Project 2 - Global High-Protein Food Production in Light of Carbon Footprint

Rachel Greenlee, Atina Karim, Douglas Barley

9/28/2020

Introduction

We choose to work with the dataset Rachel posted in last week's discussion forum.

Rachel found a dataset on Kaggle, originally from the Food and Agriculture Organization of the United Nations, that gives food production data for 245 countries. It shows what food items were produced for humans vs animals from 1961-2013. The years in the CSV file are displayed across the columns, making this a very wide dataset!

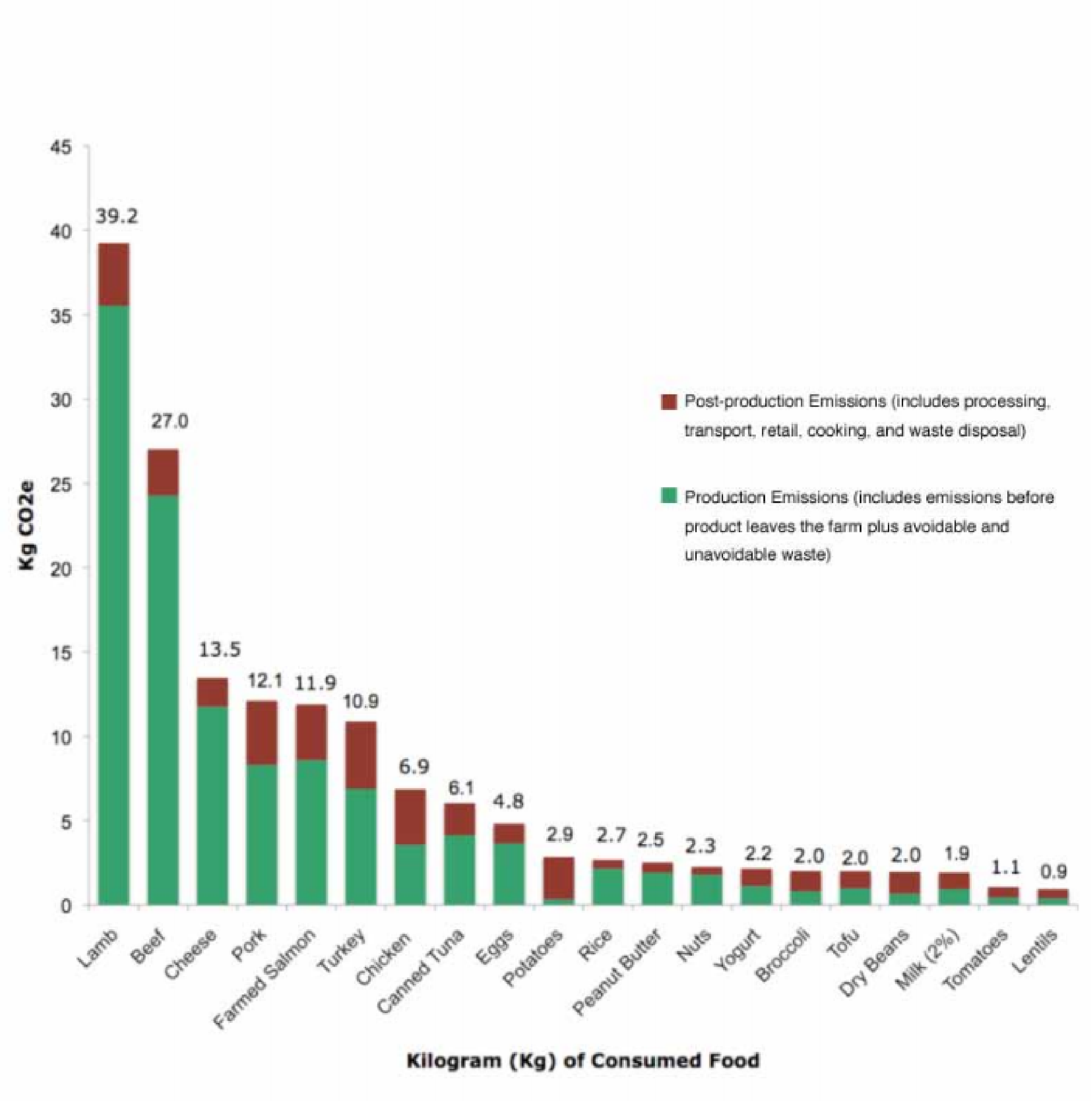
https://www.kaggle.com/dorbicycle/world-foodfeed-production

In this assignment we will compare the carbon footprint of high-protein foods. For simplicity's sake we used a separate resource to determine which food items to include. The graph below is taken from the 2011 report called "Meat Eater's Guide to Climate Change + Health" by The Environmental Working Group, a research and advocacy organization that partnered with CleanMetrics Corp, a environmental firm.

http://static.ewg.org/reports/2011/meateaters/pdf/methodology_ewg_meat_eaters_guide_to_health_and_climate_2011.pdf

Based on this I will compare legume production (legumes and dry beans) vs the top three animal meats with the highest carbon footprint, lamb (mutton), beef (bovine), and pork(pigmeat).

Figure 1. Full Lifecycle Assessment of Greenhouse Gas Emissions: Most Emissions from Common Proteins and Vegetables Occur During Production



***These include production emissions from avoidable (plate waste, spoilage) and unavoidable waste (fat and moisture loss during cooking)**

Step 1 - Import data from the provided CSV file and filter by food item

Let's load the libraries we might need.

```r
library("dplyr")
library("tidyr")
library("ggplot2")
library("rcartocolor")
library("kableExtra")
```

We've put the CSV file into my Github repository, we will read it from there, all 21,477 rows and 63 columns.

```r
food_rawdt <- read.csv(url("https://raw.githubusercontent.com/rachel-greenlee/data607_proj2/master/FAO.
```

Take a peek at the raw data.

```r
glimpse(food_rawdt)
```

```
## Rows: 21,477
## Columns: 63
## $ Area.Abbreviation <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", ...
## $ Area.Code         <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2...
## $ Area              <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afg...
## $ Item.Code         <int> 2511, 2805, 2513, 2513, 2514, 2514, 2517, 2520, 2...
## $ Item              <chr> "Wheat and products", "Rice (Milled Equivalent)",...
## $ Element.Code      <int> 5142, 5142, 5521, 5142, 5521, 5142, 5142, 5142, 5...
## $ Element           <chr> "Food", "Food", "Feed", "Food", "Feed", "Food", "...
## $ Unit              <chr> "1000 tonnes", "1000 tonnes", "1000 tonnes", "100...
## $ latitude          <dbl> 33.94, 33.94, 33.94, 33.94, 33.94, 33.94, 33.94, ...
## $ longitude         <dbl> 67.71, 67.71, 67.71, 67.71, 67.71, 67.71, 67.71, ...
## $ Y1961             <int> 1928, 183, 76, 237, 210, 403, 17, 0, 111, 45, 0, ...
## $ Y1962             <int> 1904, 183, 76, 237, 210, 403, 18, 0, 97, 45, 0, 4...
## $ Y1963             <int> 1666, 182, 76, 237, 214, 410, 19, 0, 103, 45, 0, ...
## $ Y1964             <int> 1950, 220, 76, 238, 216, 415, 20, 0, 110, 45, 0, ...
## $ Y1965             <int> 2001, 220, 76, 238, 216, 415, 21, 0, 113, 31, 0, ...
## $ Y1966             <int> 1808, 195, 75, 237, 216, 413, 22, 0, 117, 14, 16,...
## $ Y1967             <int> 2053, 231, 71, 225, 235, 454, 23, 0, 128, 19, 23,...
## $ Y1968             <int> 2045, 235, 72, 227, 232, 448, 24, 0, 130, 30, 31,...
## $ Y1969             <int> 2154, 238, 73, 230, 236, 455, 25, 0, 134, 34, 28,...
## $ Y1970             <int> 1819, 213, 74, 234, 200, 383, 26, 0, 125, 15, 9, ...
## $ Y1971             <int> 1963, 205, 71, 223, 201, 386, 26, 0, 147, 0, 13, ...
## $ Y1972             <int> 2215, 233, 70, 219, 216, 416, 27, 0, 138, 0, 13, ...
## $ Y1973             <int> 2310, 246, 72, 225, 228, 439, 27, 0, 143, 28, 6, ...
## $ Y1974             <int> 2335, 246, 76, 240, 231, 445, 28, 0, 160, 32, 0, ...
## $ Y1975             <int> 2434, 255, 77, 244, 234, 451, 29, 0, 169, 20, 0, ...
## $ Y1976             <int> 2512, 263, 80, 255, 240, 463, 37, 0, 324, 28, 10,...
## $ Y1977             <int> 2282, 235, 60, 185, 228, 439, 32, 0, 176, 24, 16,...
## $ Y1978             <int> 2454, 254, 65, 203, 234, 451, 33, 0, 225, 24, 13,...
## $ Y1979             <int> 2443, 270, 64, 198, 228, 440, 31, 0, 232, 34, 6, ...
## $ Y1980             <int> 2129, 259, 64, 202, 226, 437, 31, 0, 240, 61, 15,...
## $ Y1981             <int> 2133, 248, 60, 189, 210, 407, 29, 0, 247, 50, 0, ...
## $ Y1982             <int> 2068, 217, 55, 174, 199, 384, 27, 0, 248, 43, 0, ...
## $ Y1983                <int> 1994, 217, 53, 167, 192, 371, 28, 0, 242, 38, 0, ...
## $ Y1984                <int> 1851, 197, 51, 160, 182, 353, 26, 0, 235, 46, 0, ...
## $ Y1985                <int> 1791, 186, 48, 151, 173, 334, 25, 0, 226, 23, 0, ...
## $ Y1986                <int> 1683, 200, 46, 145, 170, 330, 23, 0, 217, 25, 0, ...
## $ Y1987                <int> 2194, 193, 46, 145, 154, 298, 23, 0, 196, 3, 0, 8...
## $ Y1988                <int> 1801, 202, 47, 148, 148, 287, 23, 0, 198, 45, 0, ...
## $ Y1989                <int> 1754, 191, 46, 145, 137, 265, 23, 0, 184, 54, 0, ...
## $ Y1990                <int> 1640, 199, 43, 135, 144, 279, 24, 0, 205, 47, 0, ...
## $ Y1991                <int> 1539, 197, 43, 132, 126, 245, 24, 0, 203, 29, 0, ...
## $ Y1992                <int> 1582, 249, 40, 120, 90, 170, 18, 0, 210, 29, 0, 5...
## $ Y1993                <int> 1840, 218, 50, 155, 141, 272, 22, 0, 210, 29, 0, ...
## $ Y1994                <int> 1855, 260, 46, 143, 150, 289, 20, 0, 211, 29, 0, ...
## $ Y1995                <int> 1853, 319, 41, 125, 159, 310, 21, 0, 212, 29, 0, ...
## $ Y1996                <int> 2177, 254, 44, 138, 108, 209, 17, 0, 213, 29, 0, ...
## $ Y1997                <int> 2343, 326, 50, 159, 90, 173, 20, 0, 214, 28, 0, 5...
## $ Y1998                <int> 2407, 347, 48, 154, 99, 192, 21, 0, 214, 28, 0, 5...
## $ Y1999                <int> 2463, 270, 43, 141, 72, 141, 17, 0, 217, 28, 0, 6...
## $ Y2000                <int> 2600, 372, 26, 84, 35, 66, 20, 0, 219, 29, 0, 61,...
## $ Y2001                <int> 2668, 411, 29, 83, 48, 93, 20, 0, 215, 29, 0, 61,...
## $ Y2002                <int> 2776, 448, 70, 122, 89, 170, 18, 0, 217, 29, 0, 7...
## $ Y2003                <int> 3095, 460, 48, 144, 63, 117, 16, 1, 347, 51, 0, 9...
## $ Y2004                <int> 3249, 419, 58, 185, 120, 231, 15, 2, 276, 50, 0, ...
## $ Y2005                <int> 3486, 445, 236, 43, 208, 67, 21, 1, 294, 29, 0, 1...
## $ Y2006                <int> 3704, 546, 262, 44, 233, 82, 11, 1, 294, 61, 0, 1...
## $ Y2007                <int> 4164, 455, 263, 48, 249, 67, 19, 0, 260, 65, 0, 1...
## $ Y2008                <int> 4252, 490, 230, 62, 247, 69, 21, 0, 242, 54, 0, 2...
## $ Y2009                <int> 4538, 415, 379, 55, 195, 71, 18, 0, 250, 114, 0, ...
## $ Y2010                <int> 4605, 442, 315, 60, 178, 82, 14, 0, 192, 83, 0, 2...
## $ Y2011                <int> 4711, 476, 203, 72, 191, 73, 14, 0, 169, 83, 0, 2...
## $ Y2012                <int> 4810, 425, 367, 78, 200, 77, 14, 0, 196, 69, 0, 2...
## $ Y2013                <int> 4895, 422, 360, 89, 200, 76, 12, 0, 230, 81, 0, 2...
```

Ok in looking at how this dataset is categorizing the crop food types, we will need to subset just the following items as they are called by this dataset:
Legumes {Pulses, Other and products; Pulses; Beans; Soyabeans}
3_Meats {Bovine Meat; Mutton & Goat Meat; Pigmeat}

Now lets filter this dataset down just to the food items we are interested in that were listed above.

```
food_dt <- food_rawdt %>%
  filter(Item %in% c("Bovine Meat", "Mutton & Goat Meat", "Pigmeat", "Pulses,
                     Other and products", "Pulses", "Beans", "Soyabeans"))
```

Step 2 - Tidy the data

First, we need to make this dataset long instead of wide, as currently the years run across columns.

```
food_dt <- food_dt %>%
  gather(Year, Amount, Y1961:Y2013)
```

Lets make a new variable that shows if the item is part of the "Pulses" or the "3_Meats" using the mutate function from dplyr and ifelse.

```
food_dt <- food_dt %>%
  mutate(food_dt, Category = ifelse(Item %in% c("Bovine Meat",
                                                "Mutton & Goat Meat", "Pigmeat"),
                                                "3_Meats", "Pulses"))
```

We can also drop some columns we don't need and reorder.

```
food_dt <- select(food_dt, -c(Area.Code, Item.Code, Element.Code, latitude,
                              longitude))

food_dt <- food_dt[, c(8, 3, 4, 2, 6, 5, 7, 1)]
```

We also want the Year variable to be an integer so we can plot it easily later.

```
food_dt$Year <- gsub("[^0-9.-]", "", food_dt$Year)

food_dt$Year <- as.numeric(food_dt$Year)
```

Step 3 - Exploratory analysis

Okay first lets take a peek at what the datset looks like now.

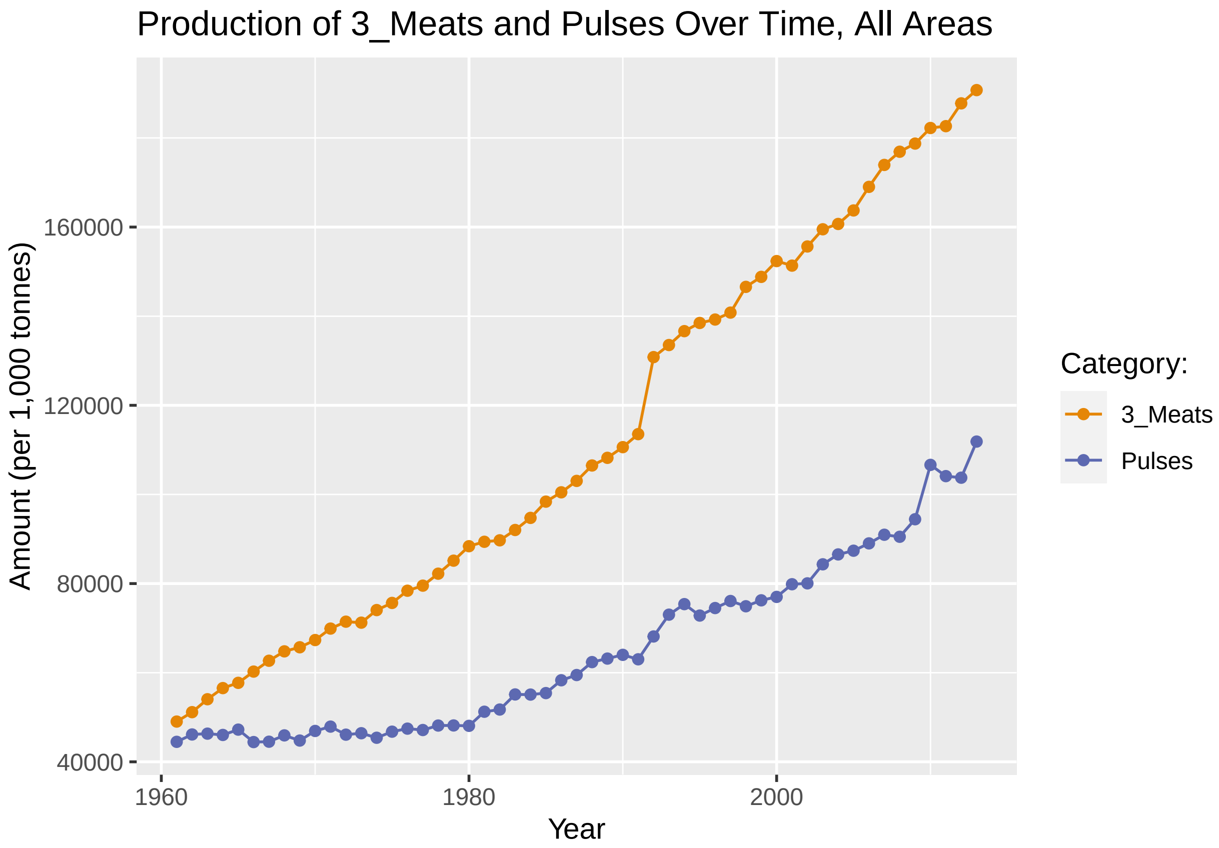
```
glimpse(food_dt)
```

```
## Rows: 63,653
## Columns: 8
## $ Category          <chr> "3_Meats", "3_Meats", "Pulses", "Pulses", "Pulses...
## $ Item              <chr> "Bovine Meat", "Mutton & Goat Meat", "Pulses", "P...
## $ Element           <chr> "Food", "Food", "Feed", "Food", "Feed", "Food", "...
## $ Area              <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afg...
## $ Year              <dbl> 1961, 1961, 1961, 1961, 1961, 1961, 1961, 1961, 1...
## $ Unit              <chr> "1000 tonnes", "1000 tonnes", "1000 tonnes", "100...
## $ Amount            <int> 43, 73, 1, 15, 0, 5, 0, 7, 13, 4, 2, 5, 4, 0, 0, ...
## $ Area.Abbreviation <chr> "AFG", "AFG", "AFG", "AFG", "ALB", "ALB", "ALB", ...
```

To start, lets combine all of the areas' total tonnes produced and see how the 3_Meats and Pulses production has changed over time for all areas.

Production of both of these food categories has increase since 1960, with the 3_Meats surpassing Pulses around 1965 and having a steady climb since then.

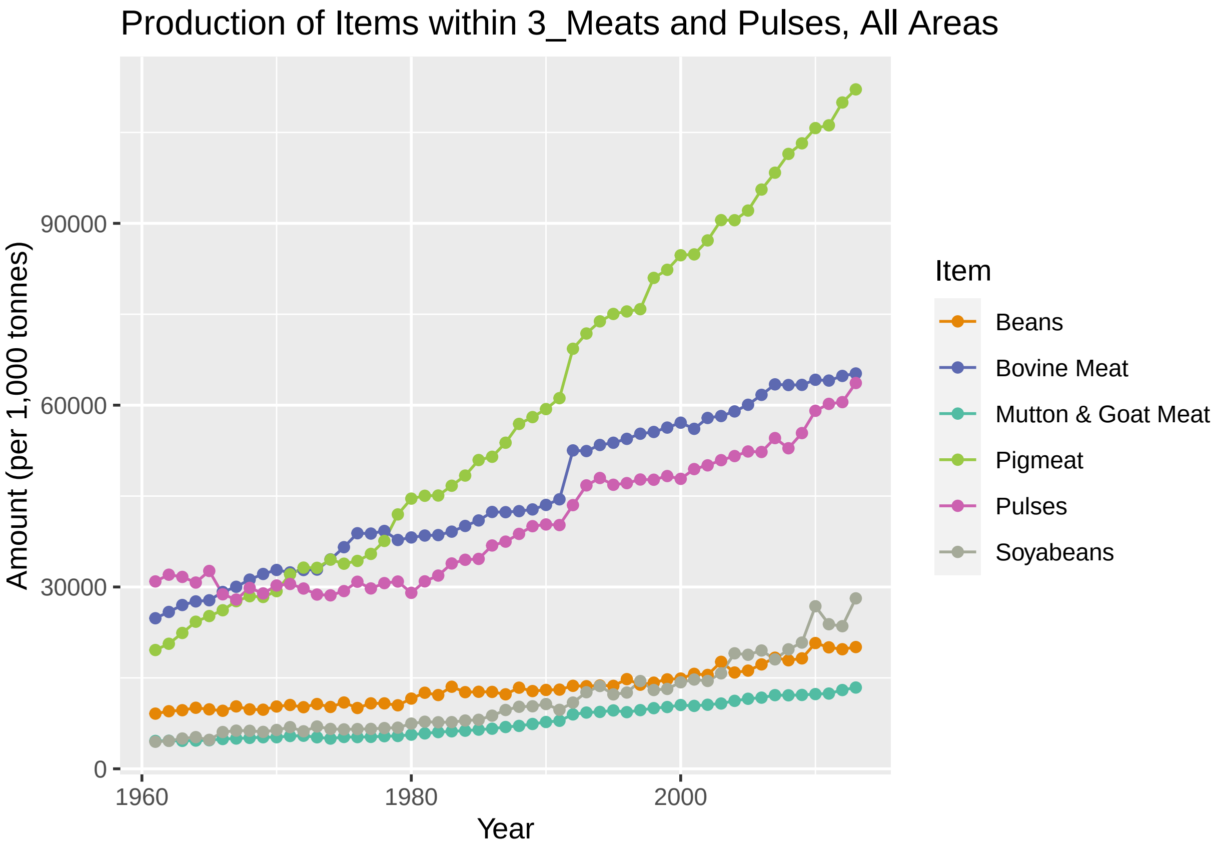
```
ggplot(food_dt, aes(x = Year, y = Amount, color=Category)) +
  stat_summary(fun=sum, geom="line") +
  stat_summary(fun=sum, geom="point") +
  labs(title="Production of 3_Meats and Pulses Over Time, All Areas") +
  ylab('Amount (per 1,000 tonnes)') +
  xlab('Year') +
  scale_color_carto_d(name = "Category: ", palette = "Vivid")
```



We can break this down and look at the types within 3_Meats and Pulses as well.

This shows that Pigmeat has had the most growth over the years of this dataset.

```
ggplot(food_dt, aes(x = Year, y = Amount, colour=Item)) +
  stat_summary(fun=sum, geom="line") +
  stat_summary(fun=sum, geom="point") +
  labs(title="Production of Items within 3_Meats and Pulses, All Areas") +
  ylab('Amount (per 1,000 tonnes)') +
  xlab('Year') +
  scale_color_carto_d(name = "Item", palette = "Vivid")
```

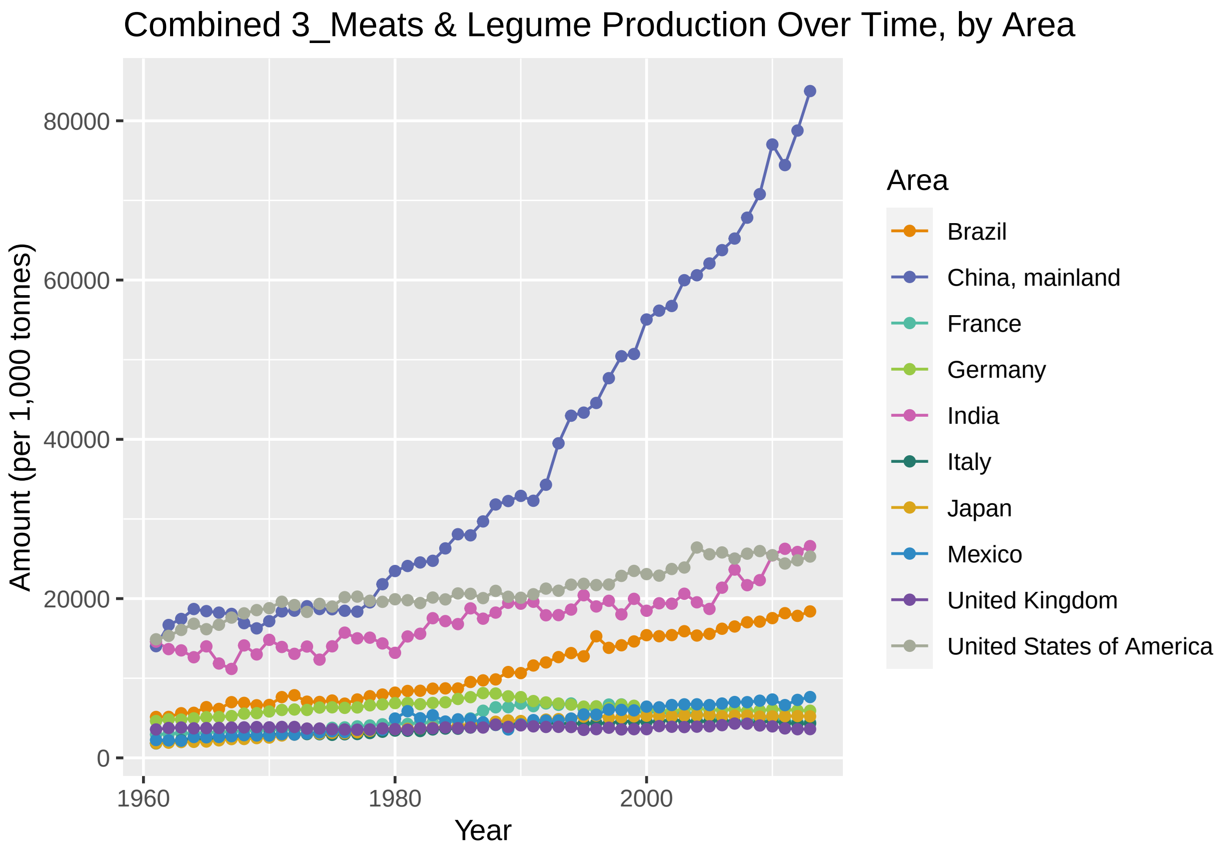


This is a lot of data, let's focus in on some of the largest producers when it comes to Legumes and/or 3_Meats. Presumably, the Areas with the largest production have the most impact on carbon footprint shifts from one food product to another. First we will find the top 10 producers of all year's combined, and Legumes & 3_Meats combines.

```
tot_production <- aggregate(food_dt$Amount, by=list(Area=food_dt$Area), FUN=sum)

tot_production <- top_n((tot_production[order(-tot_production$x),]), 10)

kable(tot_production,  format = "markdown")
```

Area	x
China, mainland	1967145
United States of America	1110256
India	930166
Brazil	581264
Germany	333710
France	252633
Mexico	251092
Japan	214020
United Kingdom	201316
Italy	195921

```r
#create a top 10 producers only subset
topareas_dt <- subset(food_dt, (Area %in% c("China, mainland", "India",
                                            "United States of America", "Brazil",
                                            "Germany", "Mexico", "France",
                                            "Japan", "Italy", "United Kingdom")))

#graph their total production over time
ggplot(topareas_dt, aes(x = Year, y = Amount, colour=Area)) +
  stat_summary(fun=sum, geom="line") +
  stat_summary(fun=sum, geom="point") +
  labs(title="Combined 3_Meats & Legume Production Over Time, by Area") +
  ylab('Amount (per 1,000 tonnes)') +
  xlab('Year') +
  scale_color_carto_d(name = "Area", palette = "Vivid")
```

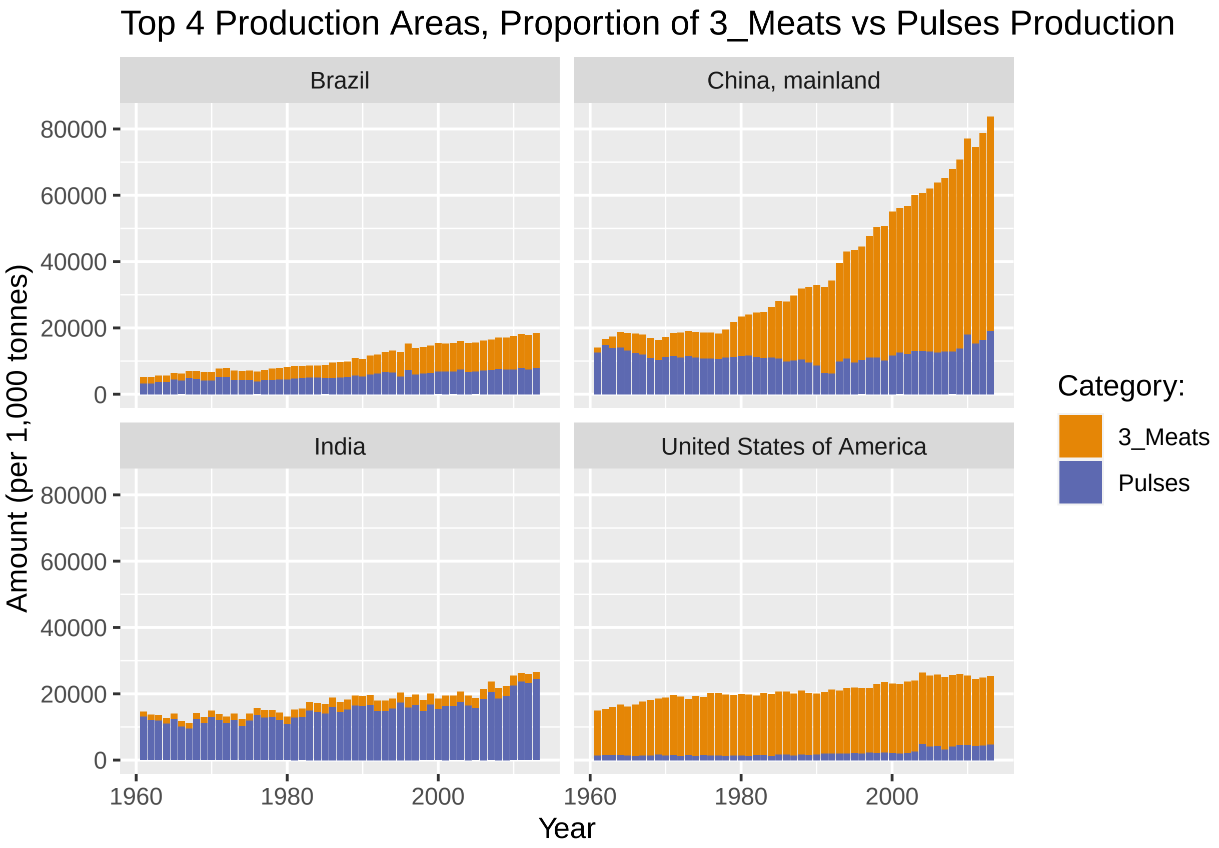


After we graph it, and see clearly there are 4 Areas with much higher production than the rest. We make a separate dataset for those countries, China, India, USA, and Brazil.

```r
top4_dt <- subset(topareas_dt, (Area %in% c("China, mainland", "India",
                                            "United States of America", "Brazil")))
```

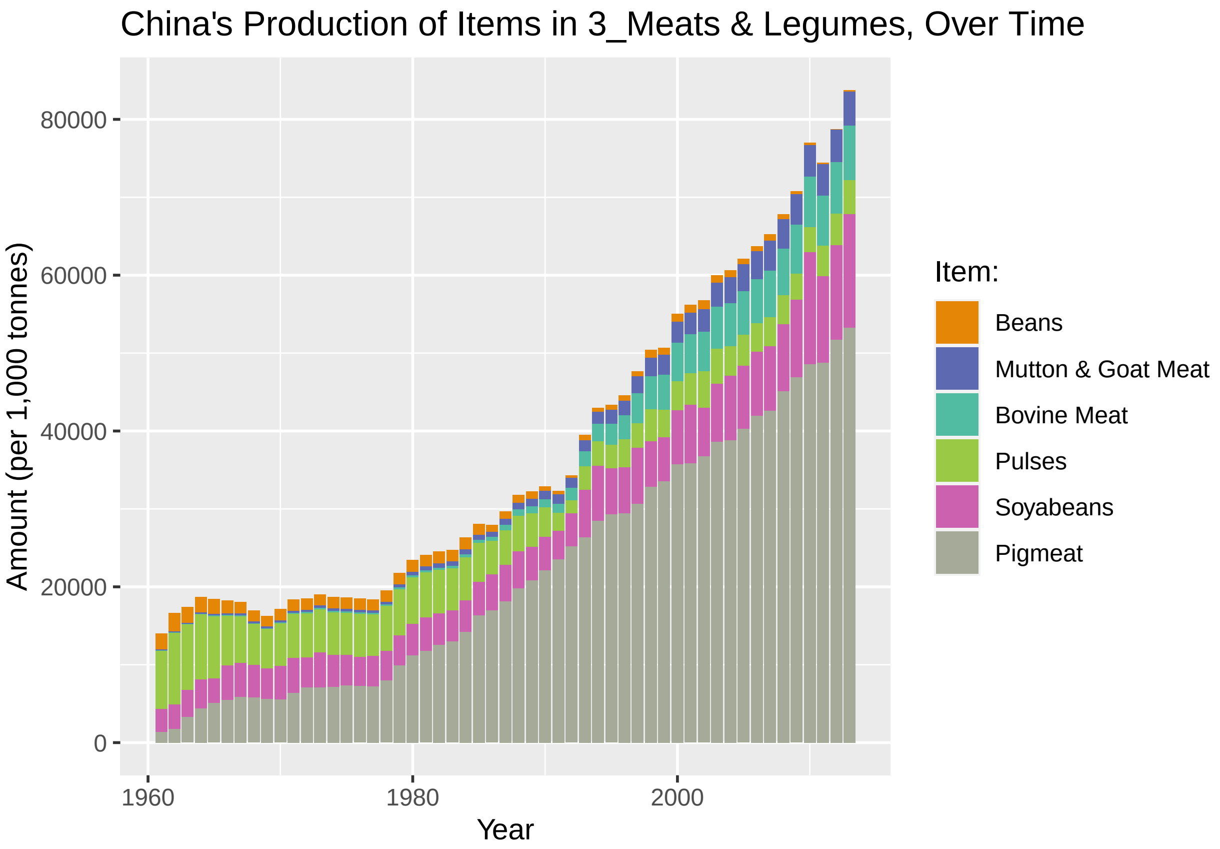
Let's look at each of these top 4 countries and their Legume v 3_Meats production over time.We can see that even by sheer tonnes produced China is hugely ahead of the other countries. China's Pulses production has stayed relatively steady over time, but the 3_Meats production has increased greatly over this time span. Closer to 1961 China was barely producing any of these 3_Meats, but as we approach 2013 the 3_Meats production explodes. We also see clearly that India, one of the country's with the highest vegetarian population, has a 3_Meats production that is miniscule compared to their Pulses production.

```
ggplot(top4_dt, aes(x = Year, y = Amount, fill = Category)) +
  geom_col() +
  facet_wrap(~Area) + labs(title = "test") +
  labs(title="Top 4 Production Areas, Proportion of 3_Meats vs Pulses Production") +
  ylab('Amount (per 1,000 tonnes)') +
  xlab('Year') +
  scale_fill_carto_d(name = "Category: ", palette = "Vivid")
```



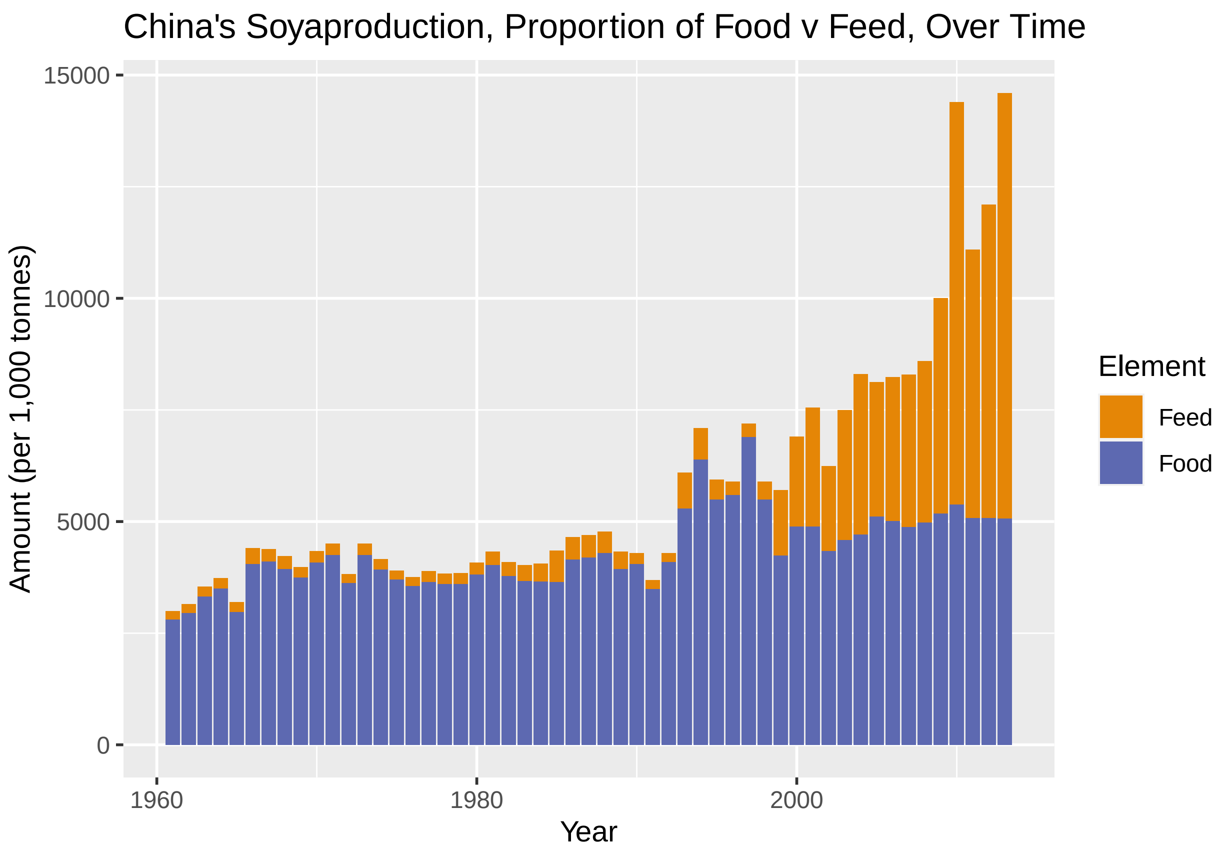
Finally, let's dig in one more time to take a closer look at China, as it's presumably the country which has the most leverage in food production with regards to lowering the overall carbon footprint. We see over time that Pigmeat grows to take up a huge proportion of these high-protein food Items.

```
#create China-only subset
china_dt <- subset(top4_dt, (Area == "China, mainland"))

ggplot(china_dt, aes(x = Year, y = Amount, fill = reorder(Item, Amount))) +
  geom_col() +
  labs(title="China's Production of Items in 3_Meats & Legumes, Over Time") +
  ylab('Amount (per 1,000 tonnes)') +
  xlab('Year') +
  scale_fill_carto_d(name = "Item: ", palette = "Vivid")
```



Soyabeans are the next largest category, but in investigating that further we see that growth doesn't appear to be for feeding humans ("Food") but rather for feeding animals ("Feed") - quite possibly the Pigs.

```
#create subset of China data with only Soyabeans
china_soya_dt <- subset(china_dt, (Item == "Soyabeans"))

ggplot(china_soya_dt, aes(x = Year, y = Amount, fill = Element)) +
  geom_col() +
  labs(title="China's Soyaproduction, Proportion of Food v Feed, Over Time") +
  ylab('Amount (per 1,000 tonnes)') +
  xlab('Year') +
  scale_fill_carto_d(name = "Element", palette = "Vivid")
```



Conclusion

There is much you could do with this large dataset, and the original poster on Kaggle used it to look more at the Food v Feed aspect of the data. In doing some exploratory analysis and graphing of this dataset, we eventually identified China as a huge producer of these food items we are looking at (Legumes & 3_Meats), and further that the amount of Pigmeat they've been producing has rapidly increased in the 1961-2013 timeframe.

From a carbon offset footprint and sheer numbers perspective, China could make a major impact in the carbon emissions by altering their food production. This could be done by reigning in their increasing production of Pigmeat and considering shifting to more environmentally-friendly protein sources such as Pulses that are consumes by humans, not necessarily animals.

If we were doing this analysis in an instance in which policy decisions were being made based on it, we'd definitely need some content experts from the food production industry to weigh in. We are positive there is nuance to this dataset that we're not aware of, and it'd be crucial to have someone more familiar with this industry on the team. For example, did domestic demand for Pigmeat drive the increase in Pigmeat production or are those exports?

Citation

Oppenheim, D. (2017, November). Who eats the food we grow?, Version 7. Retrieved October 2, 2020 from https://www.kaggle.com/dorbicycle/world-foodfeed-production.