ECON425_HW5

Charles Ancel

2024-07-01

```r
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   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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(quantmod)
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.3.3
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                           #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################
##
## Attaching package: 'xts'
```

```
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## Loading required package: TTR
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(fredr)
```

```r
fredr_set_key("89e09406a6b0e5d39d06d3b19fec5156")
```

```r
# Retrieve data from FRED
unemployment_rate <- fredr(series_id = "UNRATE")
cpi <- fredr(series_id = "CPILFESL")
hourly_earnings <- fredr(series_id = "CES0500000003")
```

```r
# Convert the data to time series objects
unemployment_ts <- xts(unemployment_rate$value, order.by = as.Date(unemployment_rate$date))
cpi_ts <- xts(cpi$value, order.by = as.Date(cpi$date))
earnings_ts <- xts(hourly_earnings$value, order.by = as.Date(hourly_earnings$date))
```

```r
# Calculate year-over-year inflation rate from CPI
inflation_rate <- diff(log(cpi_ts), lag = 12) * 100

# Calculate year-over-year wage inflation rate from hourly earnings
wage_inflation_rate <- diff(log(earnings_ts), lag = 12) * 100
```

```r
# Merge the data into a single data frame
data <- merge(unemployment_ts, inflation_rate, wage_inflation_rate, all = FALSE)
colnames(data) <- c("Unemployment_Rate", "Inflation_Rate", "Wage_Inflation_Rate")

# Convert to a tidy data frame
data_df <- data.frame(date = index(data), coredata(data))
```

```r
# Plot Inflation vs. Unemployment Rate
inflation_vs_unemployment_plot <- ggplot(data_df, aes(x = Unemployment_Rate, y = Inflation_Rate)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Inflation vs. Unemployment Rate",
       x = "Unemployment Rate (%)",
       y = "Inflation Rate (%)")

ggsave("Inflation_vs_Unemployment.png", plot = inflation_vs_unemployment_plot)
```

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

```r
# Plot Wage Inflation vs. Unemployment Rate
wage_inflation_vs_unemployment_plot <- ggplot(data_df, aes(x = Unemployment_Rate, y = Wage_Inflation_Ra
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Wage Inflation vs. Unemployment Rate",
       x = "Unemployment Rate (%)",
       y = "Wage Inflation Rate (%)")
```

```r
ggsave("Wage_Inflation_vs_Unemployment.png", plot = wage_inflation_vs_unemployment_plot)
```

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

```
## Warning: Removed 12 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```

```r
# Analyze the relationship between variables
model1 <- lm(Inflation_Rate ~ Unemployment_Rate, data = data_df)
summary(model1)
```

```
##
## Call:
## lm(formula = Inflation_Rate ~ Unemployment_Rate, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3414 -0.7152 -0.3110  0.4754  3.2524
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.28257    0.19997  21.416   <2e-16 ***
## Unemployment_Rate -0.31591    0.03166  -9.979   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.033 on 217 degrees of freedom
## Multiple R-squared:  0.3146, Adjusted R-squared:  0.3114
## F-statistic: 99.58 on 1 and 217 DF,  p-value: < 2.2e-16
```

```r
model2 <- lm(Wage_Inflation_Rate ~ Unemployment_Rate, data = data_df)
summary(model2)
```

```
##
## Call:
## lm(formula = Wage_Inflation_Rate ~ Unemployment_Rate, data = data_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3422 -0.7289 -0.4321  0.6122  6.0938
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.9731     0.2177  18.249  < 2e-16 ***
## Unemployment_Rate  -0.1585     0.0340  -4.661 5.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.097 on 205 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.09582,    Adjusted R-squared:  0.09141
## F-statistic: 21.72 on 1 and 205 DF,  p-value: 5.663e-06
```