# THE US INFLATION PHENOMENON

**AUTHOR** | Rand Sobczak Jr.

**DATE** | 16 September 2021

# Table of contents

01

# Problem Identification

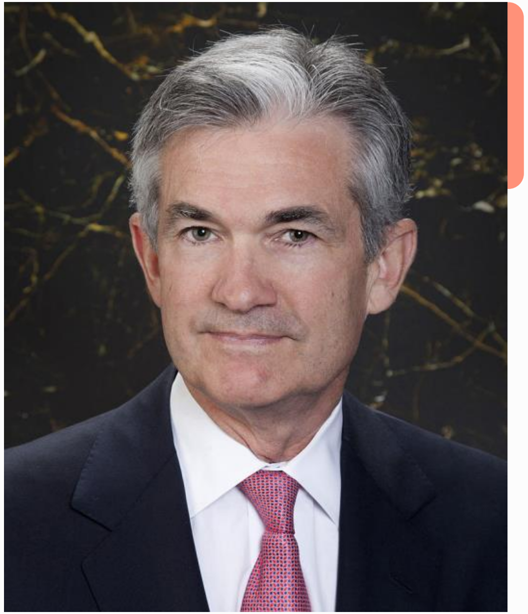
Developing a model to explain & understand the phenomenon of US Inflation


# What is Inflation?

**Inflation** is the **decline of purchasing power** of a given currency over time. **Deflation is the inverse**

# Inflation is important

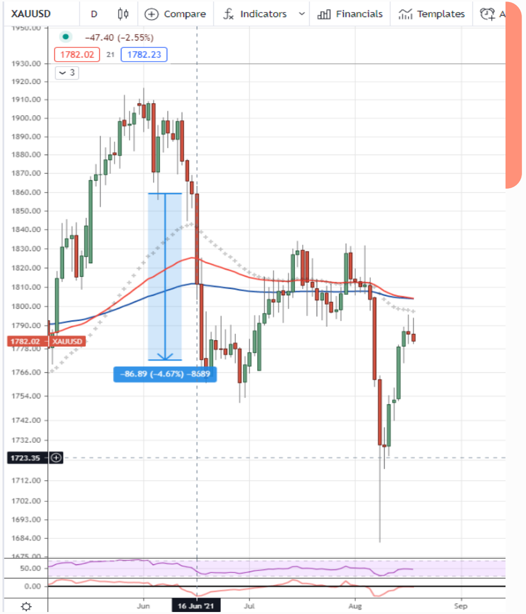
but it's a **highly debated** phenomenon in economics. Many economists maintain that **moderate** inflation **levels** are needed to **drive consumption**, assuming that higher levels of **spending are crucial** for **economic growth**



# Inflation is important

but it's a highly debated phenomenon in economics. Many economists maintain that moderate inflation levels are needed to drive consumption, assuming that higher levels of spending are crucial for economic growth

& **stabilizing it** is one of three objectives of the **Federal Reserve who's decisions move** the global **financials markets**

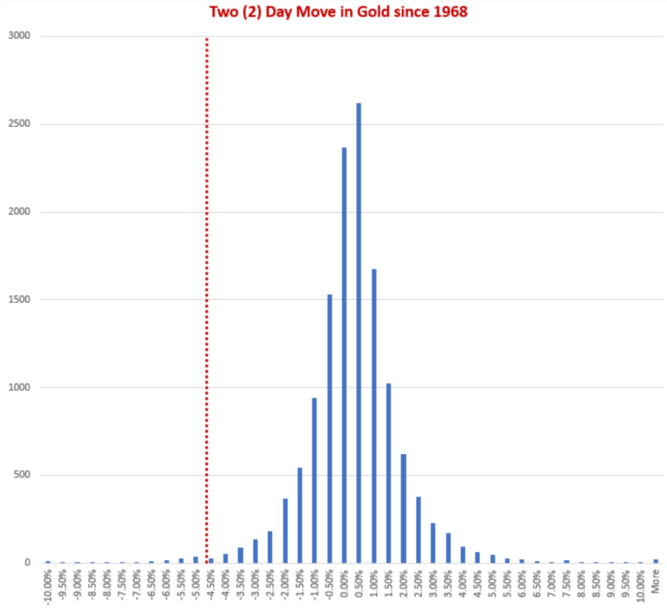



# Inflation is important

but it's a highly debated phenomenon in economics. Many economists maintain that moderate inflation levels are needed to drive consumption, assuming that higher levels of spending are crucial for economic growth

& stabilizing it is one of three objectives of the Federal Reserve who's decisions move the global financials markets

**Gold**, for example, saw **2 day drop of 4.67%** the day of & after the Fed mentioned tapering; i.e., **raising the Fed Rate in response to <u>Inflation</u>**




# Inflation is important

but it's a highly debated phenomenon in economics. Many economists maintain that moderate inflation levels are needed to drive consumption, assuming that higher levels of spending are crucial for economic growth

& stabilizing it is one of three objectives of the Federal Reserve who's decisions move the global financials markets

Gold, for example, saw 2 day drop of 4.67% the day of & after the Fed mentioned tapering; i.e., raising the Fed Rate in response to Inflation

**Math language**, a **2+ standard deviated move**

The purpose & goal of this Data Science project is to

# build a model to explain & understand the phenomenon of US Inflation

02

# Generated Deliverables

The power of API's

# Generated Deliverables

## Quandl

*Quandl is a marketplace for financial, economic and alternative data*

## Investing.com

*A financial platform & news website; one of the top 3 financial websites in the world*

## FRED

*Federal Reserve Economic Data ( FRED ) a database maintained by the Research division of the Federal Reserve Bank of St. Louis*

# Problem Identification
( cont. )

## I **shortlisted 19 variables** to determine their influence on Inflation

| Items | Reported | API | API Source | Comments |
|---|---|---|---|---|
| Inflation | Monthly | Quandl | U.S. Bureau of Labor Statistics | The target variable |
| Wages CPI | Monthly | FRED | U.S. Bureau of Labor Statistics | A component of the target variable |
| WTI | Daily | Quandl | CME | West Texas Intermediate - One of many commodities |
| Heating Oil | Daily | Investpy | Investing.com | One of many commodities |
| Copper | Daily | Investpy | Investing.com | One of many commodities |
| Sugar | Daily | Investpy | Investing.com | One of many commodities |
| Natural Gas | Daily | Investpy | Investing.com | One of many commodities |
| Cattle | Daily | Investpy | Investing.com | One of many commodities |
| Lean Hogs | Daily | Investpy | Investing.com | One of many commodities |
| Soybeans | Daily | Investpy | Investing.com | One of many commodities |
| Lumber | Daily | Investpy | Investing.com | One of many commodities |
| Capacity Utilization | Monthly | FRED | Board of Governors of the Federal Reserve | The % of resources used by corporations |
| Corn | Daily | Investpy | Investing.com | One of many commodities |
| M2 Velocity | Quarterly | FRED | Federal Reserve Bank of St. Louis | Movement of money; state of the economy proxy |
| GDP | Quarterly | FRED | U.S. Bureau of Economic Analysis | A proxy for the state of the economy |
| Wheat | Daily | Investpy | Investing.com | One of many commodities |
| PMI | Monthly | Quandl | Institute of Supply Management | Manufacturing PMI - A proxy for the economy |
| USD Index | Daily | Quandl | Intercontinental Exchange Inc | ( DXY ) Proxy for potentially importing inflation |
| Unemployment Rate | Monthly | Quandl | U.S. Bureau of Labor Statistics | A proxy for the state of the economy |
| Initial Jobless Claims | Weekly | Quandl | U.S. Employment and Training Administration | A proxy for the state of the economy |

# Target Variable

Commodities
Economic Data

## I Target variable | What we seek to understand

| Items | Reported | API | API Source | Comments |
|---|---|---|---|---|
| Inflation | Monthly | Quandl | U.S. Bureau of Labor Statistics | The target variable |

# Problem Identification
( cont. )

Target Variable

# Commodities

Economic Data

## I Commodities | Where Inflation typically shows itself

| Items | Reported | API | API Source | Comments |
|---|---|---|---|---|
| WTI | Daily | Quandl | CME | West Texas Intermediate - One of many commodities |
| Heating Oil | Daily | Investpy | Investing.com | One of many commodities |
| Copper | Daily | Investpy | Investing.com | One of many commodities |
| Sugar | Daily | Investpy | Investing.com | One of many commodities |
| Natural Gas | Daily | Investpy | Investing.com | One of many commodities |
| Cattle | Daily | Investpy | Investing.com | One of many commodities |
| Lean Hogs | Daily | Investpy | Investing.com | One of many commodities |
| Soybeans | Daily | Investpy | Investing.com | One of many commodities |
| Lumber | Daily | Investpy | Investing.com | One of many commodities |
| Corn | Daily | Investpy | Investing.com | One of many commodities |
| Wheat | Daily | Investpy | Investing.com | One of many commodities |

## Problem Identification
( cont. )

Target Variable
Commodities

# Economic Data

## I Economic Data | Variables to determine the health of the economy

| Items | Reported | API | API Source | Comments |
|---|---|---|---|---|
| Wages CPI | Monthly | FRED | U.S. Bureau of Labor Statistics | A component of the target variable |
| Capacity Utilization | Monthly | FRED | Board of Governors of the Federal Reserve | The % of resources used by corporations |
| M2 Velocity | Quarterly | FRED | Federal Reserve Bank of St. Louis | Movement of money; state of the economy proxy |
| GDP | Quarterly | FRED | U.S. Bureau of Economic Analysis | A proxy for the state of the economy |
| PMI | Monthly | Quandl | Institute of Supply Management | Manufacturing PMI - A proxy for the economy |
| USD Index | Daily | Quandl | Intercontinental Exchange Inc | ( DXY ) Proxy for potentially importing inflation |
| Unemployment Rate | Monthly | Quandl | U.S. Bureau of Labor Statistics | A proxy for the state of the economy |
| Initial Jobless Claims | Weekly | Quandl | U.S. Employment and Training Administration | A proxy for the state of the economy |

# Generated Deliverables

( cont. )

## Source Code

*This can be found at my GitHub account referenced at the end*

## Research Report

*Also can be found at my GitHub account referenced at the end*

## Presentation Report

*This one...*

03

# Data Pre-Processing

Split it up...

Data Pre-Processing

Data Cleaning

# Data Frames should talk to each other

- After pulling the data frame was **composed of variables with different lengths**

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14285 entries, 1946-01-01 to 2021-08-18
Data columns (total 19 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Wage CPI                 14276 non-null  float64
 1   WTI                      12071 non-null  float64
 2   Heating Oil              13070 non-null  float64
 3   Copper                   10423 non-null  float64
 4   Sugar                    13070 non-null  float64
 5   Natural Gas              9898 non-null   float64
 6   Cattle                   13067 non-null  float64
 7   Lean Hogs                13072 non-null  float64
 8   Soybeans                 9982 non-null   float64
 9   Lumber                   13072 non-null  float64
 10  Capacity Utilization     14016 non-null  float64
 11  Corn                     13069 non-null  float64
 12  M2 Velocity              14134 non-null  float64
 13  GDP                      14278 non-null  float64
 14  Wheat                    9984 non-null   float64
 15  PMI                      14264 non-null  float64
 16  USD Index                11256 non-null  float64
 17  Unemployment Rate        14264 non-null  float64
 18  Initial Jobless Claims   14013 non-null  float64
dtypes: float64(19)
memory usage: 2.2 MB
```

# Data Pre-Processing

Data Cleaning

## Data Frames should talk to each other

- After pulling the data frame was composed of variables with different lengths
  - **Natural Gas being the constraint**
  - **Used forward fill**

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9735 entries, 1991-04-18 to 2021-08-18
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Wage CPI                9735 non-null   float64
 1   WTI                     9735 non-null   float64
 2   Heating Oil             9735 non-null   float64
 3   Copper                  9735 non-null   float64
 4   Sugar                   9735 non-null   float64
 5   Natural Gas             9735 non-null   float64
 6   Cattle                  9735 non-null   float64
 7   Lean Hogs               9735 non-null   float64
 8   Soybeans                9735 non-null   float64
 9   Lumber                  9735 non-null   float64
 10  Capacity Utilization    9735 non-null   float64
 11  Corn                    9735 non-null   float64
 12  M2 Velocity             9735 non-null   float64
 13  GDP                     9735 non-null   float64
 14  Wheat                   9735 non-null   float64
 15  PMI                     9735 non-null   float64
 16  USD Index               9735 non-null   float64
 17  Unemployment Rate       9735 non-null   float64
 18  Initial Jobless Claims  9735 non-null   float64
dtypes: float64(19)
memory usage: 1.5 MB
```

# Data Pre-Processing

## Data Cleaning ( cont. )

# Data Frames should talk to each other ( cont. )

- Different lengths
- **Cut the data to April 1991**

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9735 entries, 1991-04-18 to 2021-08-18
Data columns (total 19 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Wage CPI                 9735 non-null   float64
 1   WTI                      9735 non-null   float64
 2   Heating Oil              9735 non-null   float64
 3   Copper                   9735 non-null   float64
 4   Sugar                    9735 non-null   float64
 5   Natural Gas              9735 non-null   float64
 6   Cattle                   9735 non-null   float64
 7   Lean Hogs                9735 non-null   float64
 8   Soybeans                 9735 non-null   float64
 9   Lumber                   9735 non-null   float64
 10  Capacity Utilization     9735 non-null   float64
 11  Corn                     9735 non-null   float64
 12  M2 Velocity              9735 non-null   float64
 13  GDP                      9735 non-null   float64
 14  Wheat                    9735 non-null   float64
 15  PMI                      9735 non-null   float64
 16  USD Index                9735 non-null   float64
 17  Unemployment Rate        9735 non-null   float64
 18  Initial Jobless Claims   9735 non-null   float64
dtypes: float64(19)
memory usage: 1.5 MB
```

# Data Pre-Processing

Data Cleaning ( cont. )

## Data Frames should talk to each other ( cont. )

- Different lengths
- Cut the Data
- **Concatenated with Inflation**
  - **Using a forward fill**
  - **Only 321 observations**

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 321 entries, 1991-04-30 to 2021-07-31
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Inflation              321 non-null    float64
 1   Wage CPI               321 non-null    float64
 2   WTI                    321 non-null    float64
 3   Heating Oil            321 non-null    float64
 4   Copper                 321 non-null    float64
 5   Sugar                  321 non-null    float64
 6   Natural Gas            321 non-null    float64
 7   Cattle                 321 non-null    float64
 8   Lean Hogs              321 non-null    float64
 9   Soybeans               321 non-null    float64
 10  Lumber                 321 non-null    float64
 11  Capacity Utilization   321 non-null    float64
 12  Corn                   321 non-null    float64
 13  M2 Velocity            321 non-null    float64
 14  GDP                    321 non-null    float64
 15  Wheat                  321 non-null    float64
 16  PMI                    321 non-null    float64
 17  USD Index              321 non-null    float64
 18  Unemployment Rate      321 non-null    float64
 19  Initial Jobless Claims 321 non-null    float64
dtypes: float64(20)
memory usage: 52.7 KB
```

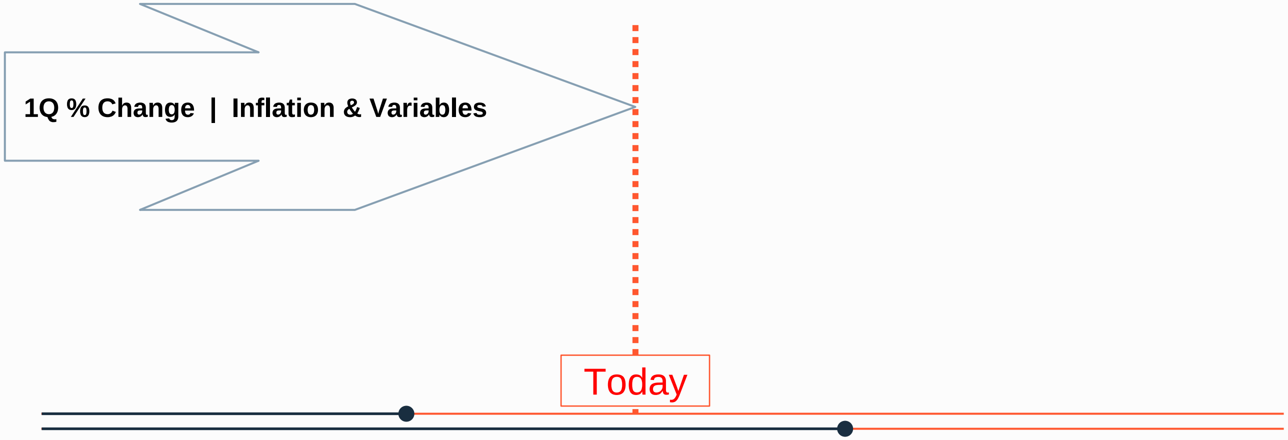
| Items | Reported | API | API Source | Comments |
|---|---|---|---|---|
| Inflation | Monthly | Quandl | U.S. Bureau of Labor Statistics | The target variable |

## Data Pre-Processing

Exploratory Data Analysis

# Investigating the Time Relationships

- **Quarter on Quarter ( for all )**
    - Compared a quarterly change on Variables & Inflation
- Month on Month ( for all )
- Quarter on Quarter for Variables ( past ) & Inflation ( forwards )
- Quarter on Quarter w/ Rolling Averages

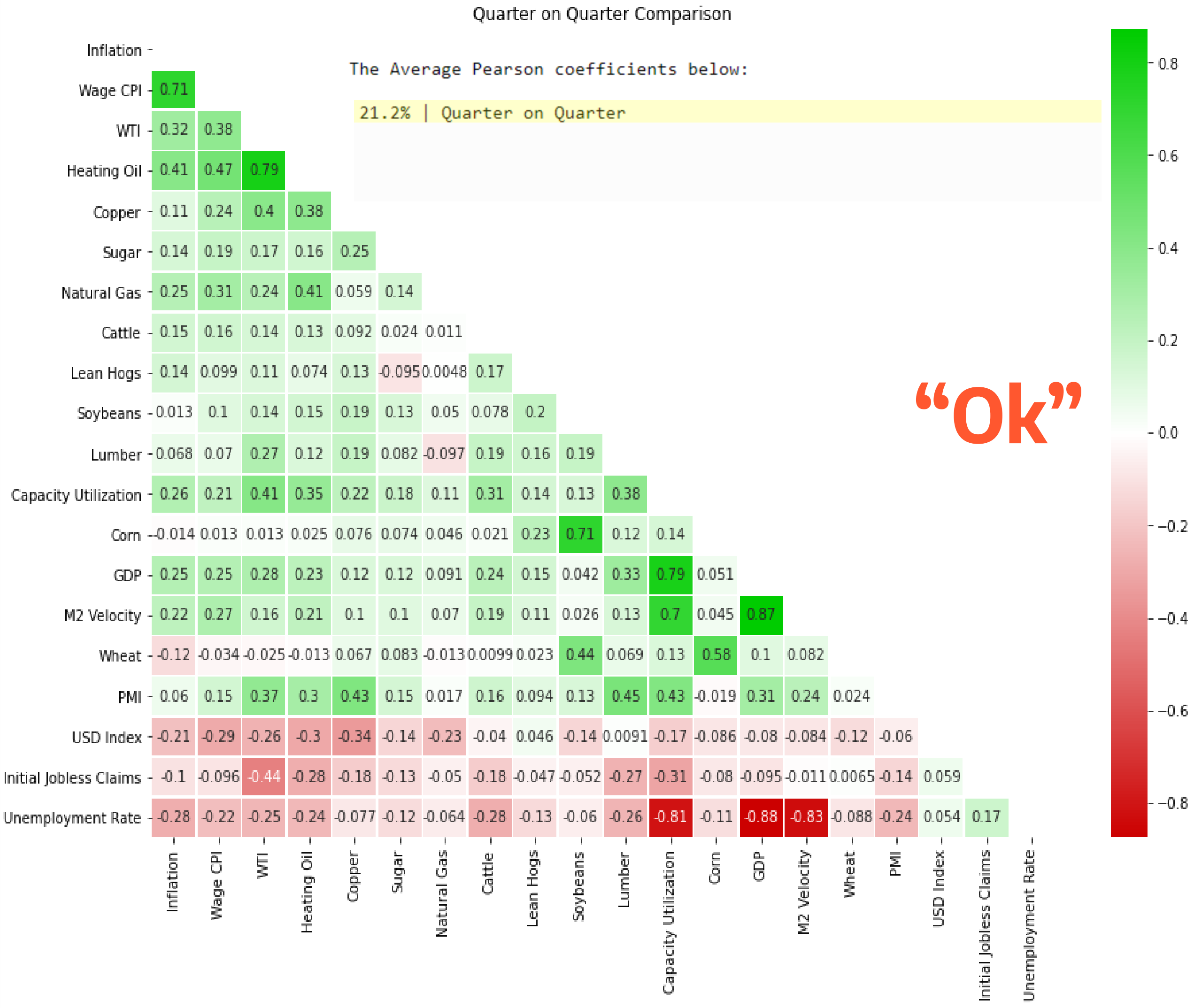


# Data Pre-Processing

**Exploratory Data Analysis**

**Quarter on Quarter ( for all )**

**Feature Correlation Heat Maps with the Pearson correlation coefficients**

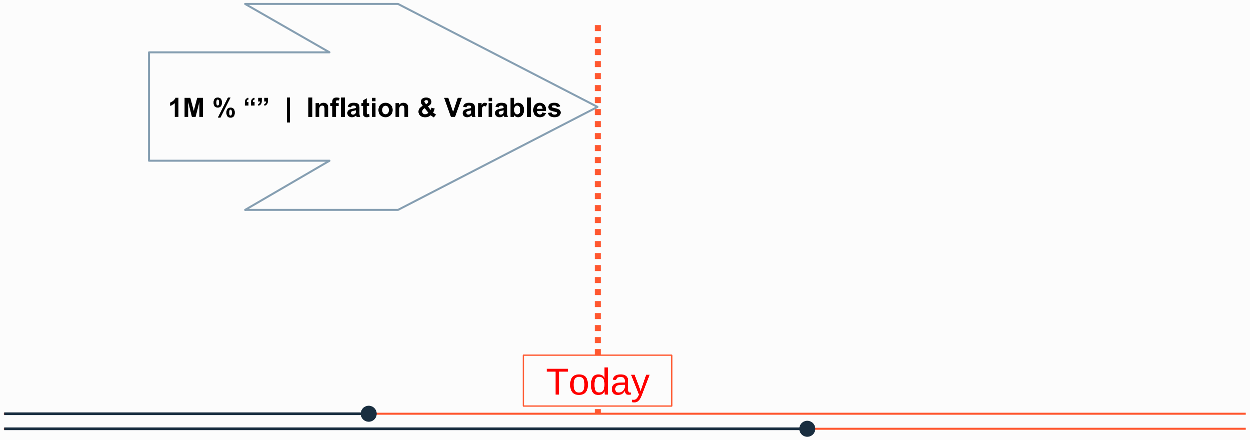


# Data Pre-Processing

### Exploratory Data Analysis ( cont. )

## Investigating the Time Relationships ( cont. )

- Quarter on Quarter ( for all )
- **Month on Month ( for all )**
    - The same as the previous but looked at monthly change
- Quarter on Quarter for Variables ( past ) & Inflation ( forwards )
- Quarter on Quarter w/ Rolling Averages



# Data Pre-Processing

**Exploratory Data Analysis**

**Month on Month ( for all )**

**Feature Correlation Heat Maps with the Pearson correlation coefficients ( cont. )**

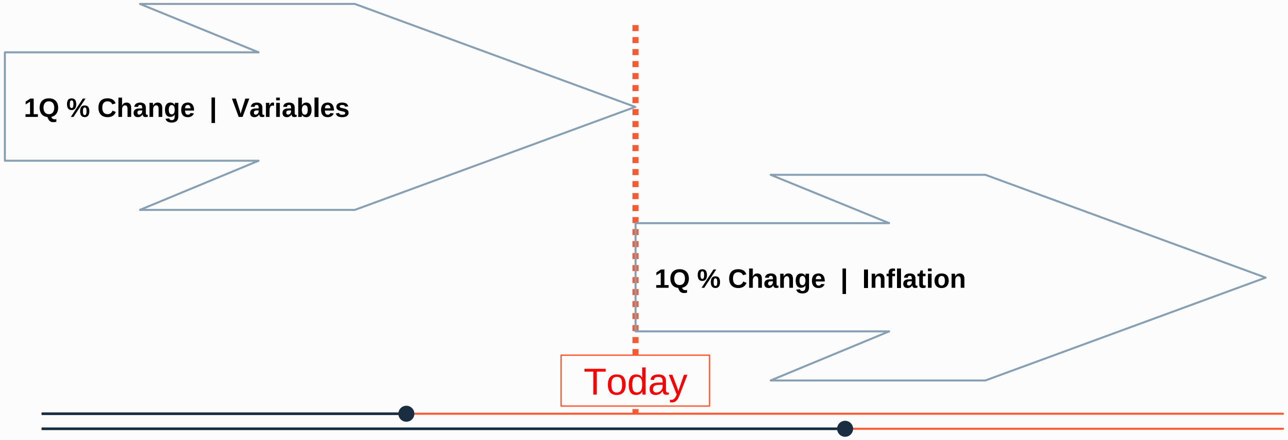

**"Worse"**

# Data Pre-Processing

Exploratory Data Analysis ( cont. )

## Investigating the Time Relationships ( cont. )

- Quarter on Quarter ( for all )
- Month on Month ( for all )
- **Q on Q for Variables ( past ) & Inflation ( forwards )**
  - Looked at a previous 1 quarter change for variables to a 1 quarter change in Inflation in the future
- Quarter on Quarter w/ Rolling Averages



# Data Pre-Processing

**Exploratory Data Analysis**

**Q on Q for Variables ( past ) & Inflation ( forwards )**

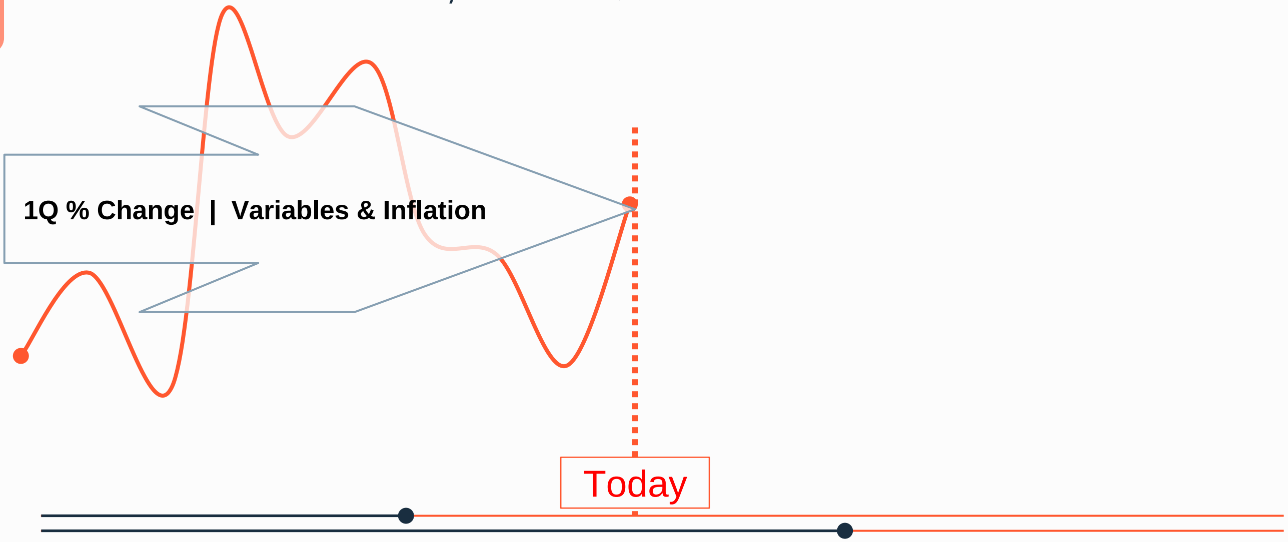
**Feature Correlation Heat Maps with the Pearson correlation coefficients ( cont. )**

# Data Pre-Processing

Exploratory Data Analysis ( cont. )

## Investigating the Time Relationships ( cont. )

- Quarter on Quarter ( for all )
- Month on Month ( for all )
- Q on Q for Variables ( past ) & Inflation ( forwards )
- **Quarter on Quarter w/ Rolling Averages**
    - Similar to # 1 albeit used a rolling average for those that were reported more often than once a Quarter as a Variable “may have had” a bad week or day when the Quarter ended



# Data Pre-Processing

**Exploratory Data Analysis**

**Quarter on Quarter w/ Rolling Averages**

**Feature Correlation Heat Maps with the Pearson correlation coefficients ( cont. )**

# Data Pre-Processing

Exploratory Data Analysis ( cont. )

## Investigating the Time Relationships ( cont. )

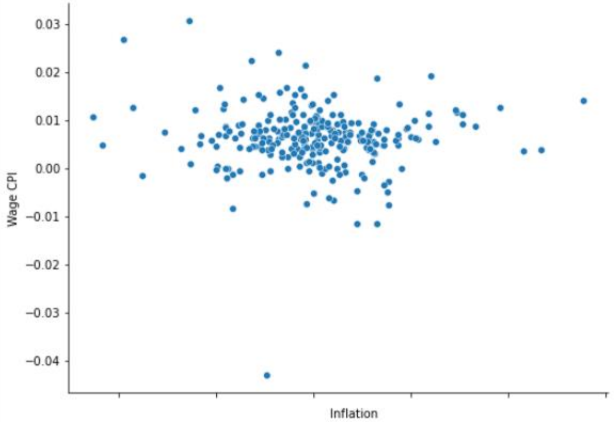
- **Scraping is also a consideration to be taken in.**
- I did not scrape on the variables; something discussed in "Next Steps"
- I did scrape, however, on Inflation; this was done on a three (3) standard deviated move whereby they were dropped; representing 1.5-2.8% of the total DataFrame

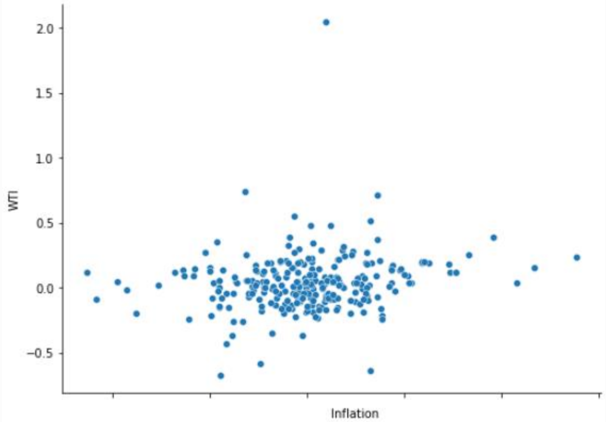
# Data Pre-Processing

**Exploratory Data Analysis ( cont. )**

## Investigating the Time Relationships ( cont. )

- Scraping is also a consideration to be taken in.
- **I did not scrape on the variables; something discussed in "Next Steps"**
- I did scrape, however, on Inflation; this was done on a three (3) standard deviated move whereby they were dropped; representing 1.5-2.8% of the total DataFrame



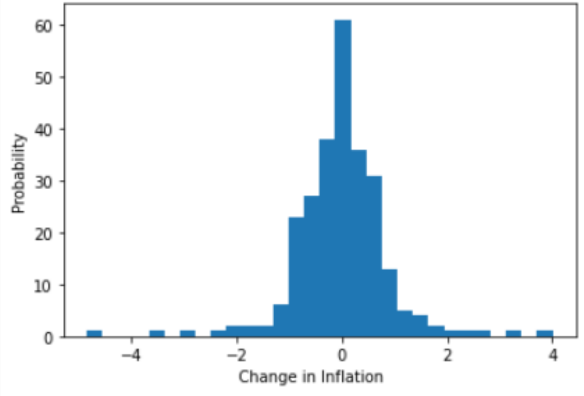



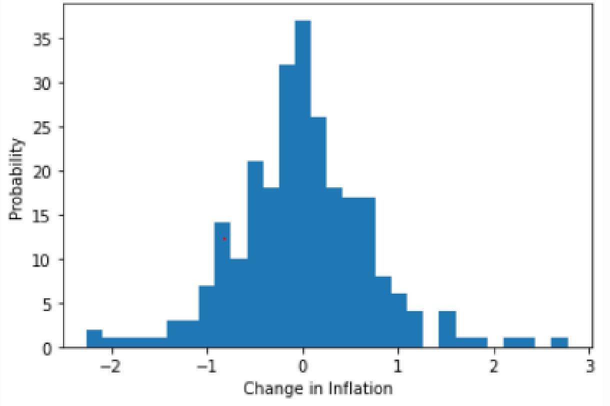
# Data Pre-Processing

Exploratory Data Analysis ( cont. )

## Investigating the Time Relationships ( cont. )

- Scraping is also a consideration to be taken in.
- I did not scrape on the variables; something discussed in "Next Steps"
- **I did scrape, however, on Inflation; this was done on a three (3) standard deviated move whereby they were dropped; representing 1.5-2.8% of the total DataFrame**



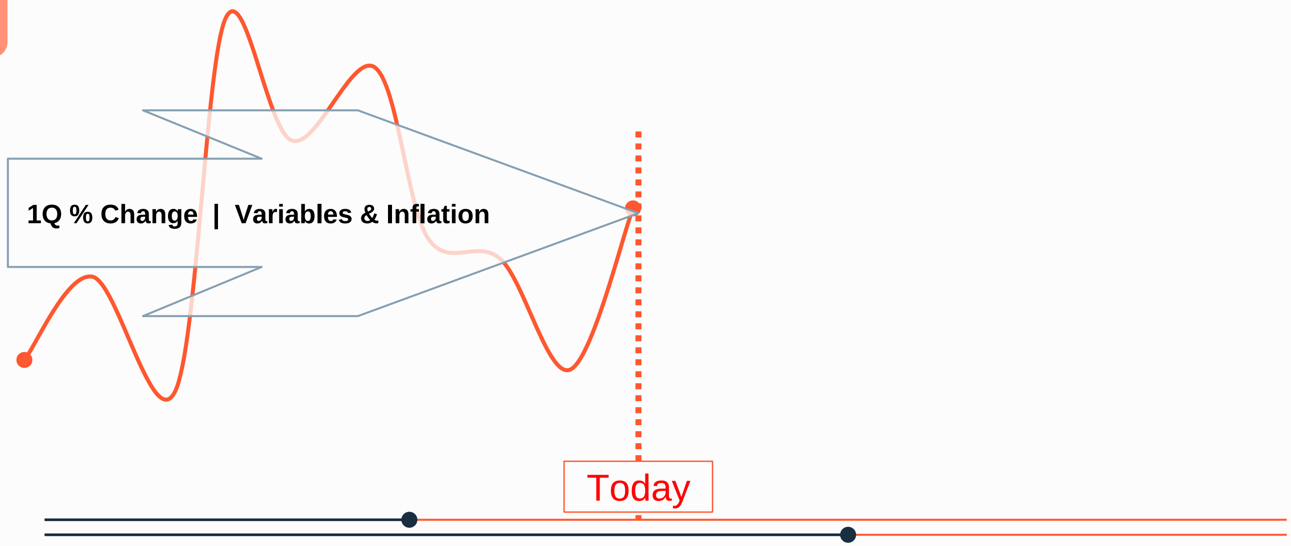



# Data Pre-Processing

Pre-Processing

## Splitting & Scaling

- **Chosen data frame**
  - The Quarter on Quarter w/ Rolling Averages was chosen
- Train, Test Split
- Scaling

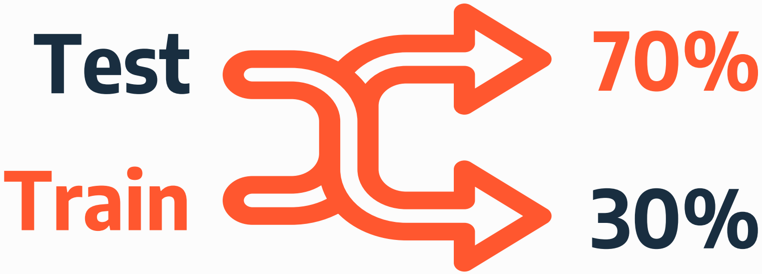


# Data Pre-Processing

Pre-Processing
( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- **Train, Test Split**
    - The data was then split for Training & Testing to be sent to different Scaling Approaches
- Scaling

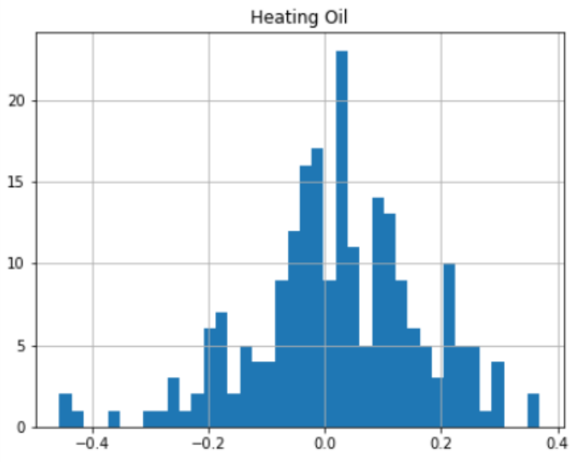


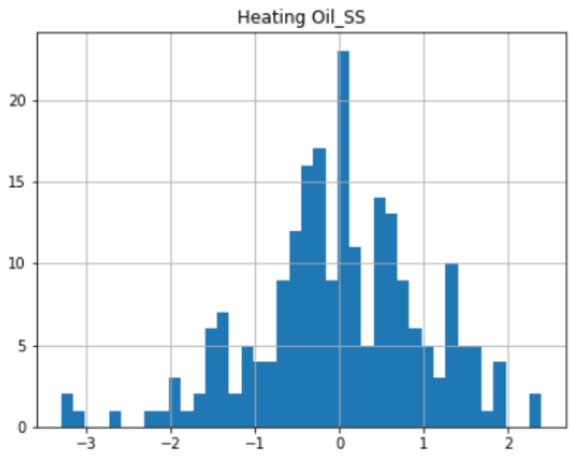
# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried to “normalize” them:
    - Standard Scaling ( SS )
    - MinMax Scaling ( MM )
    - Log Transformation ( LG )



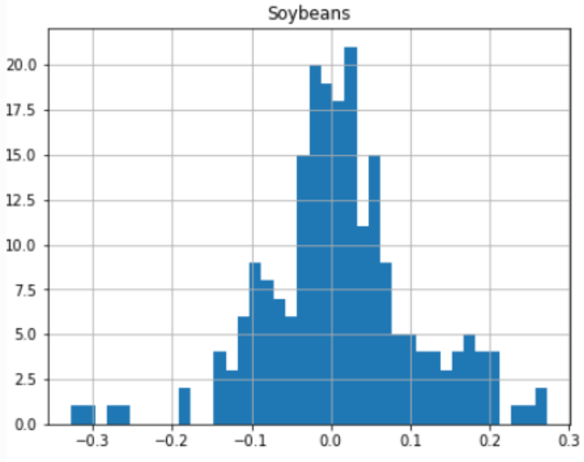



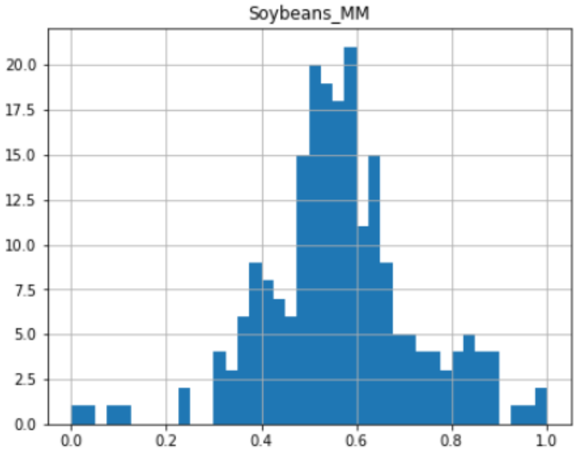
# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried to “normalize” them :
    - Standard Scaling ( SS )
    - MinMax Scaling ( MM )
    - Log Transformation ( LG )



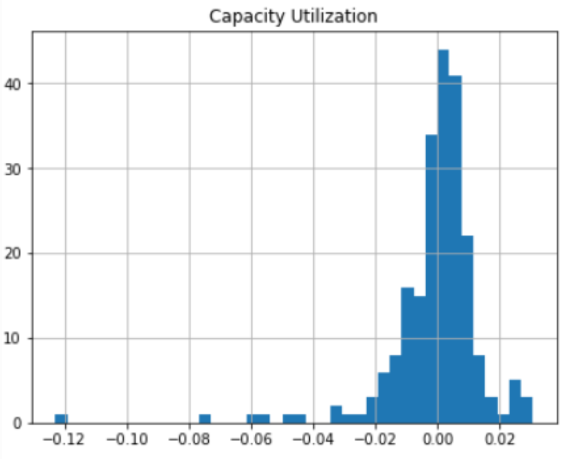



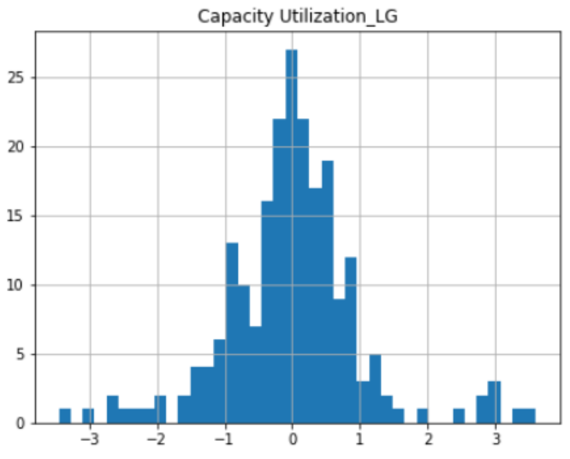
# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried to “normalize” them :
        - Standard Scaling ( SS )
        - MinMax Scaling ( MM )
        - Log Transformation ( LG )



# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried to “normalize” them :
        - Standard Scaling ( SS )
        - MinMax Scaling ( MM )
        - Log Transformation ( LG )

| | Wages CPI_SS | WTI_SS |
|---|---|---|
| count | 2.180000e+02 | 2.180000e+02 |
| mean | -4.838128e-18 | 2.750094e-17 |
| std | 1.002301e+00 | 1.002301e+00 |
| min | -6.977019e+00 | -3.783391e+00 |
| 25% | -2.671202e-01 | -5.665365e-01 |
| 50% | 1.153214e-01 | -3.959852e-02 |
| 75% | 4.280369e-01 | 6.677299e-01 |
| max | 3.074376e+00 | 2.742497e+00 |

| Wages CPI_MM | WTI_MM |
|---|---|
| 218.000000 | 218.000000 |
| 0.694134 | 0.579751 |
| 0.099718 | 0.153589 |
| 0.000000 | 0.000000 |
| 0.667559 | 0.492937 |
| 0.705608 | 0.573683 |
| 0.736719 | 0.682071 |
| 1.000000 | 1.000000 |

| Wages CPI_LG | WTI_LG |
|---|---|
| 2.180000e+02 | 2.180000e+02 |
| -2.340126e-16 | -1.018553e-17 |
| 1.002301e+00 | 1.002301e+00 |
| -4.203779e+00 | -3.308051e+00 |
| -3.922100e-01 | -6.014282e-01 |
| 2.665979e-02 | -8.488108e-02 |
| 3.947888e-01 | 6.501558e-01 |
| 4.675375e+00 | 3.071756e+00 |

# Data Pre-Processing

**Pre-Processing ( cont. )**

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried:
    - Standard Scaling ( SS )
    - MinMax Scaling ( MM )
    - Log Transformation ( LG )
  - MM posted poor results; thus removed

```
R² results for nothing scaled below
                    Test 0.254 ( nothing scaled )

R² results for X & y scaled below
SS Train | 0.3966    Test 0.2796
MM Train | 0.0424    Test -0.1085
LG Train | 0.4149    Test -23.8319

R² results for X only scaled below
SS Train | 0.4185    Test 0.254
MM Train | -0.2444   Test -0.0533
LG Train | 0.4142    Test -23.4693
```

```
MAE results for nothing scaled below
                    Test 0.563 ( nothing scaled )

MAE results for X & y scaled below
SS Train | 0.5376    Test 0.684
MM Train | 0.0811    Test 0.0943
LG Train | 0.5478    Test 1.6306

MAE results for X only scaled below
SS Train | 0.4312    Test 0.563
MM Train | 0.6711    Test 0.6112
LG Train | 0.4381    Test 1.2897
```

```
MSE results for nothing scaled below
                    Test 0.7556 ( nothing scaled )

MSE results for X & y scaled below
SS Train | 0.6034    Test 0.8602
MM Train | 0.0105    Test 0.0146
LG Train | 0.5851    Test 30.4687

MSE results for X only scaled below
SS Train | 0.3727    Test 0.571
MM Train | 0.7976    Test 0.8061
LG Train | 0.3755    Test 18.7277
```

# Data Pre-Processing

**Pre-Processing ( cont. )**

## Splitting & Scaling ( cont. )

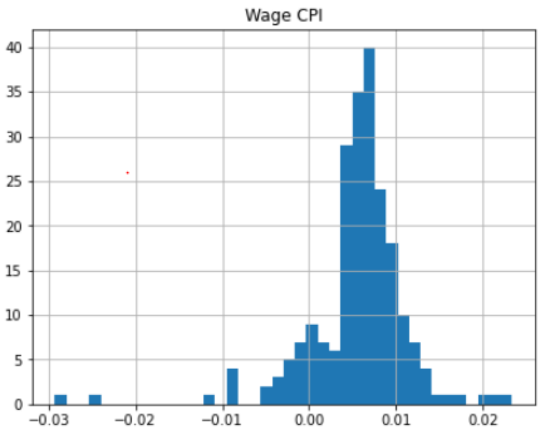
- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried:
        - Standard Scaling ( SS )
        - ~~MinMax Scaling ( MM )~~
        - Log Transformation ( LG )
    - MM posted poor results; thus removed
    - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.

# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried:
        - Standard Scaling ( SS )
        - ~~MinMax Scaling ( MM )~~
        - Log Transformation ( LG )
    - MM posted poor results; thus removed
    - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.
    - The following were sent to LG:
        - Wage CPI

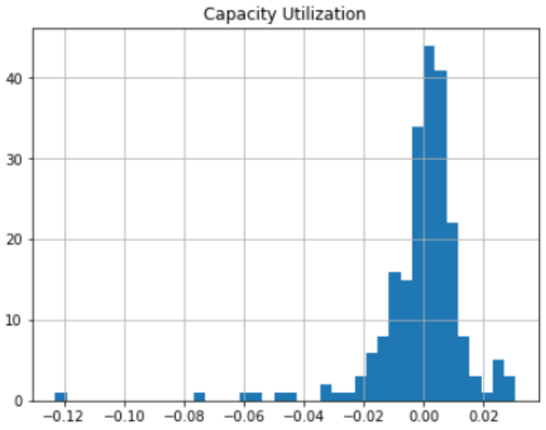


# Data Pre-Processing

**Pre-Processing ( cont. )**

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried:
        - Standard Scaling ( SS )
        - ~~MinMax Scaling ( MM )~~
        - Log Transformation ( LG )
    - MM posted poor results; thus removed
    - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.
    - The following were sent to LG:
        - Wage CPI
        - Capacity Utilization

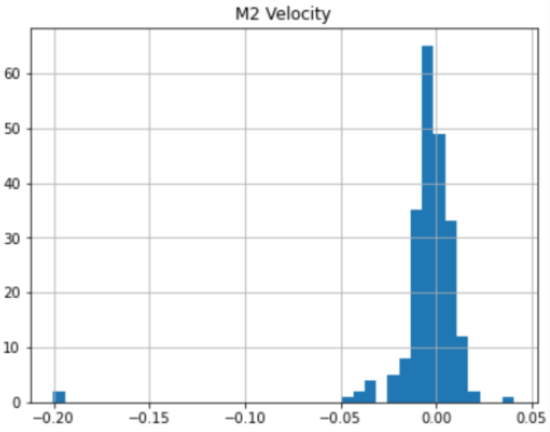


# Data Pre-Processing

**Pre-Processing ( cont. )**

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried:
    - Standard Scaling ( SS )
    - ~~MinMax Scaling ( MM )~~
    - Log Transformation ( LG )
  - MM posted poor results; thus removed
  - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.
  - The following were sent to LG:
    - Wage CPI
    - Capacity Utilization
    - M2 Velocity

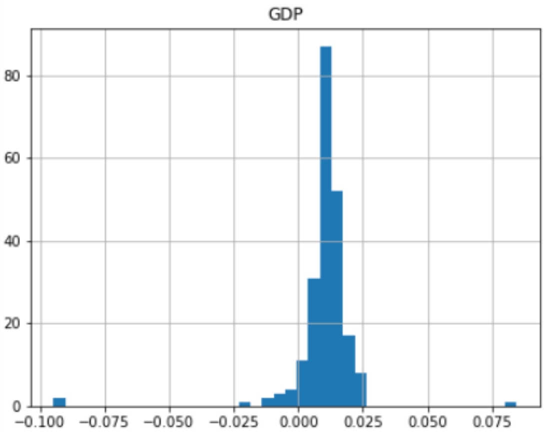


# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried:
    - Standard Scaling ( SS )
    - ~~MinMax Scaling ( MM )~~
    - Log Transformation ( LG )
  - MM posted poor results; thus removed
  - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.
  - The following were sent to LG:
    - Wage CPI
    - Capacity Utilization
    - M2 Velocity
    - GDP

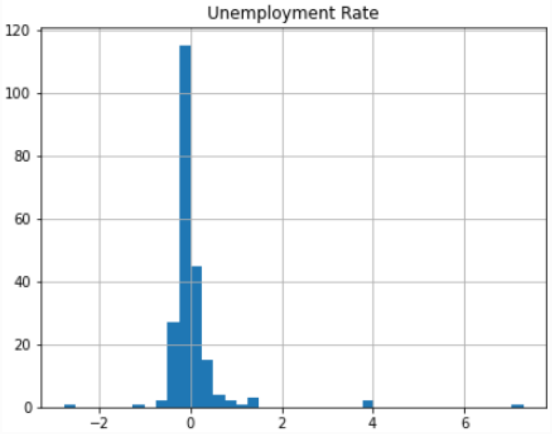


# Data Pre-Processing

### Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried:
    - Standard Scaling ( SS )
    - ~~MinMax Scaling ( MM )~~
    - Log Transformation ( LG )
  - MM posted poor results; thus removed
  - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.
  - The following were sent to LG:
    - Wage CPI
    - Capacity Utilization
    - M2 Velocity
    - GDP
    - Unemployment Rate



# Data Pre-Processing

Pre-Processing ( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried:
        - Standard Scaling ( SS )
        - ~~MinMax Scaling ( MM )~~
        - Log Transformation ( LG )
    - MM posted poor results; thus removed
    - As SS & LG posted the best result, variables were chosen to sent to either a SS or LG while keeping the y variable ( Inflation ) unscaled.
    - The following were sent to LG:
        - Wage CPI
        - Capacity Utilization
        - M2 Velocity
        - GDP
        - Unemployment Rate
        - Initial Jobless Claims

# Data Pre-Processing

**Pre-Processing ( cont. )**

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
    - 3 scaling approaches were tried:
        - Standard Scaling ( SS )
        - ~~MinMax Scaling ( MM )~~
        - Log Transformation ( LG )
    - MM posted poor results; thus removed
    - As SS & LG posted the best result, “”
    - The results of these below

$R^2$ results for nothing scaled below
Test 0.254 ( nothing scaled )

$R^2$ results for X & y scaled below
SS Train | 0.3966 Test 0.2796
~~MM Train | 0.0424 Test -0.1085~~
LG Train | 0.4149 Test -23.8319

$R^2$ results for X only scaled below
SS Train | 0.4185 Test 0.254
~~MM Train | -0.2444 Test -0.0533~~
LG Train | 0.4142 Test -23.4693

$R^2$ results for the LG & SS combination below
SS Train | 0.4067 Test -22.811

MAE results for nothing scaled below
Test 0.563 ( nothing scaled )

MAE results for X & y scaled below
SS Train | 0.5376 Test 0.684
~~MM Train | 0.0811 Test 0.0943~~
LG Train | 0.5478 Test 1.6306

MAE results for X only scaled below
SS Train | 0.4312 Test 0.563
~~MM Train | 0.6711 Test 0.6112~~
LG Train | 0.4381 Test 1.2897

MAE results for the LG & SS combination below
SS Train | 0.4377 Test 1.2751

MSE results for nothing scaled below
Test 0.7556 ( nothing scaled )

MSE results for X & y scaled below
SS Train | 0.6034 Test 0.8602
~~MM Train | 0.0105 Test 0.0146~~
LG Train | 0.5851 Test 30.4687

MSE results for X only scaled below
SS Train | 0.3727 Test 0.571
~~MM Train | 0.7976 Test 0.8061~~
LG Train | 0.3755 Test 18.7277

MSE results for the LG & SS combination below
SS Train | 0.3803 Test 18.2239

# Data Pre-Processing

Pre-Processing
( cont. )

## Splitting & Scaling ( cont. )

- Chosen data frame
- Train, Test Split
- **Scaling**
  - 3 scaling approaches were tried:
    - Standard Scaling ( SS )
    - ~~MinMax Scaling ( MM )~~
    - Log Transformation ( LG )
  - MM posted poor results; thus removed
  - As SS & LG posted the best result, “”
  - The results of these below
  - The resulting x5 Data frames went to a Random Forest Model

```
  R² results for X & y scaled below
1 SS Train | 0.3966   Test 0.2796
2 LG Train | 0.4149   Test -23.8319

  R² results for X only scaled below
3 SS Train | 0.4185   Test 0.254
4 LG Train | 0.4142   Test -23.4693

  R² results for the LG & SS combination below
5 SS Train | 0.4067   Test -22.811
```

04

# Model Description

The Random Forest

# The Random Forest Model was then used

with the goal of determining what variables best explain & understand Inflation

05

# Model Findings

What’s moving Inflation¿

# Model Findings

## Where's Inflation coming from?

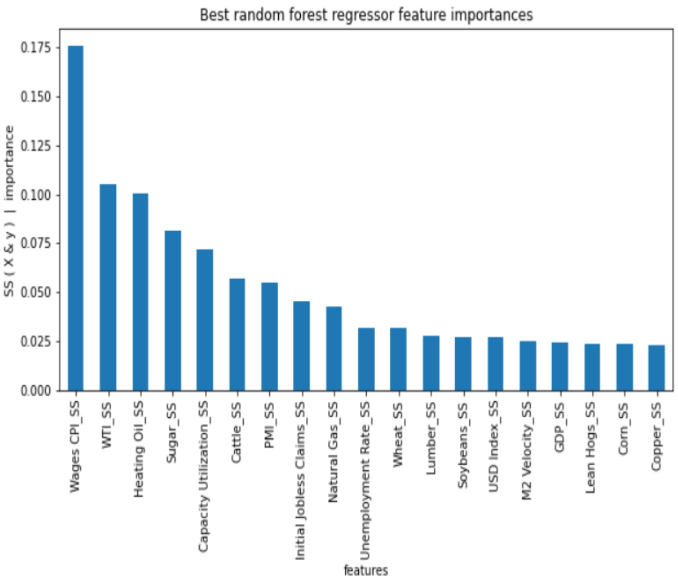
- **The standard process was taken on x5**
  - Grid Search
  - Random Forest
  - Hyperparameter search using Grid Search CV

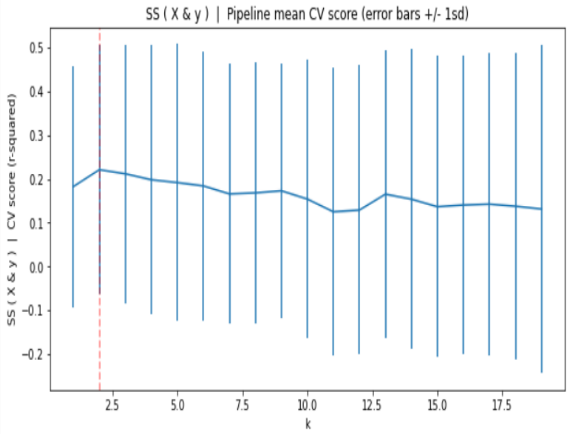
# Model Findings

( cont. )

## Where's Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on 5; other variables discounted the performance



# Model Findings

( cont. )

## Where’s Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on all; other variables discounted the performance
  - In the end, the SS approach on both X & y presented the best results with these two variables

```
R² results for X & y scaled below
SS Train | 0.2924    Test 0.424
LG Train | 0.2815    Test 0.3673

R² results for X only scaled below
SS Train | 0.2924    Test 0.3489
LG Train | 0.2778    Test 0.2979

R² results for the LG & SS combination
SS Train | 0.284    Test 0.3761
```

```
MAE results for X & y scaled below
SS Train | 0.5639    Test 0.5811
LG Train | 0.5727    Test 0.598

MAE results for X only scaled below
SS Train | 0.4515    Test 0.6127
LG Train | 0.4572    Test 0.6272

MAE results for the LG & SS combination
SS Train | 0.4572    Test 0.6272
```

```
MSE results for X & y scaled below
SS Train | 0.7076    Test 0.6877
LG Train | 0.7185    Test 0.7763

MSE results for X only scaled below
SS Train | 0.4536    Test 0.7774
LG Train | 0.4629    Test 0.8615

MSE results for the LG & SS combination
SS Train | 0.4629    Test 0.8615
```

# Model Findings

( cont. )

## Where's Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on all; other variables discounted the performance
  - In the end, the SS approach on both X & y presented the best results with these two variables
  - & showed that the process presented notable improvement from where we started
    - After rolling averages on the unscaled 19 Variables

```
A 17.0 bps increase in R²; 66.94 % increase.

A 1.81 bps increase in MAE.

A -6.79 bps decrease in MSE.
```

# Model Findings

( cont. )

## Where's Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on all; other variables discounted the performance
  - In the end, the SS approach on both X & y presented the best results with these two variables
  - & showed that the process presented notable improvement from where we started
    - After rolling averages on the unscaled 19 Variables
    - After rolling averages on the SS X & y scaled 19 Variables

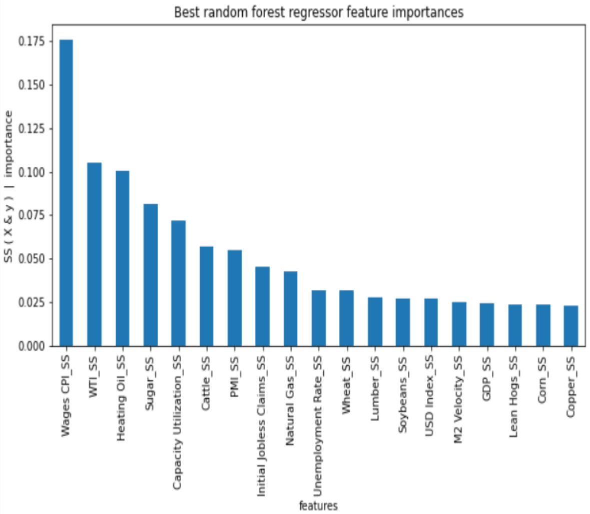
```
A 14.44 bps increase in R²; 51.64 % increase.

A -10.29 bps decrease in MAE.

A -17.24 bps decrease in MSE.
```

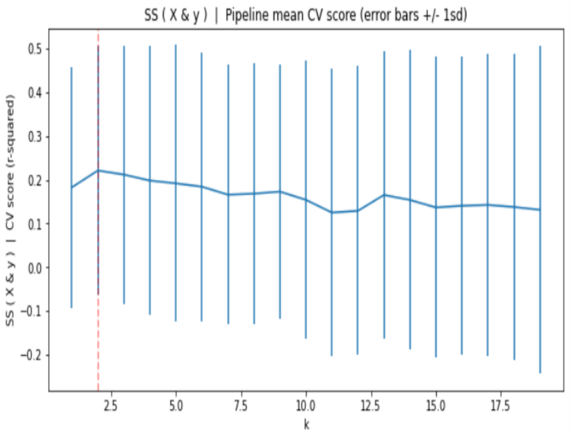
# Model Findings

( cont. )

## Where’s Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on all; other variables discounted the performance
  - In the end, the SS approach on both X & y presented the best results with these two variables
  - & showed that the process presented notable improvement from where we started
    - After rolling averages on the unscaled 19 Variables
    - After rolling averages on the SS X & y scaled 19 Variables
    - Isolated the importance to **Wages & WTI**



# Model Findings

( cont. )

## Where’s Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on all; other variables discounted the performance
  - In the end, the SS approach on both X & y presented the best results
  - & showed that the process presented notable improvement from where we started
  - So the verdict is that when you use **these two variables alone you best position yourself to understand Inflation**.

# Model Findings

( cont. )

## Where's Inflation coming from? ( cont. )

- The standard process was taken on x5
- **The results**
  - Random Forest showed a ubiquitous 1st & 2nd place to Wages CPI & WTI respectively on all; other variables discounted the performance
  - In the end, the SS approach on both X & y presented the best results
  - & showed that the process presented notable improvement from where we started
  - So the verdict is that when you use these two variables alone you best position yourself to understand Inflation.
  - While the Wages CPI is reported with Inflation itself, **we will borrow some words** to explain it on something that moves every day

*The wise words of Bill Clintons' advisor to his 1992 political campaign*

" "

*- James Carville*

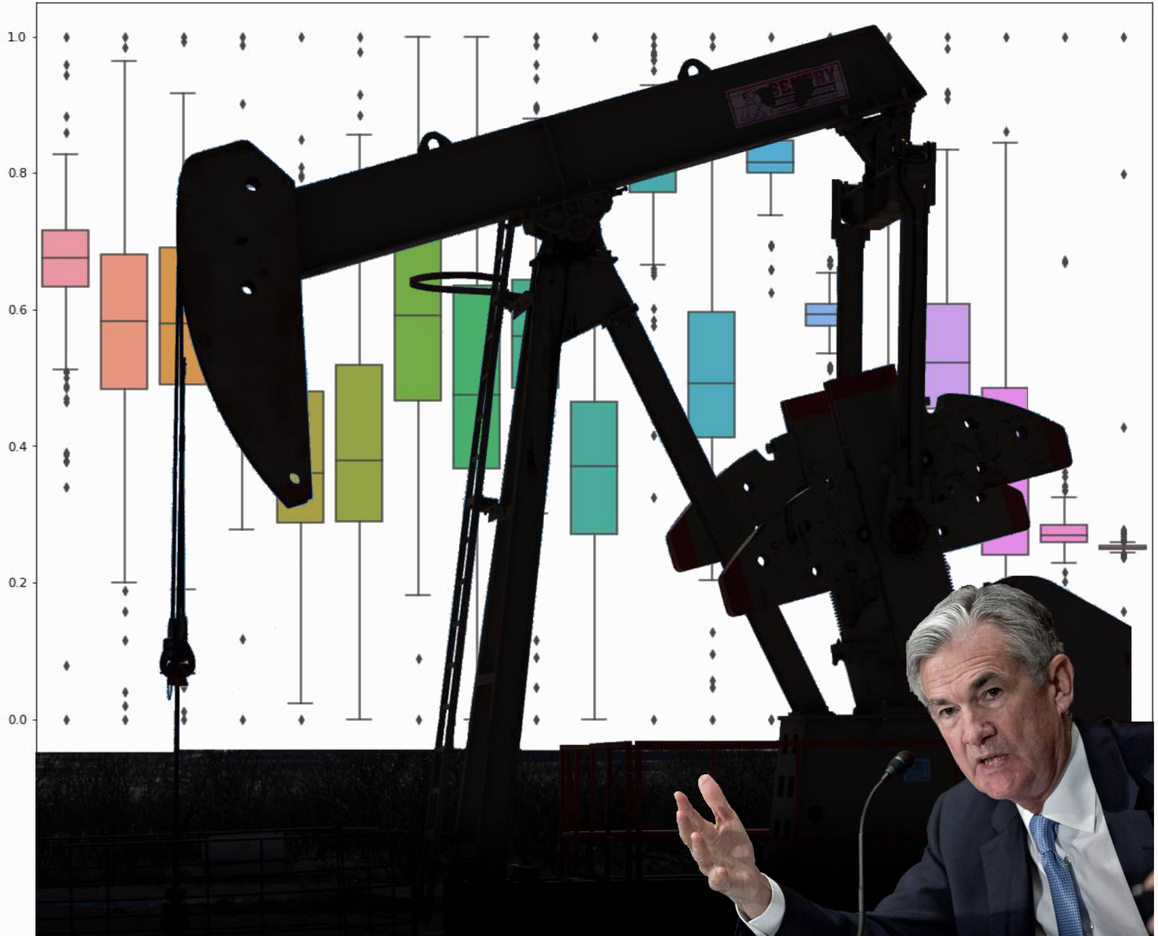
*The wise words of Bill Clintons' advisor to his 1992 political campaign*

# "It's the economy, stupid"

*- James Carville*

# Borrowed words...

# "It's Oil, silly"




# Our Conclusion

06

# Next Steps

Keep going

# Next Steps

## Variables not included

- **Steel**
  - 2008 was the furthest I could pull

# Next Steps

## Variables not included

- Steel
- **Gasoline**
    - 2005 was the furthest I could pull

# Next Steps

## Variables not included

- Steel
- Gasoline
- **US Wages Hourly Earnings**
    - Limited Data as well

# Next Steps

## Variables not included

- Steel
- Gasoline
- US Wages Hourly Earnings
- **US Dollar Index: Broad, Goods & Services**
    - Only goes until 2006

# Next Steps

## Variables not included

- Steel
- Gasoline
- US Wages Hourly Earnings
- US Dollar Index: Broad, Goods & Services
- **Growth in M2**
    - Possible overlap with M2 Velocity

# Next Steps
( cont. )

**More attention may be applicable to the below:**

- **Get more data**

# Next Steps
( cont. )

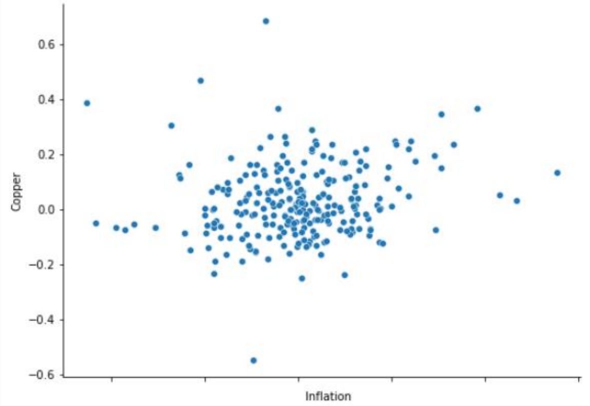
## More attention may be applicable to the below:

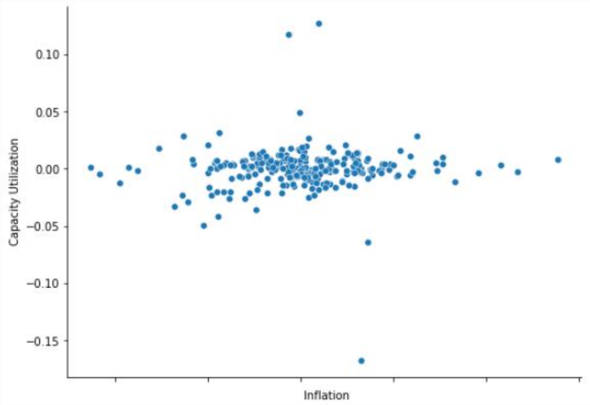
- Get more data
- **The SS & LG Divide**
  - Reassess the Variables which were chosen in the SS & LG divide; discussed in Pre-processing

# Next Steps
( cont. )

## More attention may be applicable to the below:

- Get more data
- The SS & LG Divide
- **Scrape Variables**
    - Winsorizing way present better results



# Next Steps
( cont. )

## More attention may be applicable to the below:

- Get more data
- The SS & LG Divide
- Scrape Variables
- **Predict Wages CPI Itself**
    - Develop a model to remove ourselves from the US govt's reporting

# Next Steps
( cont. )

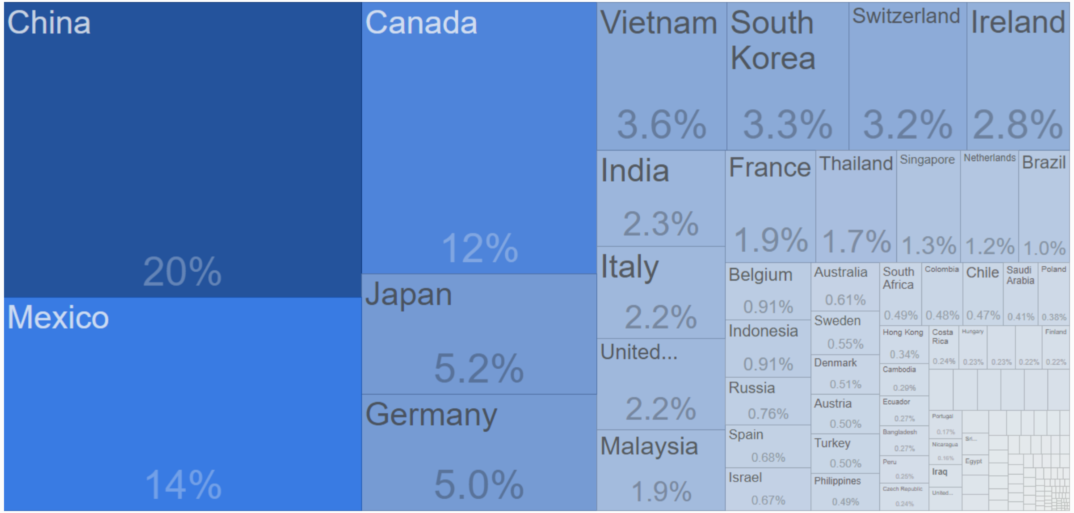
## More attention may be applicable to the below:

- Get more data
- The SS & LG Divide
- Scrape Variables
- Predict Wages CPI Itself
- **Build a Better Imported / Exported USD**
  - **The DXY doesn't correctly address the US Imports or Exports Inflation** as it's weighting is a weighted geometric mean of the:
    - Eurozone ( EUR ),
    - Japan ( JPY ),
    - United Kingdom ( GBP ),
    - Canada ( CAD ),
    - Sweden ( SEK ) &
    - Switzerland ( CHF )

# Next Steps
( cont. )

## More attention may be applicable to the below:

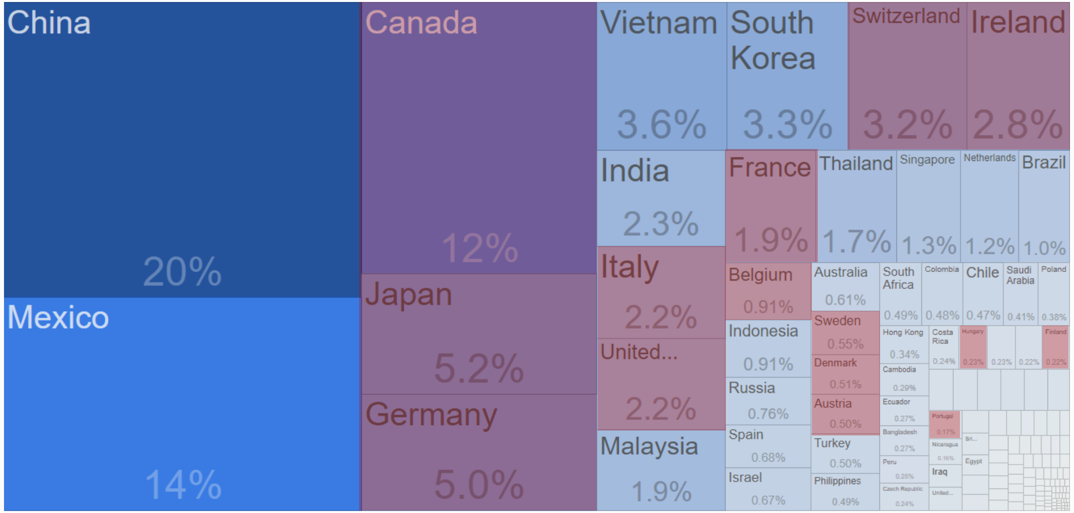
- Get more data
- The SS & LG Divide
- Scrape Variables
- Predict Wages CPI Itself
- **Build a Better Imported / Exported USD**
  - The DXY doesn't correctly address whether the US Imports or Exports Inflation as it's is a weighted geometric mean of the:
  - **Doesn't take into account the US's largest trading partner, China. Imports shown below**



# Next Steps
( cont. )

## More attention may be applicable to the below:

- Get more data
- The SS & LG Divide
- Scrape Variables
- Predict Wages CPI Itself
- **Build a Better Imported / Exported USD**
  - The DXY doesn't correctly address whether the US Imports or Exports Inflation as it's is a weighted geometric mean of the:
  - Doesn't take into account the US's largest trading partner, China. Imports shown below.
  - **It takes into account less than 40% of US Import Trade**



# Next Steps
( cont. )

## More attention may be applicable to the below:

- Get more data
- The SS & LG Divide
- Scrape Variables
- Predict Wages CPI Itself
- Build a Better Imported / Exported USD
- **Random Forest was used, while Gradient Boosting may be something to explore:**
    - **i.e. Boosting over Bagging**

# Thanks

By **Rand Sobczak Jr.**
rand.sobczak@gmail.com
+1 313 447 8634


SS ( X & y ) | Pipeline mean CV score (error bars +/- 1sd)
SS ( X & y ) | CV score (r-squared)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
2.5

# Questions¿