All code used for cleaning data, creating figures and performing analysis.

```r
# All relevant packages used.
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dplyr)
library(conflicted)
library(lubridate)

conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```r
conflict_prefer("lag", "dplyr")
```

```
## [conflicted] Will prefer dplyr::lag over any other package.
```

```r
# rm(list = ls())
# This clears the global environment
```

```r
# Importing and storing the data used into R.

silver_raw <-read_csv("silver_prices.csv", show_col_types = FALSE)
gold_raw <-read_csv("gold_prices.csv", show_col_types = FALSE)
```

```r
#Visualizing the raw data we will be using.

glimpse(silver_raw)
```

```
## Rows: 1,181
## Columns: 7
## $ Date                <dttm> 1968-01-02 09:00:00, 1968-01-30 09:00:00, 1968-02~
## $ silver_oz_usd_price <dbl> 2.17, 2.02, 2.00, 2.20, 2.26, 2.40, 2.36, 2.48, 2.~
## $ silver_kg_usd_price <dbl> 69.77, 64.94, 64.30, 70.73, 72.66, 77.16, 75.88, 7~
## $ silver_oz_gbp_price <dbl> 0.90, 0.84, 0.83, 0.92, 0.94, 1.00, 0.98, 1.04, 1.~
## $ silver_kg_gbp_price <dbl> 28.94, 27.01, 26.69, 29.58, 30.22, 32.15, 31.51, 3~
## $ silver_oz_eur_price <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ silver_kg_eur_price <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```r
glimpse(gold_raw)
```

```
## Rows: 1,181
## Columns: 7
## $ Date              <dttm> 1968-01-02 09:00:00, 1968-01-30 09:00:00, 1968-02-0~
## $ gold_oz_usd_price <dbl> 35.18, 35.19, 35.20, 35.20, 38.00, 39.10, 39.10, 41.~
## $ gold_kg_usd_price <dbl> 1131.06, 1131.38, 1131.70, 1131.70, 1221.73, 1257.09~
## $ gold_oz_gbp_price <dbl> 14.64, 14.59, 14.59, 14.66, 15.81, 16.32, 16.31, 17.~
## $ gold_kg_gbp_price <dbl> 470.69, 469.08, 469.08, 471.33, 508.30, 524.70, 524.~
## $ gold_oz_eur_price <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gold_kg_eur_price <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

```r
# Cleaning the data by selecting relevant columns for analysis and removing missing entries.

silver_cleaned <- silver_raw %>%
  select("Date", "silver_oz_usd_price") %>% drop_na()
gold_cleaned <- gold_raw %>%
  select("Date", "gold_oz_usd_price") %>% drop_na()
```

```r
# Creating new variables and data to work with for silver prices.

dates_s <- silver_cleaned$Date
price_s <- silver_cleaned$silver_oz_usd_price

month_vector_s <- c()
spec_vector <- c()
index_vector <- c()
start_price_iv1 <- c()
start_price_iv2 <- c()
end_price_iv1 <- c()

for (x in 1:length(dates_s)){
  month_vector_s <- append(month_vector_s,
  month(as.POSIXlt(dates_s[x], format="%d/%m/%Y")))
}

for (y in 1:length(month_vector_s)){
  if (month_vector_s[y] == 1 | month_vector_s[y] == 12)
      index_vector <- append(index_vector, y)
}

for (z in 1:length(month_vector_s)){
  if (month_vector_s[z] == 1 | month_vector_s[z] == 12)
      spec_vector <- append(spec_vector, month_vector_s[z])
}

for (a in 1:length(spec_vector)){
  if (spec_vector[a] == 1)
    start_price_iv1 <- append(start_price_iv1, index_vector[a])
}

i <- 1
j <- 2
```

```r
while (i < length(start_price_iv1) & j < length(start_price_iv1)){
  if ((start_price_iv1[j] - start_price_iv1[i] > 19)){
    start_price_iv2 <- append(start_price_iv2, start_price_iv1[i])
    i <- j
    j <- j + 1
  } else {
    j <- j + 1
}
}

for (b in 1:length(spec_vector)){
  if (spec_vector[b] == 12)
    end_price_iv1 <- append(end_price_iv1, index_vector[b])
}

end_price_iv2 <- end_price_iv1[-c(1,3,5,7,10,12,14,16,
18,21, 23, 25,27, 29, 31,33,35,39, 37, 42, 44,46,48,
50,53,55,57,59,62,64,66, 68, 70, 73, 75, 77,79,81,
83,85, 87, 89,91, 93, 95,97, 99, 101, 103, 105)]
```

```r
# Getting silver prices for return rate analysis

start_prices <- c()
end_prices <- c()
year <- c()

for (m in 1:length(start_price_iv2)){
  start_prices <- append(start_prices, price_s[start_price_iv2[m]])
}

for (n in 1:length(end_price_iv2)){
  end_prices <- append(end_prices, price_s[end_price_iv2[n]])
}

for (p in 1968:2023){
  year <- append(year, p)
}

difference_v <- c()

for (q in 1:length(start_prices)){
  difference_v <- append(difference_v, (end_prices[q] - start_prices[q]))
}

return_v <- c()

for (r in 1:length(difference_v)){
  return_v <- append(return_v, (difference_v[r]/start_prices[r]))
}

return_percent <- c()

for (s in 1:length(return_v)){
```

```r
  return_percent <- append(return_percent, 100*return_v[s])
}

effective_change <- c()

for(t in 1:length(return_v)){
  effective_change <- append(effective_change, 1 + return_v[t])
}

silver_final_data <- data.frame(year, start_prices,
    end_prices, return_v, effective_change, return_percent)
```

```r
# Creating new variables and data to work with for gold prices
# (the indices would be the same as silver, saving time).
# Getting gold prices for return rate analysis

dates_g <- gold_cleaned$Date
price_g <- gold_cleaned$gold_oz_usd_price

start_price_gv1 <- start_price_iv2
end_price_gv1 <- end_price_iv2

start_prices_g <- c()
end_prices_g <- c()
year_g <- c()

for (d in 1:length(start_price_gv1)){
  start_prices_g <- append(start_prices_g, price_g[start_price_gv1[d]])
}

for (e in 1:length(end_price_iv2)){
  end_prices_g <- append(end_prices_g, price_g[end_price_gv1[e]])
}

for (f in 1968:2023){
  year_g <- append(year_g, f)
}

difference_g <- c()

for (h in 1:length(start_prices_g)){
  difference_g <- append(difference_g, (end_prices_g[h] - start_prices_g[h]))
}

return_g <- c()

for (u in 1:length(difference_g)){
  return_g <- append(return_g, (difference_g[u]/start_prices_g[u]))
}

return_percent_g <- c()

for (v in 1:length(return_g)){
```

```r
  return_percent_g <- append(return_percent_g, 100*return_g[v])
}

effective_change_g <- c()

for(w in 1:length(return_v)){
  effective_change_g <- append(effective_change_g, 1 + return_g[w])
}

gold_final_data <- data.frame(year_g, start_prices_g,
    end_prices_g, return_g, effective_change_g, return_percent_g)
```

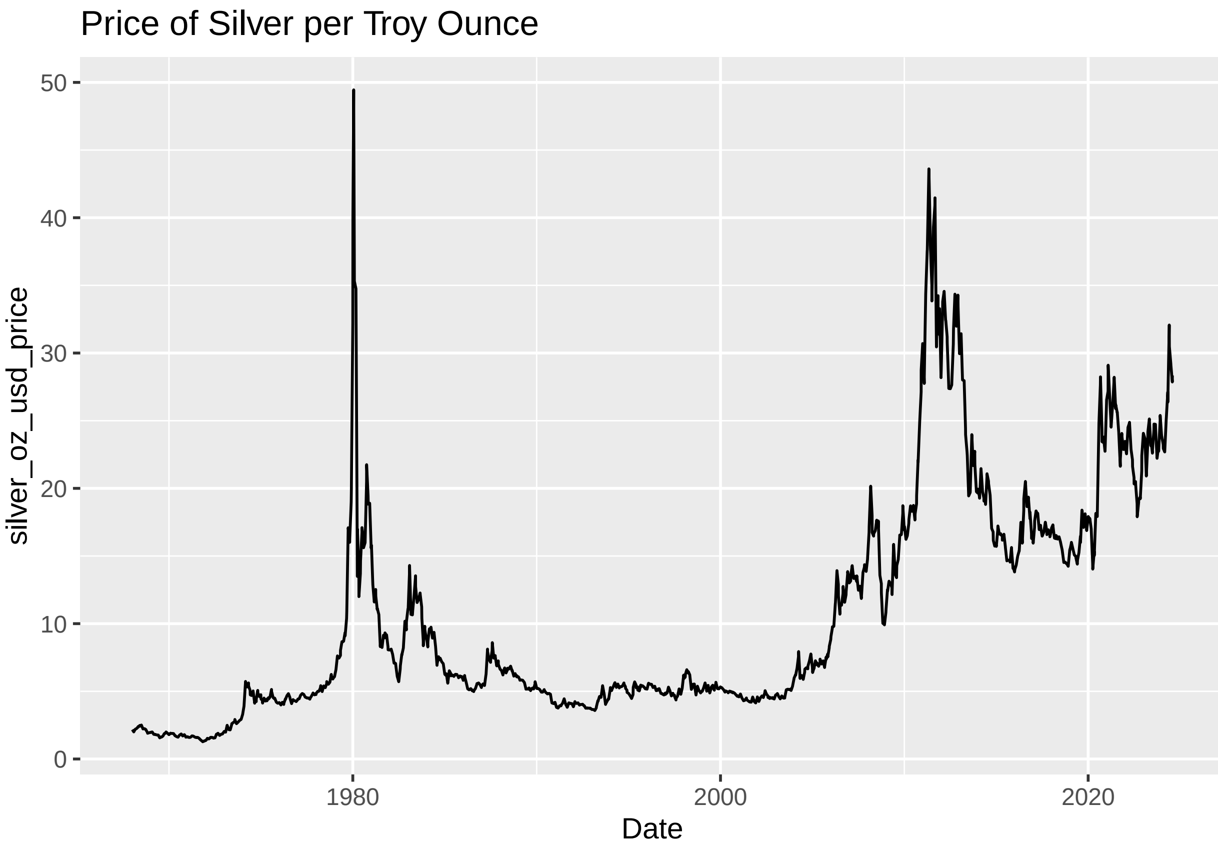
```r
# Average annual return rate of silver

product_s <- 1

for (o in 1:length(silver_final_data$effective_change)){
  product_s <- product_s * silver_final_data$effective_change[o]
}

average_annual_return_s <- product_s^(1/length(silver_final_data$effective_change)) - 1

average_annual_return_s
```

```
## [1] 0.03960043
```

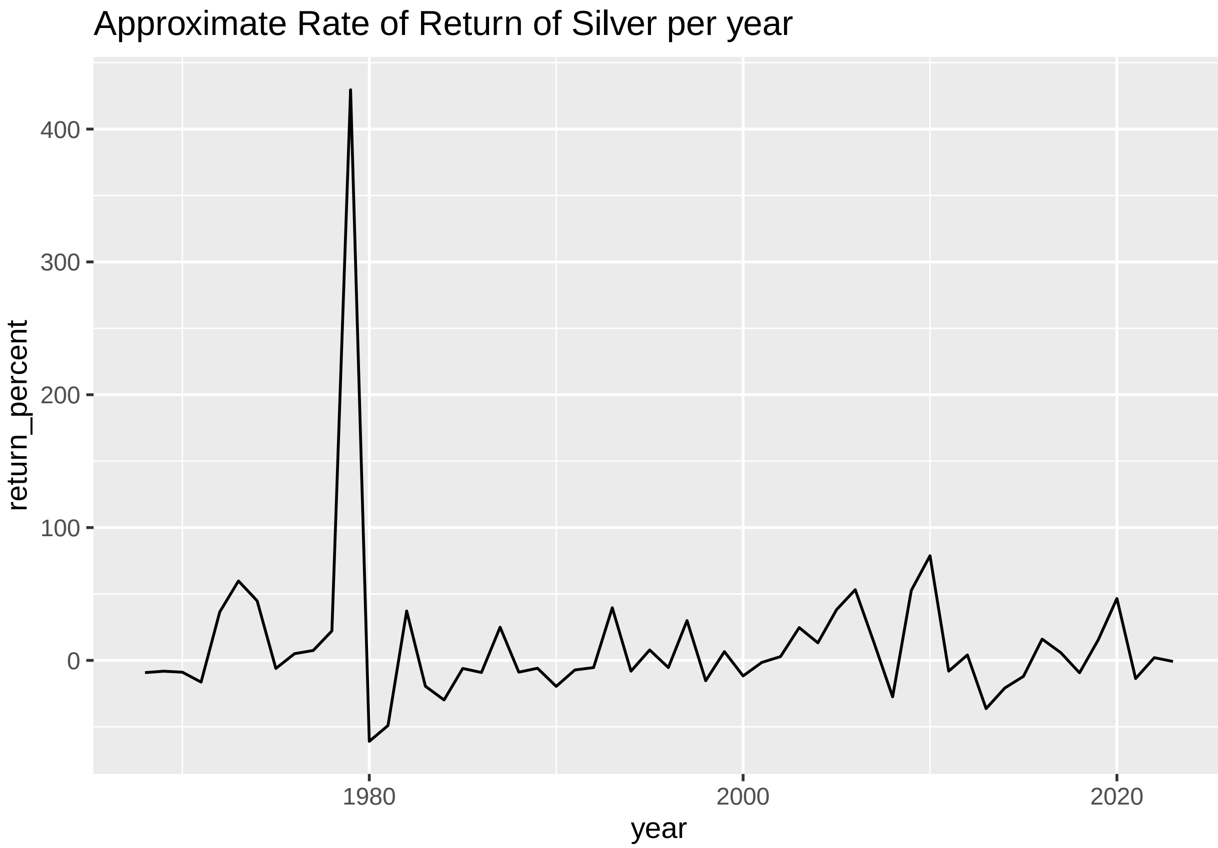
```r
# Average annual return rate of gold

product_g <- 1

for (a in 1:length(gold_final_data$effective_change)){
  product_g <- product_g * gold_final_data$effective_change[a]
}

average_annual_return_g <- product_g^(1/length(gold_final_data$effective_change)) - 1

average_annual_return_g
```

```
## [1] 0.06573133
```

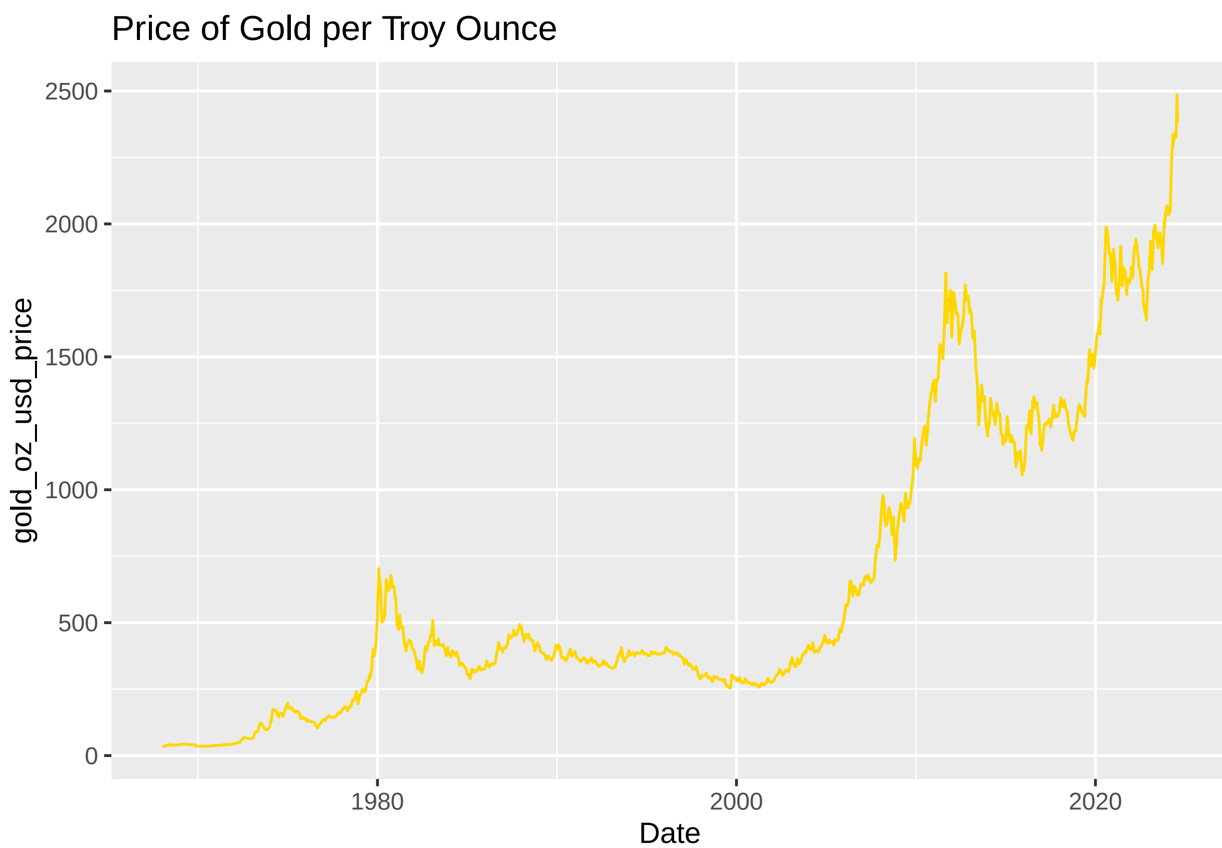
```r
# Visualizations

silver_cleaned %>% ggplot(aes(x = Date, y = silver_oz_usd_price)) +
  geom_line(color = "black") + ggtitle("Price of Silver per Troy Ounce")
```



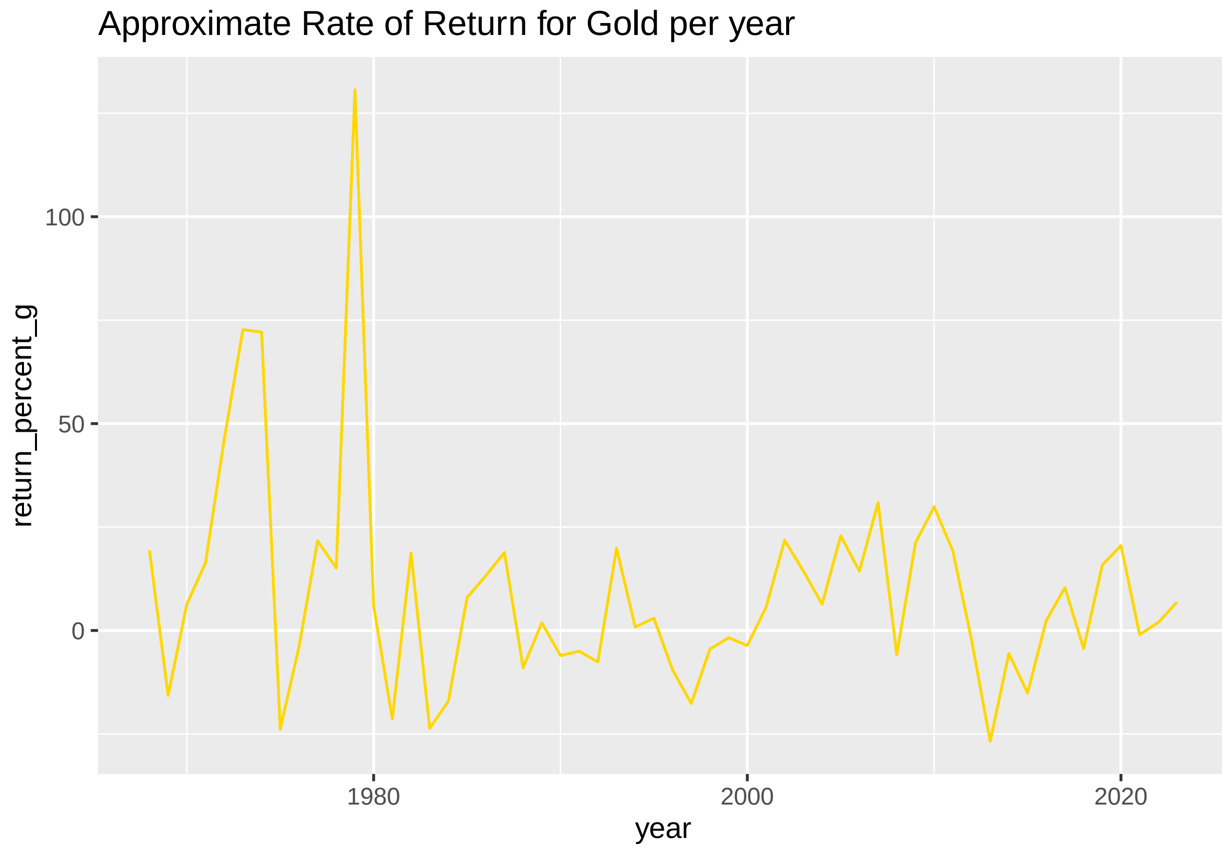

```
silver_final_data %>% ggplot(aes(x = year, y = return_percent)) +
  geom_line(color = 'black') +
  ggtitle("Approximate Rate of Return of Silver per year")
```




```
gold_cleaned %>% ggplot(aes(x = Date, y = gold_oz_usd_price)) +
  geom_line(color = "gold") + ggtitle("Price of Gold per Troy Ounce")
```




```
gold_final_data %>% ggplot(aes(x = year, y = return_percent_g)) +
  geom_line(color = 'gold') +
  ggtitle("Approximate Rate of Return for Gold per year")
```



Approximate Rate of Return for Gold per year
return_percent_g
100
50
0
1980
2000
2020
year