

ECON425_VA1_CAnce1

Charles Ance1

2024-07-06

```
# Fetching nominal interest rate and inflation rate data from FRED
nominal_rate <- fredr(series_id = "DGS10", observation_start = as.Date("2000-01-01"))
inflation_rate <- fredr(series_id = "CPIAUCSL", observation_start = as.Date("2000-01-01"))

# Calculating the real interest rate
inflation_rate <- inflation_rate %>%
  mutate(inflation_rate = (value/lag(value, 12) - 1) * 100) %>%
  filter(!is.na(inflation_rate))

nominal_rate <- nominal_rate %>%
  rename(nominal_rate = value) %>%
  filter(date %in% inflation_rate$date)

data <- left_join(nominal_rate, inflation_rate, by = "date") %>%
  mutate(real_rate = nominal_rate - inflation_rate) %>%
  filter(!is.na(real_rate))

# Plotting nominal and real interest rates
interest_rates_plot <- ggplot(data, aes(x = date)) +
  geom_line(aes(y = nominal_rate, color = "Nominal Rate")) +
  geom_line(aes(y = real_rate, color = "Real Rate")) +
  labs(title = "Nominal and Real Interest Rates (2000-2020)",
       y = "Interest Rate (%)", x = "Year") +
  scale_color_manual(values = c("Nominal Rate" = "blue", "Real Rate" = "red"))

ggsave("Nominal_vs_Real_Interest_Rates.png", plot = interest_rates_plot)
```

Saving 6.5 x 4.5 in image

```
# Fetching consumer credit data from FRED
consumer_credit <- fredr(series_id = "TOTALSL", observation_start = as.Date("2000-01-01"))

# Plotting consumer credit over time
consumer_credit_plot <- ggplot(consumer_credit, aes(x = date, y = value)) +
  geom_line(color = "blue") +
  labs(title = "Total Consumer Credit (2000-2020)",
       y = "Total Consumer Credit (Billions of USD)", x = "Year")
ggsave("Total_Consumer_Credit.png", plot = consumer_credit_plot)
```

Saving 6.5 x 4.5 in image

```
# Fetching nominal exchange rate and interest rate data from FRED
nominal_exchange_rate <- fredr(series_id = "DEXUSEU", observation_start = as.Date("2000-01-01"))
domestic_interest_rate <- fredr(series_id = "DGS10", observation_start = as.Date("2000-01-01"))
```

```

foreign_interest_rate <- fredr(series_id = "IR3TIB01EZM156N", observation_start = as.Date("2000-01-01"))

# Merging and calculating interest rate differential
data_exchange <- left_join(nominal_exchange_rate, domestic_interest_rate, by = "date") %>%
  rename(nominal_rate = value.y) %>%
  left_join(., foreign_interest_rate, by = "date") %>%
  rename(foreign_rate = value) %>%
  mutate(interest_rate_diff = nominal_rate - foreign_rate)

# Plotting nominal exchange rate and interest rate differential
exchange_rate_plot <- ggplot(data_exchange, aes(x = date)) +
  geom_line(aes(y = value.x, color = "Nominal Exchange Rate")) +
  geom_line(aes(y = interest_rate_diff, color = "Interest Rate Differential")) +
  labs(title = "Nominal Exchange Rate and Interest Rate Differential (2000-2020)",
       y = "Value", x = "Year") +
  scale_color_manual(values = c("Nominal Exchange Rate" = "blue", "Interest Rate Differential" = "red"))
ggsave("Nominal_Exchange_Rate_and_Interest_Rate_Differential.png", plot = exchange_rate_plot)

## Saving 6.5 x 4.5 in image

## Warning: Removed 63 rows containing missing values or values outside the scale range
## (`geom_line()`).

# Fetching consumer price index data from FRED
cpi <- fredr(series_id = "CPIAUCSL", observation_start = as.Date("2000-01-01"))

# Plotting CPI as a proxy for price rigidity
cpi_plot <- ggplot(cpi, aes(x = date, y = value)) +
  geom_line(color = "blue") +
  labs(title = "Consumer Price Index (2000-2020)",
       y = "CPI (1982-1984=100)", x = "Year")
ggsave("Consumer_Price_Index.png", plot = cpi_plot)

## Saving 6.5 x 4.5 in image

# Fetching GDP data from FRED
gdp <- fredr(series_id = "GDPC1", observation_start = as.Date("2000-01-01"))
potential_gdp <- fredr(series_id = "GDPPOT", observation_start = as.Date("2000-01-01"))

# Merging and calculating output gap
data_gdp <- left_join(gdp, potential_gdp, by = "date") %>%
  rename(actual_gdp = value.x, potential_gdp = value.y) %>%
  mutate(output_gap = actual_gdp - potential_gdp)

# Plotting actual and potential output
output_gap_plot <- ggplot(data_gdp, aes(x = date)) +
  geom_line(aes(y = actual_gdp, color = "Actual GDP")) +
  geom_line(aes(y = potential_gdp, color = "Potential GDP")) +
  labs(title = "Actual vs. Potential Output (2000-2020)",
       y = "Billions of Chained 2012 Dollars", x = "Year") +
  scale_color_manual(values = c("Actual GDP" = "blue", "Potential GDP" = "red"))
ggsave("Actual_vs_Potential_Output.png", plot = output_gap_plot)

## Saving 6.5 x 4.5 in image

```

```

# Fetching unemployment rate and CPI data from FRED
unemployment_rate <- fredr(series_id = "UNRATE", observation_start = as.Date("2000-01-01"))
cpi <- fredr(series_id = "CPIAUCSL", observation_start = as.Date("2000-01-01"))

# Calculating inflation rate
cpi <- cpi %>%
  mutate(inflation_rate = (value / lag(value, 12) - 1) * 100) %>%
  filter(!is.na(inflation_rate))

# Merging unemployment and inflation data
phillips_data <- left_join(unemployment_rate, cpi, by = "date") %>%
  rename(unemployment_rate = value.x, inflation_rate = inflation_rate)

# Plotting Phillips Curve
phillips_curve_plot <- ggplot(phillips_data, aes(x = unemployment_rate, y = inflation_rate)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Phillips Curve: Unemployment vs. Inflation",
       y = "Inflation Rate (%)", x = "Unemployment Rate (%)")
ggsave("Phillips_Curve.png", plot = phillips_curve_plot)

## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 13 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 13 rows containing missing values or values outside the scale range
## (`geom_point()`).

# Fetching GDP data from FRED to illustrate shocks
gdp <- fredr(series_id = "GDPC1", observation_start = as.Date("2000-01-01"))

# Simulating a supply shock event (use part of the data and repeat it)
shock_start <- 60 # Assuming the shock starts at the 60th observation
shock_length <- 12 # Length of the shock period
supply_shock <- rep(mean(gdp$value[(shock_start-5):(shock_start-1)]), shock_length)

# Assigning supply shock values to the GDP data frame
gdp$Supply_Shock <- NA
gdp$Supply_Shock[shock_start:(shock_start + shock_length - 1)] <- supply_shock

# Plotting GDP under a supply shock
supply_shock_plot <- ggplot(gdp, aes(x = date)) +
  geom_line(aes(y = value, color = "Normal GDP")) +
  geom_line(aes(y = Supply_Shock, color = "Supply Shock"), na.rm = TRUE) +
  labs(title = "Impact of Supply Shocks on GDP (2000-2020)",
       y = "GDP (Billions of Chained 2012 Dollars)", x = "Year") +
  scale_color_manual(values = c("Normal GDP" = "blue", "Supply Shock" = "red"))
ggsave("Impact_of_Supply_Shocks_on_GDP.png", plot = supply_shock_plot)

## Saving 6.5 x 4.5 in image

# Fetching data for monetary policy indicators
federal_funds_rate <- fredr(series_id = "FEDFUNDS", observation_start = as.Date("2000-01-01"))

```

```

# Fetching GDP data
gdp <- fredr(series_id = "GDPC1", observation_start = as.Date("2000-01-01"))

# Merging GDP and federal funds rate
monetary_policy_data <- left_join(gdp, federal_funds_rate, by = "date") %>%
  rename(gdp = value.x, fed_funds_rate = value.y)

# Plotting GDP and federal funds rate to illustrate monetary policy shocks
monetary_policy_shock_plot <- ggplot(monetary_policy_data, aes(x = date)) +
  geom_line(aes(y = gdp, color = "GDP")) +
  geom_line(aes(y = fed_funds_rate * 100, color = "Federal Funds Rate"), linetype = "dashed") +
  labs(title = "Impact of Monetary Policy on GDP (2000-2020)",
       y = "GDP (Billions of Chained 2012 Dollars) / Federal Funds Rate (%)", x = "Year") +
  scale_color_manual(values = c("GDP" = "blue", "Federal Funds Rate" = "red"))
ggsave("Impact_of_Monetary_Policy_on_GDP.png", plot = monetary_policy_shock_plot)

## Saving 6.5 x 4.5 in image

# Fetching GDP data to illustrate productivity shocks
gdp <- fredr(series_id = "GDPC1", observation_start = as.Date("2000-01-01"))

# Creating a vector for Positive_Shock with the same length as gdp
gdp$Positive_Shock <- NA

# Define start and end of the shock
shock_start <- 60 # Assuming the shock starts at the 60th observation
shock_length <- 12 # Length of the shock period

# Simulating a positive productivity shock
gdp$Positive_Shock[shock_start:(shock_start + shock_length - 1)] <- gdp$value[shock_start:(shock_start + shock_length - 1)]

# Plotting GDP under productivity shocks
productivity_shock_plot <- ggplot(gdp, aes(x = date)) +
  geom_line(aes(y = value, color = "Normal GDP")) +
  geom_line(aes(y = Positive_Shock, color = "Positive Shock"), na.rm = TRUE) +
  labs(title = "Impact of Productivity Shocks on GDP (2000-2020)",
       y = "GDP (Billions of Chained 2012 Dollars)", x = "Year") +
  scale_color_manual(values = c("Normal GDP" = "blue", "Positive Shock" = "green"))
ggsave("Impact_of_Productivity_Shocks_on_GDP.png", plot = productivity_shock_plot)

## Saving 6.5 x 4.5 in image

# Fetching GDP data from FRED to illustrate business cycles
gdp <- fredr(series_id = "GDPC1", observation_start = as.Date("2000-01-01"))

# Identifying peaks and troughs (using a simple method for illustration)
gdp$Phase <- ifelse(gdp$value == max(gdp$value), "Peak", ifelse(gdp$value == min(gdp$value), "Trough", "Normal"))

# Plotting business cycles
business_cycles_plot <- ggplot(gdp, aes(x = date, y = value)) +
  geom_line(color = "blue") +
  geom_point(aes(color = Phase)) +
  scale_color_manual(values = c("Peak" = "red", "Trough" = "green", "Normal" = "blue")) +
  labs(title = "Business Cycles (2000-2020)",
       y = "GDP (Billions of Chained 2012 Dollars)", x = "Year")

```

```

ggsave("Business_Cycles.png", plot = business_cycles_plot)

## Saving 6.5 x 4.5 in image
library(mFilter)

# Fetching GDP data from FRED
gdp <- fredr(series_id = "GDPC1", observation_start = as.Date("2000-01-01"))

# Applying HP filter
hp_filter <- hpfilter(gdp$value, freq = 100)
gdp$Trend <- hp_filter$trend
gdp$Cycle <- hp_filter$cycle

# Plotting HP filter results
hp_filter_plot <- ggplot(gdp, aes(x = date)) +
  geom_line(aes(y = value, color = "Actual GDP")) +
  geom_line(aes(y = Trend, color = "Trend")) +
  geom_line(aes(y = Cycle + mean(value), color = "Cycle")) +
  labs(title = "HP Filter Applied to GDP (2000-2020)",
       y = "GDP (Billions of Chained 2012 Dollars)", x = "Year") +
  scale_color_manual(values = c("Actual GDP" = "blue", "Trend" = "red", "Cycle" = "green"))
ggsave("HP_Filter_Applied_to_GDP.png", plot = hp_filter_plot)

## Saving 6.5 x 4.5 in image

```