Analyses of Food Prices in Iran and the Effect of International Sanctions on It

rahmani_pour.payam@stud.hs-fresenius.de

HS-Fresenius: Data Science for Business

Contents

Rendered at 08 December, 2022

Word count: 1947

1 Introduction

Several nations, most notably the United States and international organizations, have imposed sanctions on Iran. Prior to Russia overtaking Iran as the nation with the highest sanctions after the latter invaded the adjacent Ukrainian territory in February 2022.(Bloomberg (2022))

Since 1979, when the American embassy in Tehran was taken over, the United States has restricted trade and other activity with Iran under a number of legal pretexts (Wikipedia (2022)). Iran's economy and people have suffered greatly due to sanctions. Since 1979, the United States has been at the forefront of international attempts to use sanctions to change Iran's policies, notably its uranium enrichment program, which Western nations worry is meant to create the capacity to manufacture nuclear weapons. During a meeting in

Lausanne, Switzerland, the P5+1 and Iran reached a provisional agreement on a framework on April 2, 2015. Once it is finalized and implemented, the framework will lift most of the sanctions in exchange for Iran's nuclear program being constrained for at least ten years. The Joint Comprehensive Plan of Action, the final accord, was approved on October 18, 2015. Consequently, on January 16, 2016, UN sanctions were withdrawn. Donald Trump, the president of the United States, said that the country would leave the Iran nuclear agreement on May 8, 2018. The United States reinstituted sanctions in November 2018 and extended them to include Iran's banking sector from 2019 till now.The FATF added Iran to its blacklist on February 21, 2020.You can find more details here.

The sanctions imposed by the United States on Iran have had detrimental humanitarian effects on Iranian society. Iranians' rights to food, health, education, and other human rights have suffered due to the sanctions' broad scope.

Following the US sanctions against Iran, unprecedented inflation has also occurred in Iran, affecting the food market.

1.1 Structure

There are three primary chapters in this data analytics project that follows. The situation and the project's purpose will comply in this first part. The second section is a thorough description of all crucial procedures taken during the whole analytical process. The last section is a report for an international body that summarizes the study and answers the inquiry.

1.2 Analyziz goal

The current research aims to ascertain if the sanctions have affected Iranian family food costs and which category sees the most rise.To achieve this goal as effectively as possible, the relationship between the date of imposing sanctions and the food costs rise would be investigated.

2 Data Preparation & Overview

The datasets that will be used in this project has been made available by World Food Programme Price Database. under a public license, for more information click here and to access the data source click here

Setup and Data Import: When opening R Studio first the libraries `lubridate` ,`hms` ,`ggplot2` and `tidyverse` and its sub-libraries, which contain important functions for processing, analyzing and visualizing data, must be loaded.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(hms)
library(lubridate)
```

```
## Loading required package: timechange
```

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

```
## The following object is masked from 'package:hms':
##
##     hms
```

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

Next, the CSV file will be imported as a data frame from the website.

```
foodprices <- read.csv((url("https://data.humdata.org/dataset/6df76343-1bd9-488a-af3c-1e
```

Dataset Overview: Now by the `str()` function can learn more about the data structure of our table. Each sample in the data sets contains details on the pricing of a specific product. There are 762 items in the table, each with 14 variables.

```
str(foodprices)
```

```
## 'data.frame':    762 obs. of  14 variables:
##  $ date     : chr  "#date" "2012-04-15" "2012-04-15" "2012-04-15" ...
##  $ admin1   : chr  "#adm1+name" "Tehran" "Tehran" "Tehran" ...
```

```
##  $ admin2   : chr  "#adm2+name" "Tehran" "Tehran" "Tehran" ...
##  $ market   : chr  "#loc+market+name" "Tehran Market" "Tehran Market" "Tehran Market"
##  $ latitude : chr  "#geo+lat" "35.72" "35.72" "35.72" ...
##  $ longitude: chr  "#geo+lon" "51.4" "51.4" "51.4" ...
##  $ category : chr  "#item+type" "cereals and tubers" "meat, fish and eggs" "miscellan
##  $ commodity: chr  "#item+name" "Rice" "Eggs" "Sugar (white)" ...
##  $ unit     : chr  "#item+unit" "KG" "1.8 KG" "KG" ...
##  $ priceflag: chr  "#item+price+flag" "actual" "actual" "actual" ...
##  $ pricetype: chr  "#item+price+type" "Retail" "Retail" "Retail" ...
##  $ currency : chr  "#currency" "IRR" "IRR" "IRR" ...
##  $ price    : chr  "#value" "26860.0" "52760.0" "15890.0" ...
##  $ usdprice : chr  "#value+usd" "2.1909" "4.3034" "1.2961" ...
```

Here we can see that there are 14 different columns. The first column shows the date from the second to fifth columns about the store's location in Tehran. The next one is about the category of items and the name of the items. Furthermore, we have the items' units, price flags, and price types. The last three columns are about currency, price in IRR, and price in US dollars.

2.1 Data Cleaning & Processing

At first, we knew that the second row clarified our columns. Now we could remove this row.

```
foodprices <- foodprices[-c(1), ]
```

Checking for Missing Values:

The `is.na()` routines will be used first to identify **NA** entries, which will then be summarized for each variable separately. The second phase will show the number of fields with **NULL** values.

```
# Printing amount of missing values for each of the i = [1:14]features
for (i in 1:14) {
  message(sum(is.na(foodprices[i])),” missing values in column ”, i)
}
```

```
## 0 missing values in column 1

## 0 missing values in column 2

## 0 missing values in column 3

## 0 missing values in column 4
```

```
## 0 missing values in column 5

## 0 missing values in column 6

## 0 missing values in column 7

## 0 missing values in column 8

## 0 missing values in column 9

## 0 missing values in column 10

## 0 missing values in column 11

## 0 missing values in column 12

## 0 missing values in column 13

## 0 missing values in column 14
```

It is shown that there is not any missing value in dataset.

It is time to check the price's column for any zero value and omit them.

```
foodprices$price <- as.integer(foodprices$price)
foodprices[foodprices==0] <- NA
foodprices <- foodprices[complete.cases(foodprices),]
```

Understandably, there were 13 rows with zero values that were omitted.Checking **price** for minimum, average and maximum values.

Exploring Distinct Values: The method *n distinct()* is used to acquire a summary of how many distinct values each feature can store. It gives the total number of unique values in a column.

```
library(dplyr)
```

```
# Printing distinct values for each of the i = [1:12]features
for (i in 1:14) {
  message(n_distinct(foodprices[i])," distinct values in column ", i)
}
```

```
## 111 distinct values in column 1

## 1 distinct values in column 2

## 1 distinct values in column 3

## 1 distinct values in column 4

## 1 distinct values in column 5

## 1 distinct values in column 6

## 8 distinct values in column 7

## 34 distinct values in column 8

## 15 distinct values in column 9

## 1 distinct values in column 10

## 1 distinct values in column 11

## 1 distinct values in column 12

## 443 distinct values in column 13

## 582 distinct values in column 14
```

Variables that can already be validated as cleaned:

- admin1, admin2, market, latitude and longitude attributes are the same for every item
- pricetype and currency attributes are the same for every item
- The Priceflag attribute has two distinct values, and by exploring the table, it is evident that only ten rows have a forecasting flag

Questionable variables:

Whether there are nonsense or zero variables for the prices should be checked. Also, there is a discrepancy between **IRR Prices (444/761)** and **USD Prices (583/761)**. One scenario could be that the price is stable in the home currency, but the exchange rate has changed.

Moreover, it can be seen that there are eight different categories and 37 different items. However, some lower or upper case conditions and white spaces should be checked to determine if there is any duplicate. Furthermore, 16 different scale units must be checked to see if every item has the same unit in the time series.

```
# Declare trim function to delete leading and trailing white spaces
trim <- function (x) gsub("^\\s+|\\s+$", "", x)
# Trimming column $category and $commodity
foodprices$category <- trim(foodprices$category)
foodprices$commodity <- trim(foodprices$commodity)
```

```
## 8 distinct values in column category after trimming
```

```
## 34 distinct values in column comodity after trimming
```

```
# Converting all character of data frame to lowercase
foodprices <- mutate_all(foodprices, .funs=tolower)
```

```
## 8 distinct values in column category after conversion to lowercase
```

```
## 34 distinct values in column comodity after conversion to lowercase
```

After trimming and in the conversion to only lowercase, no values have been excluded from each of the two variables **category(8/761)** and **comodity(37/761)**.

By searching in the dataset, it could find out that the commodity and category until 2021 are stable, but for 2022 we have more items. Because of this, it would be better if we split our investigation into two different timelines.

```
## 5 distinct values in column category until 2022
```

```
## 5 distinct values in column comodity until 2022
```

There are five different items with five different categories.

For the simplicity of this project the **admin1**, **admin2**,**market**, **latitude**, **longitude**, **priceflag**, **pricetype** and **currency** variables are simply dropped from the data set.

```
foodprices <- foodprices[, -c(2:6,10:12)]
```

Identifying if commodities have the same unit along time series. Firstly knowing our unique commodities and units would be helpful.

```
unique(foodprices[foodprices$date<"2022-01-01",3])
```

```
## [1] "rice"           "eggs"           "sugar (white)"   "oil (vegetable)"
## [5] "lentils (green)"
```

```
unique(foodprices[foodprices$date<"2022-01-01",4])
```

```
## [1] "kg"     "1.8 kg" "810 g"  "900 g"
```

To check the equality of variables.

```
library(stringr)
foodprices[foodprices$date <"2022-01-01",] %>%
  group_by(item = str_extract(commodity,"^\\w+(?=\\s*)")) %>%
  summarise(isUnitSame = n_distinct(str_extract(unit,"[a-z]+$"))==1)
```

```
## # A tibble: 5 x 2
##   item    isUnitSame
##   <chr>   <lgl>
## 1 eggs    TRUE
## 2 lentils TRUE
## 3 oil     TRUE
## 4 rice    TRUE
## 5 sugar   TRUE
```

For sure, all commodities have the same unit from 2012 to 2021.

Transforming Data Types:

All column data formats should be acceptable for their context to evaluate the data correctly. In the case of this data, the majority of the variables need to be changed.

```
# Changing data types
foodprices$date <- as.Date(foodprices$date)
foodprices$category <- as.factor(foodprices$category)
foodprices$commodity <- as.factor(foodprices$commodity)
foodprices$unit <- as.factor(foodprices$unit)
foodprices$price <- as.integer(foodprices$price)
foodprices$usdprice <- as.numeric(foodprices$usdprice)
```

Next new variables **day** (classes: Mon - Sun), **month** (Jan - Dec) and **year** are created according to the date frame of the date variable.

```
### Create $day and $month and $year
foodprices$day <- weekdays(as.Date(foodprices$date))
foodprices$month <- months(as.Date(foodprices$date))
foodprices$year <- year(as.Date(foodprices$date))
```

Converting these new variables to proper characteristics.

```
# Convert to factor and integer
foodprices$day <- as.factor(foodprices$day)
foodprices$month <- as.factor(foodprices$month)
foodprices$year <- as.integer(foodprices$year)
```

Exploring Data Consistency:

```
## Minimum Price:2691
```

```
## Average Price:90742.5366336634
```

```
## Maximum Price:594000
```

The minimum,average,maximum values are realistic and consistent.

Go deep and zoom into each commodity to find the minimum price.

Table 1: Min Price For Each Item

commodity	min(price)
eggs	6085
lentils (green)	2691
oil (vegetable)	25790
rice	26860
sugar (white)	15800

It was evident that min price for two items of **eggs (6085 IRR)** and **lentils (2691 IRR)** is unrealistic.

These prices should be manipulated to solve this problem..Firstly these prices should be identified.

```
foodprices %>%
  filter_all(any_vars(. %in% c(2691,6085,29910)))
```

```
##         date            category       commodity   unit price usdprice
## 1 2020-02-15 meat, fish and eggs            eggs 1.8 kg  6085   0.1449
## 2 2020-04-15 meat, fish and eggs            eggs 1.8 kg  6085   0.1449
## 3 2020-06-15 meat, fish and eggs            eggs 1.8 kg 29910   0.7121
## 4 2020-06-15     pulses and nuts lentils (green)  900 g  2691   0.0641
##         day    month year
## 1  Saturday February 2020
## 2 Wednesday    April 2020
## 3    Monday     June 2020
## 4    Monday     June 2020
```

Changing the prices with the new value.

```
foodprices['price'][foodprices['price'] == 2691] <- 269100
foodprices['price'][foodprices['price'] == 6085] <- 244950
foodprices['price'][foodprices['price'] == 29910] <- 299100
foodprices['usdprice'][foodprices['usdprice'] == 0.1449] <- 6
foodprices['usdprice'][foodprices['usdprice'] == 0.7121] <- 7
foodprices['usdprice'][foodprices['usdprice'] == 0.0641] <- 4
```

Final Data Set Overview:

Now that the data has been thoroughly cleansed, the analysis may begin.

```
summary(foodprices)
```

```
##       date                              category             commodity
##  Min.   :2012-04-15   cereals and tubers :136   eggs            :110
##  1st Qu.:2015-05-15   miscellaneous food :131   lentils (green):110
##  Median :2018-06-15   pulses and nuts    :131   oil (vegetable):110
##  Mean   :2018-08-01   meat, fish and eggs:124   rice            :110
##  3rd Qu.:2022-05-15   oil and fats       :115   sugar (white)   :110
##  Max.   :2022-10-15   non-food           : 49   alcohol (spray):  7
##                       (Other)            : 63   (Other)         :192
##           unit          price            usdprice              day
##  kg          :297   Min.   :   700   Min.   : 0.0166   Friday   :143
##  900 g       :131   1st Qu.: 48870   1st Qu.: 1.7816   Monday   :109
##  810 g       :115   Median : 84699   Median : 2.7680   Saturday :114
##  1.8 kg      :110   Mean   :161088   Mean   : 4.3103   Sunday   :104
##  500 g       : 20   3rd Qu.:180000   3rd Qu.: 4.8467   Thursday : 96
##  cubic meter: 14   Max.   :900000   Max.   :21.4286   Tuesday  : 75
##  (Other)     : 62                                      Wednesday:108
##      month          year
##  April  : 79   Min.   :2012
##  August : 79   1st Qu.:2015
##  July   : 79   Median :2018
##  May    : 79   Mean   :2018
##  October: 79   3rd Qu.:2022
##  June   : 78   Max.   :2022
##  (Other):276
```

The finalized table would be like this.

Table 2: Food Prices Table

	date	category	commodity	unit	price	usdprice	day	month	year
2	2012-04-15	cereals and tubers	rice	kg	26860	2.1909	Sunday	April	2012
3	2012-04-15	meat, fish and eggs	eggs	1.8 kg	52760	4.3034	Sunday	April	2012
4	2012-04-15	miscellaneous food	sugar (white)	kg	15890	1.2961	Sunday	April	2012
5	2012-04-15	oil and fats	oil (vegetable)	810 g	25790	2.1036	Sunday	April	2012
6	2012-04-15	pulses and nuts	lentils (green)	900 g	27369	2.2324	Sunday	April	2012
7	2012-05-15	cereals and tubers	rice	kg	27000	2.2050	Tuesday	May	2012
8	2012-05-15	meat, fish and eggs	eggs	1.8 kg	40210	3.2838	Tuesday	May	2012
9	2012-05-15	miscellaneous food	sugar (white)	kg	15850	1.2944	Tuesday	May	2012
10	2012-05-15	oil and fats	oil (vegetable)	810 g	27880	2.2768	Tuesday	May	2012
11	2012-05-15	pulses and nuts	lentils (green)	900 g	27810	2.2711	Tuesday	May	2012

2.2 Data Analysis

As mentioned in the introduction, the current research aims to assess how US sanctions on Iran affected food prices.A trend chart is essential for our four main categories to reach this goal.A pivot table should be produced To produce this graph.

Pivot Table:

First, a new table will be created to visualize the trend change in price.

```
# New table for IRR Price group by category
foodprices_IRR <- foodprices[foodprices$date < "2022-01-01",] %>%
  group_by(date, category)%>%
  summarise(price)
```

The next step is to visualize our analysis.

```
g1 <- ggplot(data = foodprices_IRR, aes(x = date, y = price, color = category)) + geom_l
g1 + labs(title="Change Of Food Prices From 2012 To 2022 ", y="Price IRR", x="Year", cap
```

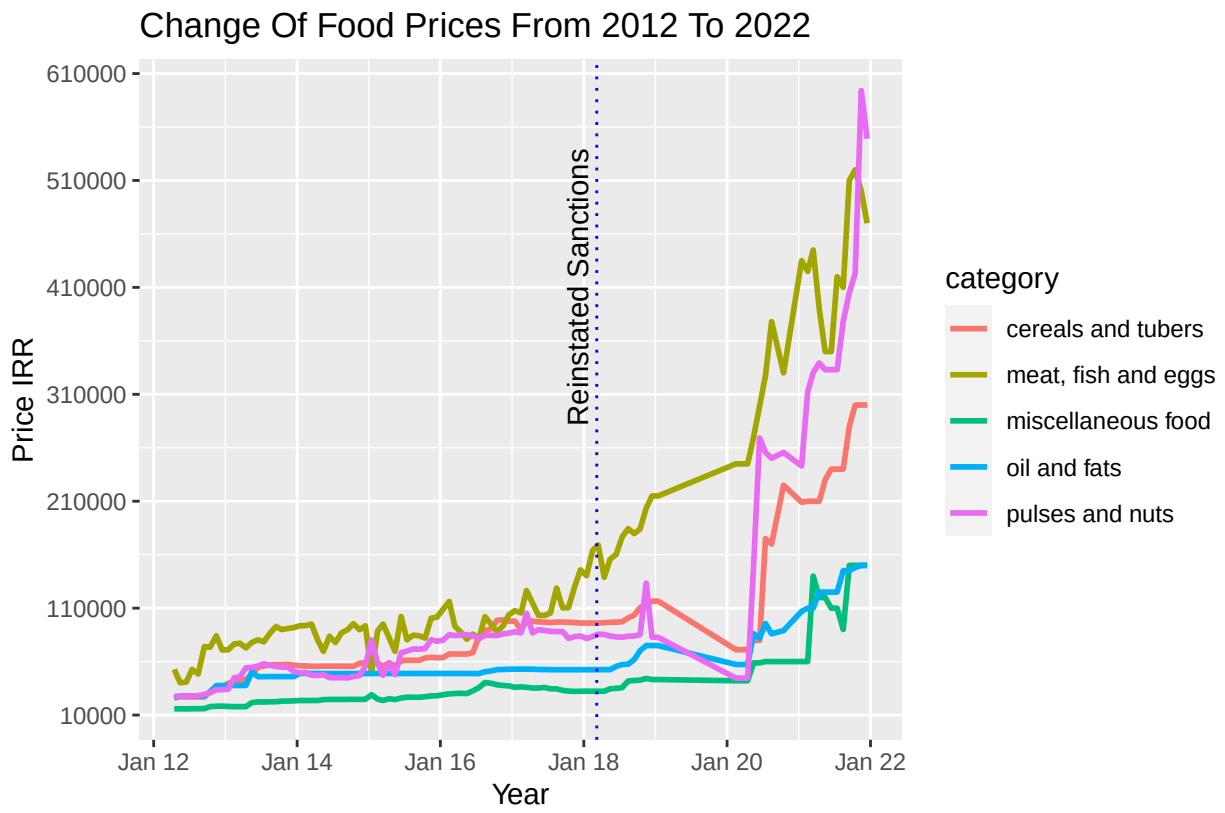

Fig.1:Food Prices Over Time

The figure depicts the price development for several food categories from 2012 to 2021. All food categories' prices have climbed significantly since the sanctions were reinstated, but the **pulses and nuts**, **meat, fish and eggs**, and **cereals and tubers** groups' price increases have been the worst.

```
# New table for IRR Price group by category
foodprices_USD <- foodprices[foodprices$date < "2022-01-01",] %>%
  group_by(date, category)%>%
  summarise(usdprice)
```

```
g1 <- ggplot(data = foodprices_USD, aes(x = date, y = usdprice, color = category)) + geo
g1 + labs(title="Change Of Food Prices From 2012 To 2022 ", y="Price USD", x="Year", cap
```

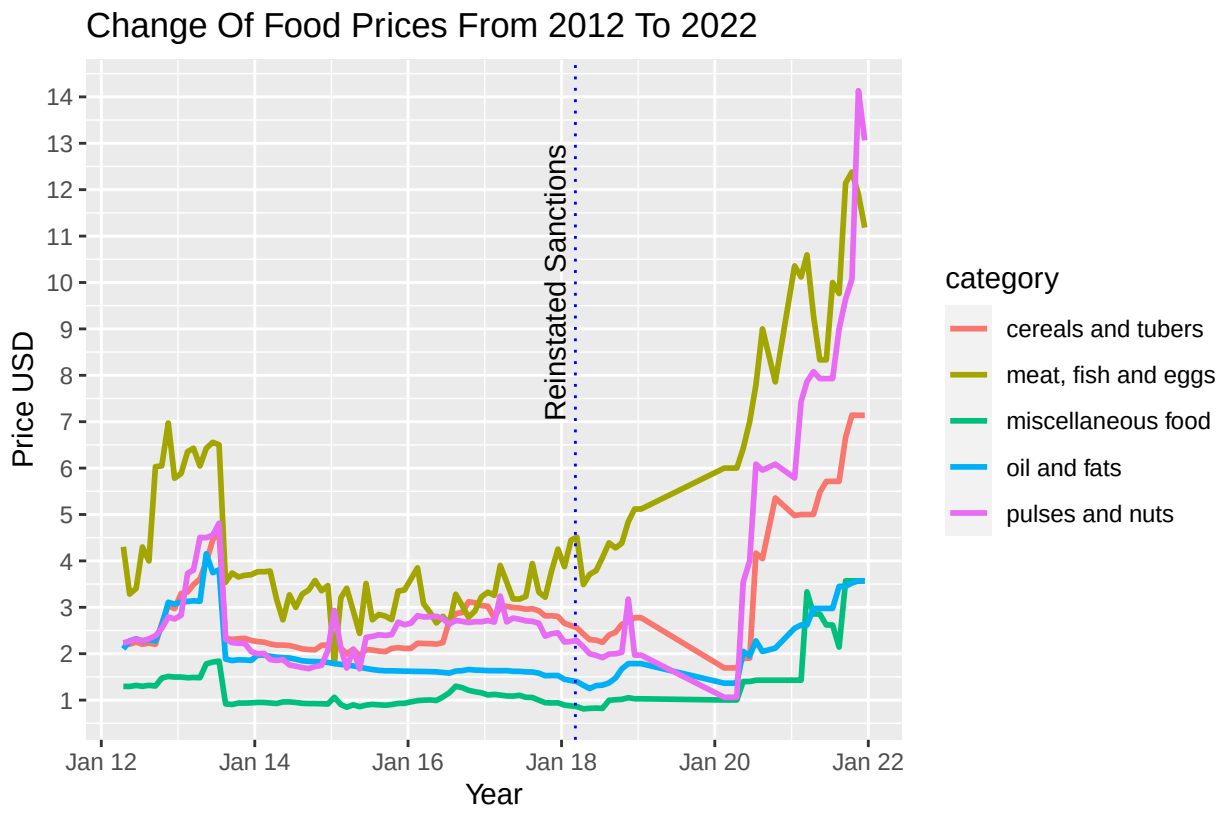

Fig.1:Food Prices Over Time

Refrences

Bloomberg. 2022. "Russia Is Now the World's Most-Sanctioned Nation." *Bloomberg.com*.

Wikipedia. 2022. "Sanctions Against Iran." *Wikipedia*. https://en.wikipedia.org/wiki/Sanctions_against_ was the most sanctioned,in Tehran and took hostages.