Story 2

Andrew Bowen

2023-09-16

Story 2

Introduction

Inflation and its economic effects have been in the news recently. The Fed's goal has been to curb inflation and maintain low unemployment. The Phillips Curve is an economic principle stating the general negative relationship between these two variables.

Data Sources

- New York Fed API
- BLS Data API - used for unemployment data

The BLS API has an R wrapper that allows for easy data pulls from the API.

```
# Setting ZFed and BLS API Keys
registration_key = Sys.getenv("BLS_API_KEY")
fed_api_key <- Sys.getenv("FED_API_KEY")
```

Getting CPI time series data

```
# Call CPI data from BLS
registration_key <- Sys.getenv("BLS_API_KEY")
series_id <- 'APU0000702111'
series_url <- glue("https://api.bls.gov/publicAPI/v2/timeseries/data/{series_id}?registrationkey={regis

response <- GET(series_url)
```

```
# Format CPI data
response_txt <- content(response, "text", encoding = "UTF-8")
json <- fromJSON(response_txt, flatten = TRUE)

cpi <- as.data.frame(json[[4]])$series.data[[1]]
head(cpi)
```

```
##   year period periodName value footnotes
## 1 2022    M12   December 1.873      NULL
## 2 2022    M11   November 1.847      NULL
## 3 2022    M10    October 1.814      NULL
## 4 2022    M09  September 1.749      NULL
## 5 2022    M08     August 1.756      NULL
## 6 2022    M07       July 1.715      NULL
```

We'll need to pull an additional 5 years of CPI data

```
series_url <= glue("https://api.bls.gov/publicAPI/v2/timeseries/data/{series_id}?registrationkey={regis
```

```
## [1] FALSE
```

```
response <- GET(series_url)
response_txt <- content(response, "text", encoding = "UTF-8")
json <- fromJSON(response_txt, flatten = TRUE)

cpi_older <- as.data.frame(json[[4]])$series.data[[1]]
```

```
# Combining our CPI data
cpi_df <- rbind(cpi, cpi_older)

# Converting dates from BLS format
cpi_df$date <- str_c(cpi_df$periodName," ", cpi_df$year)
cpi_df$date <- myd(cpi_df$date, truncated=1)

# cast types
cpi_df$cpi <- as.double(cpi$value)
```

Now let's make a simple plot of our CPI data over time

```
cpi_plot <- ggplot(cpi_df, aes(x=date, y=cpi)) + geom_line() +
  labs(x="Month", y="Consumer Price Index (CPI)", title="Consumer Price Index over Time - All US", subt
```

Grabbing Interest Rate Data

Grbabing interest rate data from the St Louis Fed API (FRED). The endpoint I'm using contains information about the Federal Effective Funds rate, which is the interest rate charged to banks borrowing overnight to meet their reserve requirements

```
# Use an environement variable for API token
url <- glue("https://api.stlouisfed.org/fred/series/observations?series_id=EFFR&api_key={fed_api_key}&f

# Pull data from FRED API
fed_response <- GET(url)
```

Let's do some basic data wrangling of our FRED API response to get it into a good format for plotting

```
# Write Response data to R dataframe
fed_response_txt <- content(fed_response, "text", encoding = "UTF-8")
fed_json <- json <- fromJSON(fed_response_txt, flatten = TRUE)
fed <- fed_json$observations
```

Let's clean up our federal funds rate data

```
# Convert types
fed$date <- as.Date(fed$date)
fed$fed_funds_rate <- as.double(fed$value)
```

```
## Warning: NAs introduced by coercion
```

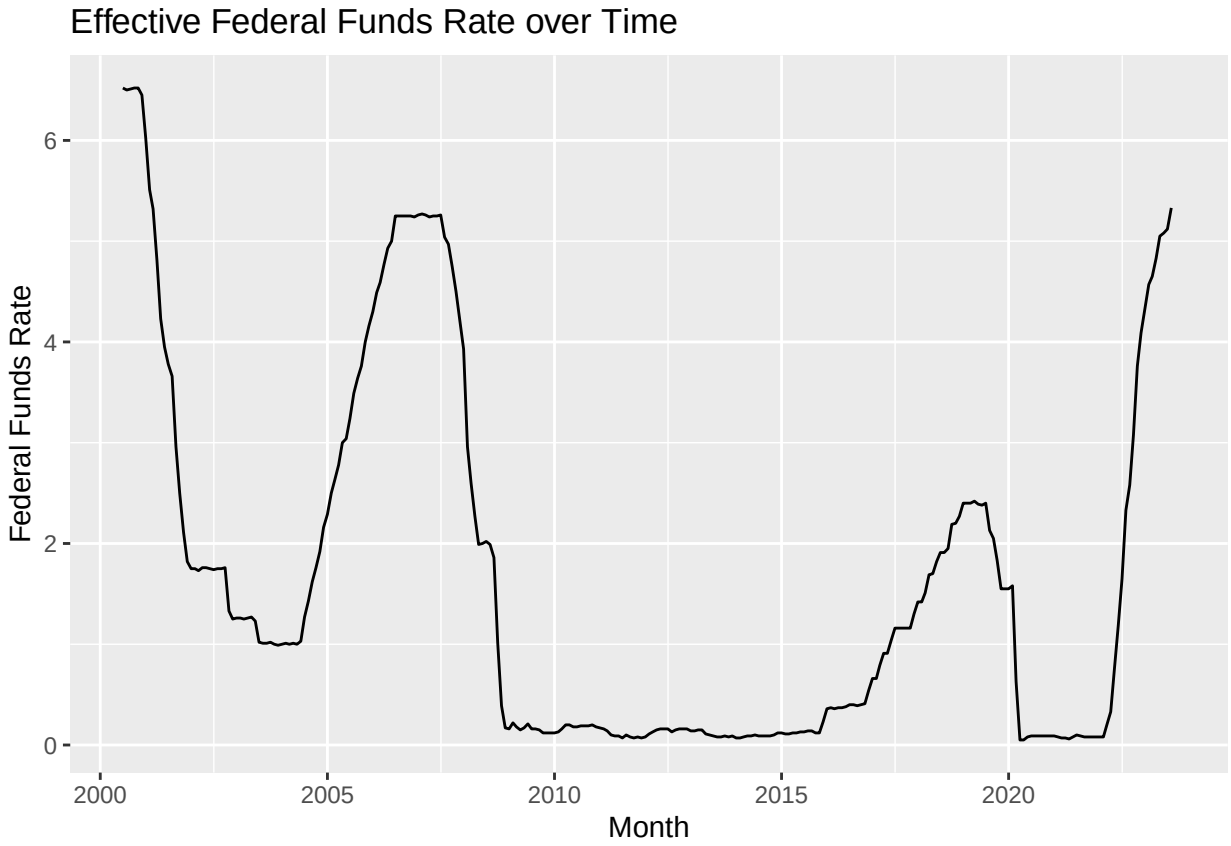
```
head(fed)
```

```
##   realtime_start realtime_end       date value fed_funds_rate
## 1     2023-09-23   2023-09-23 2000-07-01  6.52           6.52
## 2     2023-09-23   2023-09-23 2000-08-01  6.50           6.50
## 3     2023-09-23   2023-09-23 2000-09-01  6.51           6.51
## 4     2023-09-23   2023-09-23 2000-10-01  6.52           6.52
## 5     2023-09-23   2023-09-23 2000-11-01  6.52           6.52
## 6     2023-09-23   2023-09-23 2000-12-01  6.45           6.45
```

Now let's plot a simple time series of our federal funds rate date

```
# Plotting Federal Funds Rate over time
fed_rate_plot <- ggplot(fed, aes(x=date, y=fed_funds_rate)) +
  geom_line() + labs(y="Federal Funds Rate", x="Month", title ="Effective Federal Funds Rate over Time"
fed_rate_plot
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



Unemployment Data

I found a CSV file containing seasonally adjusted unemployment data from BLS here

```
unemployment <- read.csv("data/unemployment-bls.csv")
```

Let's wrangle the unemployment API data a bit

```
head(unemployment)
```

```
##     Series.ID Year Period    Label Value
## 1 LNS14000000 1998    M01 1998 Jan   4.6
## 2 LNS14000000 1998    M02 1998 Feb   4.6
## 3 LNS14000000 1998    M03 1998 Mar   4.7
## 4 LNS14000000 1998    M04 1998 Apr   4.3
## 5 LNS14000000 1998    M05 1998 May   4.4
## 6 LNS14000000 1998    M06 1998 Jun   4.5
```

Clwaning up our unemployment data before plotting

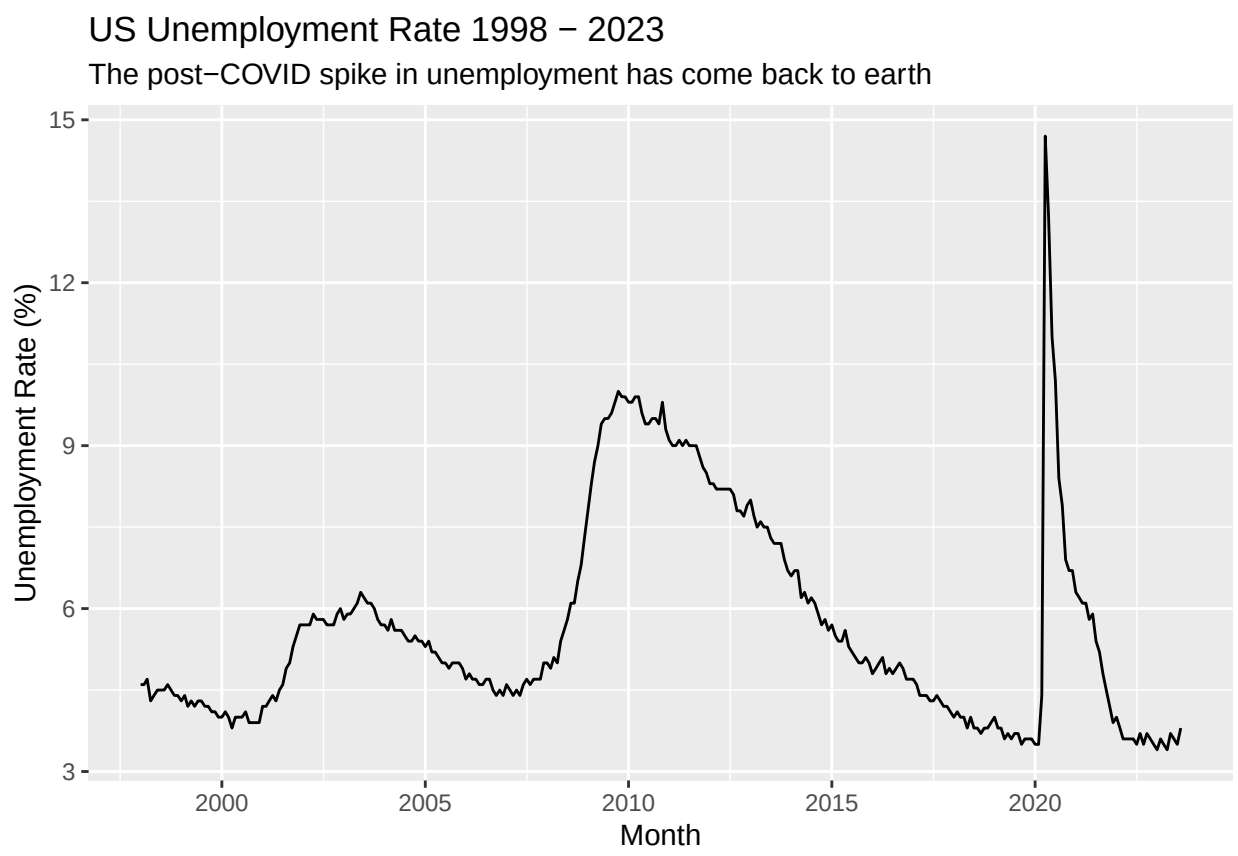
```
unemployment$date <- as.Date(parse_date_time(unemployment$Label, order = "Ym"))
unemployment$unemployment_rate <- unemployment$Value
head(unemployment)
```

```
##     Series.ID Year Period    Label Value       date unemployment_rate
## 1 LNS14000000 1998    M01 1998 Jan   4.6 1998-01-01               4.6
## 2 LNS14000000 1998    M02 1998 Feb   4.6 1998-02-01               4.6
## 3 LNS14000000 1998    M03 1998 Mar   4.7 1998-03-01               4.7
## 4 LNS14000000 1998    M04 1998 Apr   4.3 1998-04-01               4.3
## 5 LNS14000000 1998    M05 1998 May   4.4 1998-05-01               4.4
## 6 LNS14000000 1998    M06 1998 Jun   4.5 1998-06-01               4.5
```

Let's quickly plot the US unemployment rate over time (1998 - 2003)

```
unemployment_plot <- ggplot(unemployment, aes(x=date, y=unemployment_rate)) + geom_line() +
  labs(x="Month", y="Unemployment Rate (%)",
       title="US Unemployment Rate 1998 - 2023",
       subtitle="The post-COVID spike in unemployment has come back to earth")


unemployment_plot
```



Combining our Datasets

Now we can join together our datasets to overlay the data points in a single panel plot

```
dat <- merge(unemployment, fed, by=c("date", "date"))

df <- merge(dat, cpi_df, by=c("date", "date"))
```

Let's plot the time series three variables of interest:

- Consumer Price Index (CPI)
- Unemployment Rate (%)
- Federal Funds Rate

```
# Create Panel plot of our 3 metrics
labels_removed <- labs(x= NULL, title=NULL, subtitle=NULL)
grid.arrange(unemployment_plot + labels_removed,
             fed_rate_plot + labels_removed,
             cpi_plot + labels_removed + labs(x="Month"),
             ncol=1, nrow=3)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```