R Notebook: Getting Dirty with Data

# Goal today: Use R Markdown to investigate a data question!

These data are originally from Kaggle (<https://www.kaggle.com/dorbicycle/world-foodfeed-production>). Laurel Brehm has done some reformatting & cleaning, & has uploaded the results to RLadies GitHub.

The data were collected from Food and Agriculture Organization of the United Nations. We have production in 1000 tonnes of *food* (=for people) and *feed* (=for animals), per country over years. There is also data about the country location (latitude and longitude).

## Data Summary

What do these data look like?

## Warning: package 'dplyr' was built under R version 3.4.4

##   
## Attaching package: 'dplyr'

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

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

## Warning: package 'ggplot2' was built under R version 3.4.4

## Warning: package 'knitr' was built under R version 3.4.4

## # A tibble: 6 x 13  
## Area.Abbreviation Area.Code Area Item.Code Item Element.Code Element  
## <fct> <int> <fct> <int> <fct> <int> <fct>   
## 1 AFG 2 Afgh~ 2511 Wheat ~ 5142 Food   
## 2 AFG 2 Afgh~ 2805 Rice (~ 5142 Food   
## 3 AFG 2 Afgh~ 2513 Barley~ 5521 Feed   
## 4 AFG 2 Afgh~ 2513 Barley~ 5142 Food   
## 5 AFG 2 Afgh~ 2514 Maize ~ 5521 Feed   
## 6 AFG 2 Afgh~ 2514 Maize ~ 5142 Food   
## # ... with 6 more variables: Unit <fct>, latitude <dbl>, longitude <dbl>,  
## # Year <int>, KTonnes <int>, Crop <fct>

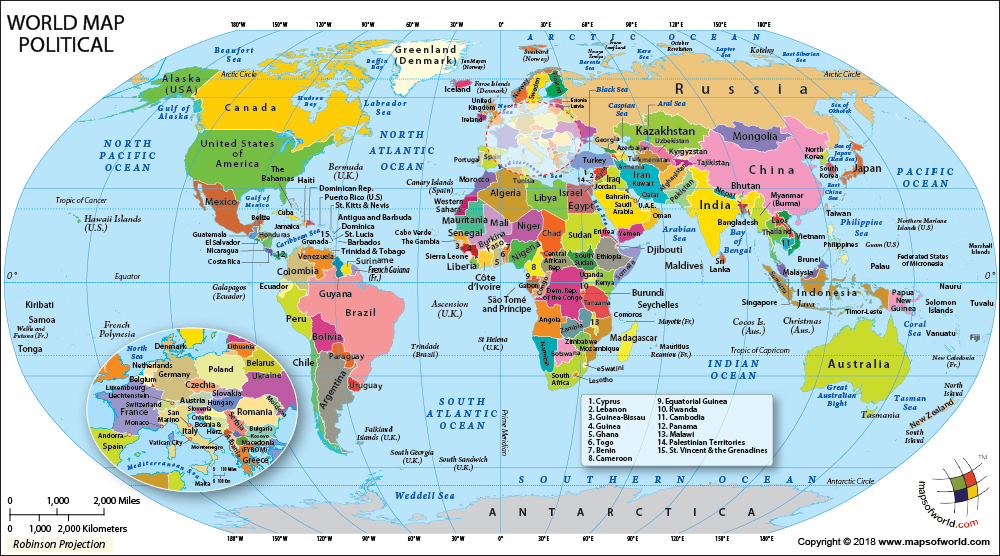
What are the countries? What are the crops & food-stuffs?

## [1] "Afghanistan"   
## [2] "Albania"   
## [3] "Algeria"   
## [4] "Angola"   
## [5] "Antigua and Barbuda"   
## [6] "Argentina"   
## [7] "Armenia"   
## [8] "Australia"   
## [9] "Austria"   
## [10] "Azerbaijan"   
## [11] "Bahamas"   
## [12] "Bangladesh"   
## [13] "Barbados"   
## [14] "Belarus"   
## [15] "Belgium"   
## [16] "Belize"   
## [17] "Benin"   
## [18] "Bermuda"   
## [19] "Bolivia (Plurinational State of)"   
## [20] "Bosnia and Herzegovina"   
## [21] "Botswana"   
## [22] "Brazil"   
## [23] "Brunei Darussalam"   
## [24] "Bulgaria"   
## [25] "Burkina Faso"   
## [26] "Cabo Verde"   
## [27] "Cambodia"   
## [28] "Cameroon"   
## [29] "Canada"   
## [30] "Central African Republic"   
## [31] "Chad"   
## [32] "Chile"   
## [33] "China, Hong Kong SAR"   
## [34] "China, Macao SAR"   
## [35] "China, mainland"   
## [36] "China, Taiwan Province of"   
## [37] "Colombia"   
## [38] "Congo"   
## [39] "Costa Rica"   
## [40] "Cote d'Ivoire"   
## [41] "Croatia"   
## [42] "Cuba"   
## [43] "Cyprus"   
## [44] "Czechia"   
## [45] "Democratic People's Republic of Korea"   
## [46] "Denmark"   
## [47] "Djibouti"   
## [48] "Dominica"   
## [49] "Dominican Republic"   
## [50] "Ecuador"   
## [51] "Egypt"   
## [52] "El Salvador"   
## [53] "Estonia"   
## [54] "Ethiopia"   
## [55] "Fiji"   
## [56] "Finland"   
## [57] "France"   
## [58] "French Polynesia"   
## [59] "Gabon"   
## [60] "Gambia"   
## [61] "Georgia"   
## [62] "Germany"   
## [63] "Ghana"   
## [64] "Greece"   
## [65] "Grenada"   
## [66] "Guatemala"   
## [67] "Guinea"   
## [68] "Guinea-Bissau"   
## [69] "Guyana"   
## [70] "Haiti"   
## [71] "Honduras"   
## [72] "Hungary"   
## [73] "Iceland"   
## [74] "India"   
## [75] "Indonesia"   
## [76] "Iran (Islamic Republic of)"   
## [77] "Iraq"   
## [78] "Ireland"   
## [79] "Israel"   
## [80] "Italy"   
## [81] "Jamaica"   
## [82] "Japan"   
## [83] "Jordan"   
## [84] "Kazakhstan"   
## [85] "Kenya"   
## [86] "Kiribati"   
## [87] "Kuwait"   
## [88] "Kyrgyzstan"   
## [89] "Lao People's Democratic Republic"   
## [90] "Latvia"   
## [91] "Lebanon"   
## [92] "Lesotho"   
## [93] "Liberia"   
## [94] "Lithuania"   
## [95] "Luxembourg"   
## [96] "Madagascar"   
## [97] "Malawi"   
## [98] "Malaysia"   
## [99] "Maldives"   
## [100] "Mali"   
## [101] "Malta"   
## [102] "Mauritania"   
## [103] "Mauritius"   
## [104] "Mexico"   
## [105] "Mongolia"   
## [106] "Montenegro"   
## [107] "Morocco"   
## [108] "Mozambique"   
## [109] "Myanmar"   
## [110] "Namibia"   
## [111] "Nepal"   
## [112] "Netherlands"   
## [113] "New Caledonia"   
## [114] "New Zealand"   
## [115] "Nicaragua"   
## [116] "Niger"   
## [117] "Nigeria"   
## [118] "Norway"   
## [119] "Oman"   
## [120] "Pakistan"   
## [121] "Panama"   
## [122] "Paraguay"   
## [123] "Peru"   
## [124] "Philippines"   
## [125] "Poland"   
## [126] "Portugal"   
## [127] "Republic of Korea"   
## [128] "Republic of Moldova"   
## [129] "Romania"   
## [130] "Russian Federation"   
## [131] "Rwanda"   
## [132] "Saint Kitts and Nevis"   
## [133] "Saint Lucia"   
## [134] "Saint Vincent and the Grenadines"   
## [135] "Samoa"   
## [136] "Sao Tome and Principe"   
## [137] "Saudi Arabia"   
## [138] "Senegal"   
## [139] "Serbia"   
## [140] "Sierra Leone"   
## [141] "Slovakia"   
## [142] "Slovenia"   
## [143] "Solomon Islands"   
## [144] "South Africa"   
## [145] "Spain"   
## [146] "Sri Lanka"   
## [147] "Sudan"   
## [148] "Suriname"   
## [149] "Swaziland"   
## [150] "Sweden"   
## [151] "Switzerland"   
## [152] "Tajikistan"   
## [153] "Thailand"   
## [154] "The former Yugoslav Republic of Macedonia"  
## [155] "Timor-Leste"   
## [156] "Togo"   
## [157] "Trinidad and Tobago"   
## [158] "Tunisia"   
## [159] "Turkey"   
## [160] "Turkmenistan"   
## [161] "Uganda"   
## [162] "Ukraine"   
## [163] "United Arab Emirates"   
## [164] "United Kingdom"   
## [165] "United Republic of Tanzania"   
## [166] "United States of America"   
## [167] "Uruguay"   
## [168] "Uzbekistan"   
## [169] "Vanuatu"   
## [170] "Venezuela (Bolivarian Republic of)"   
## [171] "Viet Nam"   
## [172] "Yemen"   
## [173] "Zambia"   
## [174] "Zimbabwe"

## Warning: package 'bindrcpp' was built under R version 3.4.4

|  |  |
| --- | --- |
| Crop | Item |
| Alcohol | Alcoholic Beverages |
| Alcohol | Beer |
| Alcohol | Beverages, Alcoholic |
| Alcohol | Beverages, Fermented |
| Alcohol | Wine |
| Beans | Beans |
| Beans | Soyabeans |
| Cocoa Beans | Cocoa Beans and products |
| Coffee | Coffee and products |
| Dairy | Butter, Ghee |
| Dairy | Cream |
| Dairy | Milk - Excluding Butter |
| Eggs | Eggs |
| Fruit | Apples and products |
| Fruit | Bananas |
| Fruit | Citrus, Other |
| Fruit | Dates |
| Fruit | Fruits - Excluding Wine |
| Fruit | Fruits, Other |
| Fruit | Grapefruit and products |
| Fruit | Grapes and products (excl wine) |
| Fruit | Lemons, Limes and products |
| Fruit | Oranges, Mandarines |
| Fruit | Pineapples and products |
| Grains | Barley and products |
| Grains | Cereals - Excluding Beer |
| Grains | Cereals, Other |
| Grains | Maize and products |
| Grains | Millet and products |
| Grains | Oats |
| Grains | Rice (Milled Equivalent) |
| Grains | Rye and products |
| Grains | Sorghum and products |
| Grains | Wheat and products |
| Honey | Honey |
| Infant food | Infant food |
| Meat | Animal fats |
| Meat | Aquatic Animals, Others |
| Meat | Bovine Meat |
| Meat | Fats, Animals, Raw |
| Meat | Meat |
| Meat | Meat, Aquatic Mammals |
| Meat | Meat, Other |
| Meat | Mutton & Goat Meat |
| Meat | Offals |
| Meat | Offals, Edible |
| Meat | Pigmeat |
| Meat | Poultry Meat |
| Miscellaneous | Miscellaneous |
| Nuts | Coconuts - Incl Copra |
| Nuts | Groundnuts (Shelled Eq) |
| Nuts | Nuts and products |
| Nuts | Treenuts |
| Oil | Coconut Oil |
| Oil | Cottonseed Oil |
| Oil | Groundnut Oil |
| Oil | Maize Germ Oil |
| Oil | Oilcrops |
| Oil | Oilcrops Oil, Other |
| Oil | Oilcrops, Other |
| Oil | Olive Oil |
| Oil | Palm Oil |
| Oil | Palmkernel Oil |
| Oil | Rape and Mustard Oil |
| Oil | Ricebran Oil |
| Oil | Sesameseed Oil |
| Oil | Soyabean Oil |
| Oil | Sunflowerseed Oil |
| Oil | Vegetable Oils |
| Pulses | Pulses |
| Pulses | Pulses, Other and products |
| Roots | Roots, Other |
| Roots | Starchy Roots |
| Seafood | Aquatic Products, Other |
| Seafood | Cephalopods |
| Seafood | Crustaceans |
| Seafood | Demersal Fish |
| Seafood | Fish, Body Oil |
| Seafood | Fish, Liver Oil |
| Seafood | Fish, Seafood |
| Seafood | Freshwater Fish |
| Seafood | Marine Fish, Other |
| Seafood | Molluscs, Other |
| Seafood | Pelagic Fish |
| Seeds | Cottonseed |
| Seeds | Rape and Mustardseed |
| Seeds | Sesame seed |
| Seeds | Sunflower seed |
| Spices | Cloves |
| Spices | Pepper |
| Spices | Spices |
| Spices | Spices, Other |
| Starch | Cassava and products |
| Starch | Plantains |
| Starch | Potatoes and products |
| Starch | Sweet potatoes |
| Starch | Yams |
| Stimulants | Stimulants |
| Sugar | Sugar & Sweeteners |
| Sugar | Sugar (Raw Equivalent) |
| Sugar | Sugar beet |
| Sugar | Sugar cane |
| Sugar | Sugar Crops |
| Sugar | Sugar non-centrifugal |
| Sugar | Sweeteners, Other |
| Tea (including mate) | Tea (including mate) |
| Veg | Aquatic Plants |
| Veg | Olives (including preserved) |
| Veg | Onions |
| Veg | Palm kernels |
| Veg | Peas |
| Veg | Pimento |
| Veg | Tomatoes and products |
| Veg | Vegetables |
| Veg | Vegetables, Other |

## Embed a plot or picture!



## Take some stuff from R and put it in text!

# in this chunk, I asked the output to be shown with echo=TRUE   
# Get the maximum value of a product, with function max().   
# Need to set na.rm=T because there are NA values in the column.  
m <- max(ds$KTonnes,na.rm=T)  
  
# Then look up which area has that value for the column KTonnes  
a <- ds[ds$KTonnes %in% m ,]$Area  
  
# And which item it is  
i <- ds[ds$KTonnes %in% m ,]$Item  
  
# And which year!  
y <- ds[ds$KTonnes %in% m ,]$Year  
  
# Then, we can use those values in text by surrounding with `` and prefacing with r

The single largest export in a year was from China, mainland in 2013 and it was 4.8929910^{8} tonnes of Vegetables.

# The goal for today is to investigate food / feed production over time by countries.

Summarise some data, plot some data, and find some patterns!

## Questions to ask:

* Pick a food type and look at where it’s grown over time
* Pick a country and find what’s been grown in it over time
* Pick a year and see what’s been grown where
* Look at differences in food (for people) vs feed (for animals) for a country or set of countries

## For advanced R users: Go forth on your own from here!

## For beginners… here are some code chunks to use to get you started.

# Data Sumamrising

# in this chunk, I asked everything to be shown with echo=TRUE   
# Here are some useful tabulation commands. These use the tidyverse.  
# This takes data and summarises for output.   
# %>% is a 'pipe' and passes commands between lines.  
  
# you can filter columns. here, we want columns where the item is sugar cane, used for food  
t2 <- ds %>%  
 filter(Item=="Sugar cane") %>%  
 filter(Element=="Food")  
  
t2

## # A tibble: 2,650 x 13  
## Area.Abbreviation Area.Code Area Item.Code Item Element.Code Element  
## <fct> <int> <fct> <int> <fct> <int> <fct>   
## 1 BGD 16 Bangl~ 2536 Suga~ 5142 Food   
## 2 BWA 20 Botsw~ 2536 Suga~ 5142 Food   
## 3 BRA 21 Brazil 2536 Suga~ 5142 Food   
## 4 BRN 26 Brune~ 2536 Suga~ 5142 Food   
## 5 KHM 115 Cambo~ 2536 Suga~ 5142 Food   
## 6 CMR 32 Camer~ 2536 Suga~ 5142 Food   
## 7 CHN 96 China~ 2536 Suga~ 5142 Food   
## 8 CHN 128 China~ 2536 Suga~ 5142 Food   
## 9 CHN 214 China~ 2536 Suga~ 5142 Food   
## 10 COD 46 Congo 2536 Suga~ 5142 Food   
## # ... with 2,640 more rows, and 6 more variables: Unit <fct>,  
## # latitude <dbl>, longitude <dbl>, Year <int>, KTonnes <int>, Crop <fct>

# you can use a | operator ('or') to look at multiple selections at once.  
# you can use the select operator to only show some columns  
# and the filter operator to remove columns even after summarising  
# here we ask for only the crops of more than 10\*1000 tonnes  
t3<- ds %>%  
 filter(Area=="Netherlands" | Area=="Belgium") %>%   
 group\_by(Area,Crop,Element,Year) %>%  
 summarise(TotalKTonnes = sum(as.numeric(KTonnes)) ) %>%  
 filter(TotalKTonnes > 10)  
  
t3

## # A tibble: 1,839 x 5  
## # Groups: Area, Crop, Element [62]  
## Area Crop Element Year TotalKTonnes  
## <fct> <fct> <fct> <int> <dbl>  
## 1 Belgium Alcohol Food 2000 2618  
## 2 Belgium Alcohol Food 2001 2590  
## 3 Belgium Alcohol Food 2002 2512  
## 4 Belgium Alcohol Food 2003 2483  
## 5 Belgium Alcohol Food 2004 2521  
## 6 Belgium Alcohol Food 2005 2722  
## 7 Belgium Alcohol Food 2006 2604  
## 8 Belgium Alcohol Food 2007 2478  
## 9 Belgium Alcohol Food 2008 2416  
## 10 Belgium Alcohol Food 2009 2455  
## # ... with 1,829 more rows

# We can make lists and use these to filter.   
# Here's list of some things eaten by people and animals  
mixed <- c("Oats","Soyabeans", "Sugar cane")  
  
# this will generate an error, but it actually works.   
#an example of tidyverse 'lazy evaluation'  
t4<- ds %>%  
 filter(Area=="Brazil" | Area=="Colombia") %>%  
 filter(Item==mixed)

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of  
## shorter object length

## Warning in `==.default`(Item, mixed): longer object length is not a  
## multiple of shorter object length

t4

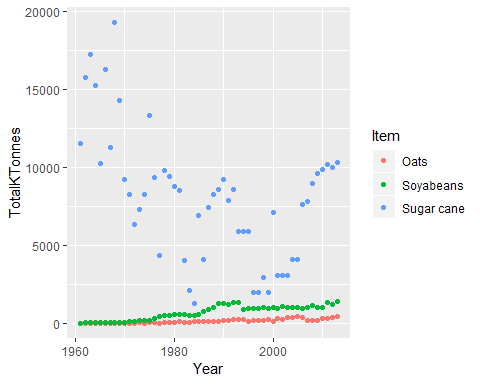
## # A tibble: 143 x 13  
## Area.Abbreviation Area.Code Area Item.Code Item Element.Code Element  
## <fct> <int> <fct> <int> <fct> <int> <fct>   
## 1 BRA 21 Braz~ 2516 Oats 5142 Food   
## 2 BRA 21 Braz~ 2536 Sugar~ 5521 Feed   
## 3 BRA 21 Braz~ 2555 Soyab~ 5521 Feed   
## 4 COL 44 Colo~ 2555 Soyab~ 5142 Food   
## 5 BRA 21 Braz~ 2536 Sugar~ 5142 Food   
## 6 BRA 21 Braz~ 2555 Soyab~ 5142 Food   
## 7 COL 44 Colo~ 2516 Oats 5142 Food   
## 8 COL 44 Colo~ 2536 Sugar~ 5521 Feed   
## 9 BRA 21 Braz~ 2516 Oats 5142 Food   
## 10 BRA 21 Braz~ 2536 Sugar~ 5521 Feed   
## # ... with 133 more rows, and 6 more variables: Unit <fct>,  
## # latitude <dbl>, longitude <dbl>, Year <int>, KTonnes <int>, Crop <fct>

# finally, we can put these together to summarise.  
# Here, we are taking mass of items grown in Brazil or Colombia  
# and taking together the sum of all tonnes by year across rows  
# (= total production in these countries, whether for food or feed)  
# (Note that I used a less lazy evaluaton method here for the filter)  
  
t5<- ds %>%  
 filter(Item %in% mixed == T) %>%  
 filter(Area=="Brazil" | Area=="Colombia") %>%  
 group\_by(Year,Item) %>%   
 summarise(TotalKTonnes = sum(as.numeric(KTonnes)) )   
  
t5

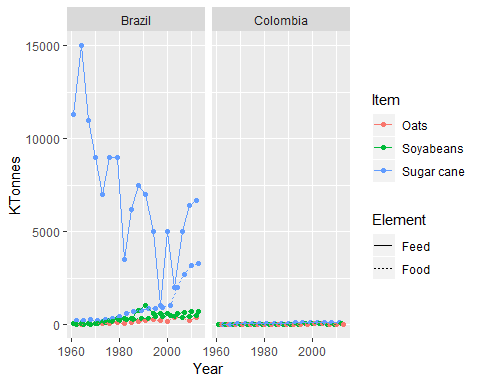
## # A tibble: 159 x 3  
## # Groups: Year [?]  
## Year Item TotalKTonnes  
## <int> <fct> <dbl>  
## 1 1961 Oats 37  
## 2 1961 Soyabeans 49  
## 3 1961 Sugar cane 11537  
## 4 1962 Oats 36  
## 5 1962 Soyabeans 68  
## 6 1962 Sugar cane 15738  
## 7 1963 Oats 33  
## 8 1963 Soyabeans 84  
## 9 1963 Sugar cane 17246  
## 10 1964 Oats 35  
## # ... with 149 more rows

%%newpage is a command for Latex…and % is the comment character for Latex. ## Plotting

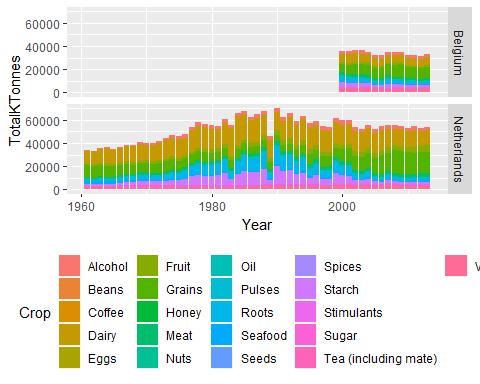
# We can easily plot these with lines, points, and bars!  
# we are using ggplot here.   
#The first line sets up what the data are & what values are mapped to things  
# in the plot ('aes' values).  
# variables that are useful include: x, y values, color, lty (=line type)  
# we can then add points or lines or bars with geom\_point, geom\_line, geom\_bar  
  
ggplot(data=t5,aes(x=Year,y=TotalKTonnes,color=Item))+  
 geom\_point()



# add connector lines  
# and make separate panels with facet\_grid   
#(explore what the ~ does by trying .~Area vs Area~.)  
ggplot(data=t4,aes(x=Year,y=KTonnes,color=Item,lty=Element))+  
 geom\_point()+  
 geom\_line()+  
 facet\_grid(.~Area)



# Here's an example of a bar plot  
# And an example of user-specified figure height / width  
ggplot(data=t3,aes(x=Year,y=TotalKTonnes,fill=Crop))+  
 geom\_bar(stat='identity')+  
 facet\_grid(Area~.)+  
 theme(legend.position="bottom")



## Tables of output

#Most of what we did earlier was tibbles:   
#this was a concious choice because the data structures were too big.  
#Where 'tables' really shine is with summary data.  
  
# here, we make a new variable with 'mutate'-- the last tibble 'verb'  
t6 <- ds %>%  
 filter(Item %in% mixed == T) %>%  
 filter(Area=="Brazil" | Area=="Colombia") %>%  
 mutate(Decade=round(((Year-5)/10),0)\*10) %>%   
 # year minus 5, divided by 10, rounded to 0 decimal places.  
 # (1969 - 5)/10=196.4, rounds to 1960  
 group\_by(Decade,Item) %>%   
 summarise(TotalKTonnes = sum(as.numeric(KTonnes)) ) %>%  
 arrange(Item,Decade)  
  
t6 <- as.data.frame(t6)  
  
kable(t6)

|  |  |  |
| --- | --- | --- |
| Decade | Item | TotalKTonnes |
| 1960 | Oats | 404 |
| 1970 | Oats | 628 |
| 1980 | Oats | 1522 |
| 1990 | Oats | 2113 |
| 2000 | Oats | 3665 |
| 2010 | Oats | 1186 |
| 1960 | Soyabeans | 811 |
| 1970 | Soyabeans | 2933 |
| 1980 | Soyabeans | 8913 |
| 1990 | Soyabeans | 9933 |
| 2000 | Soyabeans | 11735 |
| 2010 | Soyabeans | 4026 |
| 1960 | Sugar cane | 140452 |
| 1970 | Sugar cane | 76504 |
| 1980 | Sugar cane | 69605 |
| 1990 | Sugar cane | 43270 |
| 2000 | Sugar cane | 68551 |
| 2010 | Sugar cane | 30526 |

#We can also restructure the table  
# this uses matrix notation to paste together elements by column  
# I've also put the units in tonnes again  
kable(cbind(t6[1:6,1],t6[1:6,3]\*1000,t6[7:12,3]\*1000,t6[13:18,3]\*1000),  
 col.names=c("Year","Tonnes/Oats","Tonnes/Soya","Tonnes/Sugar Cane"))

|  |  |  |  |
| --- | --- | --- | --- |
| Year | Tonnes/Oats | Tonnes/Soya | Tonnes/Sugar Cane |
| 1960 | 404000 | 811000 | 140452000 |
| 1970 | 628000 | 2933000 | 76504000 |
| 1980 | 1522000 | 8913000 | 69605000 |
| 1990 | 2113000 | 9933000 | 43270000 |
| 2000 | 3665000 | 11735000 | 68551000 |
| 2010 | 1186000 | 4026000 | 30526000 |

## Cool! Adapt these code chunks to ask some questions and make a nice summary of it with Markdown.