TERM PROJECT_ ECON STATISTICS

LUIS HEREDIA AND RUMEYSA YILMAZ

2024-12-12

Summary

Load necessary libraries, otherwise the code won't run. If at any point you find a bug, please let us know. The code is presented in chunks, therefore there will be different output based on previous functions. Run one after another.

Shortcut

To see the data combined use the command View(combined_data_wide) on the command line interface. The first 4 columns will be composed of the observed and explanatory variables.Besides, the ggsave, command after each plot is executed, since it was used locally.

```
library(ggplot2)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(purrr)
```

```
## Warning: package 'purrr' was built under R version 4.4.2
```

```
library(tidyr)
library(broom)
library(modelr)
```

```
##
## Adjuntando el paquete: 'modelr'
```

```
## The following object is masked from 'package:broom':
##
##     bootstrap
```

```
library(readxl)
```

```
# Load datasets. These are the data used for the model and graphs
inflation <- read.csv("inflation.csv")
unrate <- read.csv("unrate.csv")
real_gdp <- read.csv("GDPC1.csv") |>  mutate(DATE = as.Date(DATE))
world_net_migration<- read_excel("Net_Immigration_world_data.xlsx", sheet = "Sheet1", skip = 3)
```

```
#Tidying world_net_immigration data, from the World Bank, to get only values for the US
US_Net_migration <- world_net_migration |>  filter(`Country Name` == "United States")|> select(-c(`Coun
pivot_longer(cols = starts_with("19") | starts_with("20"),
               names_to = "Year", values_to = "NetMigration") |> arrange(as.numeric(Year)) |> mutate("D
US_Net_migration
```

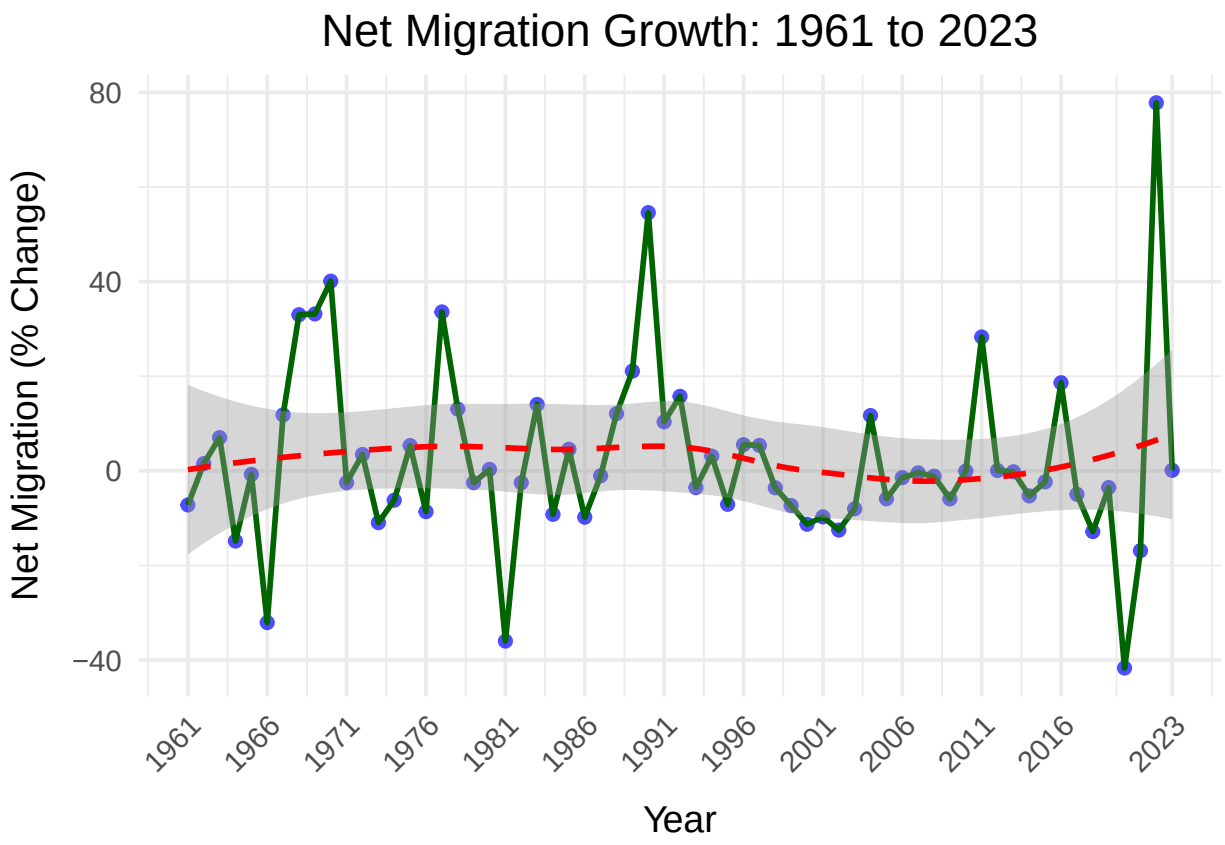
```
## # A tibble: 63 x 2
##    DATE       PercentChange
##    <date>             <dbl>
##  1 1961-01-01        -7.20
##  2 1962-01-01         1.53
##  3 1963-01-01         7.03
##  4 1964-01-01       -14.8
##  5 1965-01-01        -0.748
##  6 1966-01-01       -32.1
##  7 1967-01-01        11.8
##  8 1968-01-01        33.0
##  9 1969-01-01        33.2
## 10 1970-01-01        40.1
## # i 53 more rows
```

```
# US NET MIGRATION DATA VISUALIZATION
US_Net_migration_plot <- ggplot(US_Net_migration, aes(x = DATE, y = PercentChange)) +
  geom_point(color = "blue", alpha = 0.7, size = 2) +
  geom_line(color = "darkgreen", size = 1) +
  scale_x_date(
    date_labels = "%Y",
    breaks = c(seq(as.Date("1961-01-01"), as.Date("2020-01-01"), by = "5 years"),
               as.Date("2023-01-01"))
  ) +
  geom_smooth(method = "loess", color = "red", linetype = "dashed", size = 1) +
  labs(title = "Net Migration Growth: 1961 to 2023",
       x = "Year", y = "Net Migration (% Change)") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5),  # Center title
    axis.title.x = element_text(margin = margin(t = 10)),  # x space title
    axis.title.y = element_text(margin = margin(r = 10)),  # y space title
    axis.text.x = element_text(angle = 45, hjust = 1)  # Rotation
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
US_Net_migration_plot
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

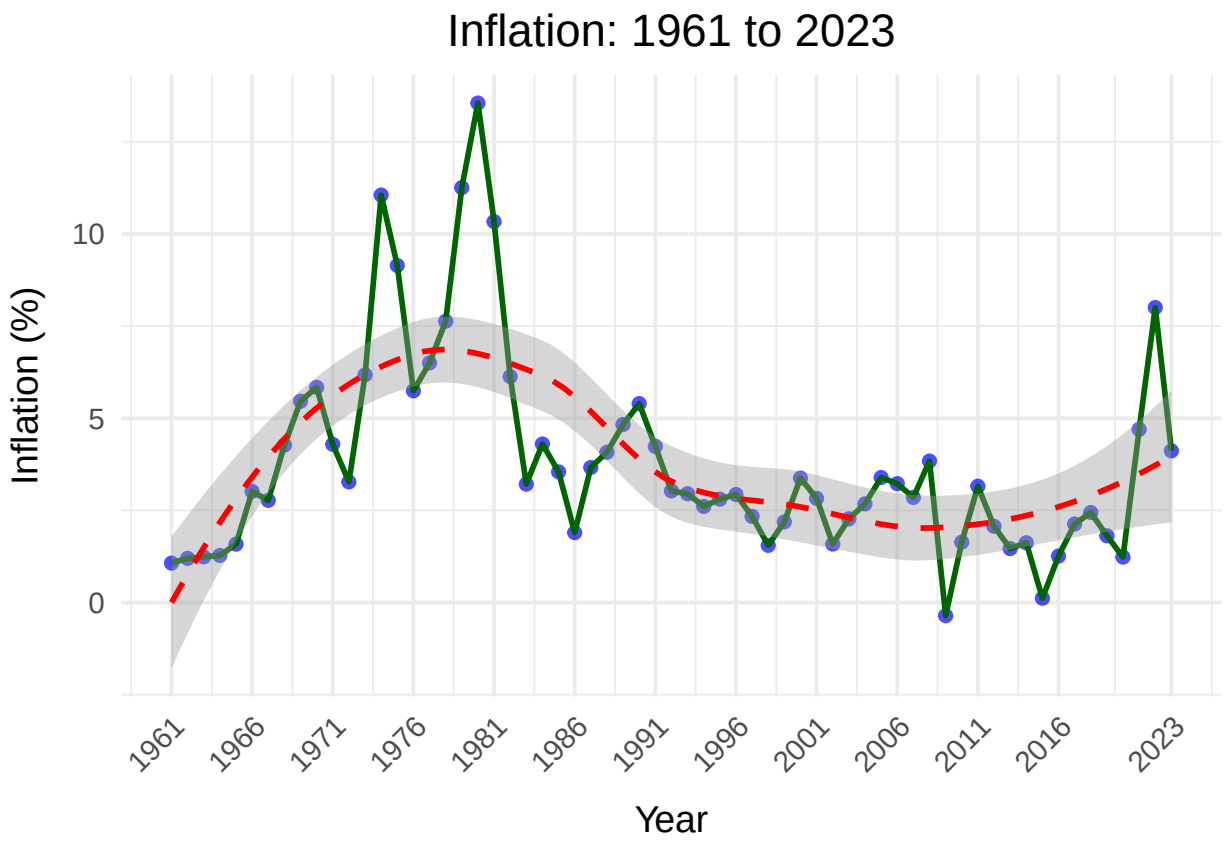



```
#You may want to save the code locally, change the file location if needed, then run the code which is
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/netmigration_plot_visualization.jpg
#  plot = US_Net_migration_plot,
#  width = 12,
 # height = 8,
 # dpi = 600
#)
```

```
#Date format
inflation <- inflation |>
  mutate(DATE = as.Date(DATE))

# inflation_plot, DATA VISUALIZATION
```

```r
inflation_plot <- ggplot(inflation, aes(x = DATE, y = FPCPITOTLZGUSA)) +
  geom_point(color = "blue", alpha = 0.7, size = 2) +
  geom_line(color = "darkgreen", size = 1) +
  scale_x_date(
    date_labels = "%Y",
    breaks = c(seq(as.Date("1961-01-01"), as.Date("2020-01-01"), by = "5 years"),
               as.Date("2023-01-01"))
  ) +
  geom_smooth(method = "loess", color = "red", linetype = "dashed", size = 1) +
  labs(title = "Inflation: 1961 to 2023",
       x = "Year", y = "Inflation (%)") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5),  # Center title
    axis.title.x = element_text(margin = margin(t = 10)),  # x space title
    axis.title.y = element_text(margin = margin(r = 10)),  # y space title
    axis.text.x = element_text(angle = 45, hjust = 1)  # Rotation
  )

inflation_plot
```

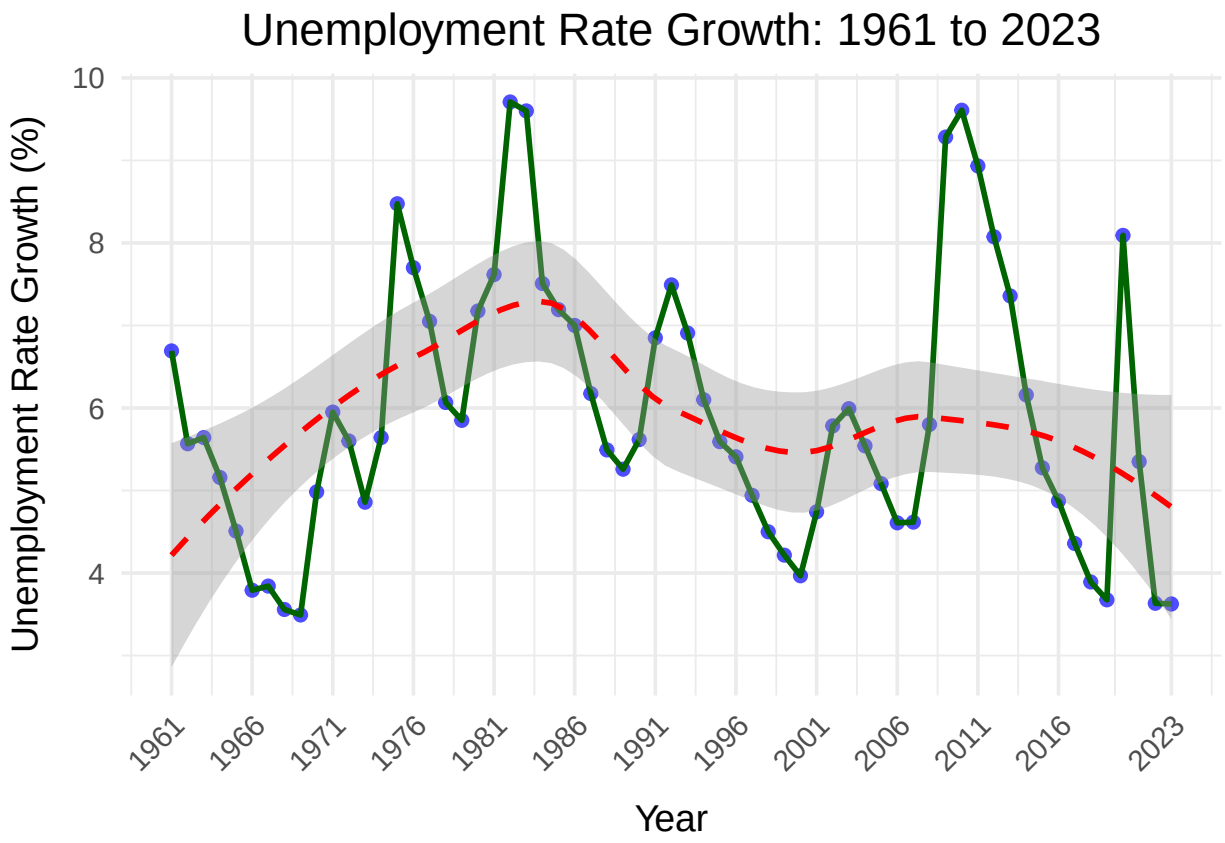
```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/inflation_plot_visualization.jpg",
#  plot = inflation_plot,
#  width = 12,
  #height = 8,
 # dpi = 600
#)
```

```r
#Date format
unrate <- unrate |>
  mutate(DATE = as.Date(DATE))

# DATA VISUALIZATION FOR UNEMPLOYMENT RATE

unrate_plot <- ggplot(unrate, aes(x = DATE, y = UNRATE)) +
  geom_point(color = "blue", alpha = 0.7, size = 2) +
  geom_line(color = "darkgreen", size = 1) +
  scale_x_date(
    date_labels = "%Y",
    breaks = c(seq(as.Date("1961-01-01"), as.Date("2020-01-01"), by = "5 years"),
               as.Date("2023-01-01"))
  ) +
  geom_smooth(method = "loess", color = "red", linetype = "dashed", size = 1) +
  labs(title = "Unemployment Rate Growth: 1961 to 2023",
       x = "Year", y = "Unemployment Rate Growth (%)") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10)),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
unrate_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

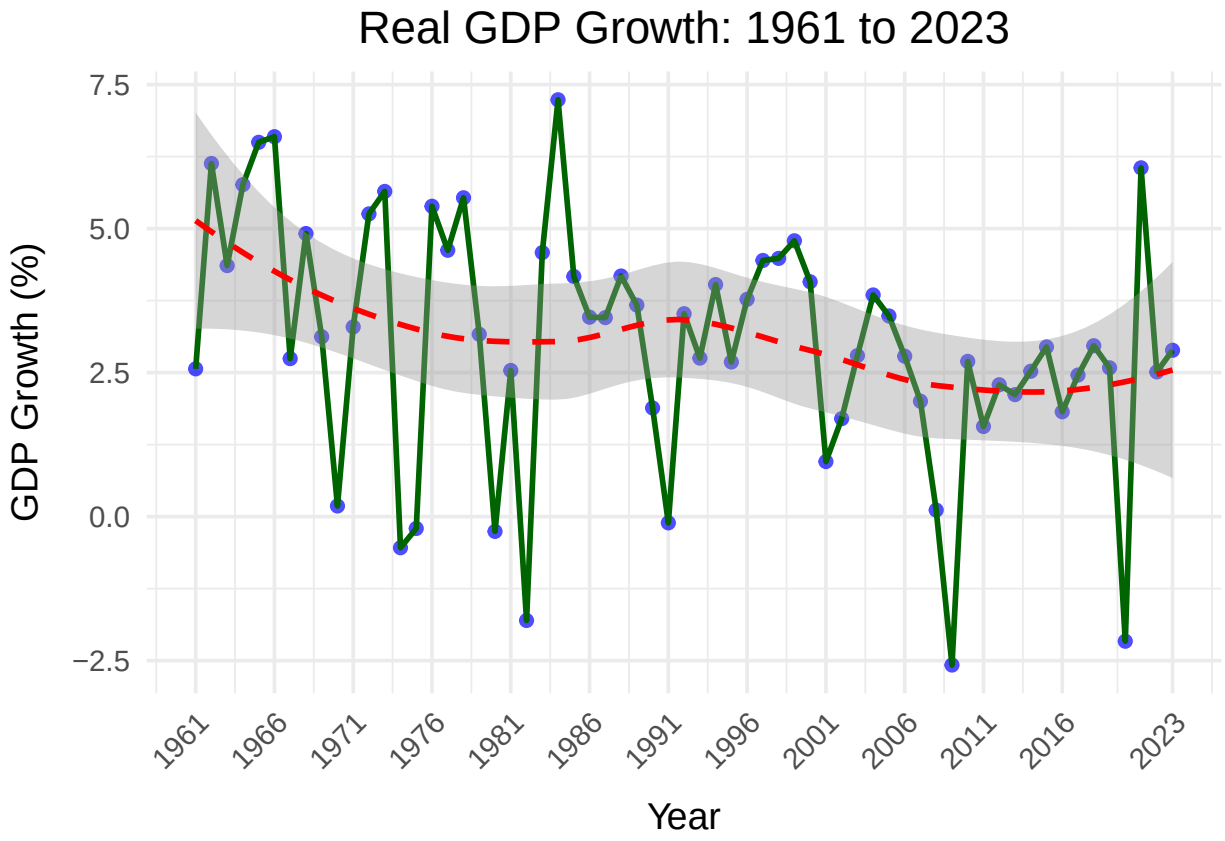



```
#ggsave(
 # filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/unrate_plot_visualization.jpg",
  #plot = unrate_plot,
 # width = 12,
  #height = 8,
  #dpi = 600
#)
```

```
#Date format
real_gdp<- real_gdp |>
  mutate(DATE = as.Date(DATE))

# Real GDP, data visualization

real_gdp_plot <- ggplot(real_gdp, aes(x = DATE, y= GDPC1_PC1)) +
  geom_point(color = "blue", alpha = 0.7, size = 2) +
  geom_line(color = "darkgreen", size = 1) +
  scale_x_date(
    date_labels = "%Y",
    breaks = c(seq(as.Date("1961-01-01"), as.Date("2020-01-01"), by = "5 years"),
               as.Date("2023-01-01"))
  ) +
  geom_smooth(method = "loess", color = "red", linetype = "dashed", size = 1) +
  labs(title = "Real GDP Growth: 1961 to 2023",
       x = "Year", y = "GDP Growth (%)") +
  theme_minimal(base_size = 14) +
```

```
  theme(
    plot.title = element_text(hjust = 0.5),  # Center title
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10)),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
real_gdp_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/real_gdp_plot_visualization.jpg",
#  plot = real_gdp_plot,
#  width = 12,
#  height = 8,
# dpi = 600
#)
```

```
# Combine all data, including US Net Migration Percent Change
combined_data <- real_gdp |>
  select(DATE, GDP_Growth = GDPC1_PC1) |>
  left_join(unrate |> select(DATE, Unemployment_Rate = UNRATE), by = "DATE") |>
  left_join(inflation |> select(DATE, inflation = FPCPITOTLZGUSA), by = "DATE") |>
  left_join(US_Net_migration |> rename(Percentage_Net_Migration_Change = PercentChange), by = "DATE") |
```

```r
  pivot_longer(
    cols = c(GDP_Growth, Unemployment_Rate, inflation, Percentage_Net_Migration_Change),
    names_to = "Indicator",
    values_to = "Value"
  )

# Checking ranges of the indicators, including Net Migration Change
combined_data_range <- combined_data |>
  group_by(Indicator) |>
  summarize(
    Min = min(Value, na.rm = TRUE),
    Max = max(Value, na.rm = TRUE)
  )

combined_data_range
```

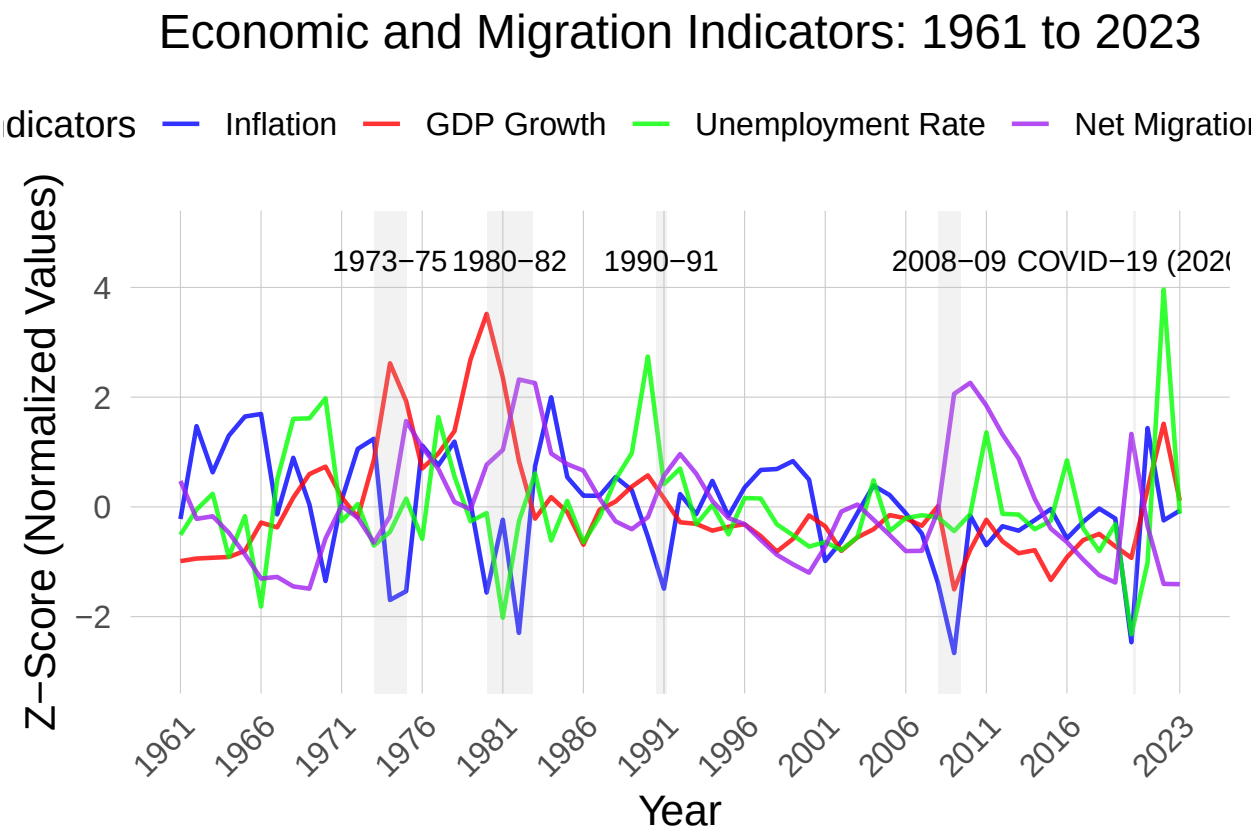
```
## # A tibble: 4 x 3
##   Indicator                          Min   Max
##   <chr>                            <dbl> <dbl>
## 1 GDP_Growth                       -2.58   7.24
## 2 Percentage_Net_Migration_Change -41.7   77.8
## 3 Unemployment_Rate                 3.49   9.71
## 4 inflation                        -0.356 13.5
```

```r
combined_data <- combined_data |>
  group_by(Indicator) |>
  mutate(Normalized_Value = (Value - mean(Value, na.rm = TRUE)) / sd(Value, na.rm = TRUE)) |>
  ungroup()
# Add recession periods, including COVID-19 recession
recession_periods <- data.frame(
  start = as.Date(c("1973-01-01", "1980-01-01", "1990-07-01", "2008-01-01", "2020-02-01")),
  end = as.Date(c("1975-01-01", "1982-11-01", "1991-03-01", "2009-06-01", "2020-04-01")),
  label = c("1973-75", "1980-82", "1990-91", "2008-09", "COVID-19 (2020)")
)

# Highlight high and low points (example for GDP Growth)
highlight_points <- combined_data |>
  filter(Indicator == "GDP Growth" & (Normalized_Value == max(Normalized_Value) |
                                      Normalized_Value == min(Normalized_Value)))

# Enhanced plot with annotations
overlay_plot_normalized <- ggplot(combined_data, aes(x = DATE, y = Normalized_Value, color = Indicator))
  # Add lines
  geom_line(size = 0.8, alpha = 0.8) +
  # Add shaded recession periods
  geom_rect(data = recession_periods,
            aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
            fill = "gray", alpha = 0.2, inherit.aes = FALSE) +
  # Annotate recession periods
  geom_text(data = recession_periods,
            aes(x = start + (end - start) / 2, y = 4.5, label = label),  # Position labels above the li
            color = "black", size = 4, inherit.aes = FALSE) +
  # Highlight high and low points
```

```
  geom_point(data = highlight_points, aes(x = DATE, y = Normalized_Value),
             color = "black", size = 3) +
  # Annotate high and low points
  geom_text(data = highlight_points, aes(x = DATE, y = Normalized_Value,
                                         label = round(Normalized_Value, 2)),
            color = "black", hjust = -0.2, vjust = -0.5, size = 4) +
  # Add labels and formatting
  labs(
    title = "Economic and Migration Indicators: 1961 to 2023",
    x = "Year",
    y = "Z-Score (Normalized Values)",
    color = "Indicators"
  ) +
  scale_color_manual(
    values = c("blue", "red", "green", "purple"),
    labels = c("Inflation", "GDP Growth", "Unemployment Rate", "Net Migration Change")
  ) +
  scale_x_date(
    date_labels = "%Y",
    breaks = c(seq(as.Date("1961-01-01"), as.Date("2020-01-01"), by = "5 years"),
               as.Date("2023-01-01"))
  ) +
  coord_cartesian(ylim = c(-3, 5)) +
  theme_minimal(base_size = 16) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
    axis.text.y = element_text(size = 12),
    legend.position = "top",
    legend.text = element_text(size = 12),
    legend.title = element_text(size = 14),
    panel.grid.major = element_line(size = 0.2, color = "gray80"),
    panel.grid.minor = element_blank()
  )
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Display the enhanced plot
overlay_plot_normalized
```




```
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/normalized_data_all_indicators.jpg"
# plot = overlay_plot_normalized,
 # width = 12,
#  height = 8,
 # dpi = 600
#)
```

```
combined_data_wide <- real_gdp |>
  select(DATE, GDP_Growth = GDPC1_PC1) |>
  left_join(unrate |> select(DATE, Unemployment_Rate = UNRATE), by = "DATE") |>
  left_join(inflation |> select(DATE, inflation = FPCPITOTLZGUSA), by = "DATE") |>
  left_join(US_Net_migration |> rename(Net_Migration = PercentChange), by = "DATE") # Add Net Migration

# Net Migration is the third column
combined_data_wide <- combined_data_wide |>
  select(DATE, GDP_Growth, Net_Migration, Unemployment_Rate, inflation)

# Check structure
head(combined_data_wide)
```

```
##         DATE GDP_Growth Net_Migration Unemployment_Rate inflation
## 1 1961-01-01    2.56534    -7.2042889          6.691667  1.070724
## 2 1962-01-01    6.12961     1.5348415          5.566667  1.198773
## 3 1963-01-01    4.35730     7.0301273          5.641667  1.239669
```

```
## 4 1964-01-01    5.76276   -14.8315462          5.158333  1.278912
## 5 1965-01-01    6.49845    -0.7483312          4.508333  1.585169
## 6 1966-01-01    6.59534   -32.0792577          3.791667  3.015075
```

```
#Summary statistics

library(psych)
```

```
## Warning: package 'psych' was built under R version 4.4.2
```

```
##
## Adjuntando el paquete: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
describe(combined_data_wide) |> slice(-1) |> select(-1) |> mutate("observations"=n)|> select(-trimmed,-m
```

```
## Warning in FUN(newX[, i], ...): ningún argumento finito para min; retornando
## Inf
```

```
## Warning in FUN(newX[, i], ...): ningun argumento finito para max; retornando
## -Inf
```

```
##                   mean    sd median    min   max observations
## GDP_Growth        3.03  2.11   2.97  -2.58  7.24           63
## Net_Migration     2.47 19.02  -1.05 -41.68 77.81           63
## Unemployment_Rate 5.92  1.63   5.62   3.49  9.71           63
## inflation         3.81  2.77   3.03  -0.36 13.55           63
```

```
# Simple regression
simple_model <- lm(GDP_Growth ~ Unemployment_Rate, data = combined_data_wide)

# Summary of the simple regression model
summary(simple_model)
```

```
##
## Call:
## lm(formula = GDP_Growth ~ Unemployment_Rate, data = combined_data_wide)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.215 -1.281  0.164  1.139  4.921
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.7035     0.9509   5.998 1.18e-07 ***
## Unemployment_Rate  -0.4512     0.1549  -2.913  0.00499 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.988 on 61 degrees of freedom
## Multiple R-squared:  0.1221, Adjusted R-squared:  0.1078
## F-statistic: 8.487 on 1 and 61 DF,  p-value: 0.004992
```

```
data_predictors <- combined_data_wide |> data_grid(Unemployment_Rate)|> add_predictions(simple_model)
data_predictors
```

```
## # A tibble: 62 x 2
##    Unemployment_Rate  pred
##                <dbl> <dbl>
##  1              3.49  4.13
##  2              3.56  4.10
##  3              3.62  4.07
##  4              3.63  4.06
##  5              3.68  4.05
##  6              3.79  3.99
##  7              3.84  3.97
##  8              3.89  3.95
##  9              3.97  3.91
## 10              4.22  3.80
## # i 52 more rows
```

```
residuals_unemployment <-combined_data_wide |> select(GDP_Growth,Unemployment_Rate)|> add_residuals(sim
residuals_unemployment
```

```
##    GDP_Growth Unemployment_Rate      resid
## 1     2.56534          6.691667 -0.1186759
## 2     6.12961          5.566667  2.9379565
## 3     4.35730          5.641667  1.1994890
## 4     5.76276          5.158333  2.3868529
## 5     6.49845          4.508333  2.8292412
## 6     6.59534          3.791667  2.6027473
## 7     2.74266          3.841667 -1.2273711
## 8     4.91452          3.558333  0.8166395
## 9     3.12248          3.491667 -1.0054827
## 10    0.18461          4.983333 -3.2702629
## 11    3.29273          5.950000  0.2740493
## 12    5.25550          5.600000  2.0788877
## 13    5.64567          4.858333  2.1343929
## 14   -0.54055          5.641667 -3.6983610
## 15   -0.20561          8.475000 -2.0849264
## 16    5.38802          7.700000  3.1589978
## 17    4.62420          7.050000  2.1018761
## 18    5.53520          6.066667  2.5691632
## 19    3.16599          5.850000  0.1021860
## 20   -0.25677          7.175000 -2.7226898
## 21    2.53769          7.616667  0.2710650
## 22   -1.80301          9.708333 -3.1258052
## 23    4.58378          9.600000  3.2121012
## 24    7.23646          7.508333  4.9209514
## 25    4.16957          7.191667  1.7111708
## 26    3.46266          7.000000  0.9177744
```

```
## 27    3.45462          6.175000  0.5374669
## 28    4.17698          5.491667  0.9514840
## 29    3.67224          5.258333  0.3414562
## 30    1.88596          5.616667 -1.2831318
## 31   -0.10831          6.850000 -2.7208806
## 32    3.52250          7.491667  1.1994708
## 33    2.75179          6.908333  0.1655413
## 34    4.02902          6.100000  1.0780243
## 35    2.68444          5.591667 -0.4959326
## 36    3.77277          5.408333  0.5096713
## 37    4.44713          4.941667  0.9734557
## 38    4.48313          4.500000  0.8101609
## 39    4.78843          4.216667  0.9876115
## 40    4.07759          3.966667  0.1639631
## 41    0.95554          4.741667 -2.6083810
## 42    1.70044          5.783333 -1.3934462
## 43    2.79561          5.991667 -0.2042693
## 44    3.84777          5.541667  0.6448357
## 45    3.48355          5.083333  0.0738004
## 46    2.78453          4.608333 -0.8395555
## 47    2.00386          4.616667 -1.6164652
## 48    0.11358          5.800000 -2.9727857
## 49   -2.57650          9.283333 -4.0910694
## 50    2.69519          9.608333  1.3272715
## 51    1.56441          8.933333 -0.1080911
## 52    2.28911          8.075000  0.2293003
## 53    2.11783          7.358333 -0.2653636
## 54    2.52382          6.158333 -0.4008537
## 55    2.94555          5.275000 -0.3777132
## 56    1.81945          4.875000 -1.6843066
## 57    2.45762          4.358333 -1.2792738
## 58    2.96650          3.891667 -0.9809694
## 59    2.58383          3.675000 -1.4614066
## 60   -2.16303          8.091667 -4.2153192
## 61    6.05506          5.350000  2.7656393
## 62    2.51238          3.633333 -1.5516580
## 63    2.88755          3.625000 -1.1802483
```

```
#This data if for unemployment residuals, you can skeep this part
joined_predictors_residuals<- data_predictors|> left_join(residuals_unemployment)|> select(Unemployment_
```

```
## Joining with `by = join_by(Unemployment_Rate)`
```

```
joined_predictors_residuals
```

```
## # A tibble: 63 x 4
##    Unemployment_Rate GDP_Growth  pred  resid
##                <dbl>      <dbl> <dbl>  <dbl>
##  1              3.49       3.12  4.13 -1.01
##  2              3.56       4.91  4.10  0.817
##  3              3.62       2.89  4.07 -1.18
##  4              3.63       2.51  4.06 -1.55
##  5              3.68       2.58  4.05 -1.46
```

```
##  6              3.79       6.60  3.99  2.60
##  7              3.84       2.74  3.97 -1.23
##  8              3.89       2.97  3.95 -0.981
##  9              3.97       4.08  3.91  0.164
## 10              4.22       4.79  3.80  0.988
## # i 53 more rows
```

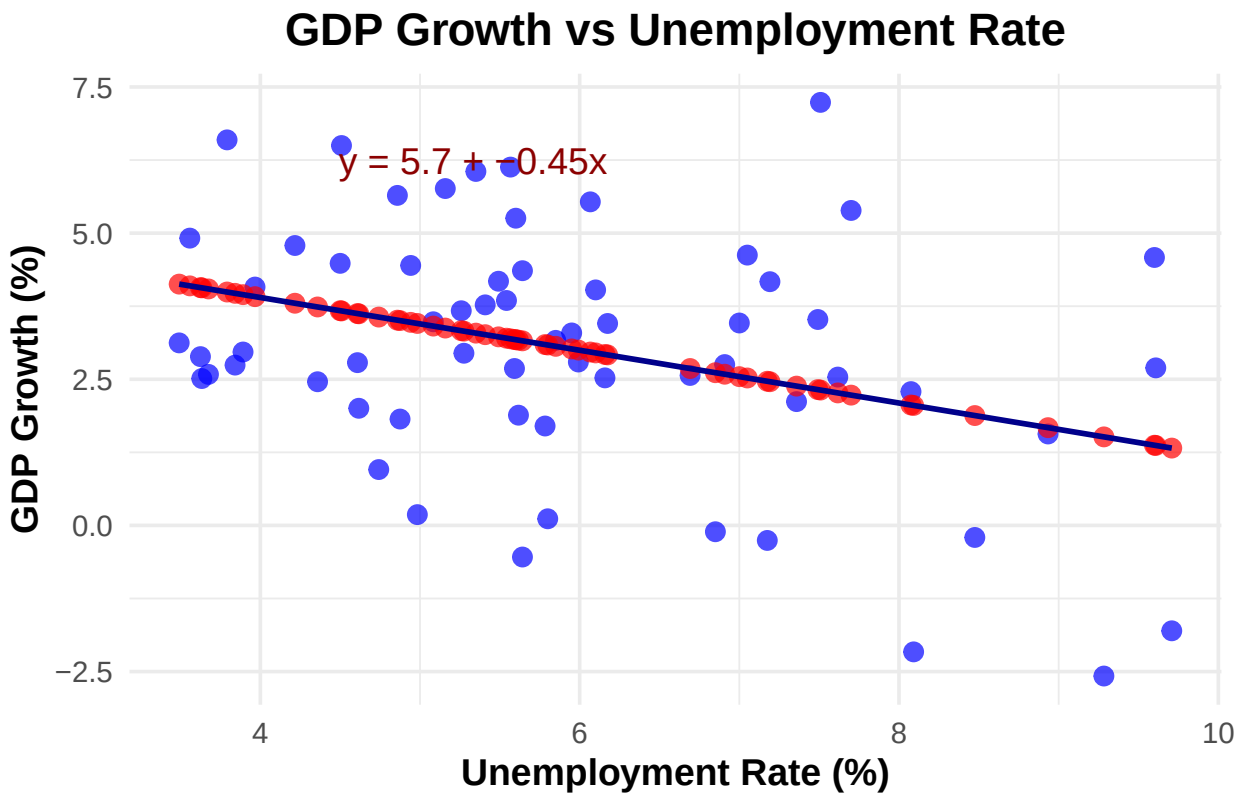
```
residuals_unemployment_density <-
residuals_unemployment |>
  ggplot(aes(x = resid)) +

  geom_density(fill = "blue", alpha = 0.5) +  # Density plot
  geom_vline(xintercept = 0, linetype = "dashed", color = "red", size = 1) +  # Line at 0
  labs(
    title = "Density plot of residuals for unemployment",
    x = "Residuals",
    y = "Density"
  ) +
  theme_minimal(base_size = 14)

residuals_unemployment_density
```


```
#ggsave(
# filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/residuals_uemployment_density.jpg",
  #plot = residuals_unemployment_density,
```

```
  #width = 12,
 # height = 8,
  #dpi = 600
#)
```

```
# Extract coefficients from the simple regression model
coefficients <- coef(simple_model)
equation <- paste0(
  "y = ", round(coefficients[1], 2),
  " + ", round(coefficients[2], 2), "x"
)

# observed, predicted data, and equation
plot_predictors <- ggplot() +
  # Observed points
  geom_point(data = combined_data_wide, aes(Unemployment_Rate, GDP_Growth),
             color = "blue", alpha = 0.7, size = 3) +
  # Predicted points
  geom_point(data = data_predictors, aes(Unemployment_Rate, pred),
             color = "red", alpha = 0.7, size = 3) +
  # Regression line
  geom_smooth(data = combined_data_wide, aes(Unemployment_Rate, GDP_Growth),
              method = "lm", color = "darkblue", se = FALSE) +
  # Add regression equation
  annotate("text",
           x = min(combined_data_wide$Unemployment_Rate) + 1,
           y = max(combined_data_wide$GDP_Growth) - 1,
           label = equation,
           hjust = 0, size = 5, color = "darkred") +
  # Labels and title
  labs(
    title = "GDP Growth vs Unemployment Rate",
    x = "Unemployment Rate (%)",
    y = "GDP Growth (%)",
    caption = "Blue: Observed Data | Red: Predicted Data"
  ) +
  # Minimal theme
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.position = "none"
  )
plot_predictors
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Blue: Observed Data | Red: Predicted Data

```
#ggsave(
# filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/UR_REGRESSION.jpg",
#plot = plot_predictors,
# width = 12,
# height = 8,
# dpi = 600
#)
```

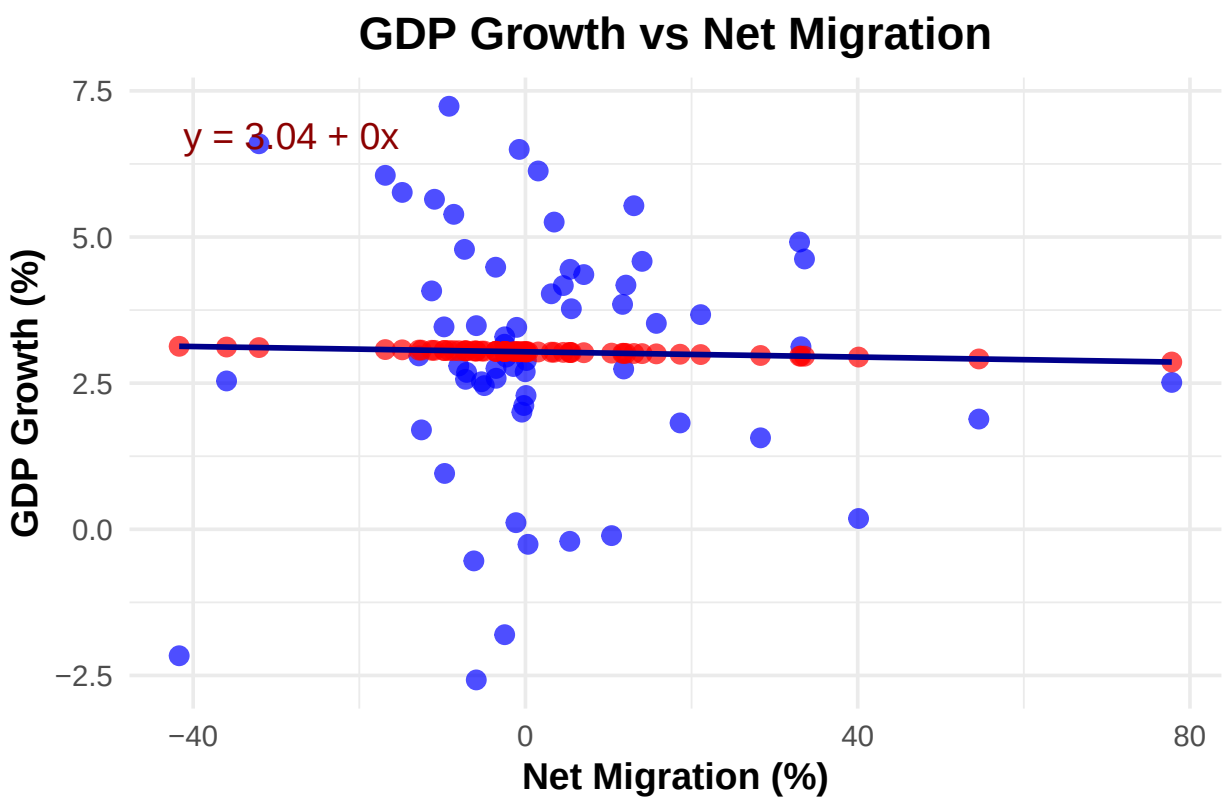
```
# Simple regression
simple_model_net_migration <- lm(GDP_Growth ~ Net_Migration, data = combined_data_wide)

# Summary of the simple regression model
summary(simple_model_net_migration)
```

```
##
## Call:
## lm(formula = GDP_Growth ~ Net_Migration, data = combined_data_wide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6266 -0.8334 -0.0965  1.3794  4.1790
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.03678    0.26960  11.264   <2e-16 ***
```

```
## Net_Migration -0.00225    0.01417  -0.159    0.874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.122 on 61 degrees of freedom
## Multiple R-squared:  0.0004133,  Adjusted R-squared:  -0.01597
## F-statistic: 0.02522 on 1 and 61 DF,  p-value: 0.8743
```

```r
# Extract coefficients from the model
coefficients <- coef(simple_model_net_migration)
equation <- paste0(
  "y = ", round(coefficients[1], 2),
  " + ", round(coefficients[2], 2), "x"
)

# Add predicted values for Net Migration
combined_data_wide <- combined_data_wide |>
  mutate(predicted_net_migration = predict(simple_model_net_migration))

# Plot observed vs. predicted with the regression line
plot_net_migration <- ggplot() +
  # Observed points
  geom_point(data = combined_data_wide, aes(x = Net_Migration, y = GDP_Growth),
             color = "blue", alpha = 0.7, size = 3) +
  # Predicted points
  geom_point(data = combined_data_wide, aes(x = Net_Migration, y = predicted_net_migration),
             color = "red", alpha = 0.7, size = 3) +
  # Regression line
  geom_smooth(data = combined_data_wide, aes(x = Net_Migration, y = GDP_Growth),
              method = "lm", color = "darkblue", se = FALSE, size = 1) +
  # Add regression equation
  annotate("text",
           x = min(combined_data_wide$Net_Migration) + 0.5,
           y = max(combined_data_wide$GDP_Growth) - 0.5,
           label = equation,
           hjust = 0, size = 5, color = "darkred") +
  # Labels and title
  labs(
    title = "GDP Growth vs Net Migration",
    x = "Net Migration (%)",
    y = "GDP Growth (%)",
    caption = "Blue: Observed Data | Red: Predicted Data"
  ) +
  # Minimal theme
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.position = "none"
  )

# Display the plot
print(plot_net_migration)
```

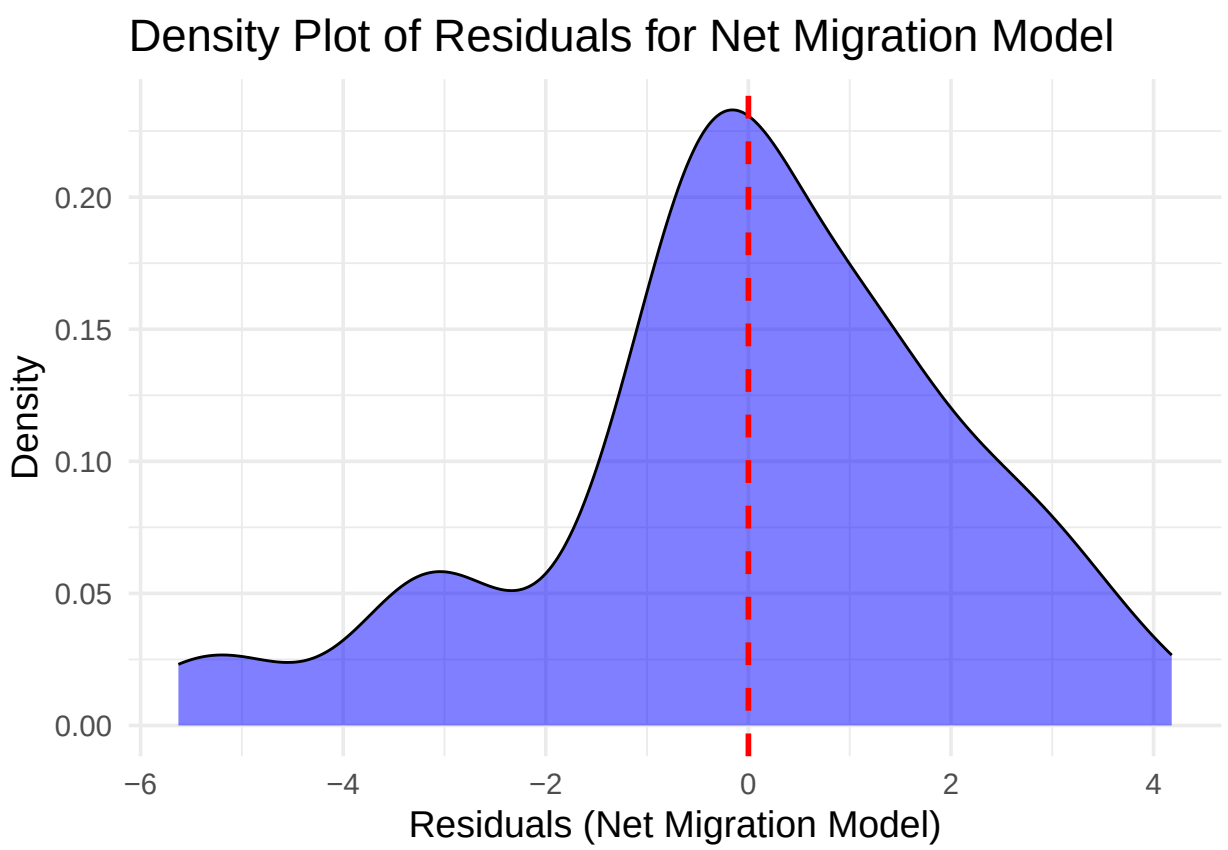
```
## `geom_smooth()` using formula = 'y ~ x'
```



Blue: Observed Data | Red: Predicted Data

```
#ggsave(
 # filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/NET_MIGRATION_REGRESSION.jpg",
 # plot = plot_net_migration,
 # width = 12,
 # height = 8,
  #dpi = 600
#)
```

```
# Calculate residuals for the Net Migration model
combined_data_wide <- combined_data_wide |>
  mutate(residuals_net_migration = GDP_Growth - predicted_net_migration)

# Plot the residuals using a density plot
net_migration_density<-
ggplot(combined_data_wide, aes(x = residuals_net_migration)) +
  geom_density(fill = "blue", alpha = 0.5) +  # Density plot
  geom_vline(xintercept = 0, linetype = "dashed", color = "red", size = 1) +  # Line at 0
  labs(
    title = "Density Plot of Residuals for Net Migration Model",
    x = "Residuals (Net Migration Model)",
    y = "Density"
  ) +
  theme_minimal(base_size = 14)
net_migration_density
```




```
#ggsave(
 # filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/NET_MIGRATION_DENSITY.jpg",
 # plot = net_migration_density,
 # width = 12,
  #height = 8,
  #dpi = 600
#)
```

```
# Simple regression for inflation
simple_model_inflation <- lm(GDP_Growth ~ inflation, data = combined_data_wide)

# Summary of the simple regression model
summary(simple_model_inflation)
```

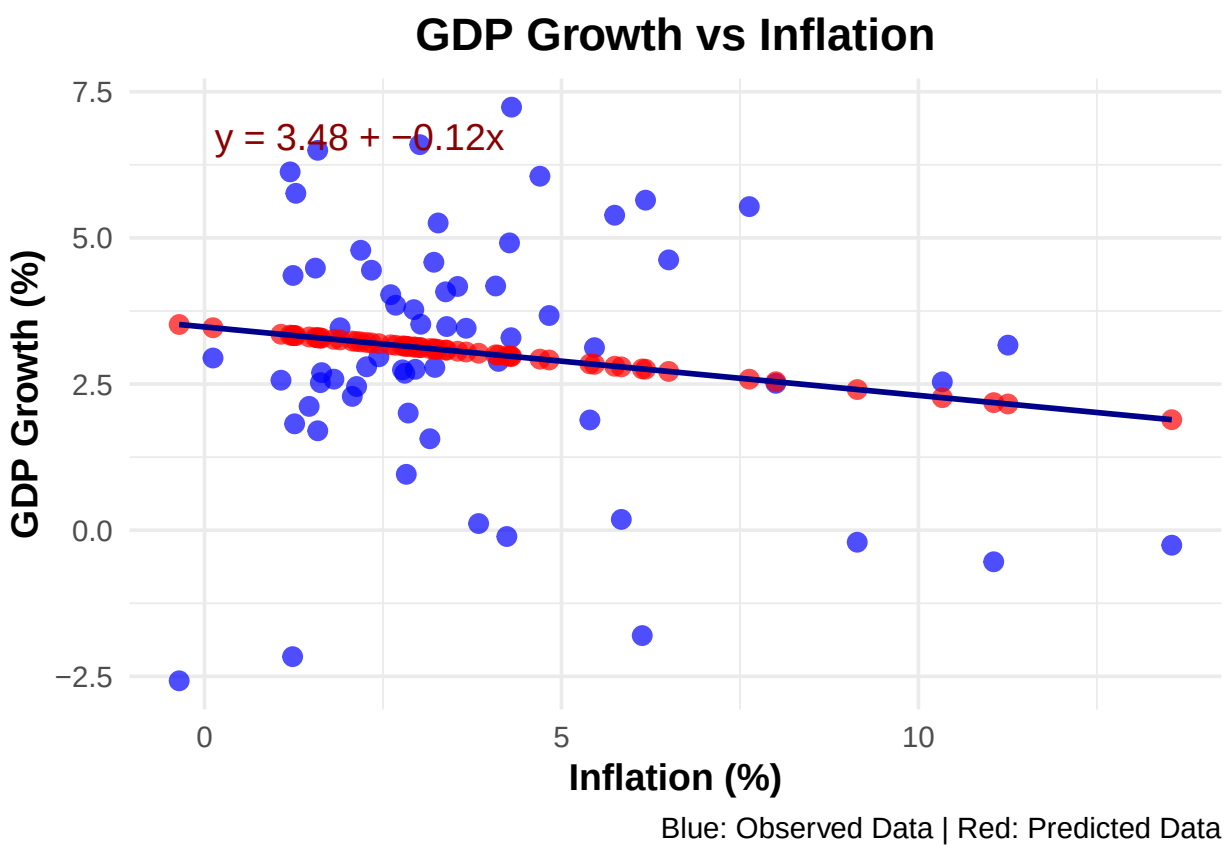
```
##
## Call:
## lm(formula = GDP_Growth ~ inflation, data = combined_data_wide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0952 -0.9523  0.2079  1.1827  4.2630
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.47706    0.45147   7.702 1.44e-10 ***
## inflation   -0.11711    0.09617  -1.218    0.228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.097 on 61 degrees of freedom
## Multiple R-squared:  0.02373,    Adjusted R-squared:  0.007731
## F-statistic: 1.483 on 1 and 61 DF,  p-value: 0.228
```

```r
# Extract coefficients from the model
coefficients_cpi <- coef(simple_model_inflation)
equation_cpi <- paste0(
  "y = ", round(coefficients_cpi[1], 2),
  " + ", round(coefficients_cpi[2], 2), "x"
)

# Add predicted values for CPI Growth
combined_data_wide <- combined_data_wide |>
  mutate(predicted_inflation = predict(simple_model_inflation))

# Plot observed vs. predicted with the regression line
plot_inflation <- ggplot() +
  # Observed points
  geom_point(data = combined_data_wide, aes(x = inflation, y = GDP_Growth),
             color = "blue", alpha = 0.7, size = 3) +
  # Predicted points
  geom_point(data = combined_data_wide, aes(x = inflation, y = predicted_inflation),
             color = "red", alpha = 0.7, size = 3) +
  # Regression line
  geom_smooth(data = combined_data_wide, aes(x = inflation, y = GDP_Growth),
              method = "lm", color = "darkblue", se = FALSE, size = 1) +
  # Add regression equation
  annotate(
    "text",
    x = min(combined_data_wide$inflation) + 0.5,
    y = max(combined_data_wide$GDP_Growth) - 0.5,
    label = equation_cpi,
    hjust = 0, size = 5, color = "darkred"
  ) +
  # Labels and title
  labs(
    title = "GDP Growth vs Inflation",
    x = "Inflation (%)",
    y = "GDP Growth (%)",
    caption = "Blue: Observed Data | Red: Predicted Data"
  ) +
  # Minimal theme
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.position = "none"
  )
```

```r
# Display the plot
print(plot_inflation)
```

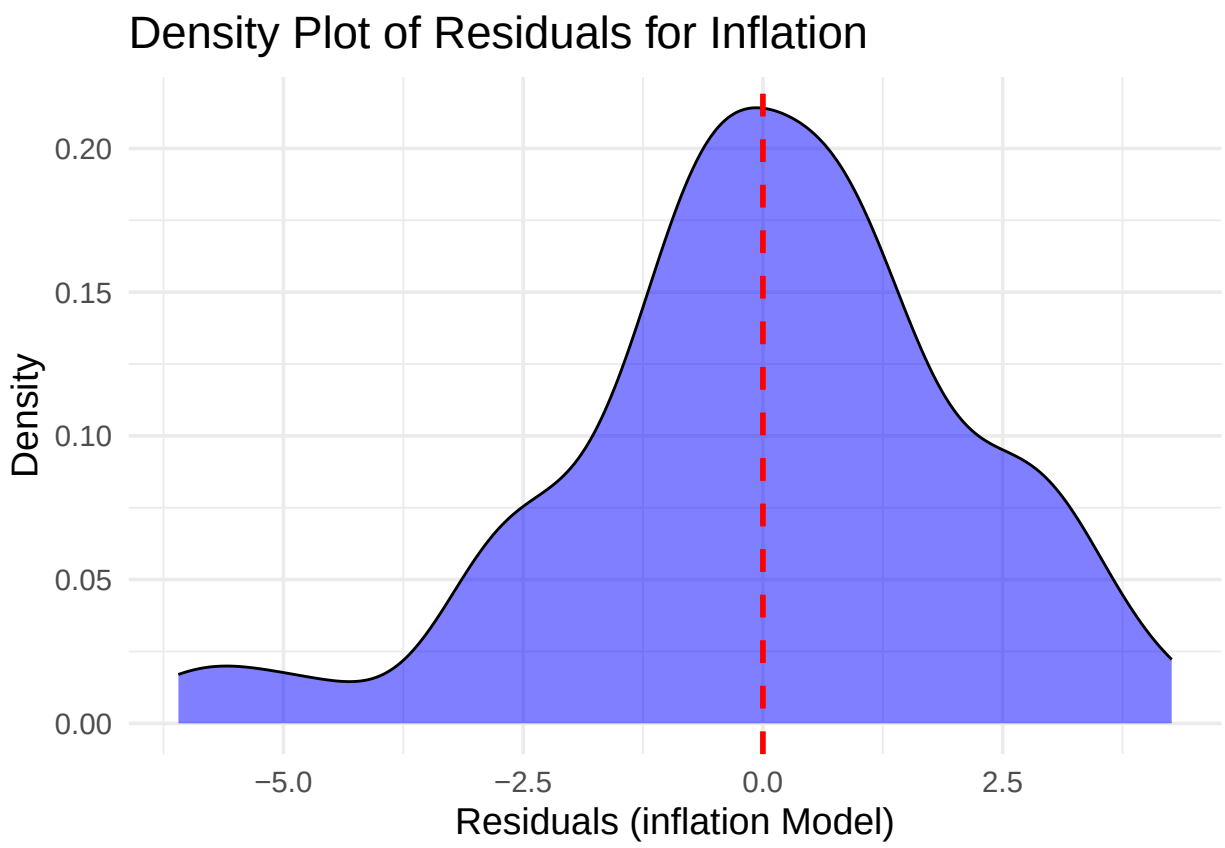
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/plot_regression_inflation.jpg",
#  plot = plot_inflation,
#  width = 12,
#  height = 8,
#  dpi = 600
#)
```

```r
# Calculate residuals for the CPI Growth model
combined_data_wide <- combined_data_wide |>
  mutate(residuals_inflation = GDP_Growth - predicted_inflation)

# Plot the residuals using a density plot
inflation_density<-
ggplot(combined_data_wide, aes(x = residuals_inflation)) +
  geom_density(fill = "blue", alpha = 0.5) +  # Density plot
  geom_vline(xintercept = 0, linetype = "dashed", color = "red", size = 1) +  # Line at 0
  labs(
    title = "Density Plot of Residuals for Inflation",
```

```
    x = "Residuals (inflation Model)",
    y = "Density"
  ) +
  theme_minimal(base_size = 14)
inflation_density
```




```
#ggsave(
 # filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/inflation_density.jpg",
 # plot = inflation_density,
 # width = 12,
 # height = 8,
 # dpi = 600
#)
```

```
# Multiple regression model with CPI Growth, Unemployment Rate, and Net Migration
multi_model <- lm(GDP_Growth ~ Net_Migration + Unemployment_Rate + inflation, data = combined_data_wide)

# Add predicted values from the multiple model to the dataset
combined_data_wide <- combined_data_wide |>
  mutate(Predicted_GDP = predict(multi_model))  # Column for multiple model predictions

head(combined_data_wide)
```

```
##         DATE GDP_Growth Net_Migration Unemployment_Rate inflation
```

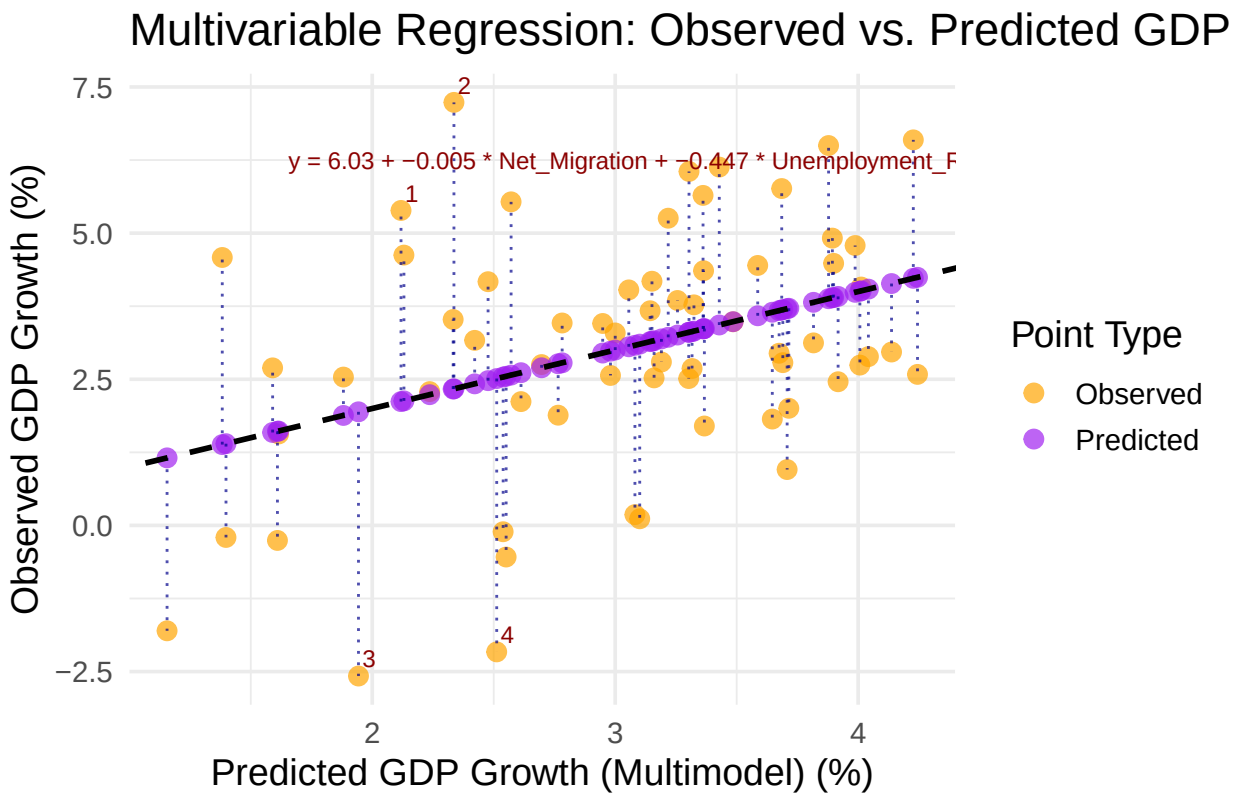
```
## 1 1961-01-01    2.56534    -7.2042889          6.691667  1.070724
## 2 1962-01-01    6.12961     1.5348415          5.566667  1.198773
## 3 1963-01-01    4.35730     7.0301273          5.641667  1.239669
## 4 1964-01-01    5.76276   -14.8315462          5.158333  1.278912
## 5 1965-01-01    6.49845    -0.7483312          4.508333  1.585169
## 6 1966-01-01    6.59534   -32.0792577          3.791667  3.015075
##   predicted_net_migration residuals_net_migration predicted_inflation
## 1                3.052989              -0.4876486            3.351668
## 2                3.033325               3.0962848            3.336672
## 3                3.020961               1.3363395            3.331883
## 4                3.070150               2.6926097            3.327287
## 5                3.038462               3.4599876            3.291421
## 6                3.108958               3.4863815            3.123962
##   residuals_inflation Predicted_GDP
## 1          -0.7863279      2.980564
## 2           2.7929381      3.428356
## 3           1.0254175      3.363733
## 4           2.4354732      3.685422
## 5           3.2070294      3.878248
## 6           3.4713775      4.227008
```

```
# Summary of the multi regression model
summary(multi_model)
```

```
##
## Call:
## lm(formula = GDP_Growth ~ Net_Migration + Unemployment_Rate +
##     inflation, data = combined_data_wide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6753 -1.2177  0.0548  1.0231  4.9002
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.031710   1.001709   6.021 1.19e-07 ***
## Net_Migration     -0.004993   0.013777  -0.362  0.71835
## Unemployment_Rate -0.447011   0.158285  -2.824  0.00646 **
## inflation         -0.089531   0.094042  -0.952  0.34496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.001 on 59 degrees of freedom
## Multiple R-squared:  0.1401, Adjusted R-squared:  0.09634
## F-statistic: 3.203 on 3 and 59 DF,  p-value: 0.0296
```

```
# Ensure no missing data
combined_data_wide <- combined_data_wide |>
  filter(!is.na(Net_Migration), !is.na(Unemployment_Rate), !is.na(inflation))

# Add predicted values and residuals from the multivariable model
combined_data_wide <- combined_data_wide |>
  mutate(
```

```
    Predicted_GDP_Multimodel = predict(multi_model),  # Predicted GDP from the model
    residuals = GDP_Growth - Predicted_GDP_Multimodel  # Residuals (Observed - Predicted)
  )

# Identify outliers by residuals (top 5% largest residuals)
outliers <- combined_data_wide |>
  filter(abs(residuals) > quantile(abs(residuals), 0.95)) |>
  mutate(outlier_id = row_number())  # Assign unique IDs to outliers

# Extract coefficients and create the equation
coefficients <- coef(multi_model)
equation <- paste0(
  "y = ", round(coefficients[1], 2),
  " + ", round(coefficients[2], 4), " * Net_Migration",
  " + ", round(coefficients[3], 4), " * Unemployment_Rate",
  " + ", round(coefficients[4], 4), " * Inflation"
)

# Plot observed vs. predicted with explicit differentiation
multi_plot <- ggplot(combined_data_wide) +
  # Observed values (y-axis)
  geom_point(aes(x = Predicted_GDP_Multimodel, y = GDP_Growth, color = "Observed"),
             alpha = 0.7, size = 3) +
  # Predicted values (x-axis)
  geom_point(aes(x = Predicted_GDP_Multimodel, y = Predicted_GDP_Multimodel, color = "Predicted"),
             alpha = 0.7, size = 3) +
  # Residual lines with blue color
  geom_segment(aes(x = Predicted_GDP_Multimodel, y = Predicted_GDP_Multimodel,
                   xend = Predicted_GDP_Multimodel, yend = GDP_Growth),
               color = "darkblue", linetype = "dotted", alpha = 0.7) +
  # Perfect fit line (45-degree)
  geom_abline(slope = 1, intercept = 0, color = "black", linetype = "dashed", size = 1) +
  # Regression equation
  annotate("text",
           x = min(combined_data_wide$Predicted_GDP_Multimodel) + 0.5,
           y = max(combined_data_wide$GDP_Growth) - 1,
           label = equation,
           hjust = 0, size = 3, color = "darkred") +
  # Highlight outliers
  geom_text(data = outliers, aes(x = Predicted_GDP_Multimodel, y = GDP_Growth, label = outlier_id),
            hjust = -0.3, vjust = -0.5, color = "darkred", size = 3) +
  # Color scale for observed and predicted points
  scale_color_manual(values = c("Observed" = "orange", "Predicted" = "purple"), name = "Point Type") +
  # Labels and title
  labs(
    title = "Multivariable Regression: Observed vs. Predicted GDP Growth",
    x = "Predicted GDP Growth (Multimodel) (%)",
    y = "Observed GDP Growth (%)",
    caption = "Orange: Observed GDP Growth | Purple: Predicted GDP Growth | Blue lines: Residuals"
  ) +
  theme_minimal(base_size = 14)
# Display the plot
print(multi_plot)
```

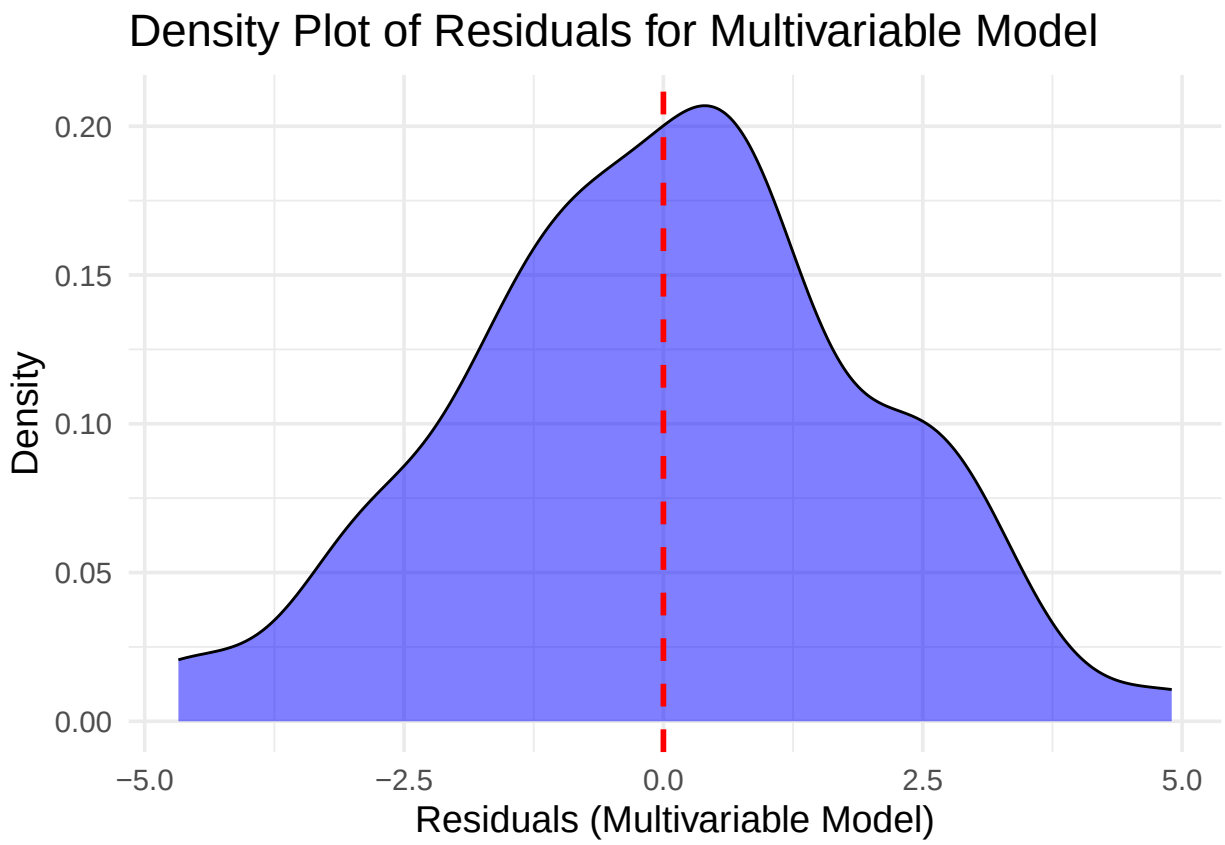


erved GDP Growth | Purple: Predicted GDP Growth | Blue lines: Residuals

```
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/multi_regression_plot_high_quality.
#  plot = multi_plot,
#  width = 12,
#  height = 8,
# dpi = 600
#)
```

```
# Add predicted GDP Growth and calculate residuals
combined_data_wide <- combined_data_wide |>
  mutate(
    Predicted_GDP = predict(multi_model),  # Predicted values
    residuals_multi_model = GDP_Growth - Predicted_GDP  # Residuals
  )

# Plot the residuals using a density plot
residual_densisty <-
ggplot(combined_data_wide, aes(x = residuals_multi_model)) +
  geom_density(fill = "blue", alpha = 0.5) +  # Density plot
  geom_vline(xintercept = 0, linetype = "dashed", color = "red", size = 1) +  # Line at 0
  labs(
    title = "Density Plot of Residuals for Multivariable Model",
    x = "Residuals (Multivariable Model)",
    y = "Density"
  ) +
  theme_minimal(base_size = 14)
```

```
residual_densisty
```




```
#ggsave(
#  filename = "C:/Users/luisc/Desktop/CODE_STATISTICS_ECON57/IMAGES/residual_multimodel.jpg",
 # plot = residual_densisty,
#  width = 12,
 # height = 8,
 # dpi = 600
#)
```