# World Food Production

Martin Oberg

## Intro

This is an analysis of the *World Food and Feed Production*. While I am not trained in global food systems, this report shows a few ideas of the kinds of questions that could be asked of these data.

## Data Organization and Cleaning

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(broom)
```

```
data_path = "D:/data/world-foodfeed-prduction/FAO.csv"
col_spec = cols(
  .default = col_double(),
  `Area Abbreviation` = col_character(),
  Area = col_character(),
  Item = col_character(),
  Element = col_character(),
  Unit = col_character()
)
data = read_csv(data_path, col_names = TRUE, col_types = col_spec, locale(encoding = 'UTF-8'))
```

```
# ggplot is having trouble with UTF.  Renaming for quick fix
data$Area[data$`Area Code` == 107] = "Cote d'Ivore"
```

## Data cleaning and organization

First, we make some convenience tibbles that relate the various codes to their data value. Using the codes could be useful shortcuts for joining tables.

```
countries =
  data %>%
  select(Area, `Area Abbreviation`, `Area Code`, latitude, longitude) %>%
  group_by(Area, `Area Abbreviation`, `Area Code`) %>%
  distinct() %>%
  ungroup()
elements =
  data %>%
  select(Element, `Element Code`) %>%
  group_by(Element, `Element Code`) %>%
  unique()
items =
  data %>%
  select(Item, `Item Code`) %>%
  group_by(Item, `Item Code`) %>%
  ungroup() %>%
  unique()
units =
  data %>%
  select(Unit) %>%
  unique()
```

```
units %>% pull()
```

```
## [1] "1000 tonnes"
```

As we see below there is only one value for Unit. This frees us from having to do any unit conversions to make comparisons. We will drop this column and remember what unit the data are in.

Here we see that there are fewer Item Codes than Item labels.

```
items %>%
  select(Item) %>%
  distinct() %>%
  nrow()
```

```
## [1] 115
```

Here we look for duplicate *Item* labels.

```
items %>%
  group_by(Item) %>%
  nest() %>%
  mutate(n = map_int(data, nrow)) %>%
  filter(n>1) %>%
  unnest(cols = data)
```

```
## # A tibble: 4 x 3
## # Groups:   Item [2]
##   Item                      `Item Code`     n
##   <chr>                           <dbl> <int>
## 1 Eggs                             2744     2
## 2 Eggs                             2949     2
## 3 Milk - Excluding Butter          2848     2
## 4 Milk - Excluding Butter          2948     2
```

There might be a reason for the duplicate Item label, but without knowing more about the data we will combine these item codes to remove the duplicate *Item* name.

```
# Two ways of cleaning
data$`Item Code`[data$Item == 'Eggs'] = 2744
data$`Item Code`[data$`Item Code` == 2948] = 2848
# Remake the table
items =
  data %>%
  select(Item, `Item Code`) %>%
  group_by(Item, `Item Code`) %>%
  ungroup() %>%
  unique()
```

```
countries %>%
  count(`Area Abbreviation`) %>%
  arrange(desc(n)) %>%
  left_join(countries)
```

```
## Joining, by = "Area Abbreviation"
```

```
## # A tibble: 174 x 6
##    `Area Abbreviatio~     n Area                  `Area Code` latitude longitude
##    <chr>              <int> <chr>                       <dbl>    <dbl>     <dbl>
##  1 CHN                    4 China, Hong Kong SAR           96     22.4     114.
##  2 CHN                    4 China, Macao SAR              128     22.2     114.
##  3 CHN                    4 China, mainland                41     35.9     104.
##  4 CHN                    4 China, Taiwan Provin~         214     23.7     121.
##  5 AZE                    2 Azerbaijan                     52     40.1      47.6
##  6 AZE                    2 Bahamas                        12     25.0     -77.4
##  7 THA                    2 Thailand                      216     15.9     101.
##  8 THA                    2 The former Yugoslav ~         154     41.6      21.8
##  9 AFG                    1 Afghanistan                     2     33.9      67.7
## 10 AGO                    1 Angola                          7    -11.2      17.9
## # ... with 164 more rows
```

It looks like there are abbreviation errors for Bahamas and Macedonia. I will also leave the 4 areas of China as they are.

```
# Bahamas BS
# The former Yugoslav Republic of Macedonia MK
data$`Area Abbreviation`[data$`Area Code` == 12] = "BS"
data$`Area Abbreviation`[data$`Area Code` == 154] = "MK"
```

```r
# Remake the table
countries =
  data %>%
  select(Area, `Area Abbreviation`, `Area Code`, latitude, longitude) %>%
  group_by(Area, `Area Abbreviation`, `Area Code`) %>%
  distinct() %>%
  ungroup()
```

For this project I will be using the term "country" to refer to the geographical entities in the Area column. While it may be incorrect in some cases, the purpose of this project is to demonstrate what can be done with this type of data and not to inform policy or make geo-political statements. A more detailed analysis about specific areas would require more sensitivity to terminology.

We will need to reshape the data into a long format.

```r
# Here we rename the year columns to be used in a long tibble.  Dropping NAs means that Countries will
data_long =
  data %>%
  rename_with( ~ gsub("Y", "", .x, fixed = TRUE)) %>%   # Remove "Y" from year columns
  select(`Area Code`, Area, `Item Code`, Item, Element, starts_with('19'), starts_with('20')) %>%
  pivot_longer(
    cols = matches('[12]'),  # select 1XXX and 2XXX column names
    names_to = 'Year',
    values_drop_na = TRUE)

data_long$Year = as.numeric(data_long$Year)
```

```r
head(data_long)
```

```
## # A tibble: 6 x 7
##   `Area Code` Area        `Item Code` Item               Element  Year value
##         <dbl> <chr>             <dbl> <chr>              <chr>   <dbl> <dbl>
## 1           2 Afghanistan        2511 Wheat and products Food     1961  1928
## 2           2 Afghanistan        2511 Wheat and products Food     1962  1904
## 3           2 Afghanistan        2511 Wheat and products Food     1963  1666
## 4           2 Afghanistan        2511 Wheat and products Food     1964  1950
## 5           2 Afghanistan        2511 Wheat and products Food     1965  2001
## 6           2 Afghanistan        2511 Wheat and products Food     1966  1808
```

We have kept the *Area Code* and *Item Code* columns to save on typing during data exploration.

# Describing the data

## Element and Item overlap

First we will look at how Items are distributed across the Food/Feed Element category.

```r
item_by_element =
  data_long %>%
  select(Item, Element) %>%
  distinct() %>%
```

```
  group_by(Element) %>%
  nest() %>%
  mutate(N_Items = map_int(data, nrow))
item_by_element %>%
  select(-data)
```

```
## # A tibble: 2 x 2
## # Groups:   Element [2]
##   Element N_Items
##   <chr>     <int>
## 1 Food        115
## 2 Feed         88
```

This table shows that there is considerable overlap between *Food* and *Feed* categories. An *Item* can be classified as *Food* and *Feed*. This is something to keep in mind as we progress through the analysis.

## Number of Items by Country

How many different Items do countries produce?

```
count_data =
  data_long %>%
  filter(value > 0) %>%
  group_by(Area, Element, Item) %>%
  summarise(x = sum(value)) %>%
  nest() %>%
  mutate(N_Items = map_int(data, nrow))
```

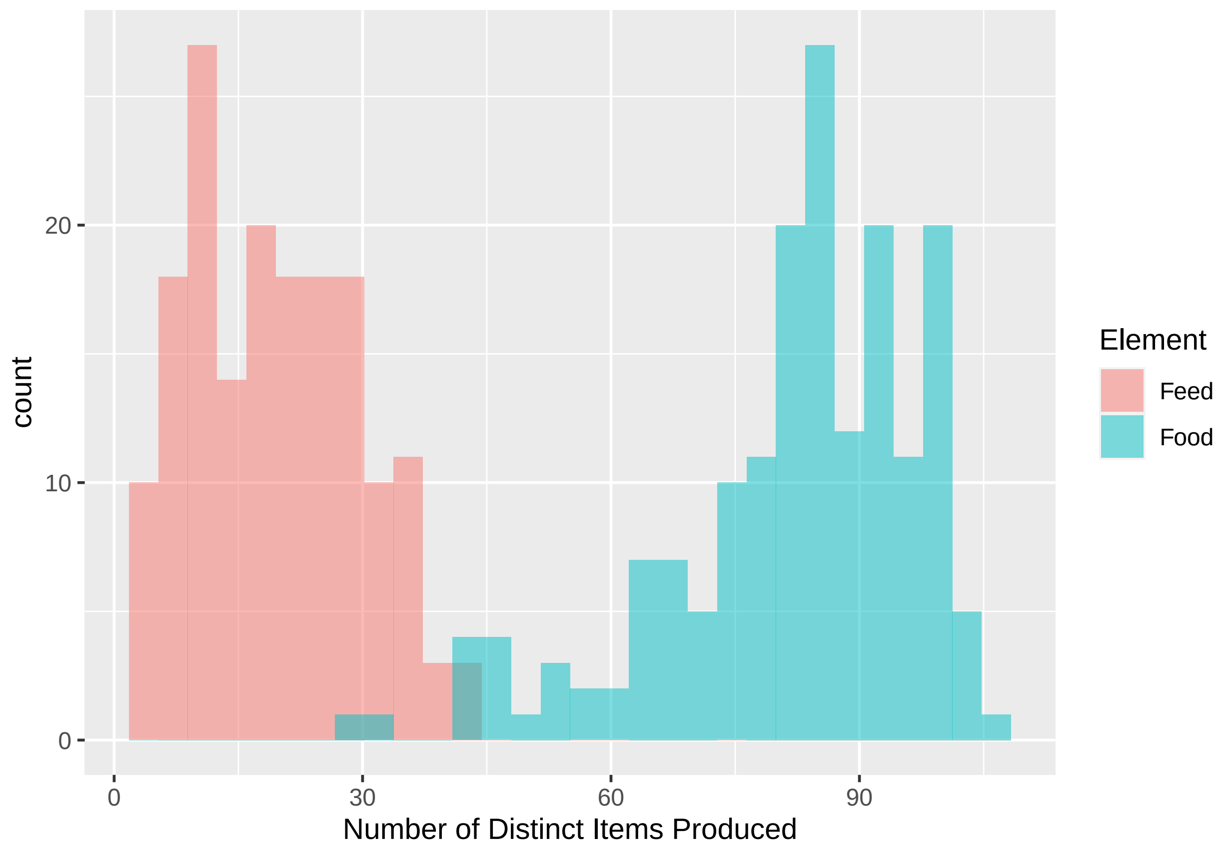
```
## `summarise()` regrouping output by 'Area', 'Element' (override with `.groups` argument)
```

```
#count_data
```

Let's have a look at the distribution of number of *Items* each country produces.

```
count_data %>%
  ggplot(aes(x=N_Items, fill=Element)) +
  geom_histogram(alpha=0.5,  position="identity") +
  xlab("Number of Distinct Items Produced")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




Countries generally produce a small number of *Feed* items and a large number of *Food* items. This could be partially explained by there being fewer *Feed* items overall. Another explanation could be that feed crops are more regionally specific.

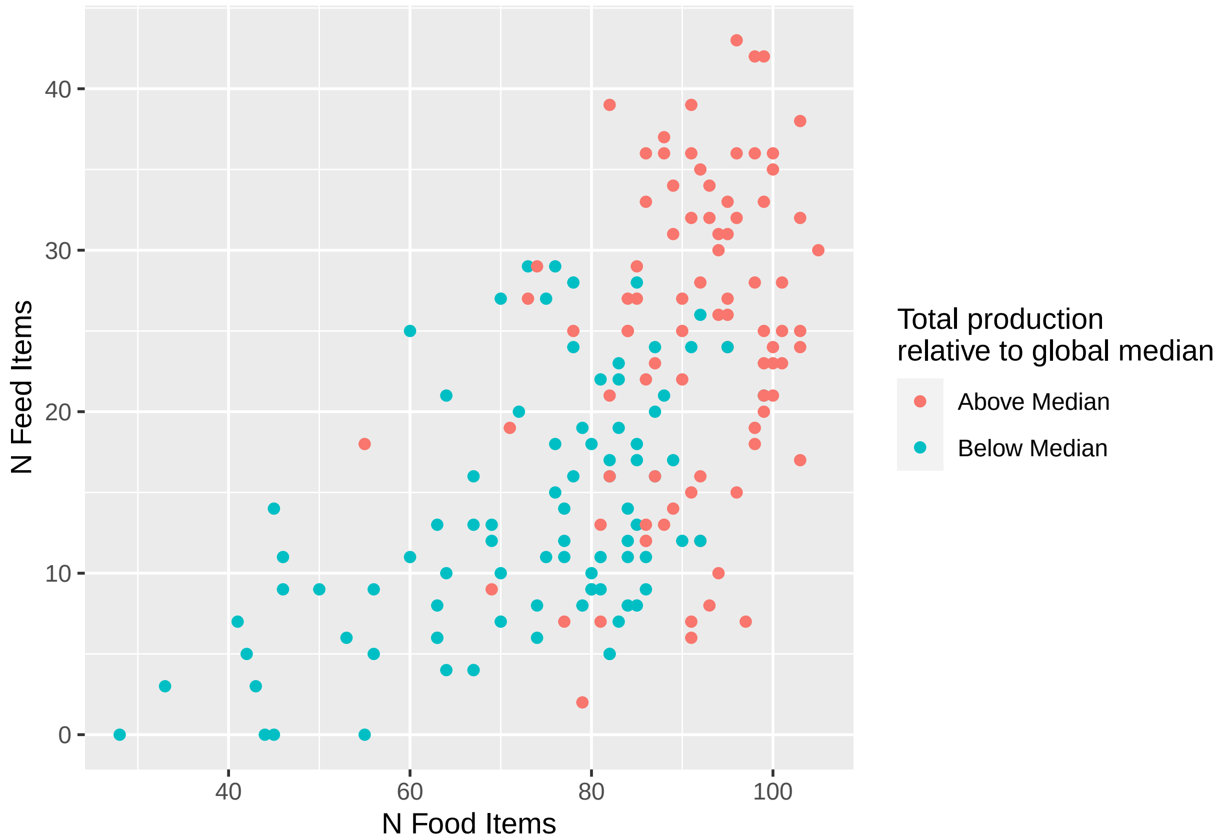
Now we can ask if there is a relationship between the number of Items produced and total production. We will look at average yearly production because countries have different amounts of yearly data.

```
total_production_by_country =
  data_long %>%
  group_by(Area) %>%
  summarise(`Average Production` = mean(value))
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
count_data %>%
  select(-data) %>%
  pivot_wider(names_from = Element, values_from = N_Items) %>%
  replace_na(list(Feed=0, Food=0)) %>%
  right_join(total_production_by_country) %>% ungroup() %>%
  mutate(quantile_rank = ntile(total_production_by_country$`Average Production`,2),
         quantile_rank = recode(quantile_rank, `1`="Below Median", `2`="Above Median")) %>%
  ggplot(aes(x=Food, y=Feed)) +
  geom_point(aes(color = factor(quantile_rank))) +
  scale_color_discrete(name="Total production\nrelative to global median")+
  labs(x = "N Food Items", y = "N Feed Items")
```

```
## Joining, by = "Area"
```




This scatter plot shows the number of Feed and Food Items produced by country and is color coded based on the total production relative to the global (median) average. Again we see that countries produce more kinds of Food items than Feed items. While no clear division exists between countries above and below median production, countries with more production produce more distinct Items.

We could follow up to see how countries around median production differ with some having more variety in terms of number of Items.

Another relationship to look in to could be number of Items produced and land area. More space could simply provide more varied conditions for more types of production. Also, whether a country is land locked could have a large effect on what it can produce. Furthermore, there is currently no categorization of whether the item is a crop, secondary item (butter, beer, oil, etc.), or sea/fisheries based.

### Amount of Feed and Food

Now we will look more closely at the amounts of production and not numbers of distinct items.

```
# Determine top 5 yearly producers
top5 =
  data_long %>%
  group_by(Area, Element) %>%
  summarise(Amount = sum(value)) %>%
  ungroup() %>%
  pivot_wider(names_from = Element, values_from = Amount) %>%
```

```
  mutate(TotFF = map2_dbl(Feed, Food, sum)) %>%
  slice_max(order_by = TotFF, n=5) %>%
  mutate(Label = Area) %>%
  select(Area, Label)
```

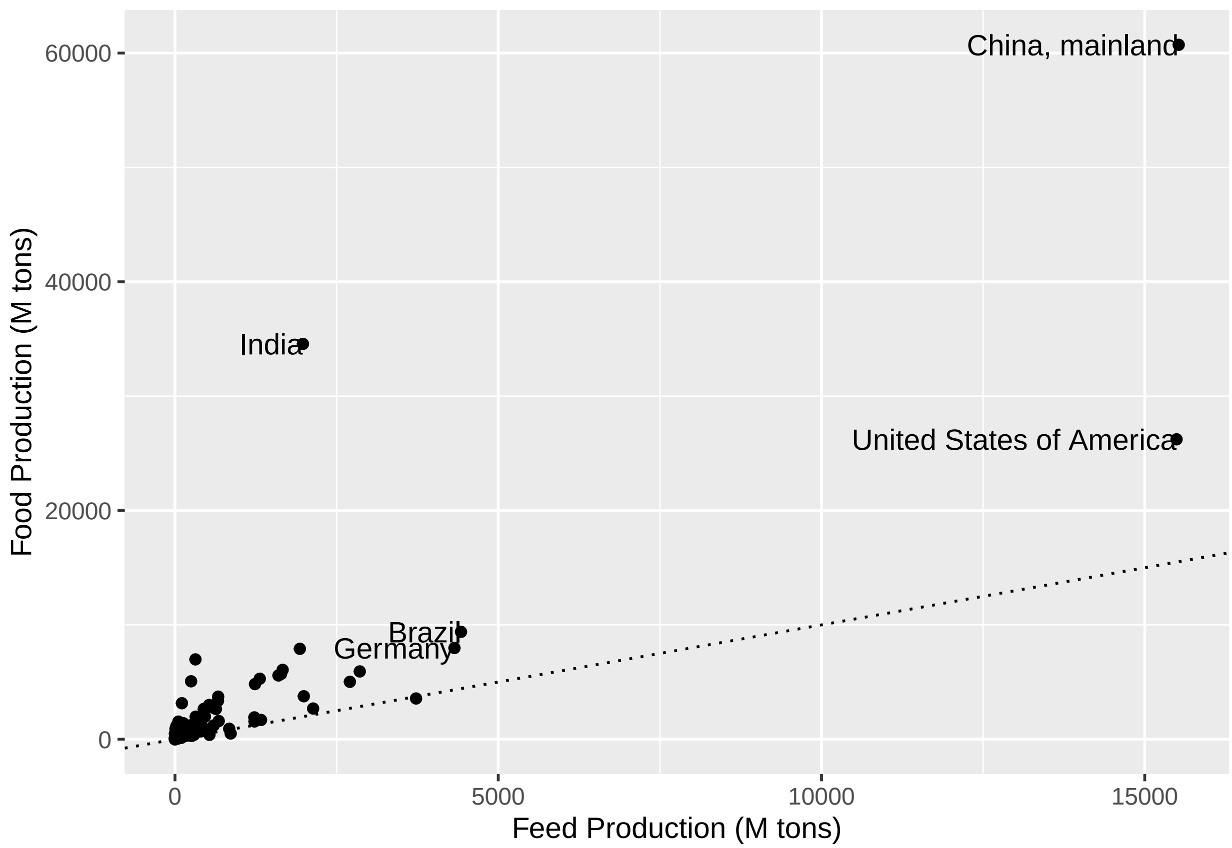
```
## ‘summarise()‘ regrouping output by ’Area’ (override with ‘.groups‘ argument)
```

```
data_long %>%
  group_by(Area, Element) %>%
  summarise(Amount = sum(value)) %>%
  ungroup() %>%
  pivot_wider(names_from = Element, values_from = Amount) %>%
  mutate(Feed = Feed / 1000,
         Food = Food / 1000) %>%
  left_join(top5) %>%
    ggplot(aes(Feed, Food, label=Label)) +
    geom_point()+
    #coord_equal()+
    geom_text(vjust = "center", hjust="right", check_overlap = FALSE) +
    xlab("Feed Production (M tons)") + # unit was originally per 1000 tonnes, and values scaled above b
    ylab("Food Production (M tons)") +
    geom_abline(intercept = 0, linetype="dotted")
```

```
## ‘summarise()‘ regrouping output by ’Area’ (override with ‘.groups‘ argument)
```

```
## Joining, by = "Area"
```

```
## Warning: Removed 169 rows containing missing values (geom_text).
```




This graph shows all countries and their total yearly Feed and Food production with top 5 overall producers labeled. The dotted line shows equal Food and Feed values. No country has produced drastically more Feed than Food. We should also look at average yearly production.

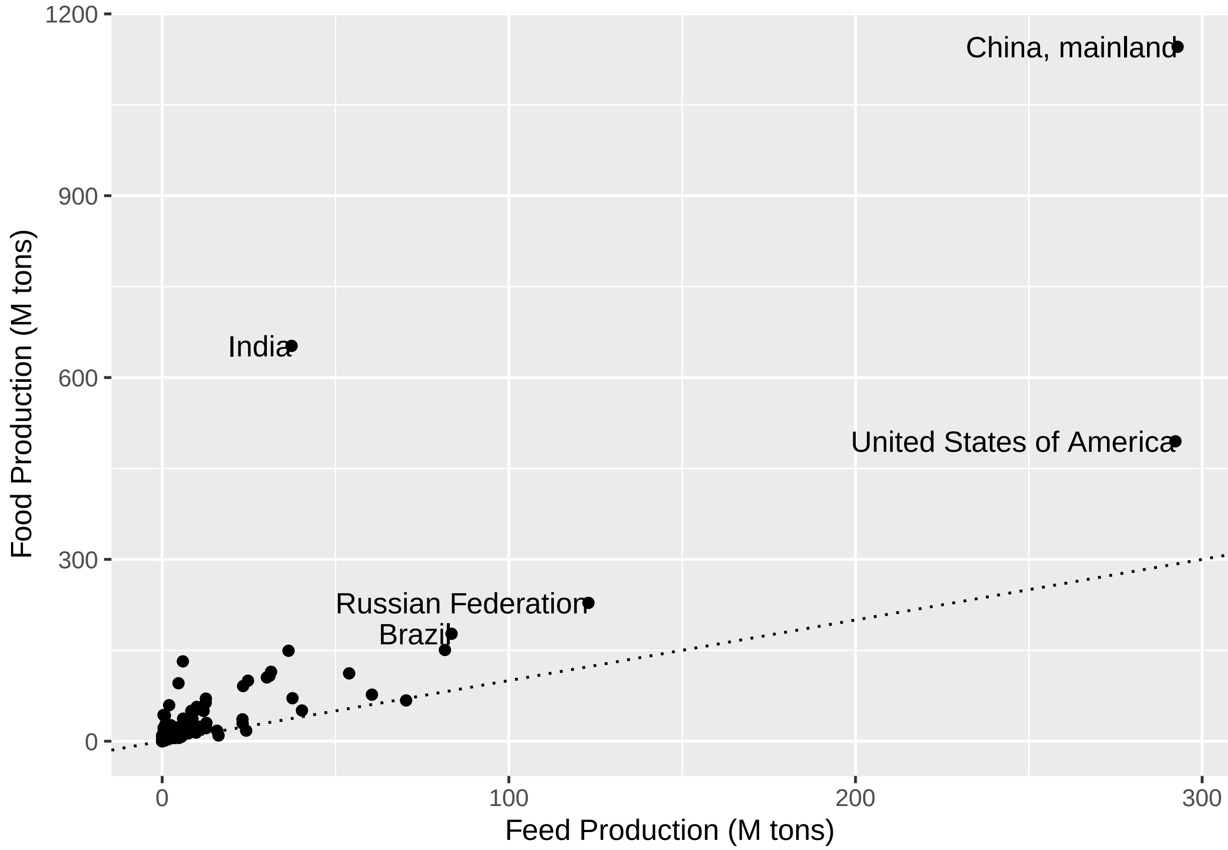
```
top5avg =
  data_long %>%
  group_by(Area, Element, Year) %>%
    summarise(yearSum = sum(value)) %>%
  ungroup() %>%
  group_by(Area, Element) %>%
    summarise(yearAvg = mean(yearSum)) %>%
  ungroup() %>%
  pivot_wider(names_from = Element, values_from = yearAvg) %>%
  mutate(TotFF = map2_dbl(Feed, Food, sum)) %>%
  slice_max(order_by = TotFF, n=5) %>%
  mutate(Label = Area) %>%
  select(Area, Label)
```

```
## `summarise()` regrouping output by 'Area', 'Element' (override with `.groups` argument)

## `summarise()` regrouping output by 'Area' (override with `.groups` argument)
```

```
data_long %>%
  group_by(Area, Element, Year) %>%
    summarise(yearSum = sum(value)) %>%
```

```
  ungroup() %>%
  group_by(Area,Element) %>%
    summarise(yearAvg = mean(yearSum)) %>%
  ungroup() %>%
  pivot_wider(names_from = Element, values_from = yearAvg) %>%
  mutate(Feed = Feed / 1000,
         Food = Food / 1000) %>%
  left_join(top5avg) %>%
    ggplot(aes(x=Feed, y=Food, label=Label)) +
    geom_point()+
    #coord_equal()+
    geom_text(vjust = "center", hjust="right", check_overlap = FALSE) +
    xlab("Feed Production (M tons)") + # unit was originally per 1000 tonnes, and values scaled above by
    ylab("Food Production (M tons)") +
    geom_abline(intercept = 0, linetype="dotted")
```

```
## `summarise()` regrouping output by 'Area', 'Element' (override with `.groups` argument)
## `summarise()` regrouping output by 'Area' (override with `.groups` argument)

## Joining, by = "Area"

## Warning: Removed 169 rows containing missing values (geom_text).
```




This looks nearly the same except the data are scaled by the number of years each country has been recorded. There are many interpretations that food systems experts could draw from this. For now, let's look at how this Food-Feed relationship changes over time.

## Percent of production as Feed

Instead of plotting Feed against Food we can calculate Feed as a percent of the total. Because *Feed* presumably refers to livestock feed and is used as an "input" for production, we will compute percent of production as feed ( *Percent Feed* ) rather than percent as food which, while statistically valid, would gloss over this relationship. With yearly data we can compute the *Percent Feed* over time to look for long term trends.

```
ratio_data =
  data_long %>%
  group_by(Area, Year, Element) %>%
  summarize(Value = sum(value, na.rm=T)) %>%
  pivot_wider(names_from = Element, values_from = Value) %>%
  mutate('Percent Feed' = Feed / (Feed + Food) * 100,
         'Total Production' = Feed + Food)
```

```
## `summarise()` regrouping output by 'Area', 'Year' (override with `.groups` argument)
```

```
head(ratio_data)
```

```
## # A tibble: 6 x 6
## # Groups:   Area, Year [6]
##   Area         Year  Feed  Food `Percent Feed` `Total Production`
##   <chr>       <dbl> <dbl> <dbl>          <dbl>              <dbl>
## 1 Afghanistan  1961   720  8761           7.59               9481
## 2 Afghanistan  1962   720  8694           7.65               9414
## 3 Afghanistan  1963   736  8458           8.01               9194
## 4 Afghanistan  1964   740  9430           7.28              10170
## 5 Afghanistan  1965   720  9753           6.87              10473
## 6 Afghanistan  1966   724  9445           7.12              10169
```

Now we can now check if there are other trends in *Percent Feed* over time. We will fit a linear model for each country that predicts *Percent Feed* as a function of *Year*. These trends would certainly be tied to societal, government, climate, and other factors which could have short term or long term effects and would be analyzed in conjunction with other data.

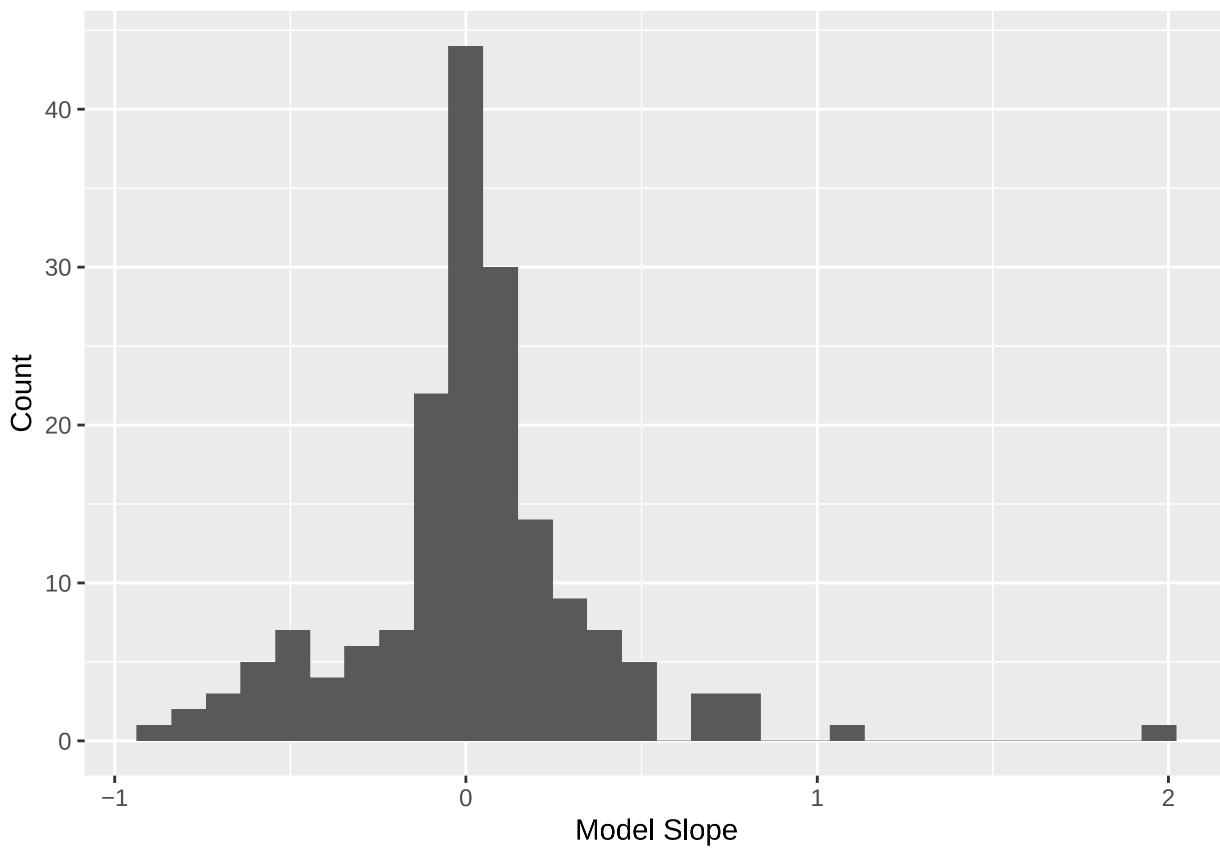
```
country_model = function(df) {
  lm(`Percent Feed`~ Year, data = df)
}

model_data =
  ratio_data %>%
  # We are making one model per country
  group_by(Area) %>%
  nest() %>%
  mutate(model  = map(data, country_model),
         tidy   = map(model, broom::tidy),
         glance = map(model, broom::glance),
         rsq    = map_dbl(glance, "r.squared"),
         #augment= map(model, broom::augment),
         Sl   = map(tidy, "estimate"),
         Slope_pctFeed = map_dbl(Sl, 2)) %>%
  select(Area, data, rsq, Slope_pctFeed) %>%
  ungroup()
head(model_data)
```

```
## # A tibble: 6 x 4
##   Area                data                 rsq Slope_pctFeed
##   <chr>               <list>             <dbl>         <dbl>
## 1 Afghanistan         <tibble [53 x 5]> 0.197       -0.0359
## 2 Albania             <tibble [53 x 5]> 0.737        0.174
## 3 Algeria             <tibble [53 x 5]> 0.573        0.192
## 4 Angola              <tibble [53 x 5]> 0.575        0.659
## 5 Antigua and Barbuda <tibble [53 x 5]> 0.150       -0.0294
## 6 Argentina           <tibble [53 x 5]> 0.306       -0.122
```

The slope of the model represents how much on average *Percent Feed* changes over time. $R^2$ (rsq) is a measure of how closely the data follow that trend, with 1 being a perfect fit and 0 representing no relation to the model at all.

```
model_data %>%
  select(Slope_pctFeed) %>%
  ggplot( aes(x=Slope_pctFeed)) +
  geom_histogram() +
  xlab("Model Slope") +
  ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




Here we see that model slope is normally distributed with most countries having near 0 slope, i.e. no relationship between *Percent Feed* and *Year*. Positive and negative slopes correspond to *Percent Feed* increasing and decreasing over time.

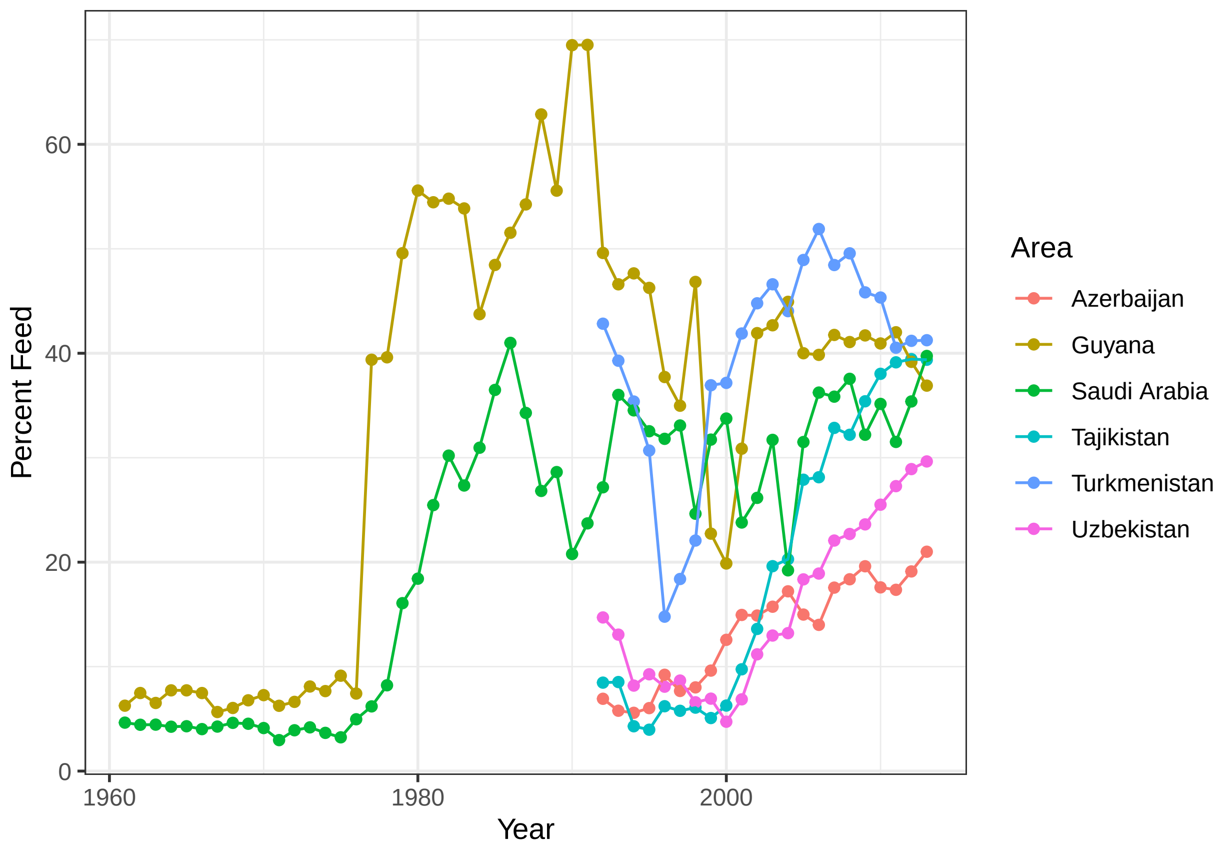
Let us have a look at the countries that have the largest changes over time.

```
model_data %>%
  select(Area, Slope_pctFeed, rsq) %>%
  arrange(desc(abs(Slope_pctFeed))) %>%
  head()
```

```
## # A tibble: 6 x 3
##   Area               Slope_pctFeed    rsq
##   <chr>                      <dbl>  <dbl>
## 1 Tajikistan                 2.01  0.886
## 2 Uzbekistan                 1.05  0.704
## 3 Lithuania                 -0.852 0.632
## 4 Turkmenistan               0.828 0.291
## 5 Serbia                    -0.811 0.0526
## 6 Russian Federation        -0.800 0.516
```

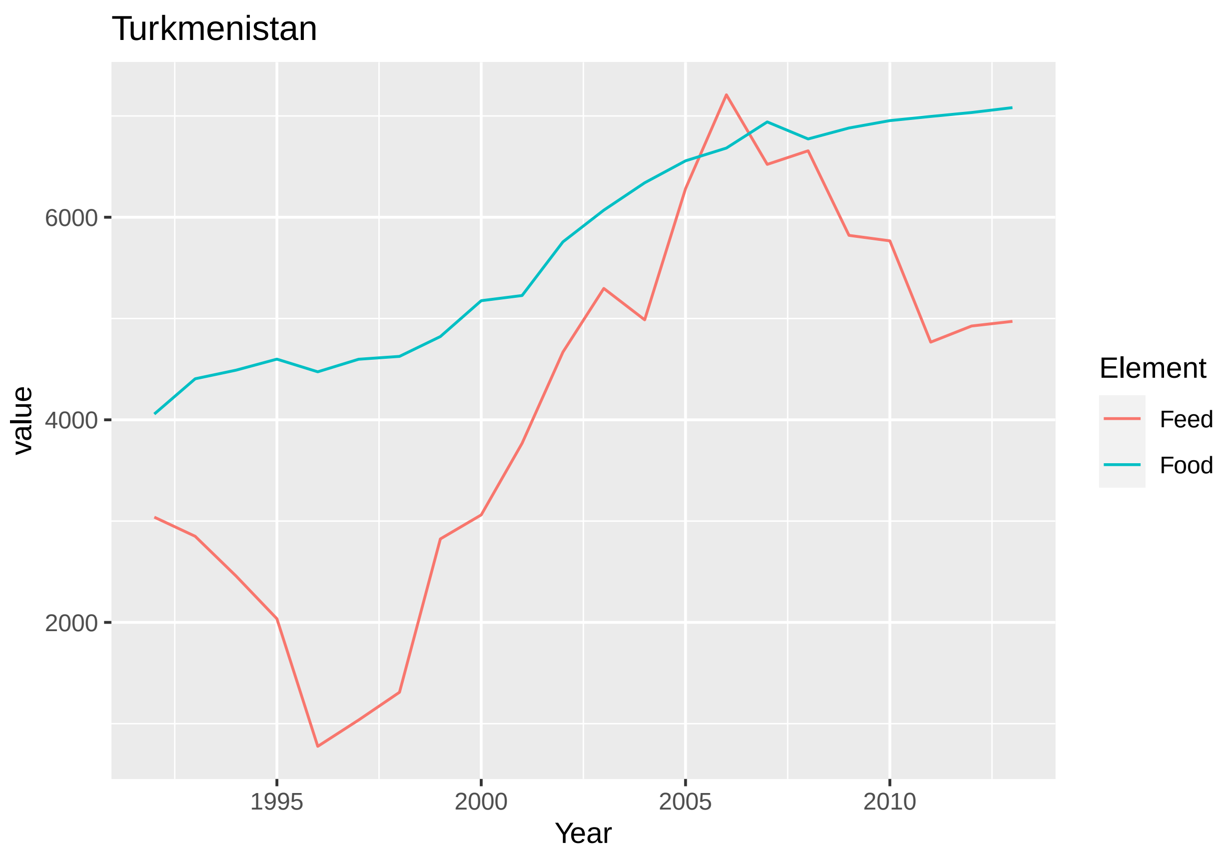
Looking at $R^2$, we see that some models are good (Tajikistan) and others are not so great (Serbia and Turkmenistan). Graphing some of these will provide some insight.

```
model_data %>%
  arrange(desc(Slope_pctFeed)) %>%
  head() %>%
  select(Area, data, rsq, Slope_pctFeed) %>%
  unnest(data) %>%
  ggplot(aes(x=Year, y=`Percent Feed`, color =Area)) +
  #ggtitle(Area[[1]]) +
  #facet_grid(vars(Area)) +
  geom_point(aes(color=Area)) +
  geom_line() +
  theme_bw()
```




Here we see that Turkmenistan had a drop in Percent Feed around 1995. Let us have a closer look.

```
ratio_data %>%
  filter(Area == "Turkmenistan") %>%
  select(-`Percent Feed`, -`Total Production`) %>%
  pivot_longer(cols = starts_with('F'), names_to = "Element" ) %>%
  ggplot(aes(x=Year, y=value, group=Element, color=Element)) +
  geom_line() +
  ggtitle("Turkmenistan")
```

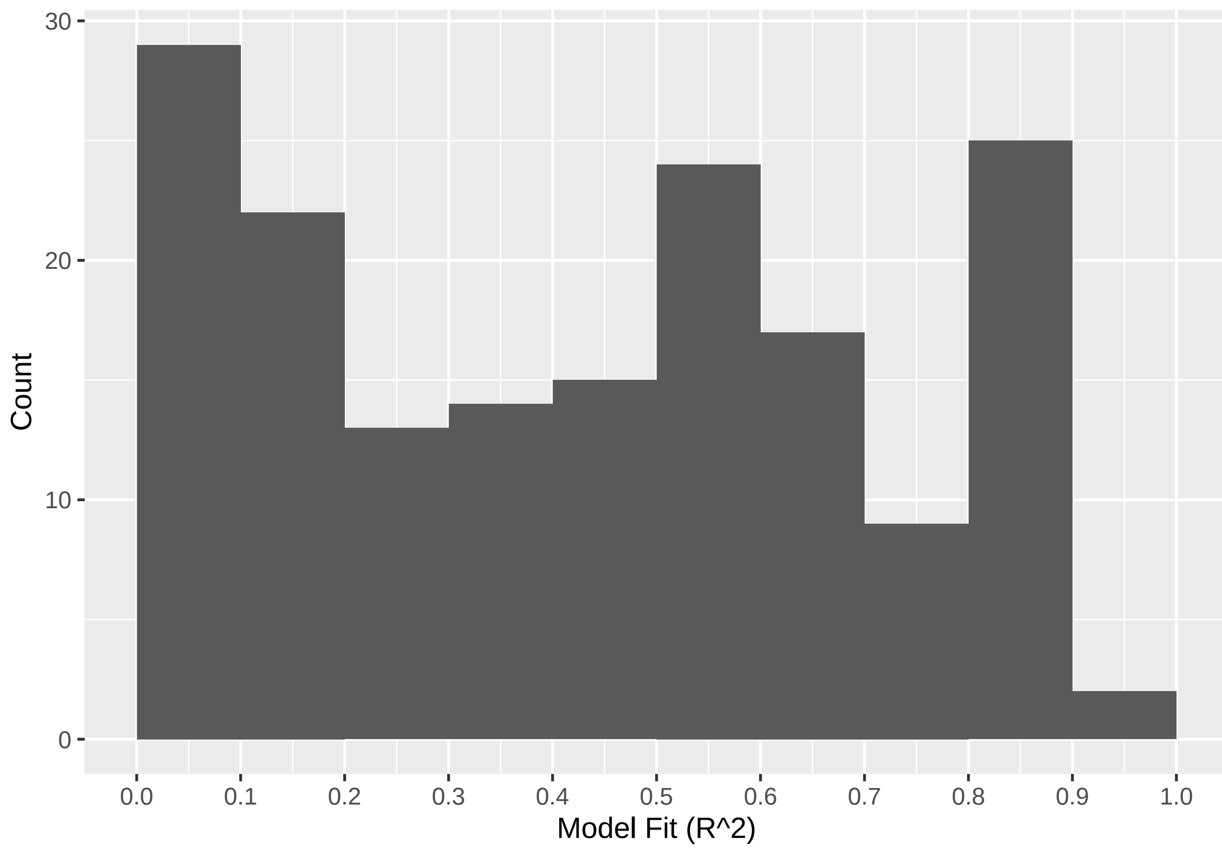



Here we see that the drop in *Percent Feed* is from a drop in Feed production, and not an increase in Food, which has seen steady increases.

Now we will discuss how good all of the models are.

```
model_data %>%
  select(rsq) %>%
  ggplot( aes(x=rsq)) +
  geom_histogram(bins=11, boundary=T) +
  scale_x_continuous(breaks=seq(0,1.,0.1))+
  xlab("Model Fit (R^2)") +
  ylab("Count")
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

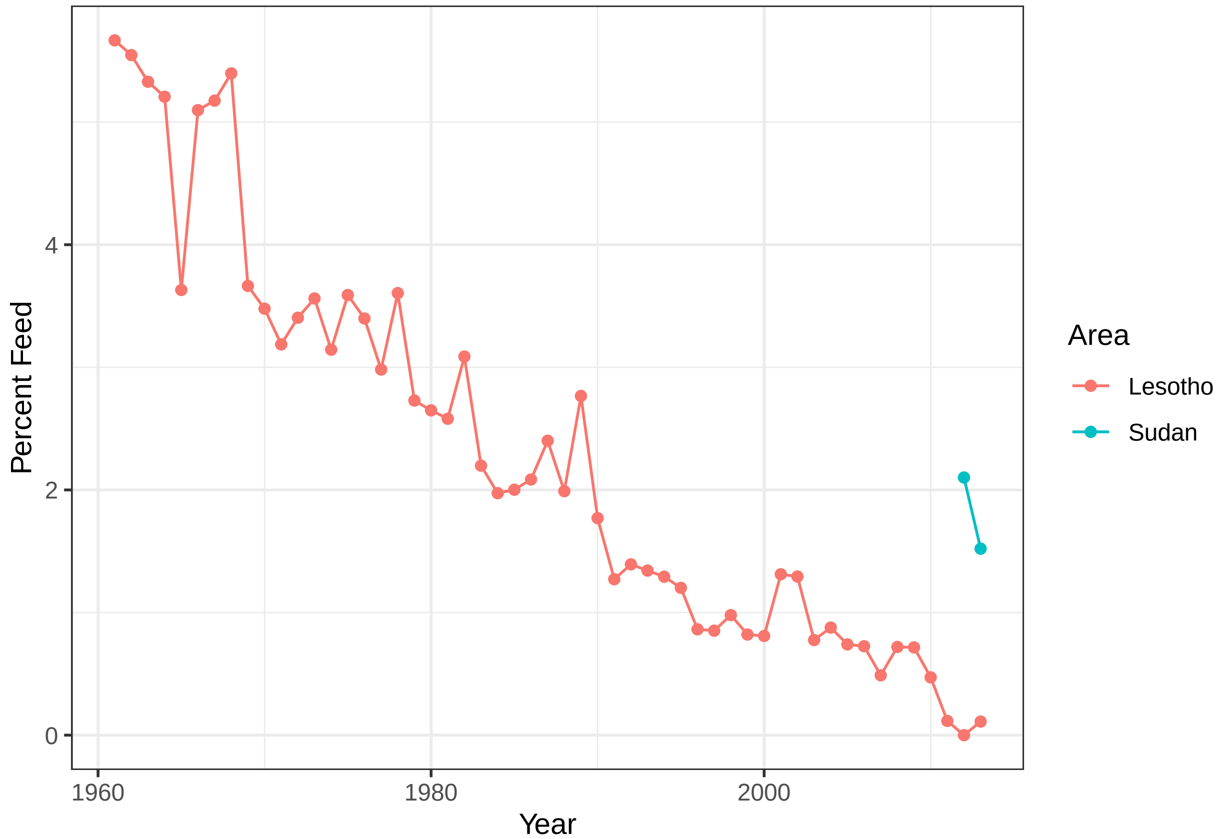



The histogram above shows that model fit as evaluated by $R^2$ is generally evenly, i.e. uniformly, distributed with only 2 countries showing a near 'perfect' fit. Of course, looking at a single linear model ignores year by year changes that could be the result of many different factors.

#### The best fit are not always informative

After looking at countries showing the largest changes we can also look at the models with the best fit.

```
model_data %>%
  filter(rsq>0.9) %>%
  arrange(desc(rsq)) %>%
  unnest(cols=c(data)) %>%
  ggplot(aes(x=Year, y=`Percent Feed`, color=Area)) +
  geom_point(aes(color=Area)) +
  geom_line() +
  theme_bw()
```



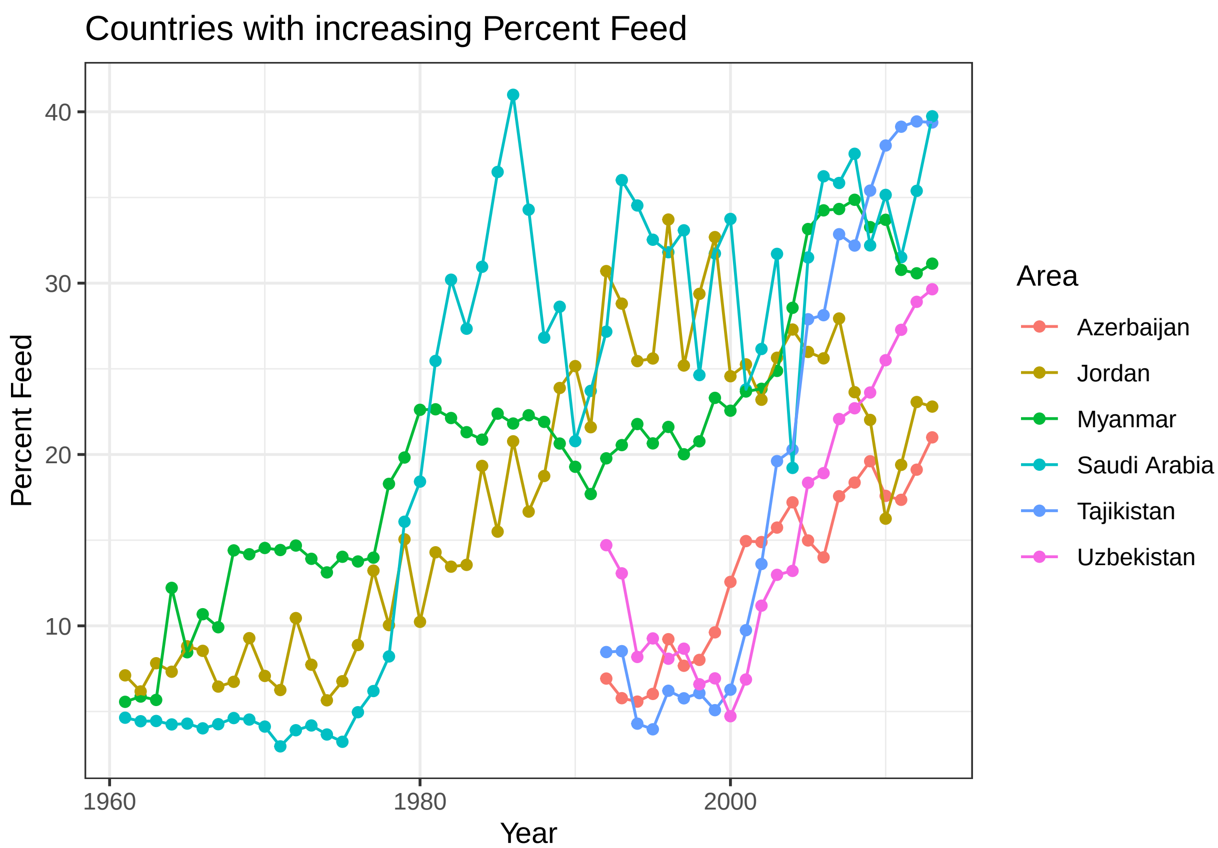

The countries with the best fit models are not particularly ground breaking. There are only 2 years of data for Sudan which makes for a limited model. Lesotho is more interesting as it has Percent Feed values that are highly correlated with Year. However, he range of values between 0 and 5% show that Feed has never been a large production area. Comparing this to the previous plots with *Percent Feed* ranging from 0 to 40, the range for Lesotho is nearly an order of magnitude less.

### Focus on Good Models

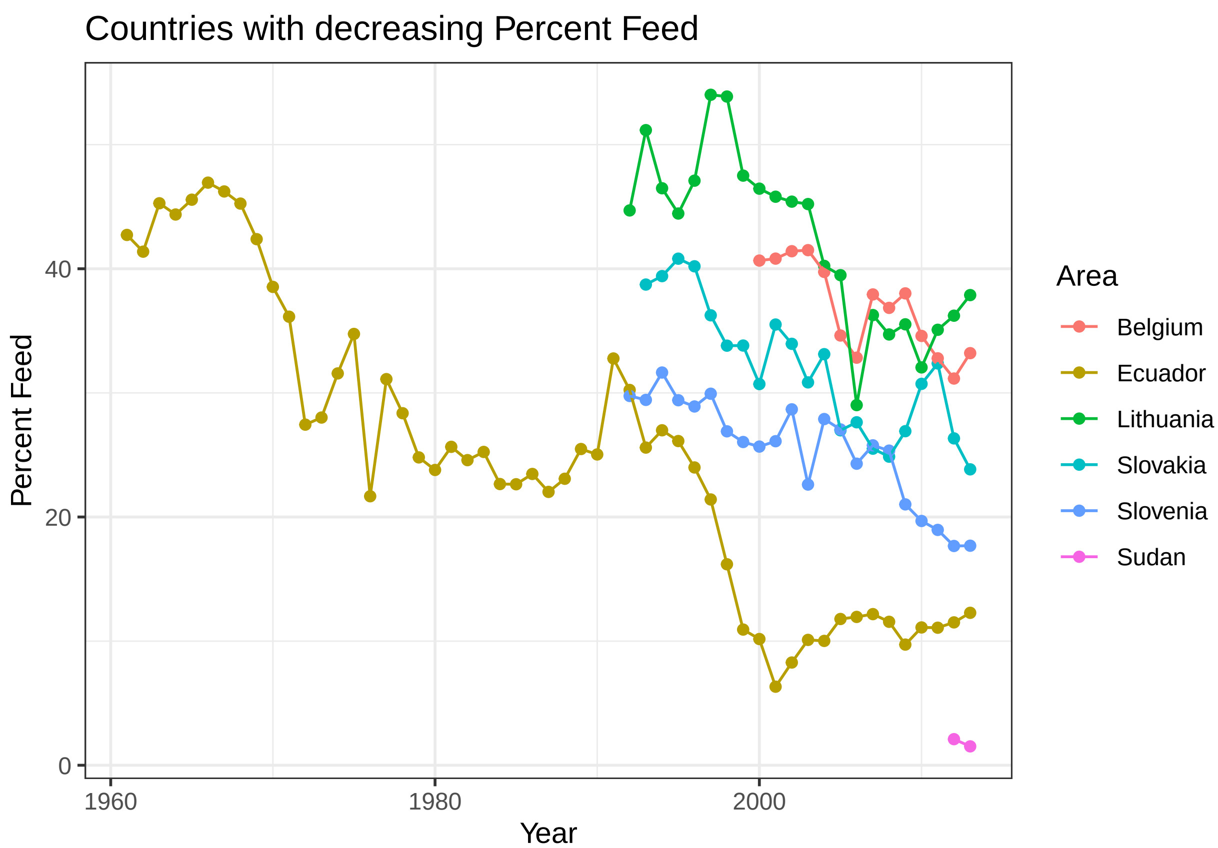
For now let us focus on models with a reasonably good fit.

```
good_model_data =
  model_data %>%
  filter(abs(rsq) > 0.6)
```

```
good_model_data %>%
  arrange(desc(Slope_pctFeed)) %>%
  head() %>%
  #filter(Area == 'Tajikistan') %>%
  select(Area, data, rsq, Slope_pctFeed) %>%
  unnest(data) %>%
  ggplot(aes(x=Year, y=`Percent Feed`, color =Area)) +
  geom_point(aes(color=Area)) +
  geom_line() +
  ggtitle("Countries with increasing Percent Feed")+
  theme_bw()
```




```
good_model_data %>%
  arrange(-desc(Slope_pctFeed)) %>%
  head() %>%
  #filter(Area == 'Tajikistan') %>%
  select(Area, data, rsq, Slope_pctFeed) %>%
  unnest(data) %>%
  ggplot(aes(x=Year, y=`Percent Feed`, color =Area)) +
  geom_point(aes(color=Area)) +
  geom_line() +
  ggtitle("Countries with decreasing Percent Feed")+
  theme_bw()
```




From these figures we can surmise that the *Percent Feed* variable captures how tied the two categories of production are.

We should also model Total Production over time.

## Analysis of Top Producers

Which countries are the biggest producers of which crops? Here we consider a country to be a top producer of an *Item* if it has been among the top 3 producers for any year.

```
top_producers =
  data_long %>%
  group_by(Item, `Item Code`, Year)  %>%
  slice_max(value, n=3, with_ties = FALSE) %>%
  ungroup() %>%
  group_by(`Item Code`, Item) %>%
  distinct(Area) %>%
  ungroup()


top_producers
```

```
## # A tibble: 764 x 3
##    Area                         `Item Code` Item
##    <chr>                              <dbl> <chr>
```

```
##  1 United States of America        2924 Alcoholic Beverages
##  2 France                          2924 Alcoholic Beverages
##  3 Germany                         2924 Alcoholic Beverages
##  4 China, mainland                 2924 Alcoholic Beverages
##  5 Russian Federation              2924 Alcoholic Beverages
##  6 Brazil                          2924 Alcoholic Beverages
##  7 United States of America        2946 Animal fats
##  8 Germany                         2946 Animal fats
##  9 United Kingdom                  2946 Animal fats
## 10 Poland                          2946 Animal fats
## # ... with 754 more rows
```

Now we can look at how many *Items* a country is a producer of.

```
top_producers %>%
  select(Area) %>%
  count(Area) %>%
  arrange(desc(n))
```

```
## # A tibble: 79 x 2
##    Area                         n
##    <chr>                    <int>
##  1 China, mainland             94
##  2 United States of America    73
##  3 India                       65
##  4 Germany                     48
##  5 Japan                       38
##  6 Brazil                      32
##  7 Russian Federation          29
##  8 France                      25
##  9 Indonesia                   23
## 10 United Kingdom              23
## # ... with 69 more rows
```

This makes it easy to find what countries are top producers of only 1 *Item*.

```
top_producers %>%
  select(Area) %>%
  count(Area) %>%
  arrange(desc(n)) %>%
  filter(n == 1) %>%
  left_join(top_producers) %>%
  select(-n)
```

```
## Joining, by = "Area"
```

```
## # A tibble: 17 x 3
##    Area                                     `Item Code` Item
##    <chr>                                          <dbl> <chr>
##  1 Angola                                          2614 Citrus, Other
##  2 Belarus                                         2537 Sugar beet
##  3 Botswana                                        2562 Palm kernels
```

```
##   4 Burkina Faso                                  2657 Beverages, Fermented
##   5 Central African Republic                      2562 Palm kernels
##   6 China, Hong Kong SAR                          2769 Aquatic Animals, Others
##   7 Congo                                         2562 Palm kernels
##   8 Democratic People's Republic of Korea         2764 Marine Fish, Other
##   9 El Salvador                                   2782 Fish, Liver Oil
##  10 Guinea                                        2614 Citrus, Other
##  11 Ireland                                       2782 Fish, Liver Oil
##  12 Lithuania                                     2570 Oilcrops, Other
##  13 Madagascar                                    2642 Cloves
##  14 Nepal                                         2642 Cloves
##  15 Rwanda                                        2615 Bananas
##  16 Timor-Leste                                   2562 Palm kernels
##  17 Yemen                                         2562 Palm kernels
```

We can also look at what *Items* a single country is a top producer of.

```
top_producers %>%
  select(Area) %>%
  count(Area) %>%
  arrange(desc(n)) %>%
  filter(Area == "Canada") %>%
  left_join(top_producers) %>%
  select(-n)
```

```
## Joining, by = "Area"
```

```
## # A tibble: 11 x 3
##    Area   `Item Code` Item
##    <chr>        <dbl> <chr>
##  1 Canada        2513 Barley and products
##  2 Canada        2520 Cereals, Other
##  3 Canada        2743 Cream
##  4 Canada        2781 Fish, Body Oil
##  5 Canada        2782 Fish, Liver Oil
##  6 Canada        2613 Grapefruit and products
##  7 Canada        2680 Infant food
##  8 Canada        2768 Meat, Aquatic Mammals
##  9 Canada        2928 Miscellaneous
## 10 Canada        2516 Oats
## 11 Canada        2558 Rape and Mustardseed
```

Another question is what percent of world production is accounted for by the top producers.

```
# How much do the top producers make?
data_top_producers =
  data_long %>%
  group_by(`Item Code`) %>%
  inner_join(top_producers) %>%
  summarise(top = sum(value, na.rm=TRUE))
```

```
## Joining, by = c("Area", "Item Code", "Item")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# How much of each item is made?
all_production =
  data_long %>%
  group_by(`Item Code`) %>%
  summarise(`Total Units` = sum(value, na.rm=TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#
top_production =
  data_top_producers %>%
  left_join(all_production) %>%
  left_join(items) %>%
  mutate(pct_top = top/`Total Units`*100) %>%
  arrange(pct_top)
```

```
## Joining, by = "Item Code"
```

```
## Joining, by = "Item Code"
```

```
#top_production %>%
#  select(-top, -`Total Units`)
```

The value of *pct_top* represents how much of the world production is produced by countries that have been among the top 3 producers over time. This is a measure of how concentrated world wide production is.

The following table shows the top produced *Items* and the percent production accounted for by the top producers.

```
top_production %>%
  slice_max(order_by = `Total Units`, n=6) %>%
  select(-top)
```

```
## # A tibble: 6 x 4
##   `Item Code` `Total Units` Item                        pct_top
##         <dbl>         <dbl> <chr>                         <dbl>
## 1        2905      64884281 Cereals - Excluding Beer       43.7
## 2        2848      45014120 Milk - Excluding Butter        27.9
## 3        2918      24179916 Vegetables                     55.8
## 4        2907      22711529 Starchy Roots                  51.5
## 5        2514      19960640 Maize and products             57.0
## 6        2511      19194671 Wheat and products             41.8
```

No item is completely dominated by the top producers, however, maize and products comes the closest with 57% of production accounted for by top producers. Top global production takes effort from many people.

Now we look at which items are produced by few countries.

```r
top_production %>%
  slice_max(order_by = pct_top, n=10) %>%
  select(-top)
```

```
## # A tibble: 10 x 4
##    `Item Code` `Total Units` Item                    pct_top
##          <dbl>         <dbl> <chr>                     <dbl>
##  1        2775        253722 Aquatic Plants            100
##  2        2562           396 Palm kernels               99.7
##  3        2961        277596 Aquatic Products, Other    98.0
##  4        2537        246095 Sugar beet                 96.6
##  5        2559        288112 Cottonseed                 95.0
##  6        2782          1211 Fish, Liver Oil            94.6
##  7        2769         23870 Aquatic Animals, Others    93.0
##  8        2533       6079204 Sweet potatoes             91.3
##  9        2642           910 Cloves                     90.8
## 10        2541        532950 Sugar non-centrifugal      88.8
```

These are all quite small compared to the top producers. Sweet potatoes, at 6 million (1000 ton) units, is still only 1/3 that of the smallest top produced item in the previous table.

Now a look at the Items that are produced the least by the top producers.

```r
top_production %>%
  slice_min(order_by = pct_top, n=10) %>%
  select(-top)
```

```
## # A tibble: 10 x 4
##    `Item Code` `Total Units` Item                      pct_top
##          <dbl>         <dbl> <chr>                       <dbl>
##  1        2848      45014120 Milk - Excluding Butter      27.9
##  2        2736        523927 Offals, Edible               35.7
##  3        2945        523927 Offals                       35.7
##  4        2919      14420114 Fruits - Excluding Wine      40.1
##  5        2909       5861502 Sugar & Sweeteners           41.1
##  6        2914       2267192 Vegetable Oils               41.6
##  7        2542       4708762 Sugar (Raw Equivalent)       41.7
##  8        2511      19194671 Wheat and products           41.8
##  9        2745         49164 Honey                        42.2
## 10        2905      64884281 Cereals - Excluding Beer     43.7
```

This table shows which items are produced the least by the top producers, loosely meaning items that are most uniformly produced by all countries.

Some patterns that I see are that Milk and Offals are lowest on this list. This could be because they are perishable and not easily exported. There are probably food systems patterns that can be seen from this table, especially in conjunction with import/export data.

# End

This document shows how to get started with the World Food and Feed data set. There are many more directions to follow with a little guidance from some domain knowledge. I hope I have shown some novel ideas to some readers and helped raise new directions for analysis to others.