Food Inflation Study

Jessica Woods

2023-12-20

Food inflation rate:

$$\text{Inflation Rate} = \left(\frac{\text{Current Food Price Index} - \text{Base Food Price Index}}{\text{Base Food Price Index}} \right) \times 100$$

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

In this R document I will be specifically just cleaning data and correlating data points that are relevant to modeling, looking at the distribution of the data and saving those back to a new CSV file to be visually assessed in Python and modeled in R.

Below is the loaded CSV files needed to start cleaning for modeling and analysis work. The link to get the food inflation information is https://microdata.worldbank.org/index.php/catalog/4483 from December 2019 to December 2023 to complete 4 years of data on internation food inflation. "Monthly food price estimates by product and market 25 countries, 1353 markets, 2007/01/01-2023/12/01, version 2023/12/11" -The World Bank of Microdata website. There are two other factors in the analysis work, that is conflict (gun violence and war) and the US stock Market prices from December 2019 to December 2023.

The current HO: "The occurrence of conflict in key food-producing regions, coupled with fluctuations in the US stock market, significantly influences international food inflation rates. Higher instances of conflict and volatility in the US stock market are expected to correlate with increased food inflation on a global scale." The current Alternative HO : "The impact of conflict and US stock market fluctuations on international food inflation rates may not exhibit a significant correlation."

```
# Food inflation data
food_data <- read.csv("Food_inflation_2019_2023.csv", header=TRUE, sep=",")
# time to clean this up
clean_food_data <- subset(food_data, select = -X)
clean_food_data <-na.omit(clean_food_data)
clean_food_data <- subset(clean_food_data, select = -Market)
clean_food_data <-subset(clean_food_data, select = -Currency)
#fix the date
clean_food_data$Date <-as.Date(clean_food_data$Date, format = "%Y-%m-%d")
```

Narrowing down the top most conflicted countries in the past 4 years according to Wikipedia https://en.wikipedia.org/wiki/List_of_ongoing_armed_conflicts sited. Mexico, Ukraine, Afghanistan, Syria (Syrian Arab Republic), Ethiopia, Yemen Out of those listed I found 3 in the data set to work with.

```r
conflicted_countries_food_data <- clean_food_data %>%
  filter(Country %in% c("Afghanistan", "Syrian Arab Republic", "Yemen, Rep."))

conflicted_countries_food_data <- conflicted_countries_food_data %>%
  filter(Open != 0 & Close != 0 & High != 0)
confliced_countries_food_data <- na.omit(conflicted_countries_food_data[c("Open", "Close")])
```

```r
#Isolate Food Price Index
conflicted_countries_food_price_index<-conflicted_countries_food_data[conflicted_countries_food_data$Pr
conflicted_countries_food_data <- conflicted_countries_food_data %>%
  filter(Product != "food_price_index")
str(conflicted_countries_food_price_index)
```

```
## 'data.frame':    11123 obs. of  8 variables:
##  $ Country: chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Region : chr  "Badakhshan" "Badakhshan" "Badakhshan" "Badakhshan" ...
##  $ Product: chr  "food_price_index" "food_price_index" "food_price_index" "food_price_index" ...
##  $ Date   : Date, format: "2019-12-01" "2019-12-01" ...
##  $ Open   : num  1.12 1.12 1.13 1.13 1.13 1.13 1.13 1.13 1.23 1.23 ...
##  $ High   : num  1.13 1.13 1.14 1.14 1.14 1.14 1.2 1.2 1.25 1.25 ...
##  $ Low    : num  1.11 1.11 1.12 1.12 1.12 1.12 1.12 1.12 1.21 1.21 ...
##  $ Close  : num  1.13 1.13 1.13 1.13 1.13 1.13 1.2 1.2 1.25 1.25 ...
```

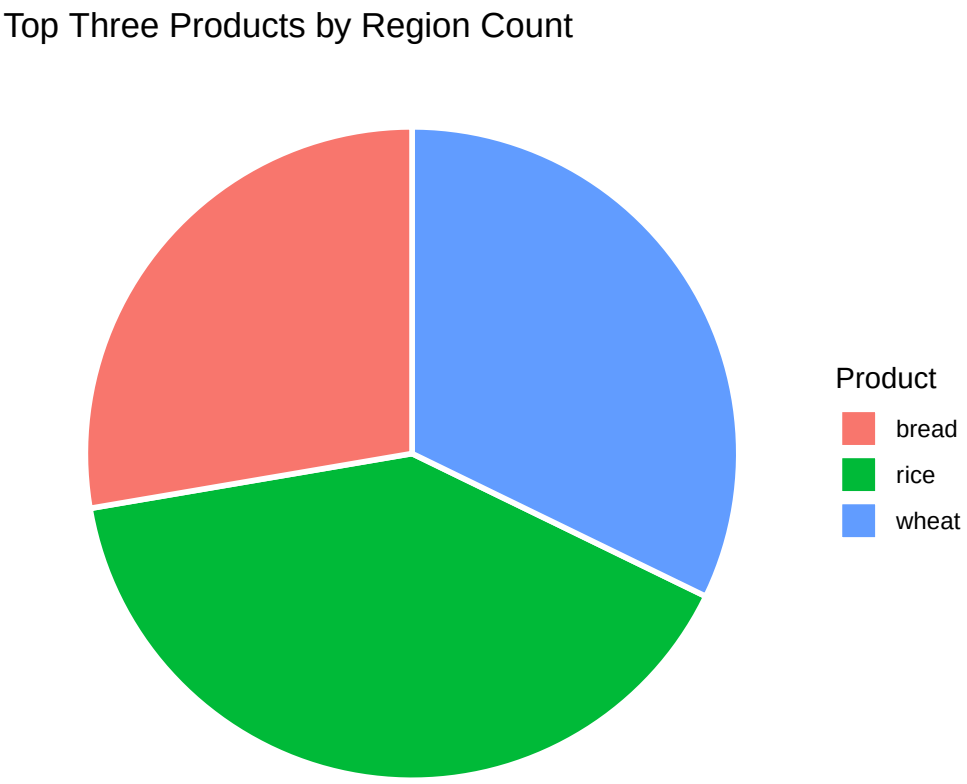
```r
# factor and find top products from data, naming convientions are so important as I work through this
top_products <- conflicted_countries_food_data
#  factor  products, regions, countries
top_products$Product <- as.factor(top_products$Product)
top_products$Country <- as.factor(top_products$Country)
# Convert 'Region' to a factor within each country
top_products$Region <- as.factor(top_products$Region)
top_products$Region <- factor(
  top_products$Region,
  levels = unique(top_products$Region)
)

# I want to know which regions have the highest fluctuation in their markets
top_products$Fluctuation <- top_products$Close - top_products$Open
# Find observations with the highest fluctuations
top_fluctuations <- top_products[order(top_products$Fluctuation, decreasing = TRUE), ]
head(top_fluctuations)
```

```
##                     Country      Region                             Product
## 69874 Syrian Arab Republic Deir-ez-Zor livestock_sheep_two_year_old_male
## 98045 Syrian Arab Republic       Idleb livestock_sheep_two_year_old_male
## 74774 Syrian Arab Republic Deir-ez-Zor livestock_sheep_two_year_old_male
## 73549 Syrian Arab Republic Deir-ez-Zor livestock_sheep_two_year_old_male
## 74770 Syrian Arab Republic Deir-ez-Zor livestock_sheep_two_year_old_male
## 73545 Syrian Arab Republic Deir-ez-Zor livestock_sheep_two_year_old_male
##             Date    Open    High     Low   Close Fluctuation
## 69874 2023-12-01 2449085 3261065 2060071 3261065    811980.1
## 98045 2023-08-01 2843081 3522899 2699200 3522899    679817.8
## 74774 2023-12-01 2591094 3243195 2139313 3243195    652101.2
```

```
## 73549 2023-12-01 2613006 3256648 2141895 3256648    643642.5
## 74770 2023-08-01 2217816 2779033 1977785 2779033    561217.1
## 73545 2023-08-01 2266728 2820968 2082072 2820968    554239.3
```

```r
product_region_count <- top_products %>%
  group_by(Product) %>%
  summarise(Region_Count = n_distinct(Region)) %>%
  arrange(desc(Region_Count))

# Product with the most regions
product_with_most_regions <- product_region_count[which.max(product_region_count$Region_Count), ]
top_three_products <- product_region_count %>%
  slice_head(n = 3)
# Extract the top three products
top_three_products_list <- top_three_products$Product

# Filter the original data to get regions and countries for the top three products
regions_countries_top_three <- top_products %>%
  filter(Product %in% top_three_products_list) %>%
  distinct(Product, Region, Country)


# Create a pie chart for the top three regions
ggplot(top_three_products, aes(x = "", y = Region_Count, fill = Product)) +
  geom_bar(stat = "identity", width = 1, color = "white", size = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Top Three Products by Region Count", fill = "Product", x = NULL, y = NULL) +
  theme_void() +
  theme(legend.position = "right") +
  scale_fill_discrete(name = "Product")
```

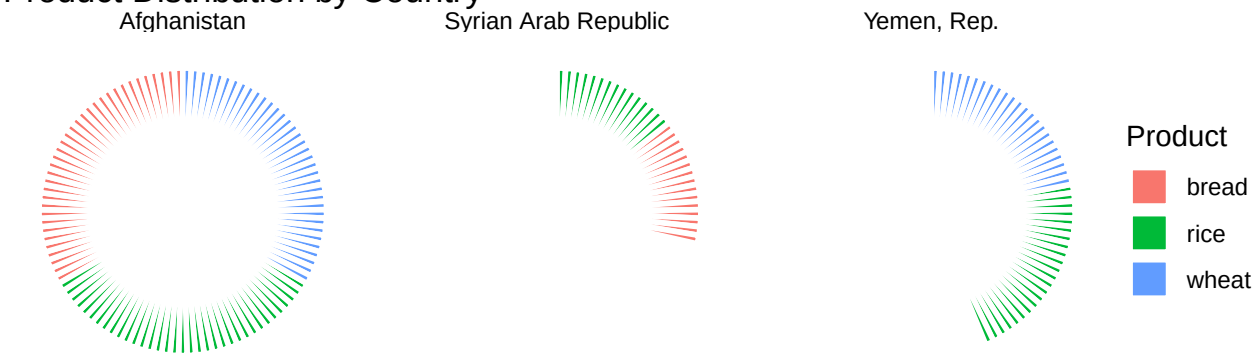
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
top_three_products
```

```
## # A tibble: 3 x 2
##   Product Region_Count
##   <fct>          <int>
## 1 rice              71
## 2 wheat             57
## 3 bread             49
```

```
unique_countries <- unique(regions_countries_top_three$Country)

product_counts <- regions_countries_top_three %>%
  group_by(Country, Product, Region) %>%
  summarise(Count = n(), .groups = "drop")

# Create a pie chart for each country showing product distribution within regions
ggplot(product_counts, aes(x = "", y = Count, fill = Product)) +
  geom_bar(stat = "identity", width = 1, color = "white", size = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Product Distribution by Country", fill = "Product", x = NULL, y = NULL) +
  theme_void() +
  theme(legend.position = "right") +
  facet_wrap(~ Country)
```




```
num_unique_regions <- top_products %>%
  summarise(Num_Regions = n_distinct(Region))

num_unique_regions
```

```
##   Num_Regions
## 1          71
```

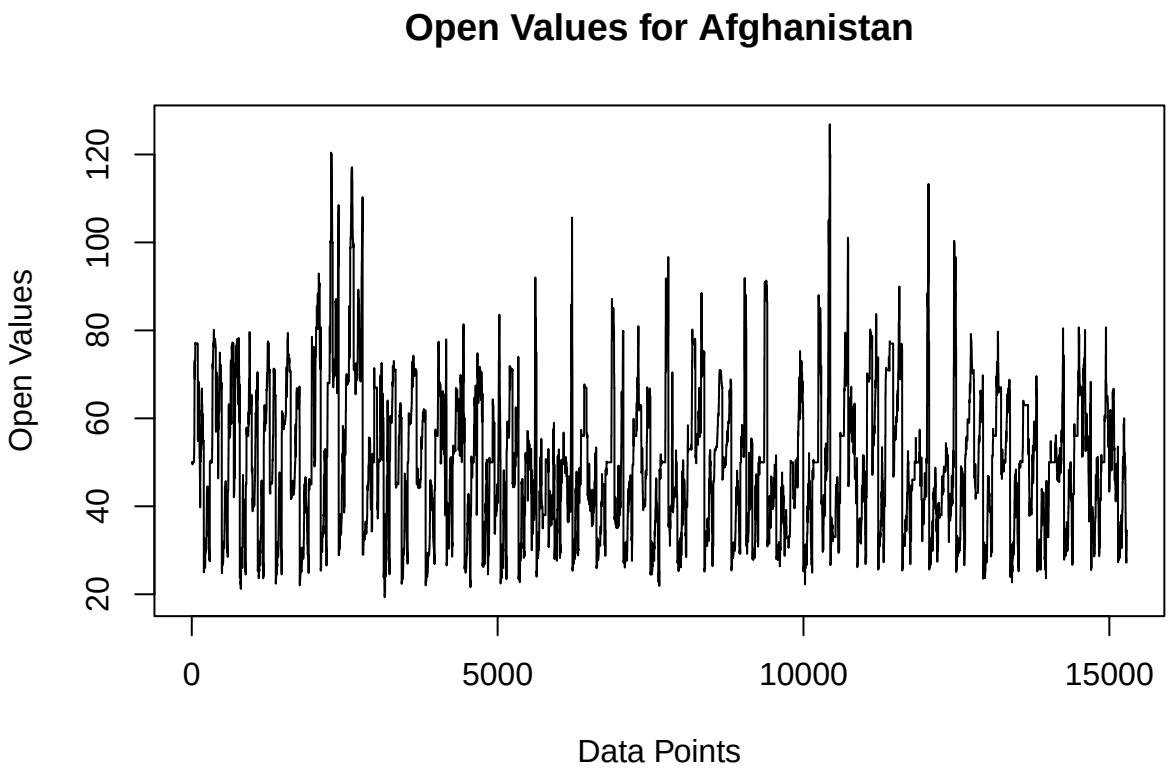
This second set of data is for the main products I am evaluating in each country. Which is Rice, Wheat and Bread.

```
# drop all product data that is not wheat, rice or bread and factor data
conflicted_countries_food_data_main_food <- conflicted_countries_food_data %>%
  filter(Product %in% c("rice", "wheat", "bread"))
#str(conflicted_countries_food_data_main_food)
conflicted_countries_food_data_main_food$Product <- as.factor(conflicted_countries_food_data_main_food$
conflicted_countries_food_data_main_food$Country <- as.factor(conflicted_countries_food_data_main_food$
# Convert 'Region' to a factor within each country
conflicted_countries_food_data_main_food$Region <- as.factor(conflicted_countries_food_data_main_food$R
conflicted_countries_food_data_main_food$Region <- factor(
  conflicted_countries_food_data_main_food$Region,
  levels = unique(conflicted_countries_food_data_main_food$Region)
)
str(conflicted_countries_food_data_main_food)
```

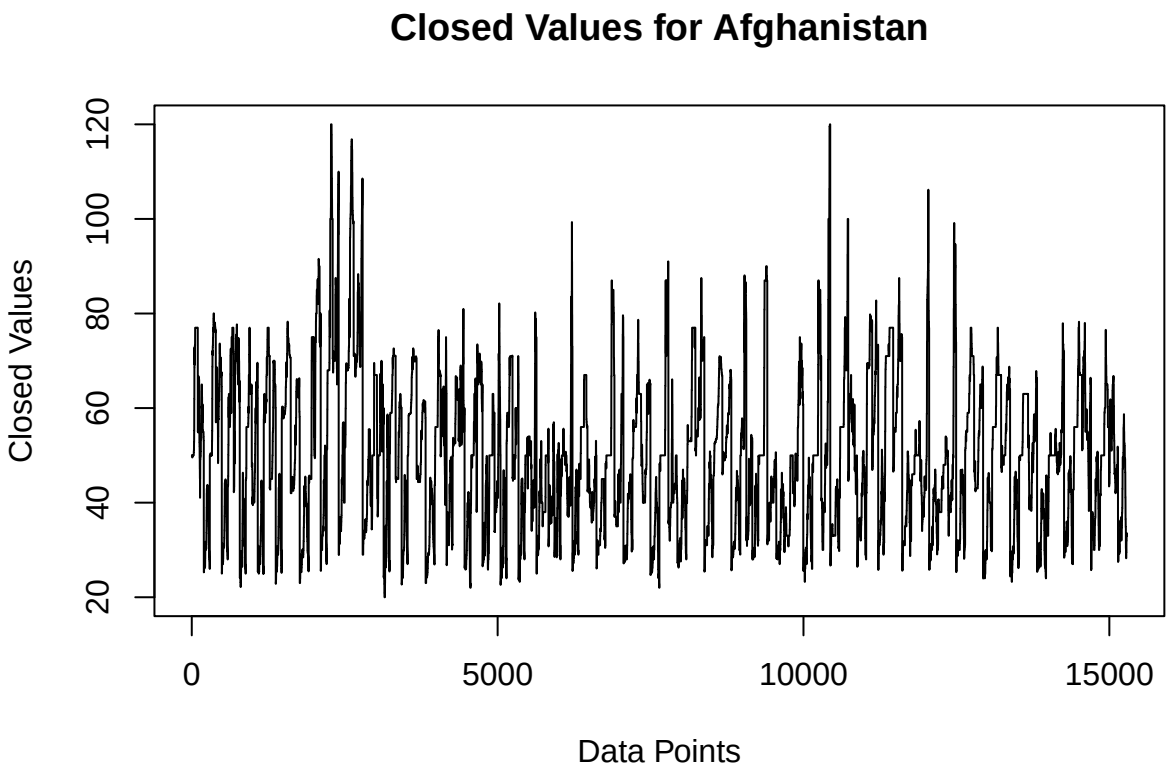
```
## 'data.frame':    27342 obs. of  8 variables:
##  $ Country: Factor w/ 3 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Region : Factor w/ 71 levels "Badakhshan","Badghis",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Product: Factor w/ 3 levels "bread","rice",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date   : Date, format: "2019-12-01" "2019-12-01" ...
##  $ Open   : num  49.8 49.8 49.8 49.8 49.9 ...
##  $ High   : num  49.9 49.9 49.9 49.9 50 ...
##  $ Low    : num  49.7 49.7 49.7 49.7 49.8 ...
##  $ Close  : num  49.8 49.8 49.9 49.9 49.9 ...
```

Afghanistan data on Food Inflation open and close, the difference between the two. Its important to look at the distribution of the data to do further analysis and cleaning of the data. If the differences between open and close prices themselves follow a normal distribution, it could suggest a certain level of regularity and randomness in price movements.

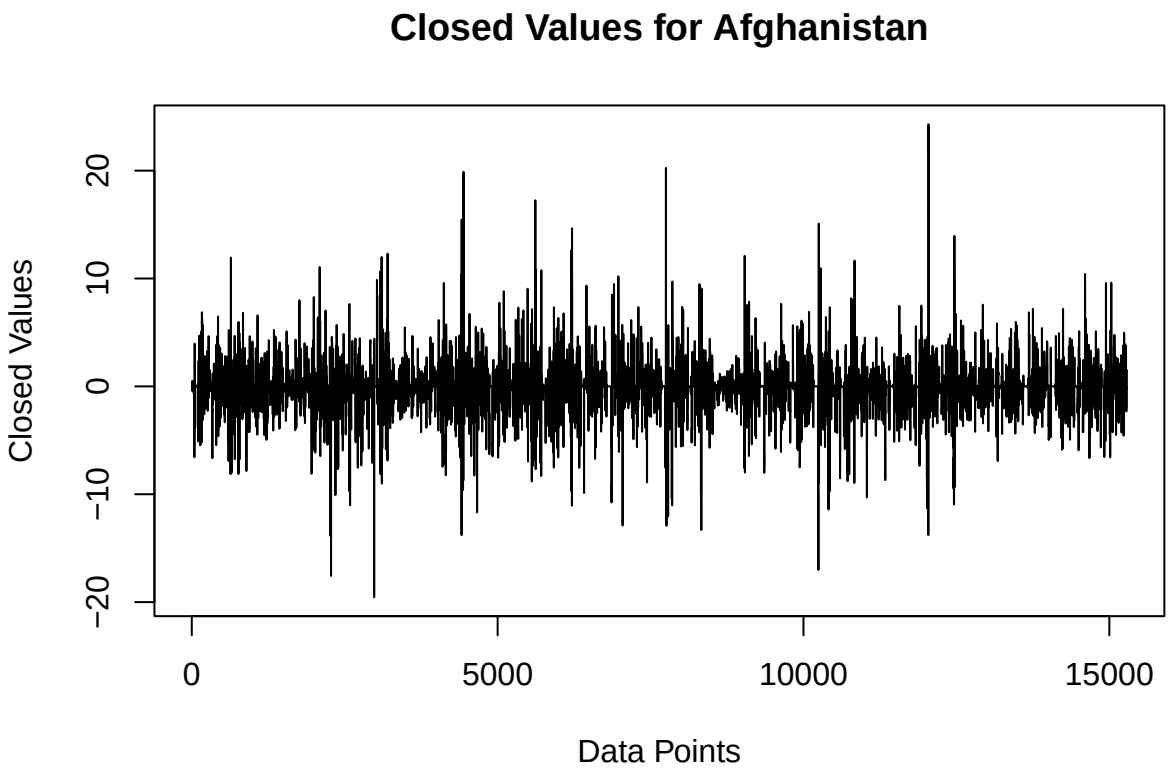
```
#switched values to evaluate the main foods we figured out above.
data_afghanistan <- conflicted_countries_food_data_main_food[conflicted_countries_food_data_main_food$C

open_values_afghanistan <- data_afghanistan$Open
closed_values_afghanistan <- data_afghanistan$Close
difference_afghanistan <- open_values_afghanistan - closed_values_afghanistan


# Create a sequence for x-axis (assuming you want a sequence of numbers as x-axis)
x <- seq(length(open_values_afghanistan))
x2 <-seq(length(closed_values_afghanistan))
x3 <- seq(length(difference_afghanistan))
# Plotting Open values for Afghanistan
plot(x, open_values_afghanistan, type = "l", xlab = "Data Points", ylab = "Open Values", main = "Open V
```



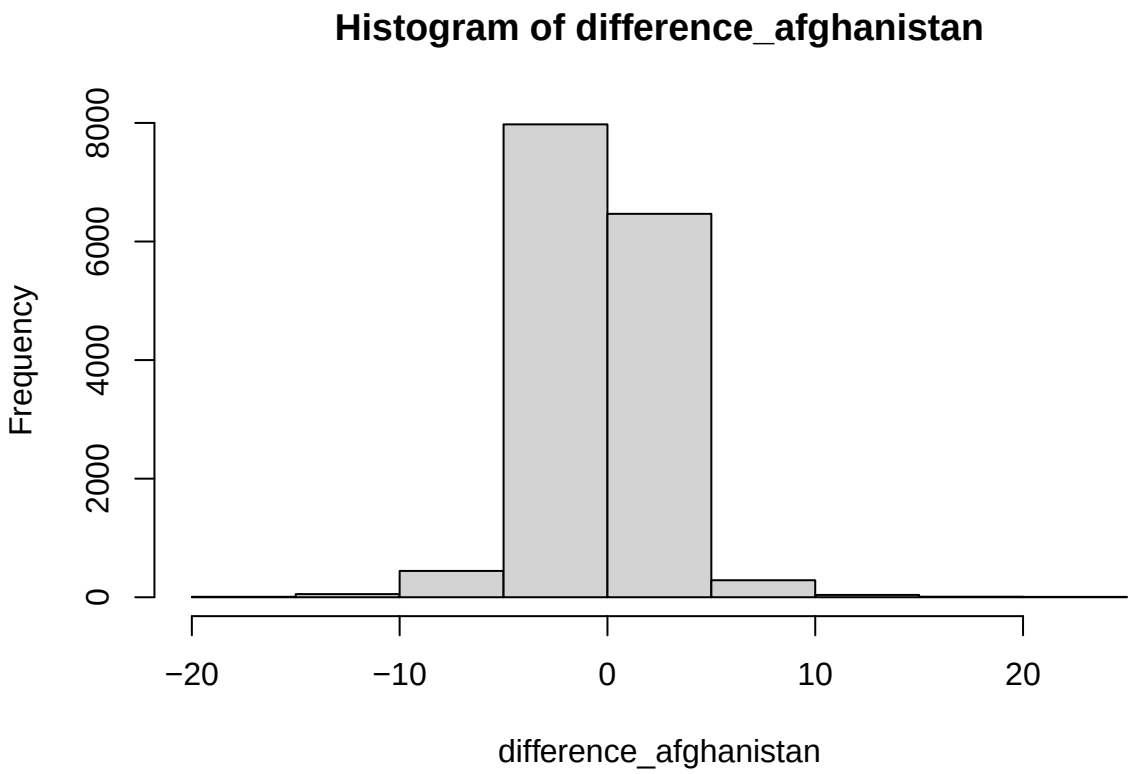

```
plot(x2, closed_values_afghanistan, type = "l", xlab = "Data Points", ylab = "Closed Values", main = "C
```



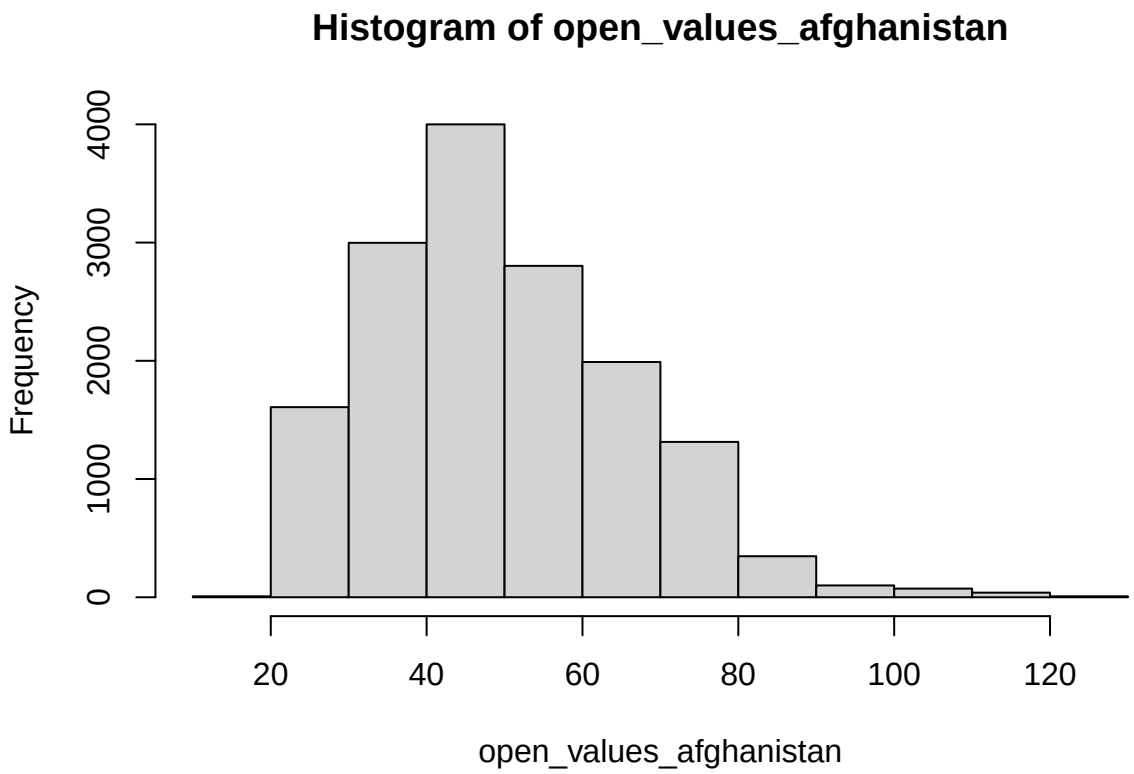

```
plot(x3, difference_afghanistan, type = "l", xlab = "Data Points", ylab = "Closed Values", main = "Clos
```



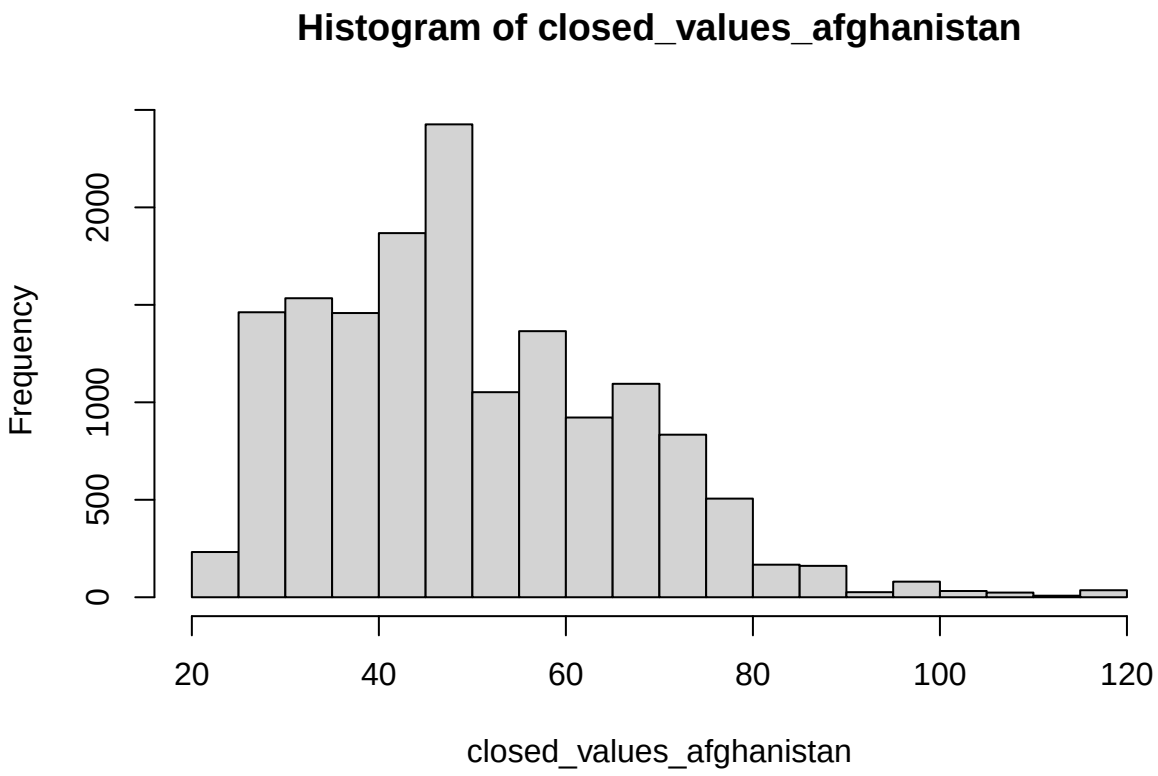

```
#seems to be normally distributed, this helps narrow down appropriate models to examine Afghanistan spe
hist(difference_afghanistan)
```



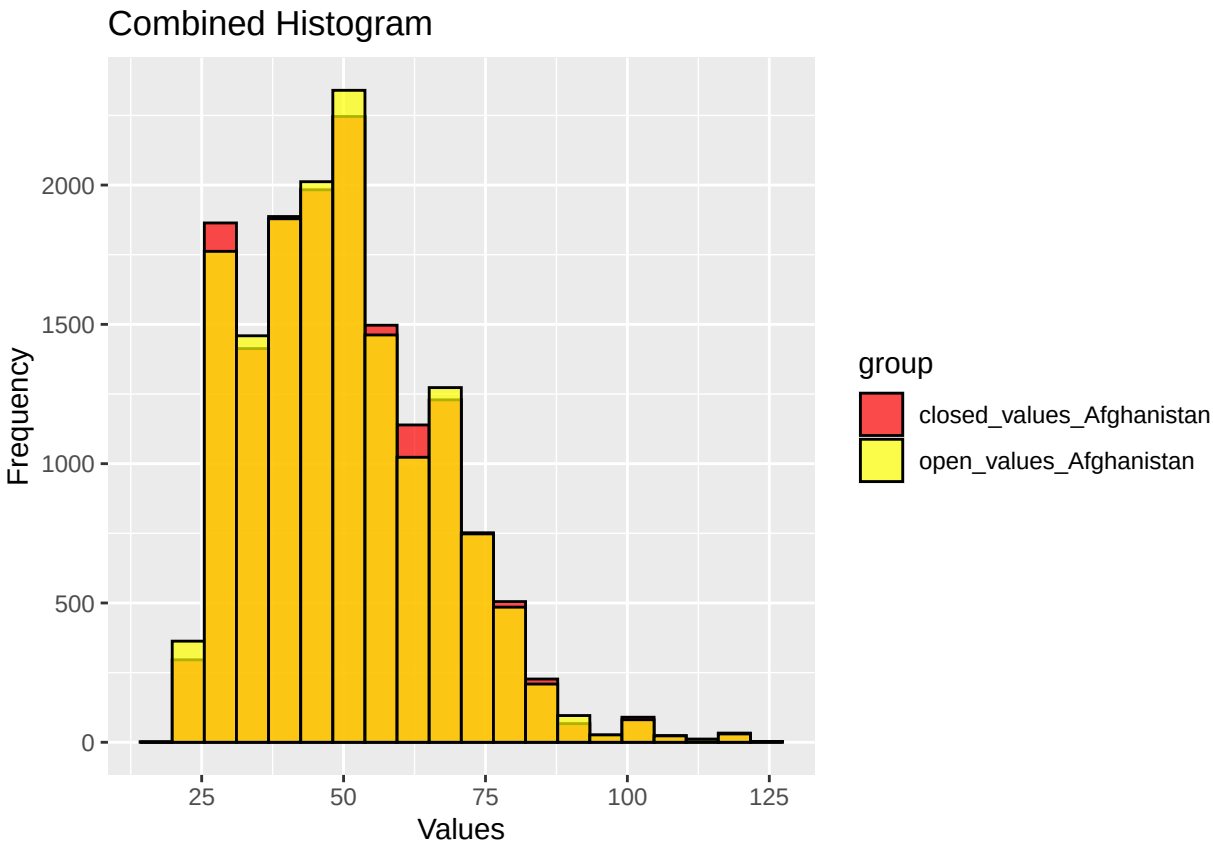

```
# open and close are distributed to the right
hist(open_values_afghanistan)
```




```
hist(closed_values_afghanistan)
```




```
combined_data <- data.frame(value = c(open_values_afghanistan, closed_values_afghanistan), group = rep(

ggplot(combined_data, aes(x = value, fill = group)) +
  geom_histogram(position = "identity", alpha = 0.7, bins = 20, color = "black") +
  labs(title = "Combined Histogram", x = "Values", y = "Frequency") +
  scale_fill_manual(values = alpha(c("red", "yellow"), .2))
```



In financial markets, a normal distribution of price index differences might support the idea of market efficiency, suggesting that market prices reflect all available information, and arbitrage opportunities might be limited. Furthermore, the questioning factor still remains that the open and close histograms are right skewed, which indicate otherwise. Its considered a bullish market indication.

I recalculated the index values to hold true to the individual products that we are using instead of using the "basket f goods" price index listed above in the original data set. I need to have strong comparisons per region to fully understand inflation in conflicted areas.

```
data_afghanistan <- data_afghanistan %>%
  mutate(MonthYear = format(Date, "%Y-%m"))
grouped_data_afghanistan <- data_afghanistan %>%
  group_by(MonthYear, Product, Region, Open, Close) %>%
  summarize(
    CurrentPrice = mean((Open + Close) / 2),
  )
```

```
## `summarise()` has grouped output by 'MonthYear', 'Product', 'Region', 'Open'.
## You can override using the `.groups` argument.
```

```
#print(grouped_data_afghanistan)

# Calculate the base price for each region
base_prices <- grouped_data_afghanistan %>%
  group_by(Region) %>%
  summarize(
    BasePrice = mean(CurrentPrice)
```

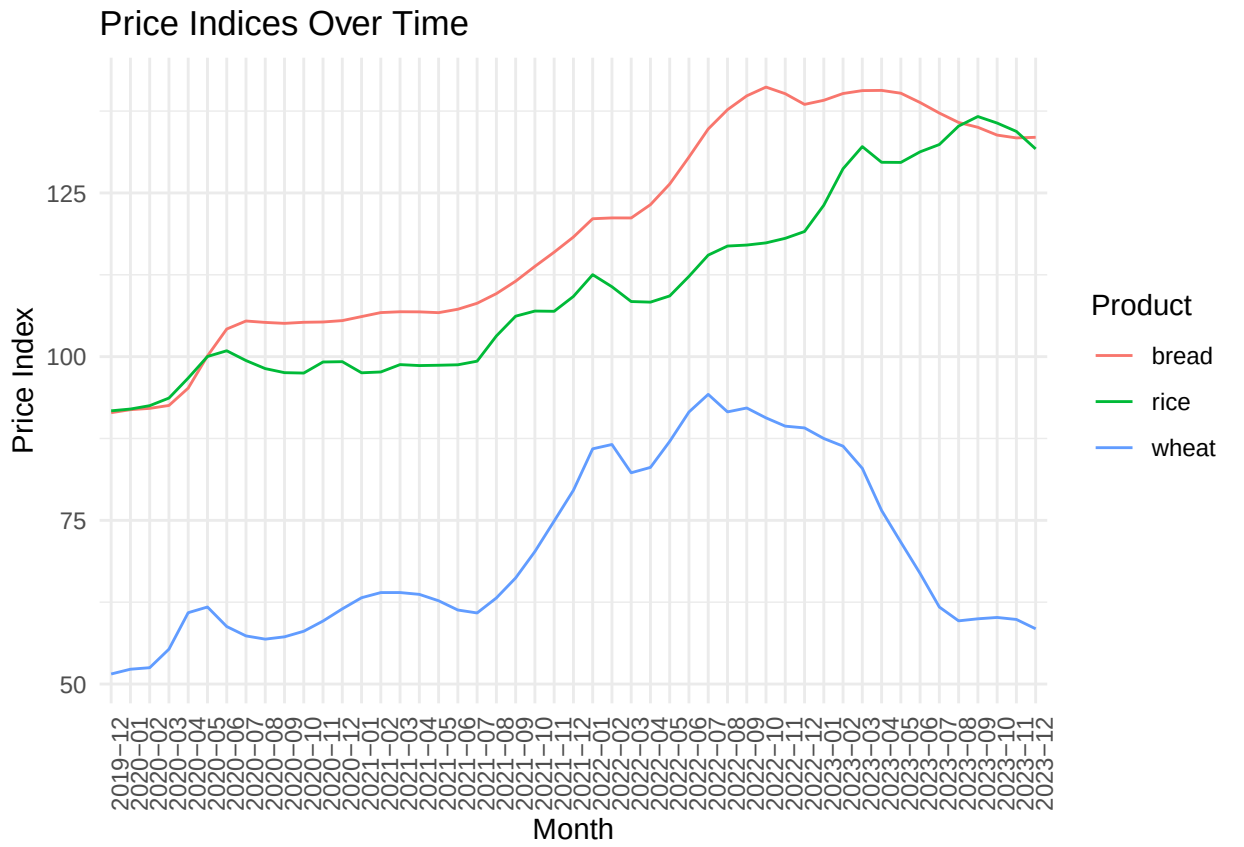
```
  )
# Merge the base prices back into the grouped_data_afghanistan
final_data_afghanistan <- merge(grouped_data_afghanistan, base_prices, by = "Region")
#print(final_data_afghanistan)

# recalculate the price index

new_index_afghanistan <- final_data_afghanistan %>%
  group_by(MonthYear, Product) %>%
  summarize(
    PriceIndex = mean((CurrentPrice / BasePrice) * 100)  # Price Index formula
  )
```

```
## `summarise()` has grouped output by 'MonthYear'. You can override using the
## `.groups` argument.
```

```
print(new_index_afghanistan)
```

```
## # A tibble: 147 x 3
## # Groups:   MonthYear [49]
##    MonthYear Product PriceIndex
##    <chr>     <fct>        <dbl>
##  1 2019-12   bread         91.4
##  2 2019-12   rice          91.7
##  3 2019-12   wheat         51.5
##  4 2020-01   bread         91.9
##  5 2020-01   rice          92.0
##  6 2020-01   wheat         52.3
##  7 2020-02   bread         92.1
##  8 2020-02   rice          92.5
##  9 2020-02   wheat         52.5
## 10 2020-03   bread         92.5
## # i 137 more rows
```

```
ggplot(new_index_afghanistan, aes(x = MonthYear, y = PriceIndex, group = Product, color = Product)) +
  geom_line() +
  labs(title = "Price Indices Over Time",
       x = "Month",
       y = "Price Index") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
# these three data points are important to the final set of data for Afghanistan
# Base price, current price, and price index
# merge them into one data set
# all the cleaning for Afghanistan is done-
final_data_afghanistan <- merge(final_data_afghanistan, new_index_afghanistan, by = c("MonthYear", "Proc
str(final_data_afghanistan)
```

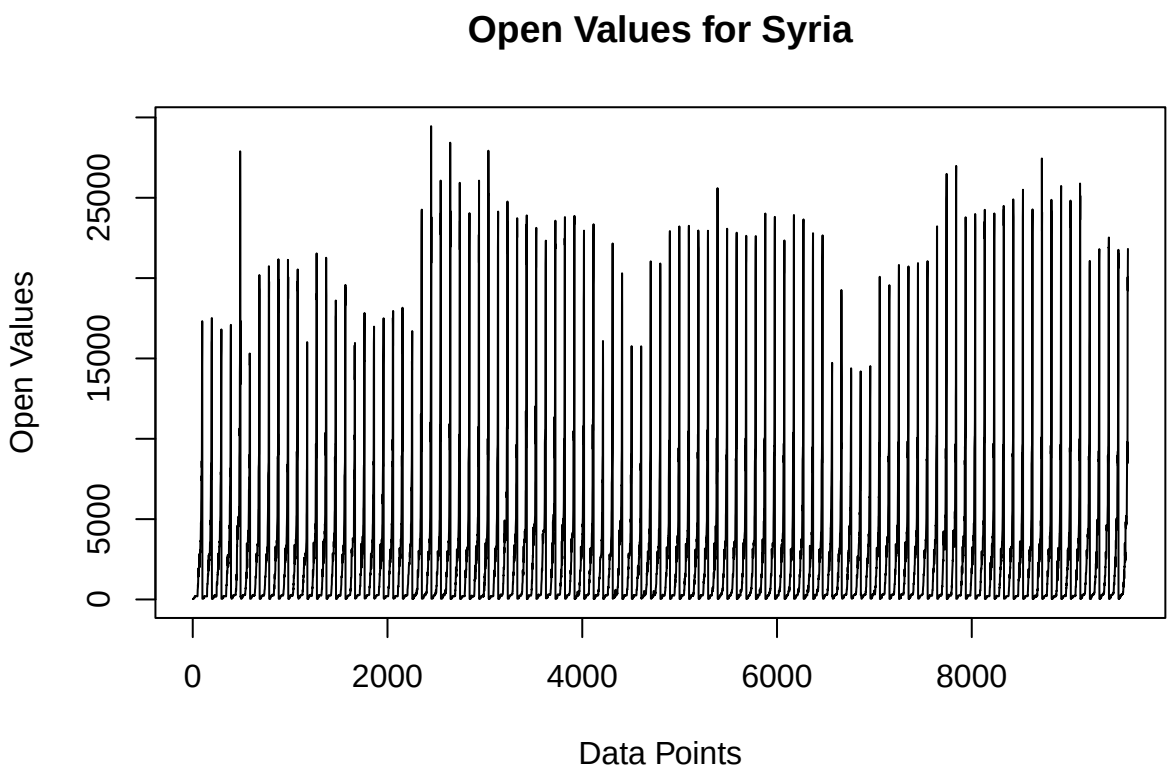
```
## 'data.frame':    6027 obs. of  8 variables:
##  $ MonthYear   : chr  "2019-12" "2019-12" "2019-12" "2019-12" ...
##  $ Product     : Factor w/ 3 levels "bread","rice",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Region      : Factor w/ 71 levels "Badakhshan","Badghis",..: 1 1 20 6 30 4 31 8 24 8 ...
##  $ Open        : num  49.8 50 48.3 51.7 50.8 ...
##  $ Close       : num  49.8 50 48.1 51.7 51 ...
##  $ CurrentPrice: num  49.8 50 48.2 51.7 50.9 ...
##  $ BasePrice   : num  53 53 46.1 67.1 50.2 ...
##  $ PriceIndex  : num  91.4 91.4 91.4 91.4 91.4 ...
```

Similarly after looking at AFghanistan we want to do the the same thing for the other two countries. This is just to look at how the data is distributed in the market per country out of the three different countries that we are evaluating.Below we will evaluate the Syrian Arab Republic data, leaving Yemen last. I am trying to break this down so you can follow along regardless of your background. The coolest thing I have learned in College about being intelligent or beign able to interpret data... is that its pointless if you can explain it well enough for anyone else to understand.

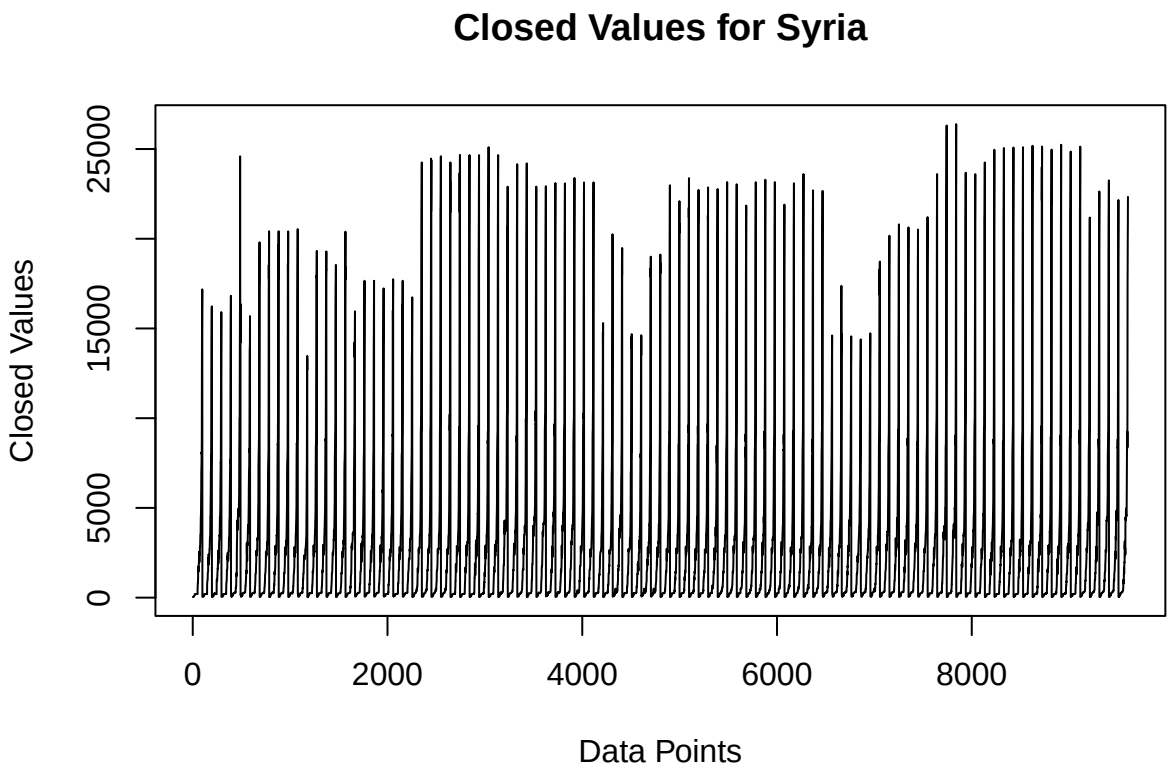
```r
data_Syria <- conflicted_countries_food_data_main_food[conflicted_countries_food_data_main_food$Country
str(data_Syria)
```

```
## 'data.frame':    9604 obs. of  8 variables:
##  $ Country: Factor w/ 3 levels "Afghanistan",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Region : Factor w/ 71 levels "Badakhshan","Badghis",..: 36 36 36 36 36 36 36 36 36 36 ...
##  $ Product: Factor w/ 3 levels "bread","rice",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date   : Date, format: "2019-12-01" "2020-01-01" ...
##  $ Open   : num  35.5 35.1 35.5 50.4 38.6 ...
##  $ High   : num  38.4 36.8 42.3 53.1 42.3 ...
##  $ Low    : num  32.7 33.5 34.6 40.6 35.2 ...
##  $ Close  : num  35.5 35.5 42.3 40.6 42.3 ...
```

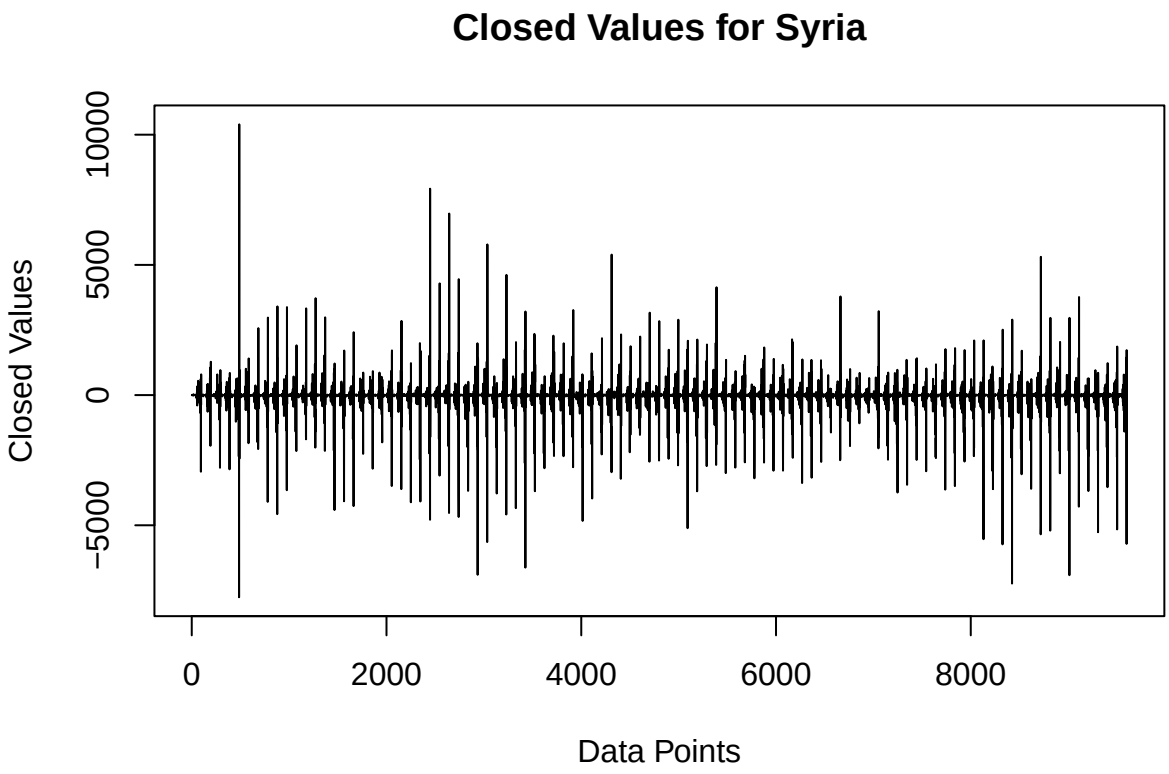
```r
open_values_Syria <- data_Syria$Open
closed_values_Syria <- data_Syria$Close
difference_Syria <- open_values_Syria - closed_values_Syria


x <- seq(length(open_values_Syria))
x2 <-seq(length(closed_values_Syria))
x3 <- seq(length(difference_Syria))
# Plotting Open values for Syria
plot(x, open_values_Syria, type = "l", xlab = "Data Points", ylab = "Open Values", main = "Open Values
```



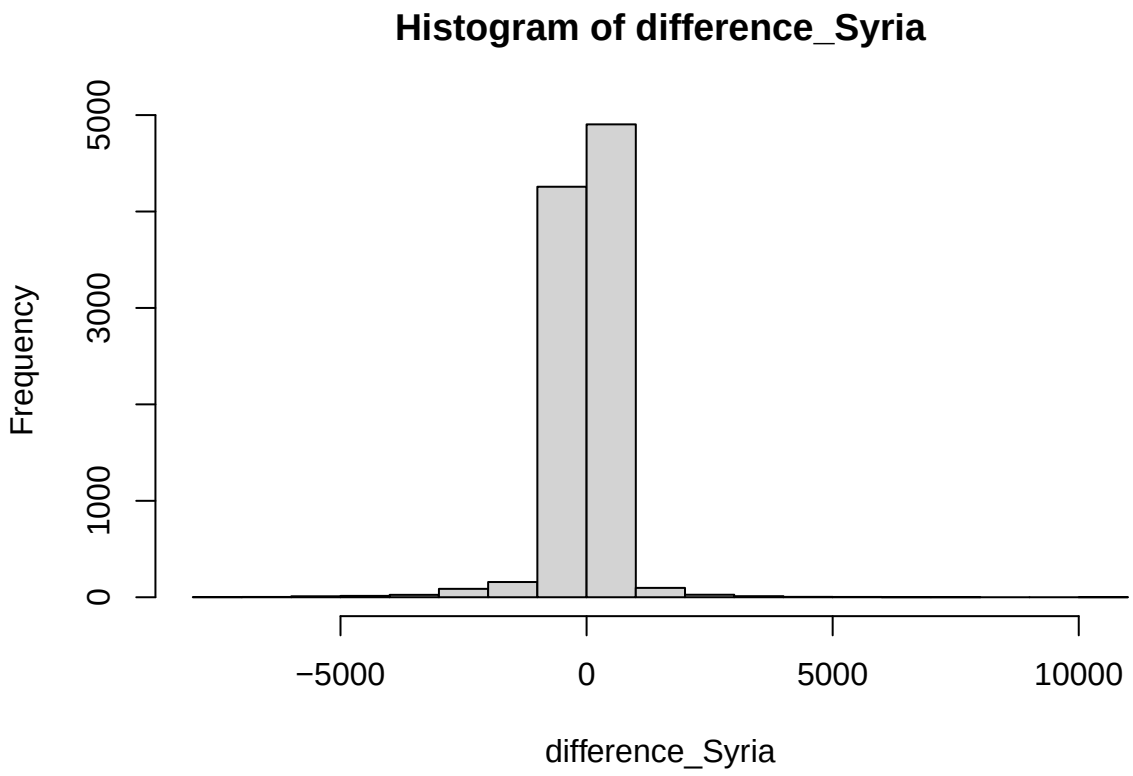
```
plot(x2, closed_values_Syria, type = "l", xlab = "Data Points", ylab = "Closed Values", main = "Closed V
```



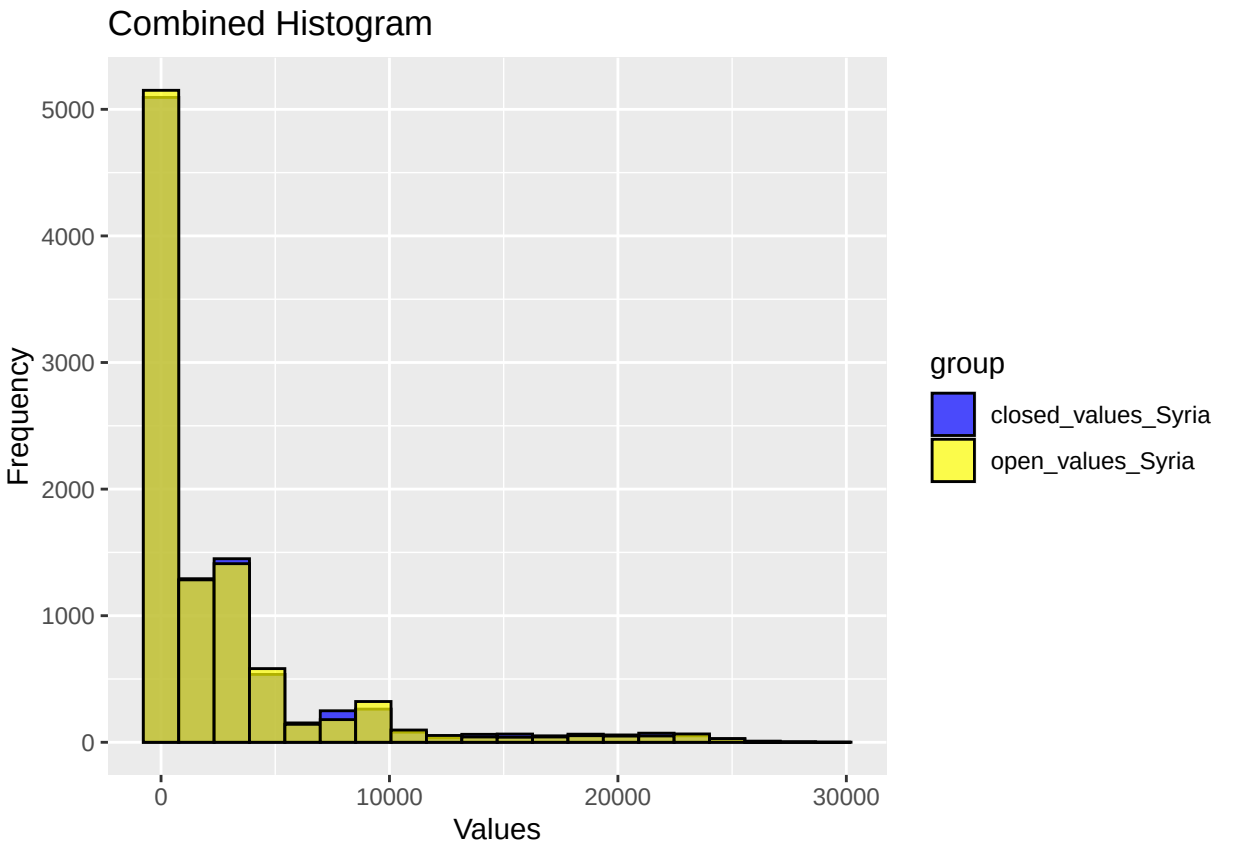

```
plot(x3, difference_Syria, type = "l", xlab = "Data Points", ylab = "Closed Values", main = "Closed Valu
```




```
#seems to be normally distributed, this helps narrow down appropriate models to examine Syria specifica
hist(difference_Syria)
```




```
combined_data <- data.frame(value = c(open_values_Syria, closed_values_Syria), group = rep(c("open_valu

ggplot(combined_data, aes(x = value, fill = group)) +
  geom_histogram(position = "identity", alpha = 0.7, bins = 20, color = "black") +
  labs(title = "Combined Histogram", x = "Values", y = "Frequency") +
  scale_fill_manual(values = alpha(c("blue", "yellow"), .2))
```



Below is the conflict data which is from https://acleddata.com/data-export-tool/. This requires a access key to get this data, it seems to be the most up to date data on conflicts. That coincides with the Dates from the Food inflation.

```
# Conflict data
```

Below is the the US Stock Market Data which is from https://www.ers.usda.gov/data-products/wheat-data/ and Yahoo Stocks for rought rice https://finance.yahoo.com/quote/ZR%3DF/history?period1=1575158400&period2=1703030400&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true. With US Stocks.. need the names of each actual stock that dominates these regions to represent the coefficients of wheat which is my target data. It would essentially look like this: wheat = target, samples = [wheat company 1, wheat company 2, wheat company 3, wheat company 4] with all of the open and close data listed in the US stock markets. Then I will find which coefficients or features are significant in the target data of wheat in that country. You take wheat data from that country in the above Food_inflation_2019_2023.csv that is loaded and that is the target, then building out a data set of different companies of wheat in the US stock market to create the other half of the data. I learned how to do this between Advanced Statistics and Machine Learning. Its kind of important when looking at Linear Regression Models.

```
# US Stock Market Data
```