
##  Length:1138281     Min.   :  1.0   Length:1138281     Min.   :2511
##  Class :character   1st Qu.: 63.0   Class :character   1st Qu.:2561
##  Mode  :character   Median :120.0   Mode  :character   Median :2640
##                     Mean   :125.4                      Mean   :2694
##                     3rd Qu.:188.0                      3rd Qu.:2782
##                     Max.   :276.0                      Max.   :2961
##
##      Item            Element.Code    Element             Unit
##  Length:1138281     Min.   :5142   Length:1138281     Length:1138281
##  Class :character   1st Qu.:5142   Class :character   Class :character
##  Mode  :character   Median :5142   Mode  :character   Mode  :character
##                     Mean   :5212
##                     3rd Qu.:5142
##                     Max.   :5521
##
##     latitude        longitude           Year             Production
##  Min.   :-40.90   Min.   :-172.10   Length:1138281     Min.   :  -246
##  1st Qu.:  6.43   1st Qu.: -11.78   Class :character   1st Qu.:     0
##  Median : 20.59   Median :  19.15   Mode  :character   Median :     3
##  Mean   : 20.45   Mean   :  15.79                      Mean   :   371
##  3rd Qu.: 41.15   3rd Qu.:  46.87                      3rd Qu.:    49
##  Max.   : 64.96   Max.   : 179.41                      Max.   :489299
##                                                        NA's   :117450
```

```
str(agriculture_data_long)
```

```
## tibble [1,138,281 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Area.Abbreviation: chr [1:1138281] "AFG" "AFG" "AFG" "AFG" ...
##  $ Area.Code        : int [1:1138281] 2 2 2 2 2 2 2 2 2 2 ...
##  $ Area             : chr [1:1138281] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Item.Code        : int [1:1138281] 2511 2511 2511 2511 2511 2511 2511 2511 2511 2511 ...
##  $ Item             : chr [1:1138281] "Wheat and products" "Wheat and products" "Wheat and products"
```

```
##  $ Element.Code     : int [1:1138281] 5142 5142 5142 5142 5142 5142 5142 5142 5142 5142 ...
##  $ Element          : chr [1:1138281] "Food" "Food" "Food" "Food" ...
##  $ Unit             : chr [1:1138281] "1000 tonnes" "1000 tonnes" "1000 tonnes" "1000 tonnes" ...
##  $ latitude         : num [1:1138281] 33.9 33.9 33.9 33.9 33.9 ...
##  $ longitude        : num [1:1138281] 67.7 67.7 67.7 67.7 67.7 ...
##  $ Year             : chr [1:1138281] "Y1961" "Y1962" "Y1963" "Y1964" ...
##  $ Production       : int [1:1138281] 1928 1904 1666 1950 2001 1808 2053 2045 2154 1819 ...
```

```r
# Convert years to integers
agriculture_data_long$Year <- as.integer(gsub("Y", "", agriculture_data_long$Year))

# Remove rows with NA in the Production column
agriculture_data_long <- agriculture_data_long %>%
  drop_na(Production)

# Display a summary of the updated dataset
summary(agriculture_data_long)
```

```
##  Area.Abbreviation    Area.Code         Area             Item.Code
##  Length:1020831     Min.   :  1.0   Length:1020831     Min.   :2511
##  Class :character   1st Qu.: 59.0   Class :character   1st Qu.:2561
##  Mode  :character   Median :117.0   Mode  :character   Median :2641
##                     Mean   :120.4                      Mean   :2694
##                     3rd Qu.:176.0                      3rd Qu.:2782
##                     Max.   :276.0                      Max.   :2961
##      Item            Element.Code    Element             Unit
##  Length:1020831     Min.   :5142   Length:1020831     Length:1020831
##  Class :character   1st Qu.:5142   Class :character   Class :character
##  Mode  :character   Median :5142   Mode  :character   Mode  :character
##                     Mean   :5209
##                     3rd Qu.:5142
##                     Max.   :5521
##     latitude        longitude           Year        Production
##  Min.   :-40.90   Min.   :-172.10   Min.   :1961   Min.   :  -246
##  1st Qu.:  4.21   1st Qu.: -24.01   1st Qu.:1975   1st Qu.:     0
##  Median : 17.06   Median :  17.68   Median :1989   Median :     3
##  Mean   : 17.77   Mean   :  13.60   Mean   :1988   Mean   :   371
##  3rd Qu.: 36.20   3rd Qu.:  46.87   3rd Qu.:2002   3rd Qu.:    49
##  Max.   : 64.96   Max.   : 179.41   Max.   :2013   Max.   :489299
```

```r
# Identify rows where production is negative
rows_with_negative_production <- which(agriculture_data_long$Production < 0)
print(rows_with_negative_production)
```

```
## [1] 493075 493076
```

```r
# Filter out rows with negative production
agriculture_data_filtered <- agriculture_data_long %>%
  filter(Production >= 0)

# Display a summary of the filtered dataset
summary(agriculture_data_filtered)
```

```
##  Area.Abbreviation    Area.Code          Area              Item.Code
##  Length:1020829     Min.   :  1.0   Length:1020829     Min.   :2511
##  Class :character   1st Qu.: 59.0   Class :character   1st Qu.:2561
##  Mode  :character   Median :117.0   Mode  :character   Median :2641
##                     Mean   :120.4                      Mean   :2694
##                     3rd Qu.:176.0                      3rd Qu.:2782
##                     Max.   :276.0                      Max.   :2961
##      Item            Element.Code    Element              Unit
##  Length:1020829     Min.   :5142   Length:1020829     Length:1020829
##  Class :character   1st Qu.:5142   Class :character   Class :character
##  Mode  :character   Median :5142   Mode  :character   Mode  :character
##                     Mean   :5209
##                     3rd Qu.:5142
##                     Max.   :5521
##     latitude        longitude           Year        Production
##  Min.   :-40.90   Min.   :-172.10   Min.   :1961   Min.   :     0
##  1st Qu.:  4.21   1st Qu.: -24.01   1st Qu.:1975   1st Qu.:     0
##  Median : 17.06   Median :  17.68   Median :1989   Median :     3
##  Mean   : 17.77   Mean   :  13.60   Mean   :1988   Mean   :   371
##  3rd Qu.: 36.20   3rd Qu.:  46.87   3rd Qu.:2002   3rd Qu.:    49
##  Max.   : 64.96   Max.   : 179.41   Max.   :2013   Max.   :489299
```

```
# Select relevant columns for analysis
agriculture_analysis_data <- agriculture_data_filtered %>%
  select(Area,Item,Element,Unit,Year,Production)

# Display a summary of the analysis dataset
summary(agriculture_analysis_data)
```

```
##      Area               Item              Element              Unit
##  Length:1020829     Length:1020829     Length:1020829     Length:1020829
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       Year        Production
##  Min.   :1961   Min.   :     0
##  1st Qu.:1975   1st Qu.:     0
##  Median :1989   Median :     3
##  Mean   :1988   Mean   :   371
##  3rd Qu.:2002   3rd Qu.:    49
##  Max.   :2013   Max.   :489299
```

```
# Display unique values in the 'Element' column
unique(agriculture_data_filtered$Element)
```

```
## [1] "Food" "Feed"
```

```
# Convert 'Element' column to a factor
agriculture_analysis_data<- agriculture_analysis_data %>%
  mutate(Element = factor(Element))
```

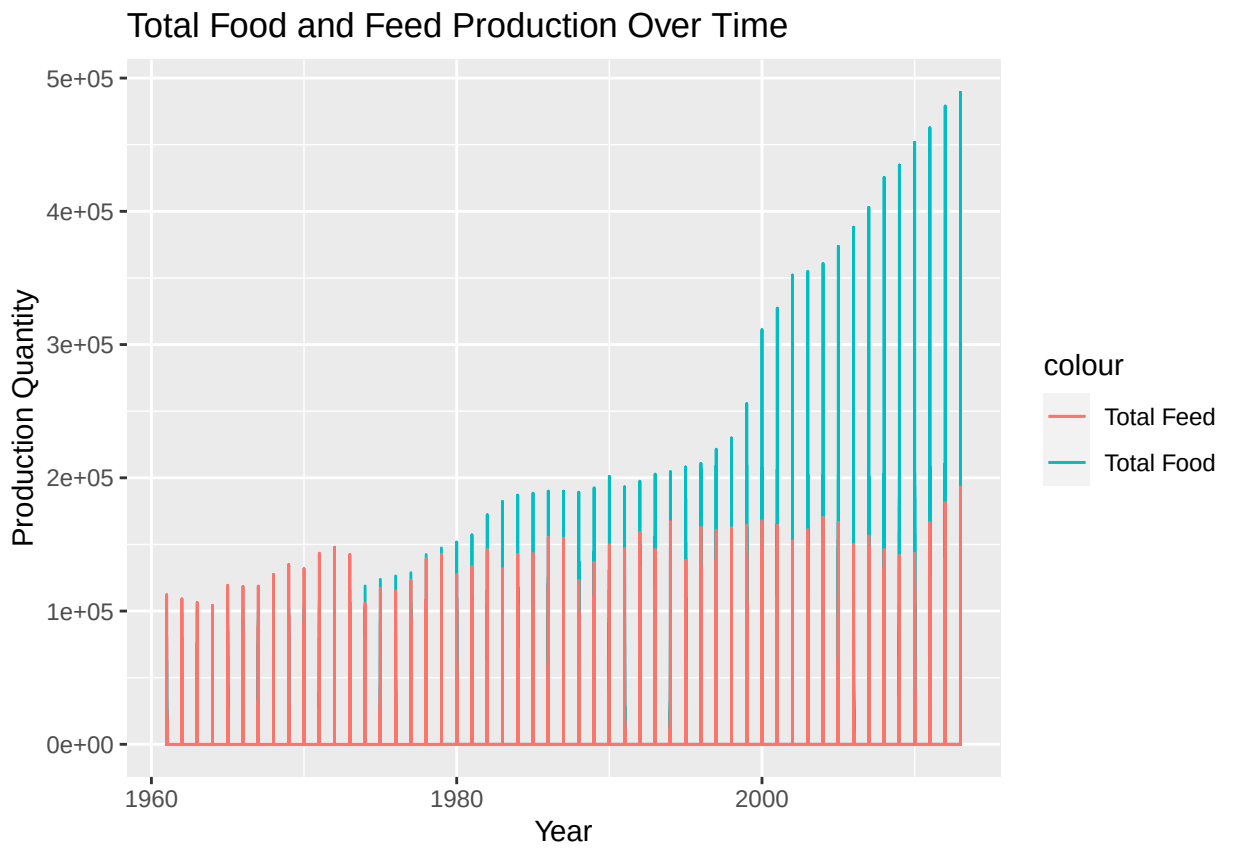
```r
# Aggregate data by Area, Item, Unit, Year, and Element
aggregated_data <- agriculture_analysis_data %>%
  group_by(Area, Item, Unit, Year, Element) %>%
  summarise(Production = sum(Production), .groups = "drop")

# Display a summary of the aggregated data
summary(aggregated_data)
```

```
##      Area               Item               Unit                Year
##  Length:999314      Length:999314      Length:999314      Min.   :1961
##  Class :character   Class :character   Class :character   1st Qu.:1975
##  Mode  :character   Mode  :character   Mode  :character   Median :1989
##                                                           Mean   :1988
##                                                           3rd Qu.:2002
##                                                           Max.   :2013
##  Element         Production
##  Feed:176766   Min.   :     0
##  Food:822548   1st Qu.:     0
##                Median :     3
##                Mean   :   379
##                3rd Qu.:    47
##                Max.   :489299
```

```r
# Pivot the data to wider format for better visualization
aggregated_data <- pivot_wider(
  data = aggregated_data,
  names_from = Element,
  values_from = Production,
  values_fill = 0
)
```

EXPLORATORY DATA ANALYSIS

```r
# Visualize the trends of Food and Feed production over time
ggplot(aggregated_data, aes(x = Year)) +
  geom_line(aes(y = Food, color = "Total Food")) +
  geom_line(aes(y = Feed, color = "Total Feed")) +
  labs(title = "Total Food and Feed Production Over Time",
       x = "Year", y = "Production Quantity")
```



#The data shows that food production has gone up a lot more than feed production over these years. There's a big difference in how much food and feed we produce

```
#Top 5 food producers since 1961

# Calculate top 5  food producers

largest_food_producers <- aggregated_data %>%
  group_by(Area) %>%
  summarize(Food = sum(Food, na.rm = TRUE)) %>%
  arrange(desc(Food))

# Display the top 5 largest food producers
top_n_producers <- 5
top_food_producers <- head(largest_food_producers, n = top_n_producers)

# Convert production to million tons for visualization
top_food_producers$Total_Food_Million_Tons <- top_food_producers$Food / 1e6

# Visualize the top 5 food producers in a bar chart
ggplot(top_food_producers, aes(x = reorder(Area, Total_Food_Million_Tons), y = Total_Food_Million_Tons)
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = paste("Top", top_n_producers, "Largest Food Producers"),
       x = "Country/Region", y = "Total Food Production(in  millon Tonnes)")
```

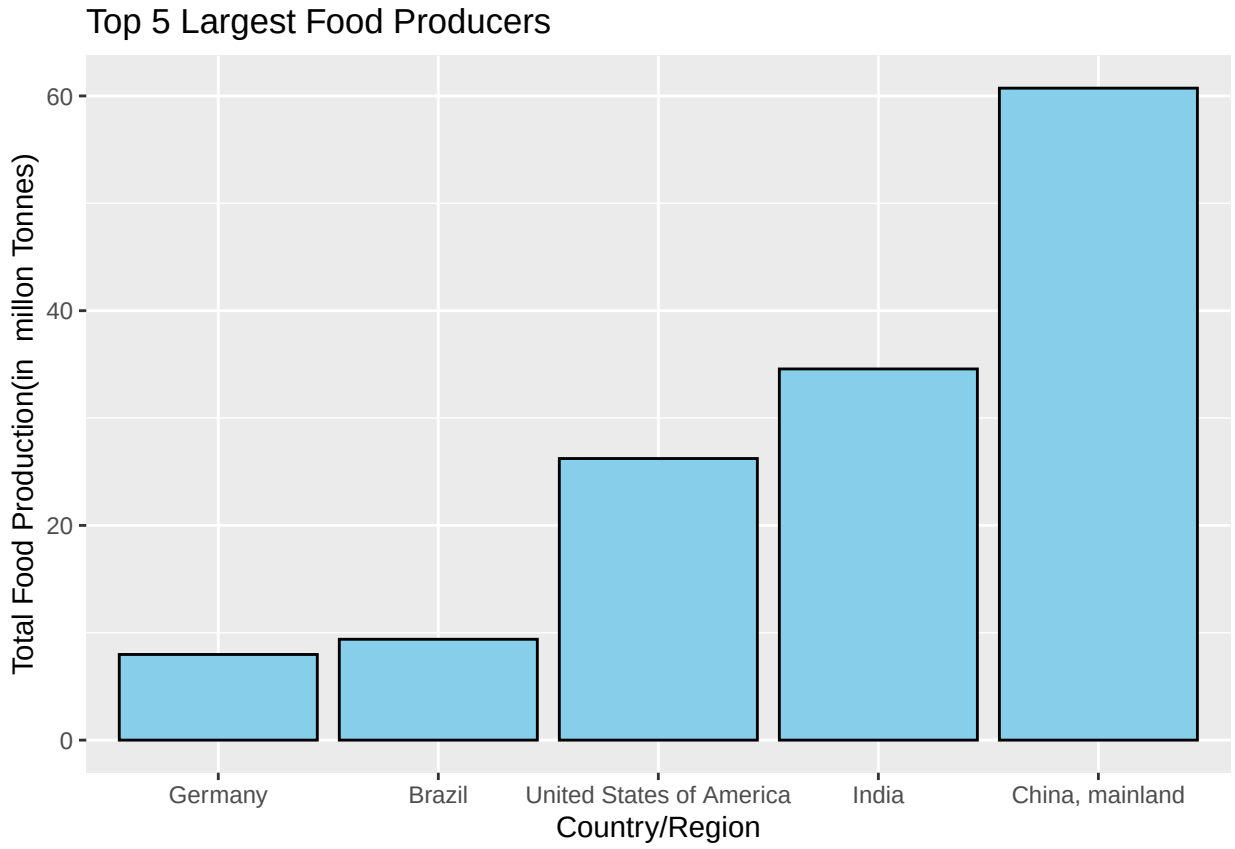


Figure 1: China, India, and the USA are the largest food producers. Among the top five, Germany and Brazil contribute the smallest amounts.

```r
# Similar steps for largest feed producers
largest_feed_producers <- aggregated_data %>%
  group_by(Area) %>%
  summarize(Feed = sum(Feed, na.rm = TRUE)) %>%
  arrange(desc(Feed))

# Display the top N largest food producers (e.g., top 5)
top_n_producers <- 5
top_feed_producers <- head(largest_feed_producers, n = top_n_producers)

# Convert production to million tons
top_feed_producers$Total_Feed_Million_Tons <- top_feed_producers$Feed / 1e6

# Visualization (bar chart)
ggplot(top_feed_producers, aes(x = reorder(Area, Total_Feed_Million_Tons), y = Total_Feed_Million_Tons)
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = paste("Top", top_n_producers, "Largest Feed Producers"),
       x = "Country/Region", y = "Total Feed Production(in  millon Tonnes)")
```

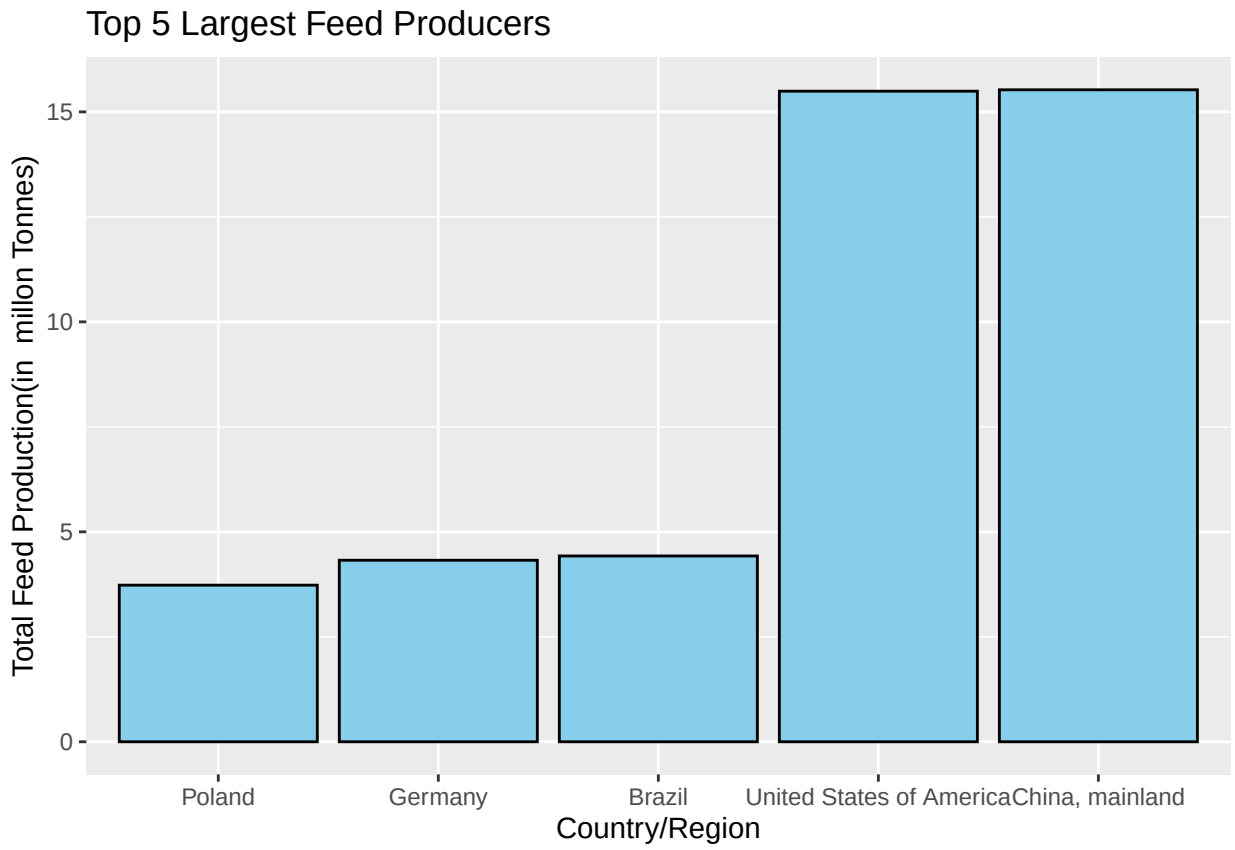


Figure 2: The United States and China top the list as the biggest producers of animal feed, with Poland, Germany, and Brazil lagging behind with significantly smaller production volumes.

```r
# Filter data for 'Food' element
food_data <- agriculture_analysis_data %>% filter(Element == "Food")
```

```r
# Group by item, calculate the total production for each item
item_production <- food_data %>%
  group_by(Item) %>%
  summarize(Total_Production = sum(Production, na.rm = TRUE)) %>%
  arrange(desc(Total_Production)) %>%
  head(5)  # Select the top 5 items

# Filter the original data for the top 10 food items
top_food_data <- food_data %>% filter(Item %in% item_production$Item)

top_food_data_aggregated <- top_food_data %>%
  group_by(Year, Item) %>%
  summarize(Total_Production = sum(Production, na.rm = TRUE))
```

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

```r
# Create a line plot for the aggregated production of top food items
ggplot(top_food_data_aggregated, aes(x = Year, y = Total_Production/1e6, color = Item)) +
  geom_line() +
  labs(title = "Aggregated Production Trends of Top 5 Food Items",
       x = "Year", y = "Total Production (Millions of Tons)",
       color = "Food Item") +
  theme(legend.position = "right")
```

#We will now investigate the leading items produced for animal feed

```r
# Filter data for 'Feed' element
feed_data <- agriculture_analysis_data %>% filter(Element == "Feed")

# Group by item, calculate the total production for each item
item_production_feed <- feed_data %>%
  group_by(Item) %>%
  summarize(Total_Production = sum(Production, na.rm = TRUE)) %>%
  arrange(desc(Total_Production)) %>%
  head(5)  # Select the top 10 items

# Filter the original data for the top 10 feed items
top_feed_data <- feed_data %>% filter(Item %in% item_production_feed$Item)

top_feed_data_aggregated <- top_feed_data %>%
  group_by(Year, Item) %>%
  summarize(Total_Production = sum(Production, na.rm = TRUE))
```

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

```r
# Create a line plot for the aggregated production of top feed items
ggplot(top_feed_data_aggregated, aes(x = Year, y = Total_Production/1e6, color = Item)) +
  geom_line() +
  labs(title = "Aggregated Production Trends of Top 5 Feed Items",
```

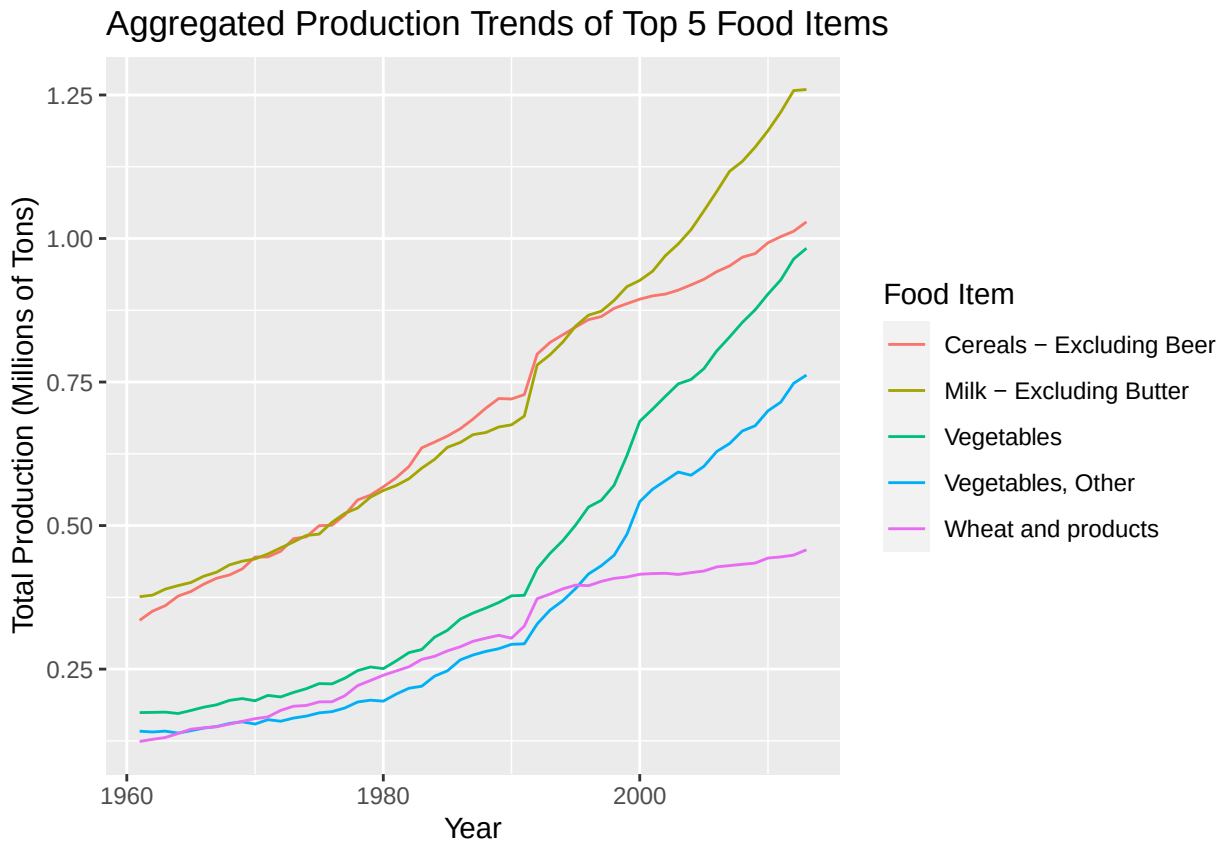


Figure 3: Since 1961, cereals and milk have topped the charts as the food items with the highest production.

```
      x = "Year", y = "Total Production (Millions of Tons)",
      color = "Feed Item") +
 theme(legend.position = "right")
```

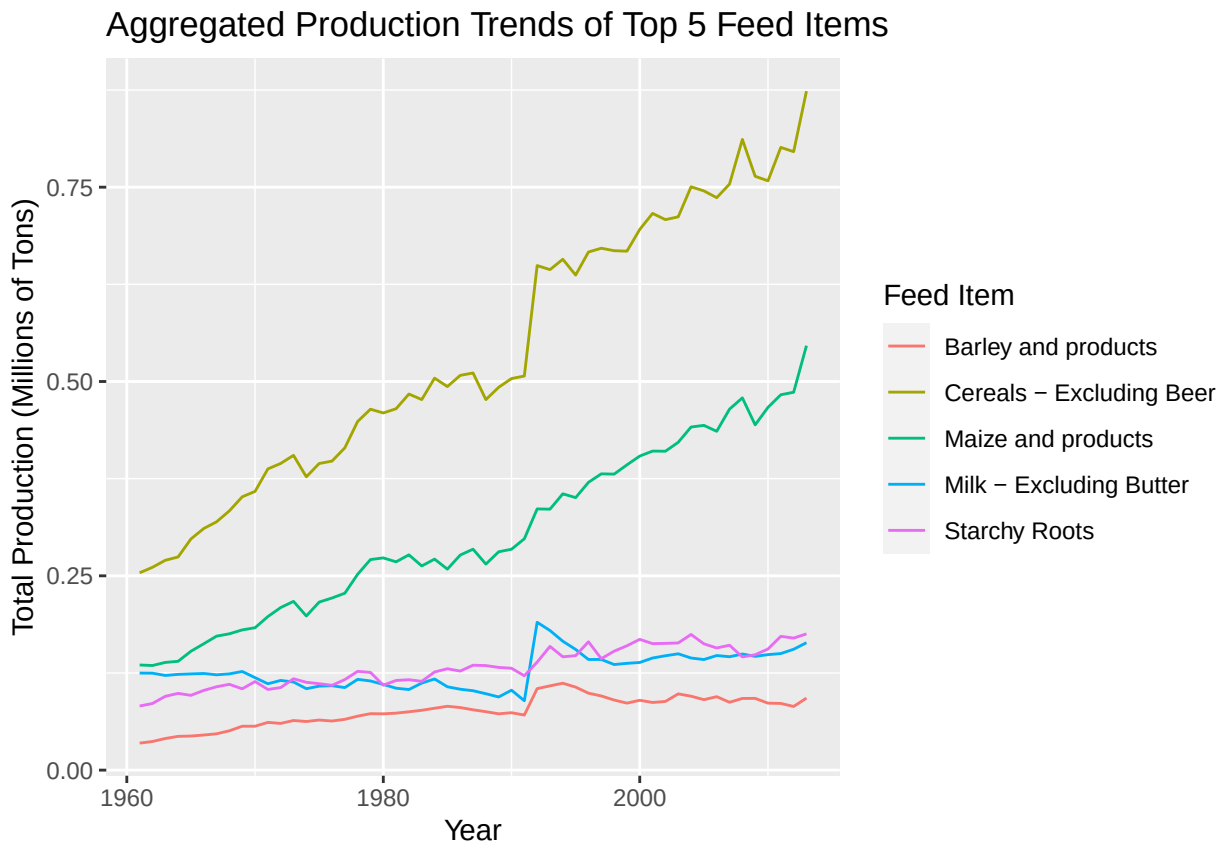


Figure 4: The production of cereals and maize as feed has sharply increased since the 1980s, while other feed items like barley, milk, and starchy roots have seen only modest growth.