U.S. Food Imports

2022-11-19

```
install.packages("tidyr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/vkoyya/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'tidyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'tidyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\vkoyya\AppData\Local\R\win-library\4.2\00LOCK\tidyr\libs\x64\tidyr.dll
## to C:\Users\vkoyya\AppData\Local\R\win-library\4.2\tidyr\libs\x64\tidyr.dll:
## Permission denied

## Warning: restored 'tidyr'

##
## The downloaded binary packages are in
##  C:\Users\vkoyya\AppData\Local\Temp\RtmpY7LNOf\downloaded_packages
```

```
install.packages("factoextra", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/vkoyya/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'factoextra' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\vkoyya\AppData\Local\Temp\RtmpY7LNOf\downloaded_packages
```

```
install.packages("dplyr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/vkoyya/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\vkoyya\AppData\Local\R\win-library\4.2\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\vkoyya\AppData\Local\R\win-library\4.2\dplyr\libs\x64\dplyr.dll:
## Permission denied
```

```
## Warning: restored 'dplyr'
```

```
##
## The downloaded binary packages are in
##  C:\Users\vkoyya\AppData\Local\Temp\RtmpY7LNOf\downloaded_packages
```

```
install.packages("knitr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/vkoyya/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
##
##   There is a binary version available but the source version is later:
##       binary source needs_compilation
## knitr   1.40   1.41             FALSE
```

```
## installing the source package 'knitr'
```

```
install.packages("ggplot2", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/vkoyya/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\vkoyya\AppData\Local\Temp\RtmpY7LNOf\downloaded_packages
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.2.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
##
## 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
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.2.2
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
library(knitr)
library(ggplot2)
```

Project Introduction

American consumers need a wide selection of high-quality, convenient foods. The American food basket represents an increasing share of tropical crops, spices, and imported gourmet goods as Americans have become wealthier and more ethnically diverse. Seasonal and climatic factors drive U.S. imports of popular types of fruits and vegetables and tropical products, such as cocoa and coffee. Additionally, a significant portion of U.S. imports can be attributed to intra-industry trade, in which American agricultural-processing companies outsource some of their operations overseas and import goods that have undergone varying degrees of processing from their foreign-market subsidiaries.

Data Source

https://catalog.data.gov/dataset/u-s-food-imports

Project Objective

By analyzing the aforementioned data, we will be able to determine the factual data, such as the amount of growth and income that occurred each year for each food category.

Project Outcomes

1)To find statistics of the data. 2)Finding linear regression. 3)polynomial regression discovery. 4)Plotting the data. 5)looking for clustering in data.

```
#Reading the data into a dataframe
coredf =  read.csv("C:/Users/Public/Food_data.csv")

#The dataframe has been modified
coredf = coredf %>% rename("id" = "X","Item_Type" = "X.1","Amount_Type" = "X.2","2021" = "X2021", "2020

#Dataframe structure before transformation
str(coredf)
```

```
## 'data.frame':    18 obs. of  27 variables:
##  $ Food_categories: chr  "Total foods 1/" "Foods" "Foods" "Foods" ...
##  $ id             : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Item_Type      : chr  "U.S. imports" "Live meat animals" "Meats" "Fish and shellfish 2/" ...
##  $ Amount_Type    : chr  "Million $" "Million $" "Million $" "Million $" ...
##  $ 2021           : chr  "166,946.5" "2,299.9" "13,196.0" "24,198.5" ...
##  $ 2020           : chr  "146,406.1" "2,158.0" "10,389.4" "21,285.7" ...
##  $ 2019           : chr  "141,720.7" "2,253.4" "9,668.7" "21,797.7" ...
##  $ 2018           : chr  "139,587.6" "2,029.4" "9,251.2" "22,272.6" ...
##  $ 2017           : chr  "131,143.3" "2,016.3" "8,875.2" "21,324.2" ...
##  $ 2016           : chr  "123,153.3" "2,103.3" "8,587.2" "19,257.8" ...
##  $ 2015           : chr  "121,117.9" "2,774.9" "9,991.7" "18,513.6" ...
##  $ 2014           : chr  "119,666.2" "3,009.9" "8,940.3" "20,053.5" ...
##  $ 2013           : chr  "109,711.4" "2,193.0" "6,529.9" "17,784.3" ...
##  $ 2012           : chr  "106,809.5" "2,196.3" "6,245.2" "16,467.5" ...
##  $ 2011           : chr  "102,857.9" "1,893.3" "5,755.3" "16,459.4" ...
##  $ 2010           : chr  "86,983.8" "2,014.0" "5,087.9" "14,516.7" ...
##  $ 2009           : chr  "77,261.6" "1,661.5" "4,612.1" "12,933.9" ...
##  $ 2008           : chr  "84,543.8" "2,277.1" "5,059.8" "13,912.0" ...
##  $ 2007           : chr  "76,843.8" "2,596.4" "5,367.4" "13,434.6" ...
##  $ 2006           : chr  "70,861.5" "2,173.9" "5,243.7" "13,112.3" ...
##  $ 2005           : chr  "64,391.6" "1,673.3" "5,752.0" "11,840.2" ...
##  $ 2004           : chr  "58,428.9" "1,139.4" "5,718.5" "11,106.3" ...
##  $ 2003           : chr  "52,473.6" "1,278.0" "4,426.9" "10,859.9" ...
##  $ 2002           : chr  "46,680.8" "1,725.2" "4,283.5" "9,963.3" ...
##  $ 2001           : chr  "43,930.1" "1,772.0" "4,256.2" "9,663.3" ...
##  $ 2000           : chr  "43,339.3" "1,420.3" "3,827.7" "9,879.8" ...
##  $ 1999           : chr  "41,400.9" "1,190.6" "3,260.5" "8,859.8" ...
```

```
#Transforming the dataframe with structure

coredf1 = gather(coredf, key = "Year", value = "Amount", "2021", "2020", "2019","2018","2017","2016","2
```

```
#Renaming the columns with the the actual values
coredf1$Food_categories[coredf1$Food_categories == 'Total foods 1/'] = 'Total foods'
coredf1$Food_categories[coredf1$Food_categories == 'Subtotal foods 4/'] = 'Subtotal foods'

#Casting the values with desired types
coredf1$Year = as.numeric(coredf1$Year)
coredf1$Amount = gsub(",", "", coredf1$Amount)
coredf1$Amount = as.double(coredf1$Amount)

#Dataframe structure after transformation
str(coredf1)
```

```
## 'data.frame':    414 obs. of  6 variables:
##  $ Food_categories: chr  "Total foods" "Foods" "Foods" "Foods" ...
##  $ id             : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Item_Type      : chr  "U.S. imports" "Live meat animals" "Meats" "Fish and shellfish 2/" ...
##  $ Amount_Type    : chr  "Million $" "Million $" "Million $" "Million $" ...
##  $ Year           : num  2021 2021 2021 2021 2021 ...
##  $ Amount         : num  166947 2300 13196 24199 2462 ...
```

```
#Redefining and removal of the NA values from the dataframe
master1df = select(coredf1,id,Year,Item_Type,Amount,Food_categories)
master1df = na.omit(master1df)

#Details of Food categories and its item types:
distinct(master1df, Food_categories,Item_Type)
```

```
##                  Item_Type Food_categories
## 1             U.S. imports     Total foods
## 2        Live meat animals           Foods
## 3                    Meats           Foods
## 4    Fish and shellfish 2/           Foods
## 5                    Dairy           Foods
## 6                Vegetables           Foods
## 7                    Fruits           Foods
## 8                      Nuts           Foods
## 9  Coffee, tea, and spices           Foods
## 10                  Grains           Foods
## 11          Vegetable oils           Foods
## 12         Sugar and candy           Foods
## 13     Cocoa and chocolate           Foods
## 14   Other edible products           Foods
## 15            Beverages 3/           Foods
## 16                 Animals  Subtotal foods
## 17                  Plants  Subtotal foods
## 18            Beverages 3/  Subtotal foods
```

```
#Partitioning the master dataframe based on Food categories
Total_foods = subset(master1df, Food_categories=='Total foods')
#Total Food dataframe statistics
kable(summary(select(Total_foods,id,Year,Amount)),row.names = FALSE,caption = "Total foods")
```

Table 1: Total foods

id	Year	Amount
Min. :1	Min. :1999	Min. : 41401
1st Qu.:1	1st Qu.:2004	1st Qu.: 61410
Median :1	Median :2010	Median : 86984
Mean :1	Mean :2010	Mean : 93750
3rd Qu.:1	3rd Qu.:2016	3rd Qu.:122136
Max. :1	Max. :2021	Max. :166947

```
Foods = subset(master1df, Food_categories== "Foods")
#Foods dataframe statistics
kable(summary(select(Foods,id,Year,Amount)),row.names = FALSE,caption = "Foods")
```

Table 2: Foods

id	Year	Amount
Min. : 2.0	Min. :1999	Min. : 717
1st Qu.: 5.0	1st Qu.:2004	1st Qu.: 2460
Median : 8.5	Median :2010	Median : 5084
Mean : 8.5	Mean :2010	Mean : 6696
3rd Qu.:12.0	3rd Qu.:2016	3rd Qu.: 9779
Max. :15.0	Max. :2021	Max. :24199

```
Subtotal_foods = subset(master1df, Food_categories== "Subtotal foods")
#Subtotal_foods dataframe statistics
kable(summary(select(Subtotal_foods,id,Year,Amount)),row.names = FALSE,caption = "Subtotal foods")
```

Table 3: Subtotal foods

id	Year	Amount
Min. :16	Min. :1999	Min. : 4433
1st Qu.:16	1st Qu.:2004	1st Qu.: 13559
Median :17	Median :2010	Median : 22725
Mean :17	Mean :2010	Mean : 31250
3rd Qu.:18	3rd Qu.:2016	3rd Qu.: 35973
Max. :18	Max. :2021	Max. :106623

```
#Finding linear regression of each Food categories

lmTotal_foods = lm(Amount~Year, data = Total_foods)
#linear regression of Total foods category
summary(lmTotal_foods)
```

```
##
## Call:
## lm(formula = Amount ~ Year, data = Total_foods)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -10934  -2208   -669   1926  12091
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.107e+07  3.085e+05  -35.89   <2e-16 ***
## Year         5.555e+03  1.535e+02   36.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4883 on 21 degrees of freedom
## Multiple R-squared:  0.9842, Adjusted R-squared:  0.9835
## F-statistic:  1310 on 1 and 21 DF,  p-value: < 2.2e-16
```

```
lmall_foods = lm(Amount~Year, data = Foods)
#linear regression of Food category
summary(lmall_foods)
```

```
##
## Call:
## lm(formula = Amount ~ Year, data = Foods)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8761.2 -2799.8  -460.7  2440.0 13137.4
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -790842.05   74271.60  -10.65   <2e-16 ***
## Year            396.79      36.95   10.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4398 on 320 degrees of freedom
## Multiple R-squared:  0.2649, Adjusted R-squared:  0.2626
## F-statistic: 115.3 on 1 and 320 DF,  p-value: < 2.2e-16
```

```
lmSubtotal_foods = lm(Amount~Year, data = Subtotal_foods)
#linear regression of Subtotal foods category
summary(lmSubtotal_foods)
```

```
##
## Call:
## lm(formula = Amount ~ Year, data = Subtotal_foods)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34098 -13794  -6449  11973  55004
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3690596.2   810837.5  -4.552  2.3e-05 ***
## Year            1851.7      403.4   4.590  2.0e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22230 on 67 degrees of freedom
## Multiple R-squared:  0.2392, Adjusted R-squared:  0.2279
## F-statistic: 21.07 on 1 and 67 DF,  p-value: 2.002e-05
```

```
#Polynomial regression discovery for each Food categories

#Logic of polynomial regression of Total foods category
lm(Amount ~ poly(Year, 2, raw = TRUE),data = Total_foods)
```

```
##
## Call:
## lm(formula = Amount ~ poly(Year, 2, raw = TRUE), data = Total_foods)
##
## Coefficients:
##                (Intercept)  poly(Year, 2, raw = TRUE)1
##                  2.299e+08                   -2.343e+05
## poly(Year, 2, raw = TRUE)2
##                  5.965e+01
```

```
#Logic of polynomial regression of Food category
lm(Amount ~ poly(Year, 2, raw = TRUE),data = Foods)
```

```
##
## Call:
## lm(formula = Amount ~ poly(Year, 2, raw = TRUE), data = Foods)
##
## Coefficients:
##                (Intercept)  poly(Year, 2, raw = TRUE)1
##                  1.642e+07                   -1.673e+04
## poly(Year, 2, raw = TRUE)2
##                  4.261e+00
```

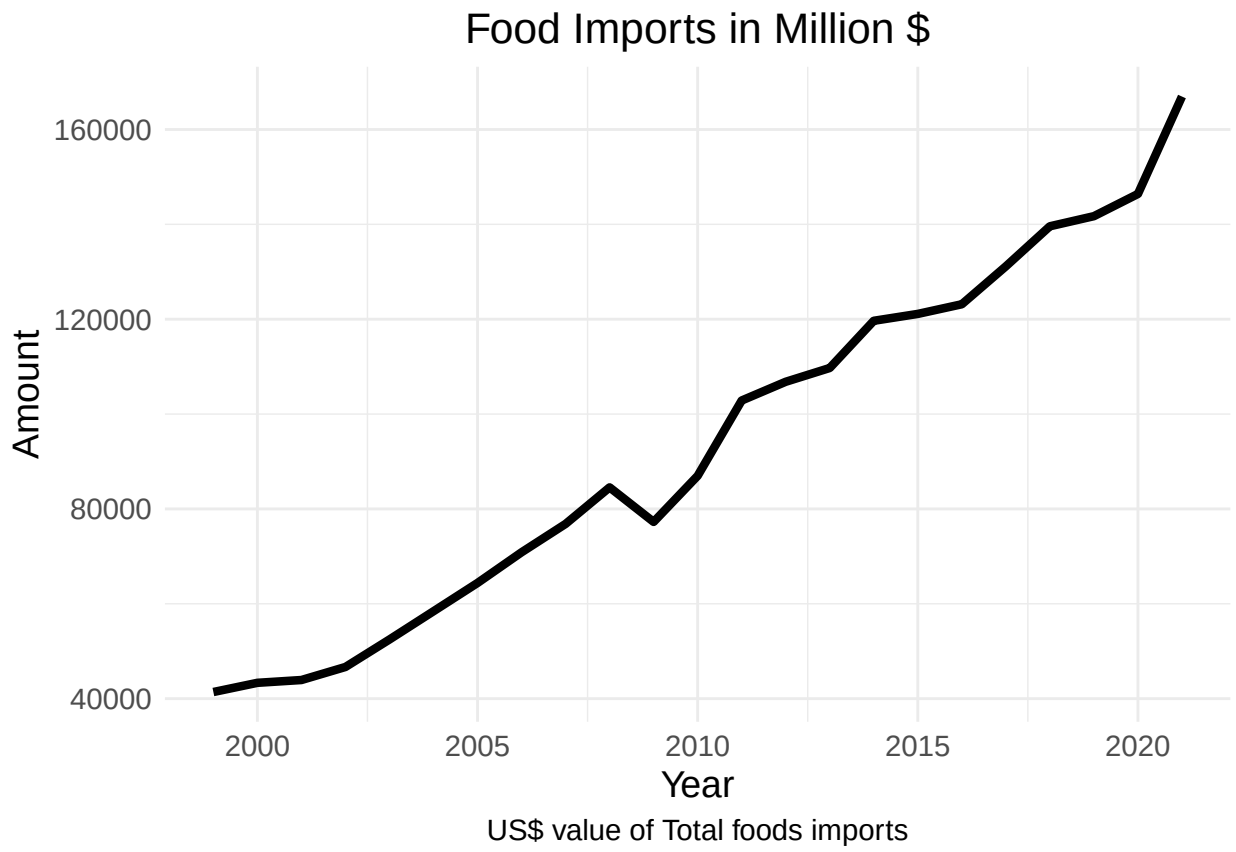
```
#Logic of polynomial regression of Subtotal foods category
lm(Amount ~ poly(Year, 2, raw = TRUE),data = Subtotal_foods)
```

```
##
## Call:
## lm(formula = Amount ~ poly(Year, 2, raw = TRUE), data = Subtotal_foods)
##
## Coefficients:
##                (Intercept)  poly(Year, 2, raw = TRUE)1
##                  7.664e+07                   -7.808e+04
## poly(Year, 2, raw = TRUE)2
##                  1.988e+01
```

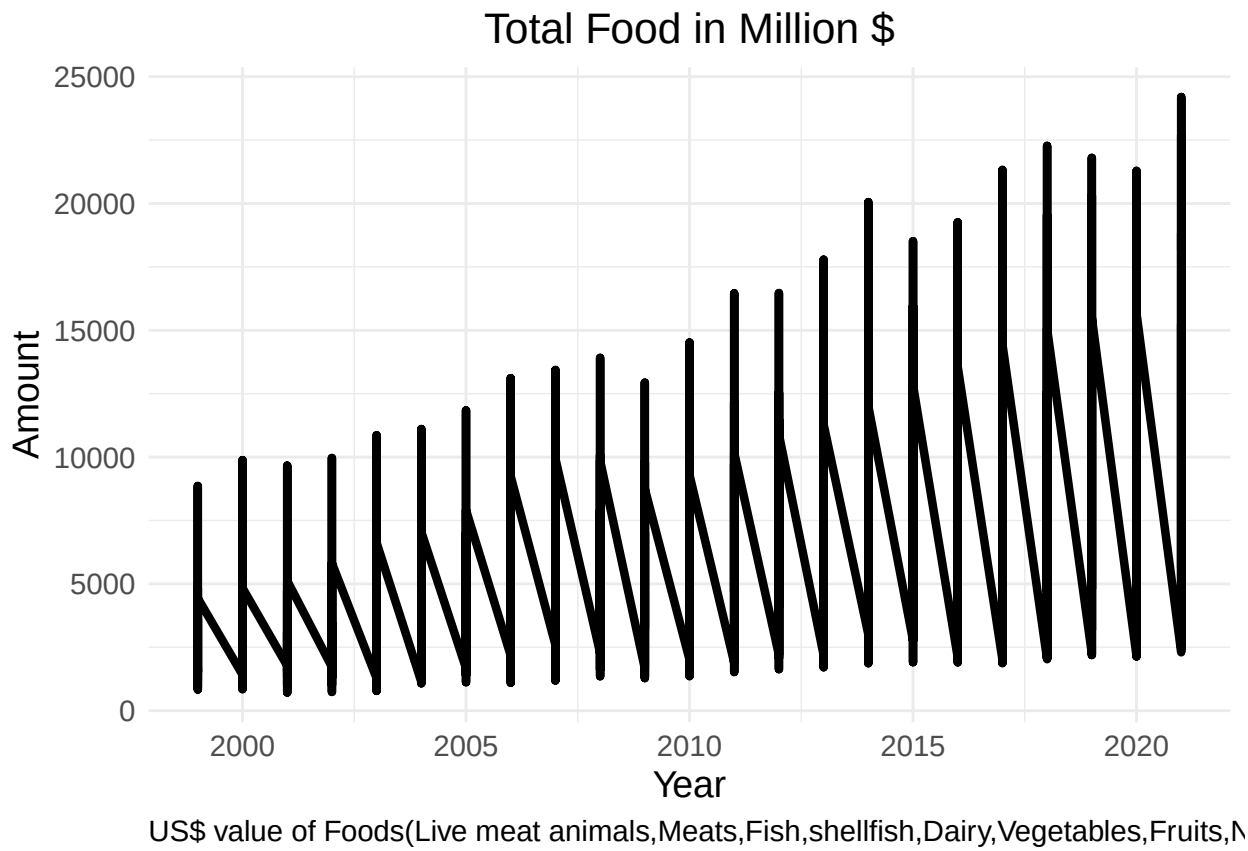
```
#Plotting the data for each Food categories

#Logic for Plotting Year vs Amount of Total foods category
ggplot(Total_foods,aes(x=Year,y=Amount)) +geom_line(size=1.5) +
```

```
    labs(x = "Year", y = "Amount",caption="US$ value of Total foods imports") +
    ggtitle(paste0("Food Imports in Million $"))+
    theme_minimal()+
    theme(legend.position = "right",
          plot.caption = element_text(hjust = 0.5),
          plot.title = element_text(hjust = 0.5,size=16),
          text = element_text(size=14))+
    scale_fill_brewer(palette="Set3")
```

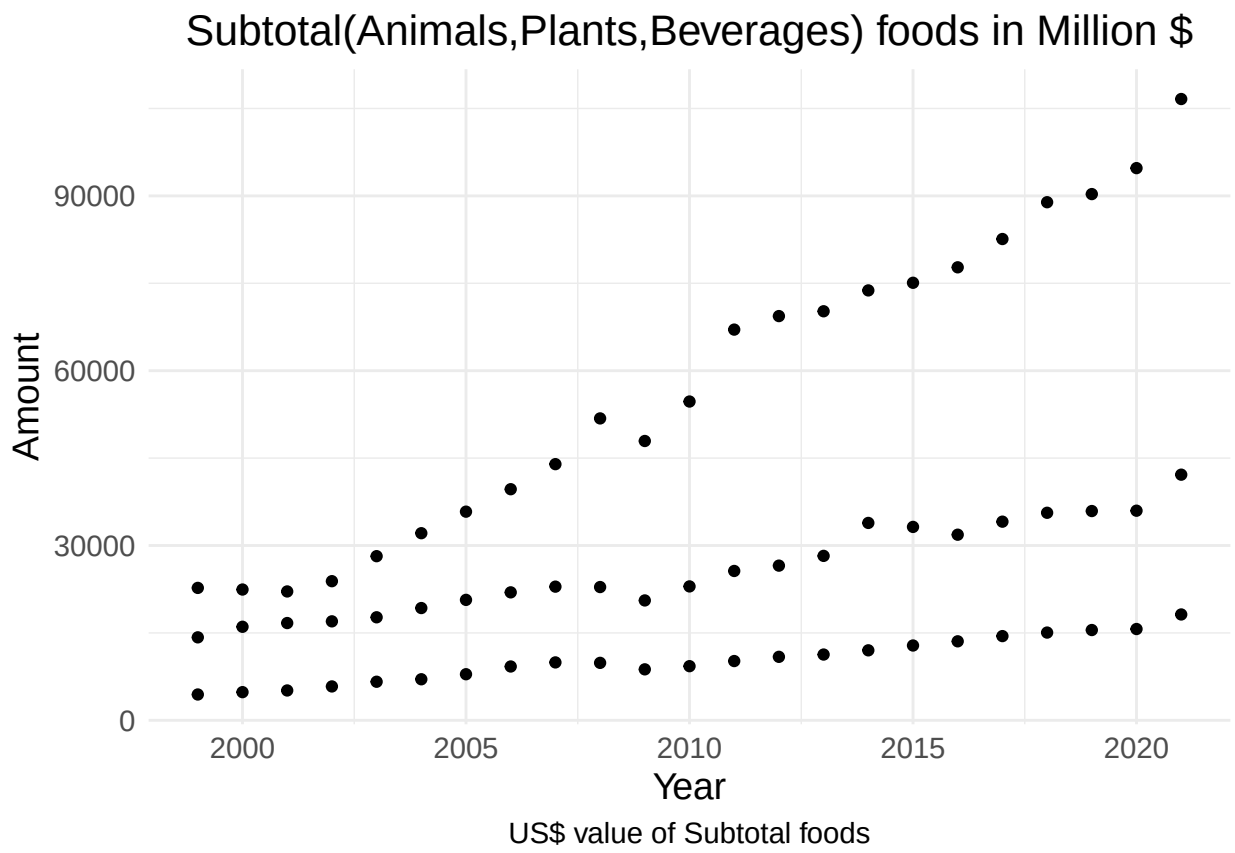
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```



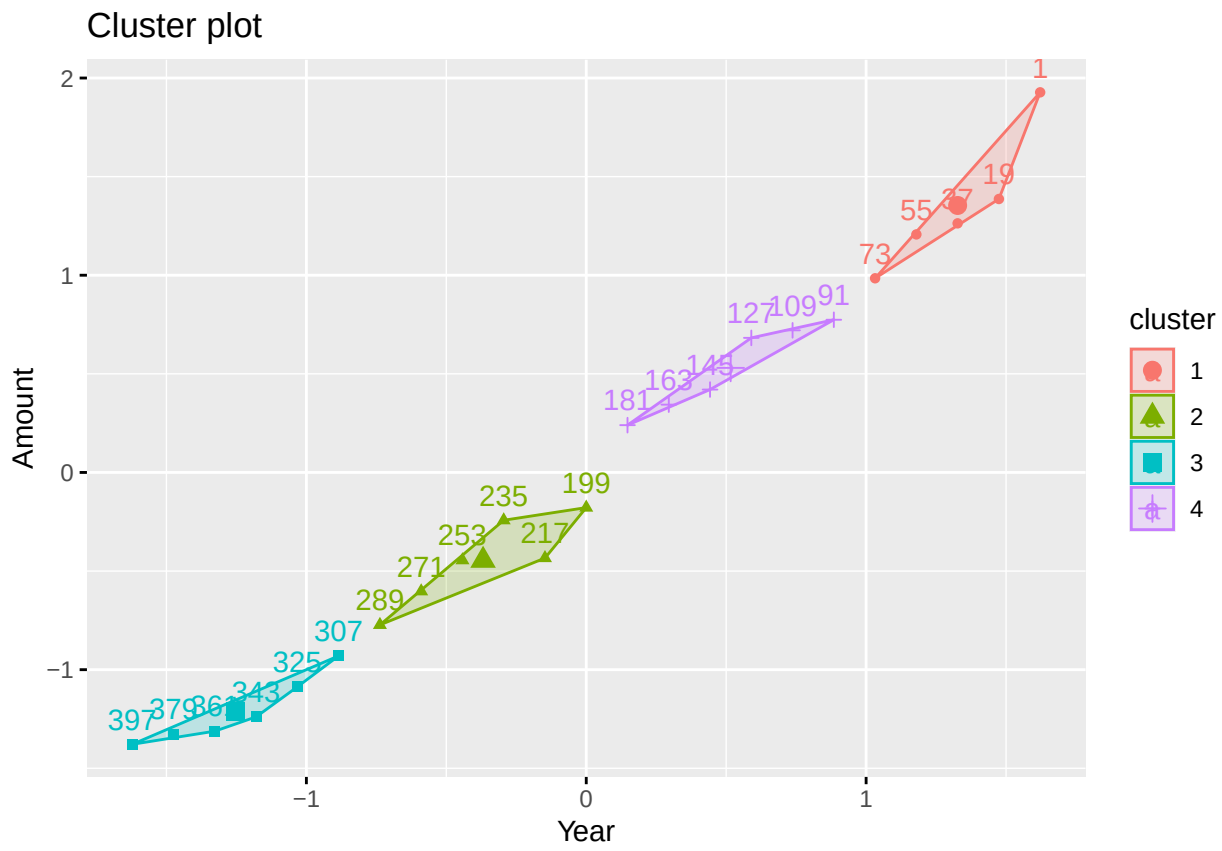

```
#Logic for Plotting Year vs Amount of Foods category
ggplot(Foods,aes(x=Year,y=Amount)) +geom_line(size=1.5) +
    labs(x = "Year", y = "Amount",caption="US$ value of Foods(Live meat animals,Meats,Fish,shellfish,Da
    ggtitle(paste0("Total Food in Million $"))+
    theme_minimal()+
    theme(legend.position = "right",
          plot.caption = element_text(hjust = 0),
          plot.title = element_text(hjust = 0.5,size=16),
          text = element_text(size=14))+
    scale_fill_brewer(palette="Set3")
```



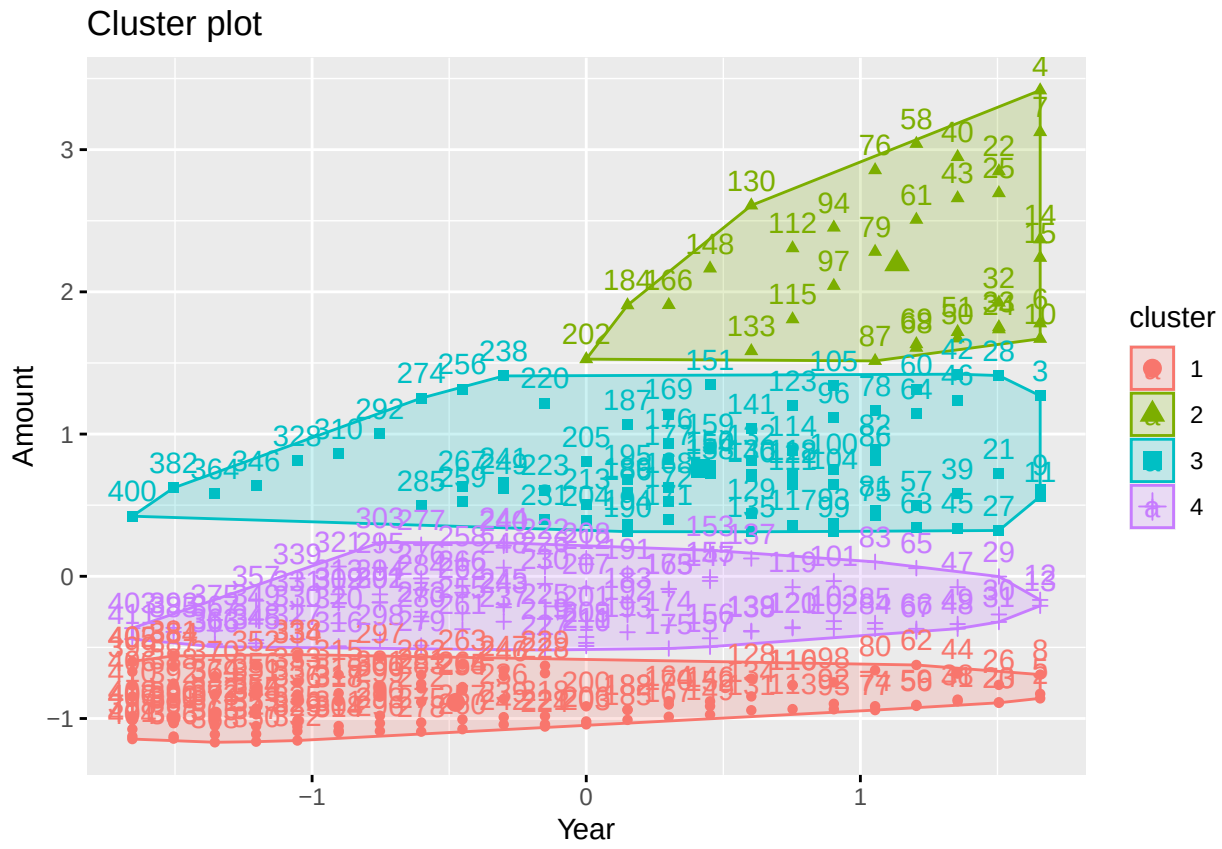

```
#Logic for Plotting Year vs Amount of Subtotal foods category
ggplot(Subtotal_foods,aes(x=Year,y=Amount)) +geom_point(size=1.5) +
    labs(x = "Year", y = "Amount",caption="US$ value of Subtotal foods") +
    ggtitle(paste0("Subtotal(Animals,Plants,Beverages) foods in Million $"))+
    theme_minimal()+
    theme(legend.position = "right",
          plot.caption = element_text(hjust = 0.5),
          plot.title = element_text(hjust = 0.5,size=16),
          text = element_text(size=14))+
    scale_fill_brewer(palette="Set3")
```



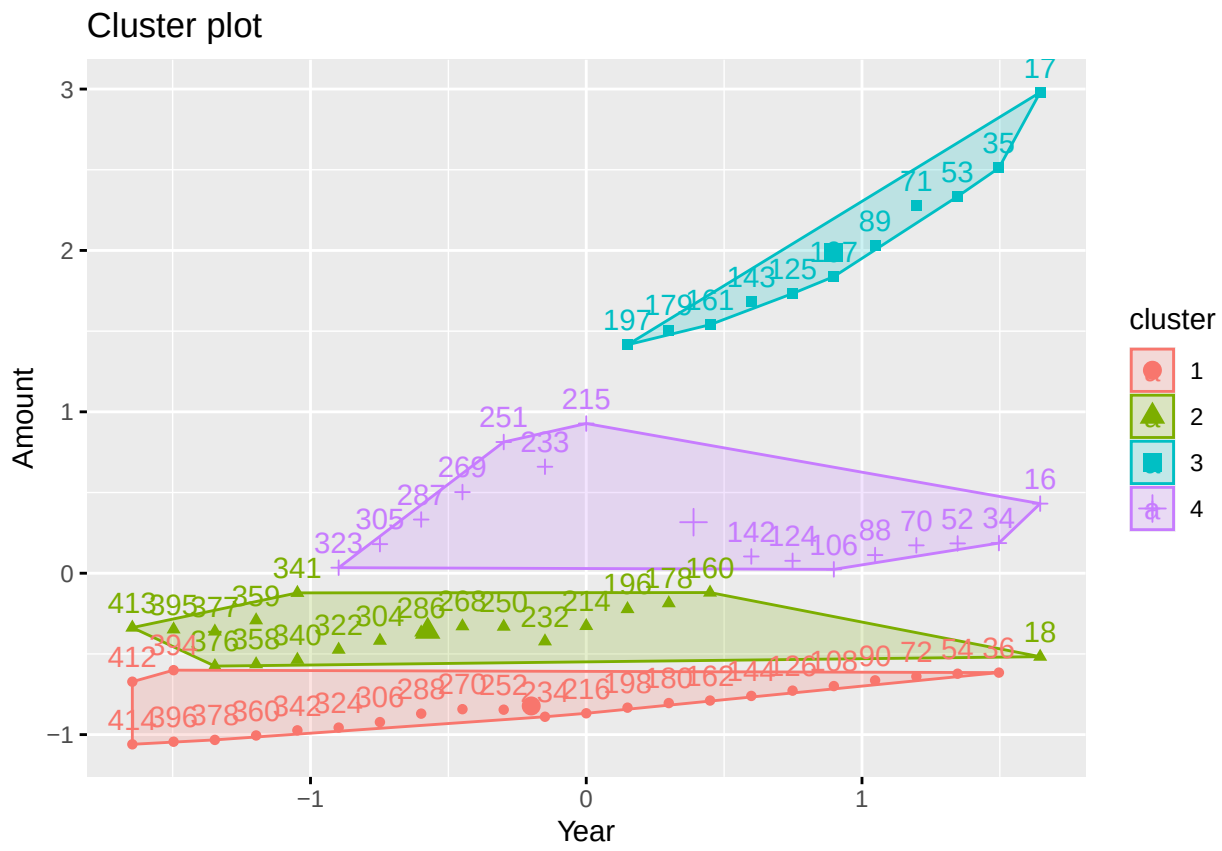

```
#Looking for clustering in data based on for each Food categories

#Total_foods clustering
kdf = select(Total_foods,Year,Amount)
km = kmeans(kdf, centers = 4)
fviz_cluster(km, data = kdf)
```




```
#Foods clustering
kdf1 = select(Foods,Year,Amount)
km1 = kmeans(kdf1, centers = 4)
fviz_cluster(km1, data = kdf1)
```




```
#Subtotal foods clustering
kdf2 = select(Subtotal_foods,Year,Amount)
km2 = kmeans(kdf2, centers = 4)
fviz_cluster(km2, data = kdf2)
```


Cluster plot
Amount
Year
cluster
1
2
3
4