# R Notebook

Haiti GDP composition, the data is from Wolrd Bank

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

```
##
## Attaching package: '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)
data_clean <- read_csv("data_clean.csv")
```

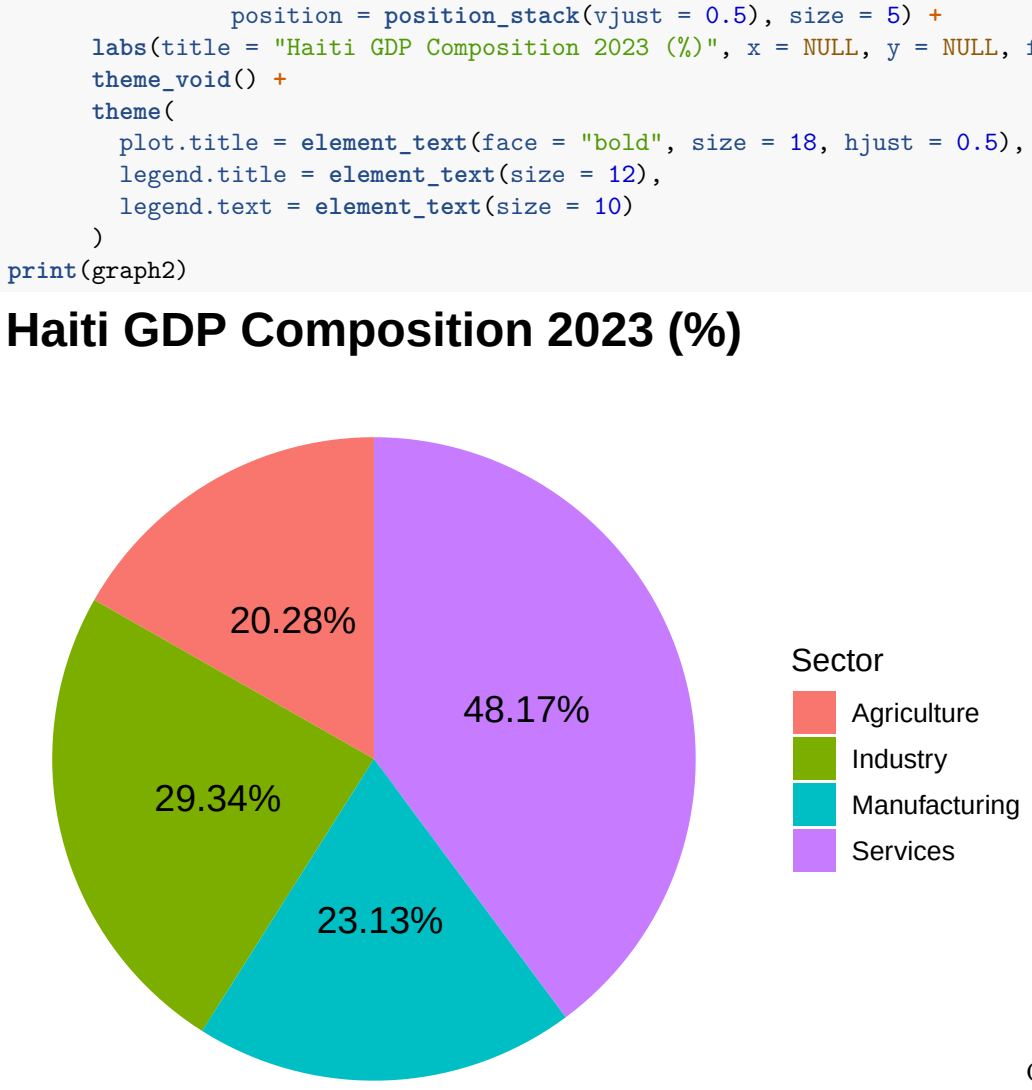
```
## Rows: 450 Columns: 7
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Series_Name, Series_Code, Country_Name, Country_Code, Factor
## dbl (2): Year, Value
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gdp_data <- data.frame(
  Sector = c("Agriculture", "Services", "Industry", "Manufacturing"),
  Percentage = c(20.28, 48.17, 29.34, 23.13)
)

graph1 <- ggplot(gdp_data, aes(x = Sector, y = Percentage, fill = Sector)) +
  geom_bar(stat = "identity", width = 0.7, show.legend = FALSE) +
  geom_text(aes(label = paste0(Percentage, "%")),
            vjust = -0.5, size = 5) +
  scale_y_continuous(limits = c(0, 100)) +
  labs(title = "GDP Composition %", x = "", y = "") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 18, hjust = 0),
    axis.text = element_text(size = 12),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank()
  )
graph2 <- ggplot(gdp_data, aes(x = "", y = Percentage, fill = Sector)) +
      geom_bar(stat = "identity", width = 1) +
      coord_polar(theta = "y") +
```

```
      geom_text(aes(label = paste0(Percentage, "%")),
                position = position_stack(vjust = 0.5), size = 5) +
      labs(title = "Haiti GDP Composition 2023 (%)", x = NULL, y = NULL, fill = "Sector") +
      theme_void() +
      theme(
        plot.title = element_text(face = "bold", size = 18, hjust = 0.5),
        legend.title = element_text(size = 12),
        legend.text = element_text(size = 10)
      )
print(graph2)
```




Comparison of GDP Haiti and

```
unique(data_clean$Factor)
```

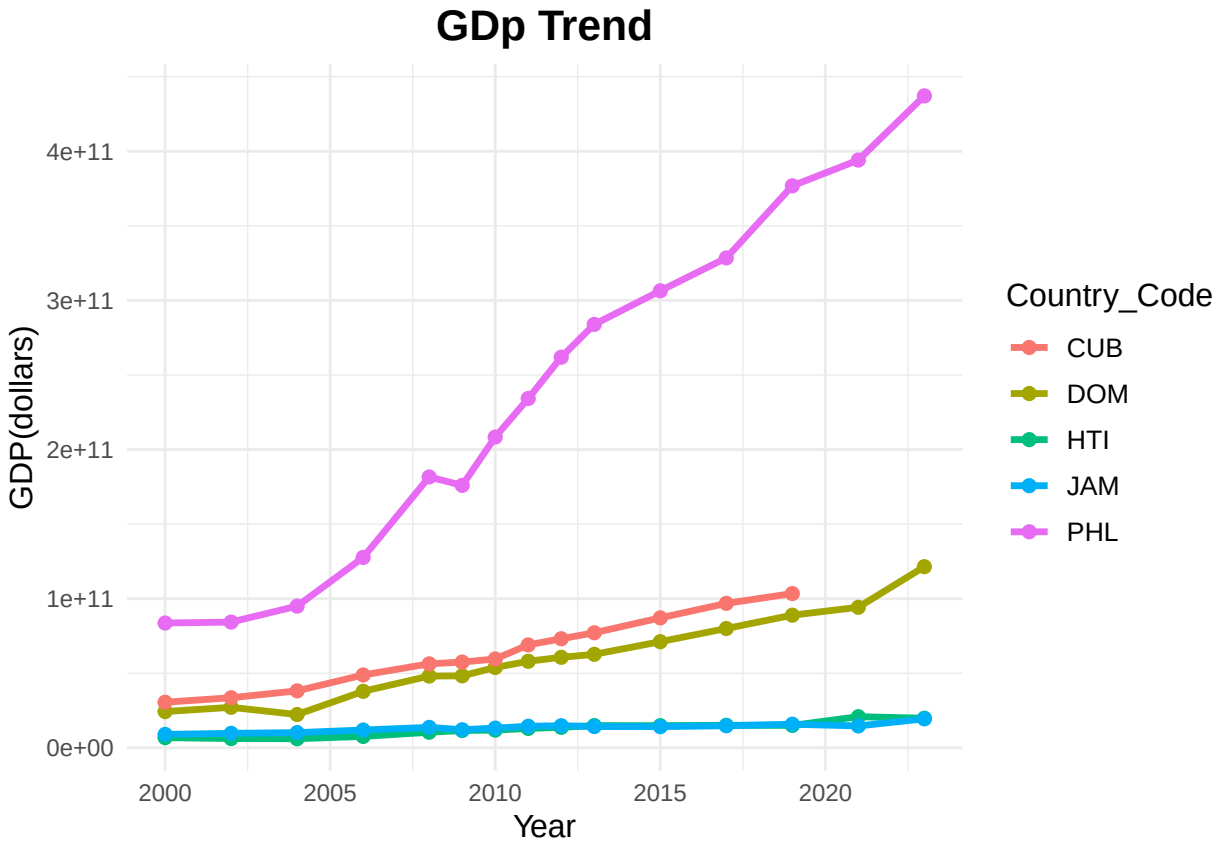
```
## [1] "Life_expectancy"    "GDP"                "Inflation"
## [4] "Foreign_investment" "National_income"    "Unemployment"
```

```
GDP_data <- data_clean[data_clean$Factor == "GDP", ]
ggplot(GDP_data, aes(x = Year, y = Value, color = Country_Code)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "GDp Trend",
       x = "Year", y = "GDP(dollars)") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
## 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.
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




Comparison of life expectency

```
life_expectancy_data <- data_clean[data_clean$Factor == "Life_expectancy", ]

# Create the plot
plot_life <- ggplot(life_expectancy_data, aes(x = Year, y = Value, color = Country_Code)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Life Expectancy Trend",
       x = "Year", y = "Life Expectancy (Years)") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
```

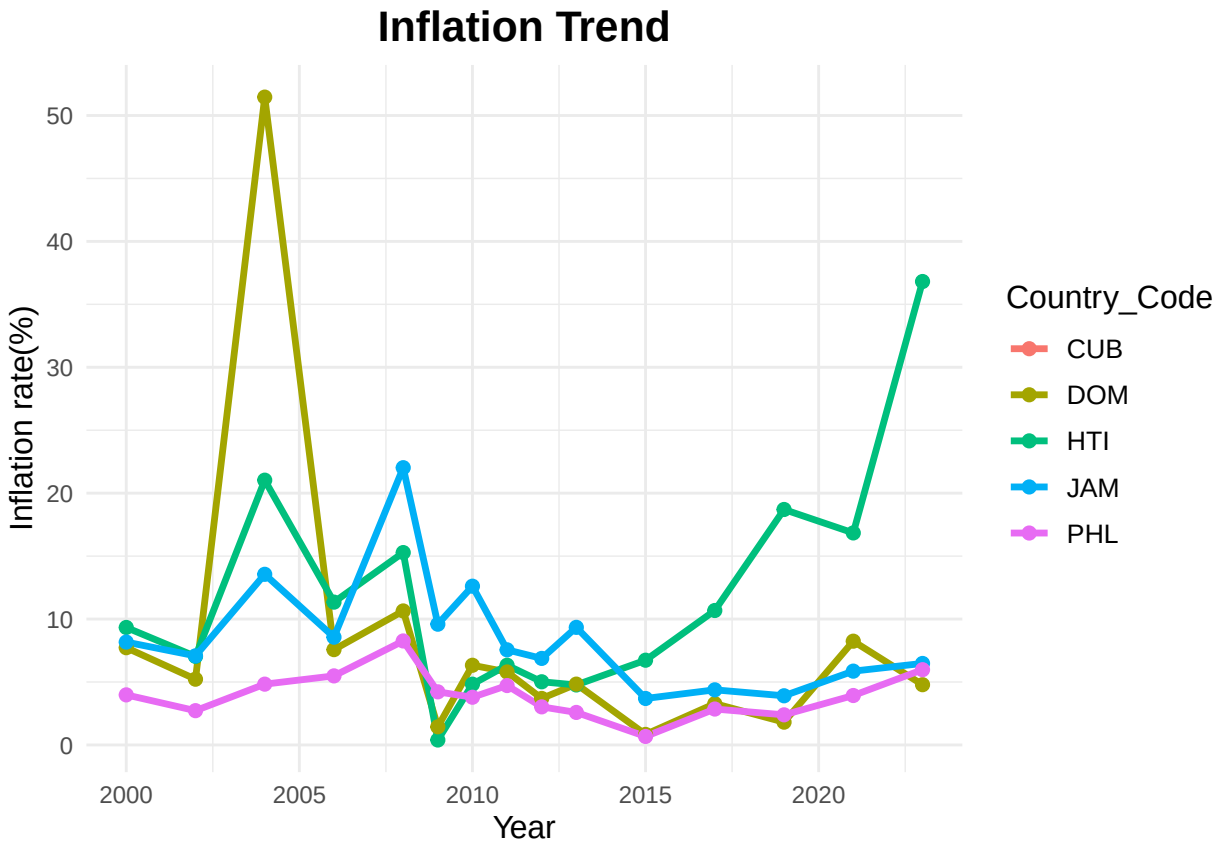
```
  )
```

Comparison of Inflation rate

```
Inflation_data <- data_clean[data_clean$Factor == "Inflation", ]
ggplot(Inflation_data, aes(x = Year, y = Value, color = Country_Code)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Inflation Trend",
       x = "Year", y = "Inflation rate(%)") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



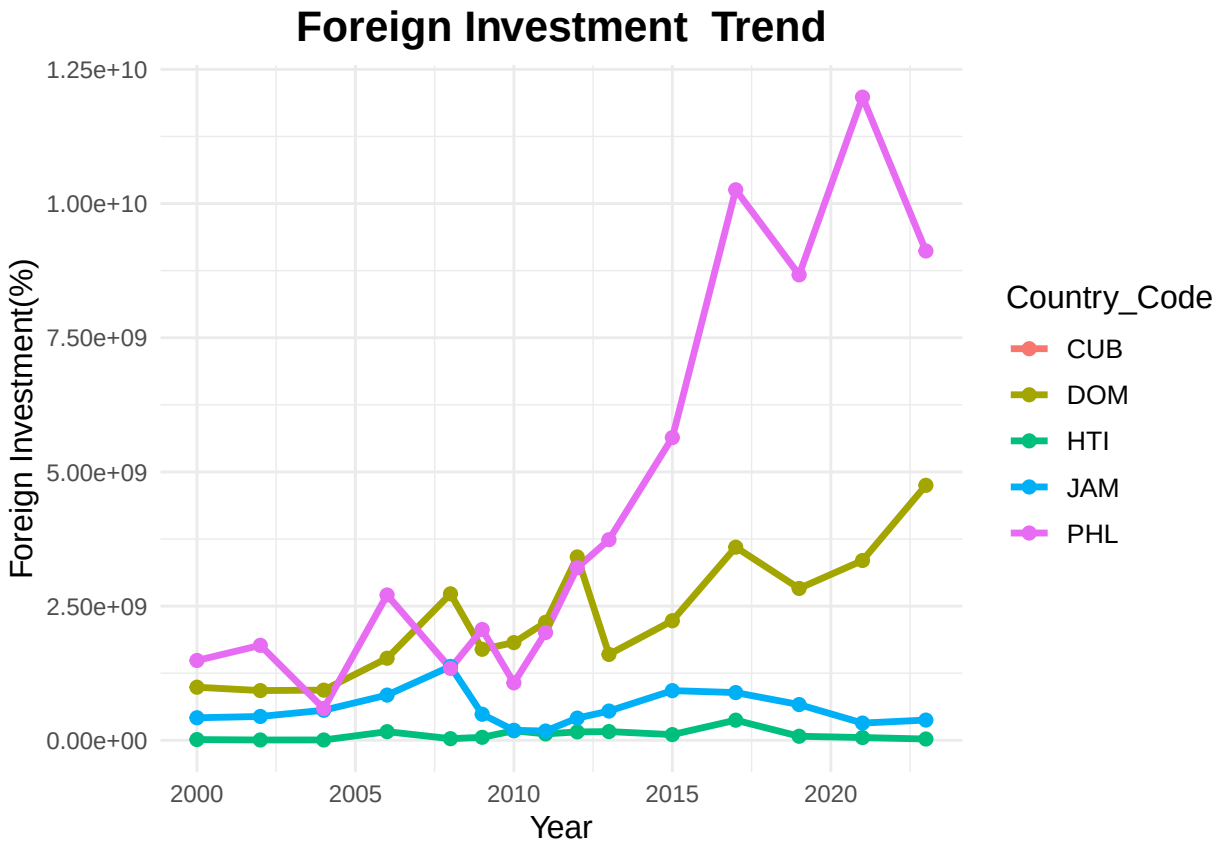

Comparison of Foreign_investment

```
Invest_data <- data_clean[data_clean$Factor == "Foreign_investment", ]
ggplot(Invest_data, aes(x = Year, y = Value, color = Country_Code)) +
  geom_line(size = 1.2) +
```

```
  geom_point(size = 2) +
  labs(title = "Foreign Investment  Trend",
       x = "Year", y = "Foreign Investment(%)") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



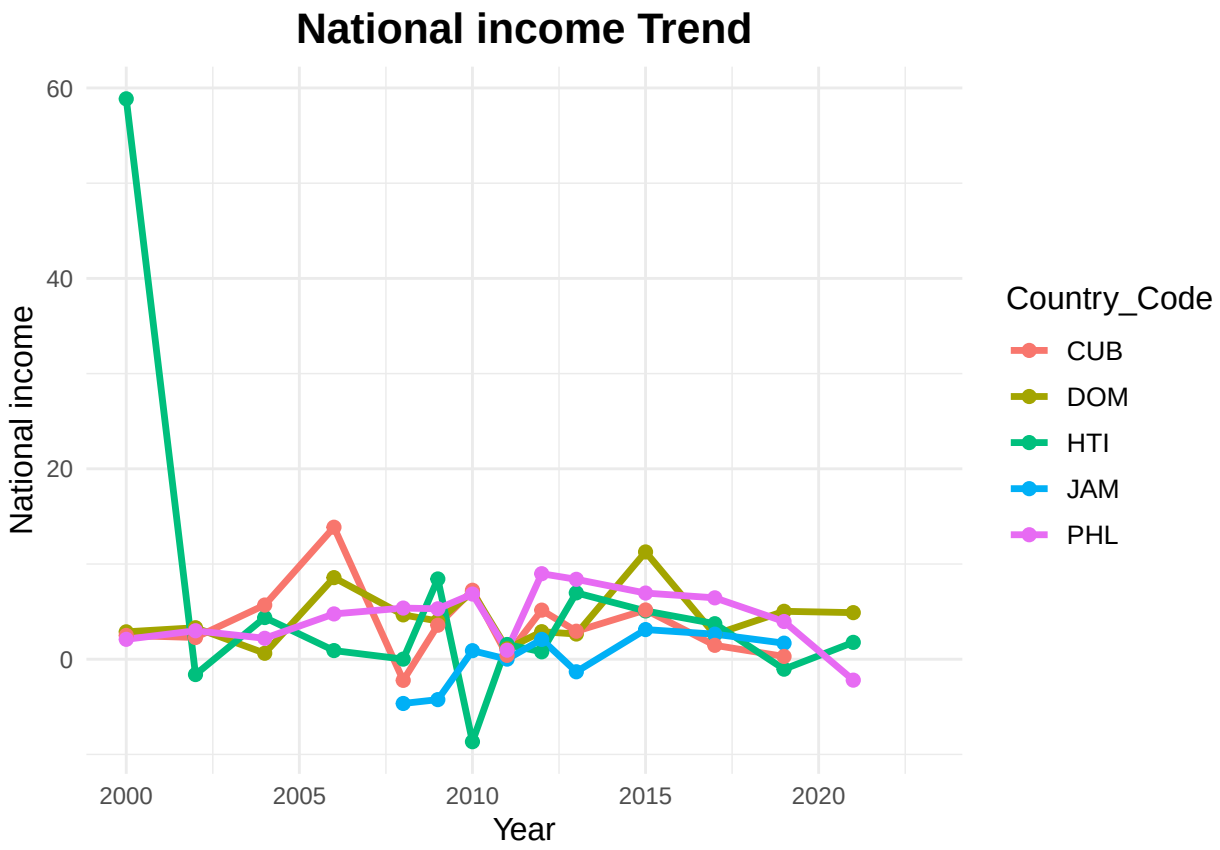

Comparison of Haiti's national income

```
Income_data <- data_clean[data_clean$Factor == "National_income", ]
ggplot(Income_data, aes(x = Year, y = Value, color = Country_Code)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "National income Trend",
       x = "Year", y = "National income") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
```

```
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
## Warning: Removed 11 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 11 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

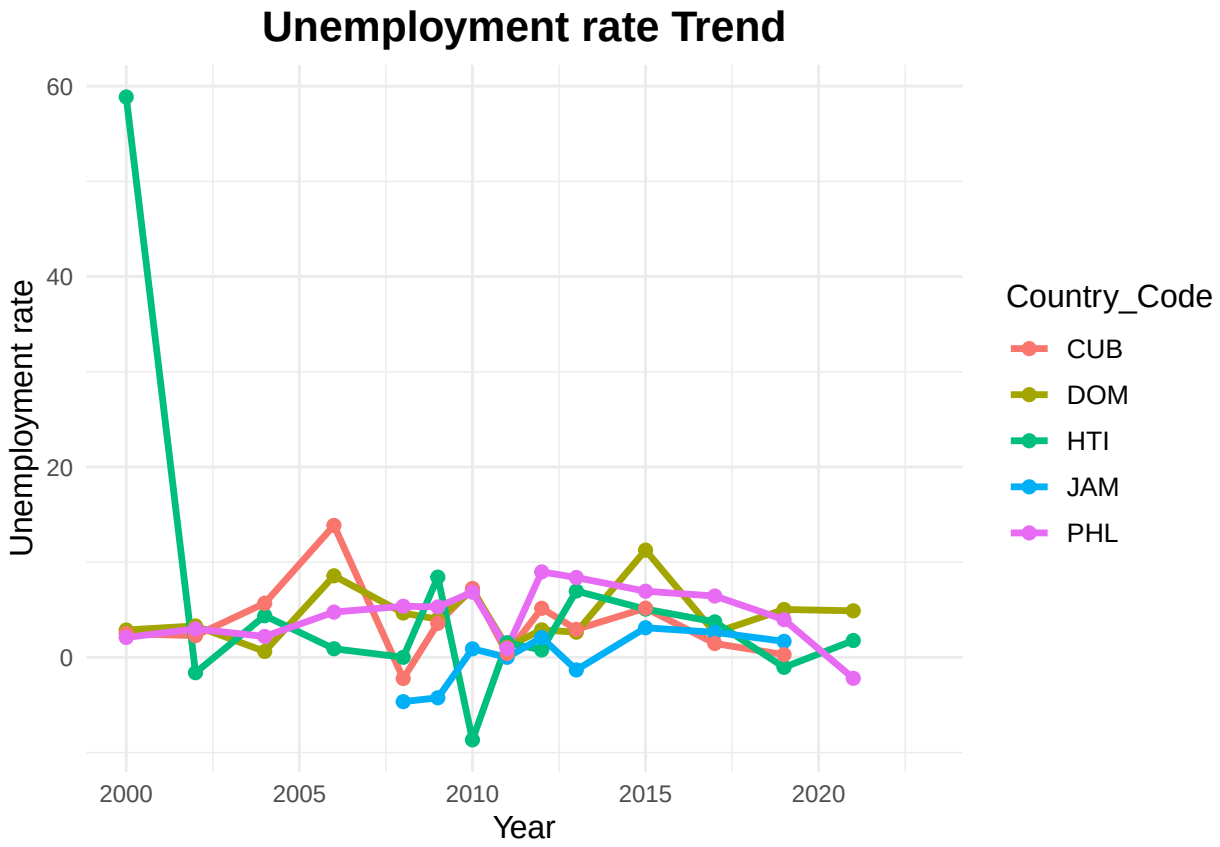



Comparison of Haiti's unemployment rate

```
Job_data <- data_clean[data_clean$Factor == "Unemployment", ]
ggplot(Income_data, aes(x = Year, y = Value, color = Country_Code)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Unemployment rate Trend",
       x = "Year", y = "Unemployment rate") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
## Warning: Removed 11 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 11 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
predict_data <- read.csv("predict_data_clean.csv")
head(predict_data)
```

```
##         Series.Name    Series.Code Country.Name Country.Code Factor Year
## 1 GDP (current US$) NY.GDP.MKTP.CD        Haiti          HTI    GDP 1999
## 2 GDP (current US$) NY.GDP.MKTP.CD        Haiti          HTI    GDP 2000
## 3 GDP (current US$) NY.GDP.MKTP.CD        Haiti          HTI    GDP 2001
## 4 GDP (current US$) NY.GDP.MKTP.CD        Haiti          HTI    GDP 2002
## 5 GDP (current US$) NY.GDP.MKTP.CD        Haiti          HTI    GDP 2003
## 6 GDP (current US$) NY.GDP.MKTP.CD        Haiti          HTI    GDP 2004
##        Value
## 1 4153725884
## 2 6813566099
## 3 6331970324
## 4 6205847214
## 5 5071947798
## 6 6087360684
```

```
GDP_data1 <- predict_data[predict_data$Factor == "GDP", ]
gdp_ts <- ts(GDP_data1$Value, start = 1999, frequency = 1)
library(forecast)
```

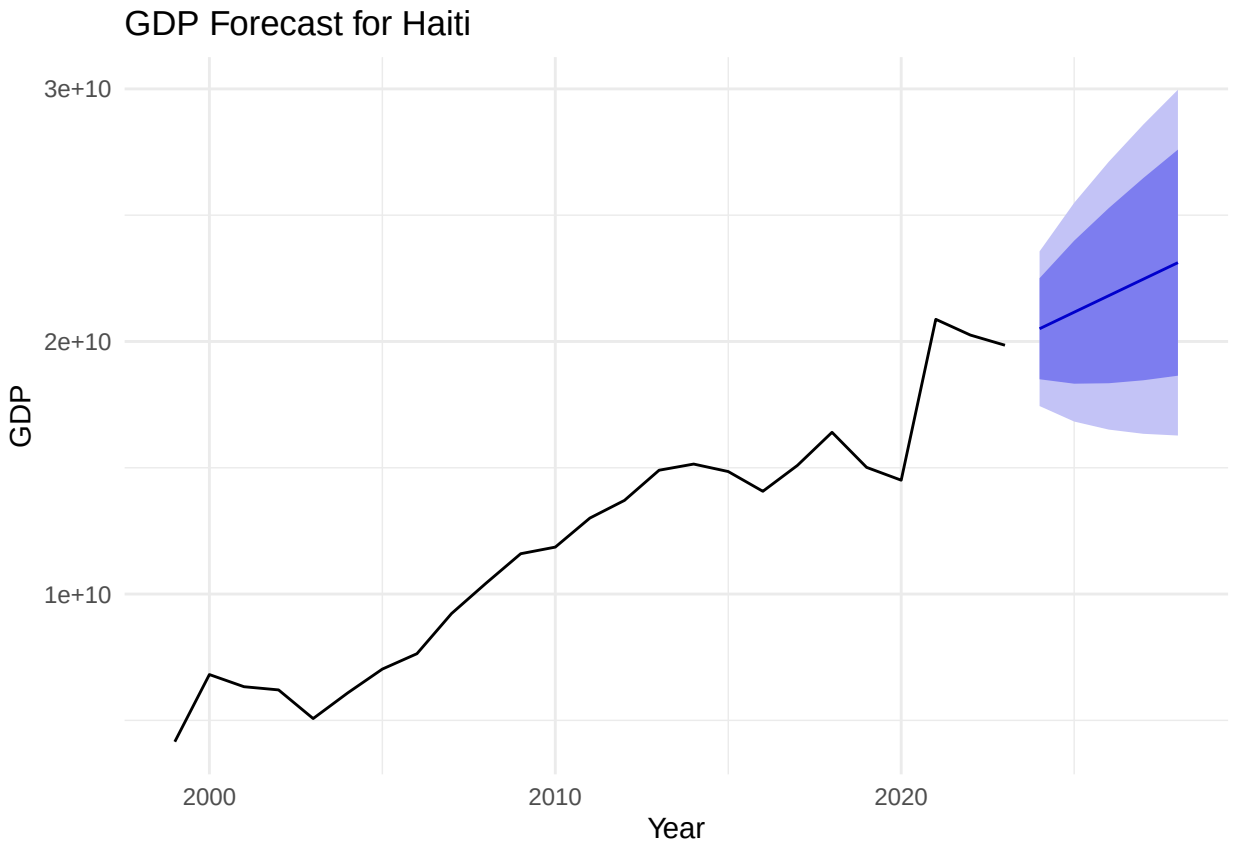
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Fit an ARIMA model
arima_model <- auto.arima(gdp_ts)

# Forecast the next 5 years
gdp_forecast <- forecast(arima_model, h = 5)

# Plot the forecast
gdp_forecast <- forecast(arima_model, h = 5)

# Plot the forecast
autoplot(gdp_forecast) +
  labs(title = "GDP Forecast for Haiti",
       x = "Year", y = "GDP") +
  theme_minimal()
```

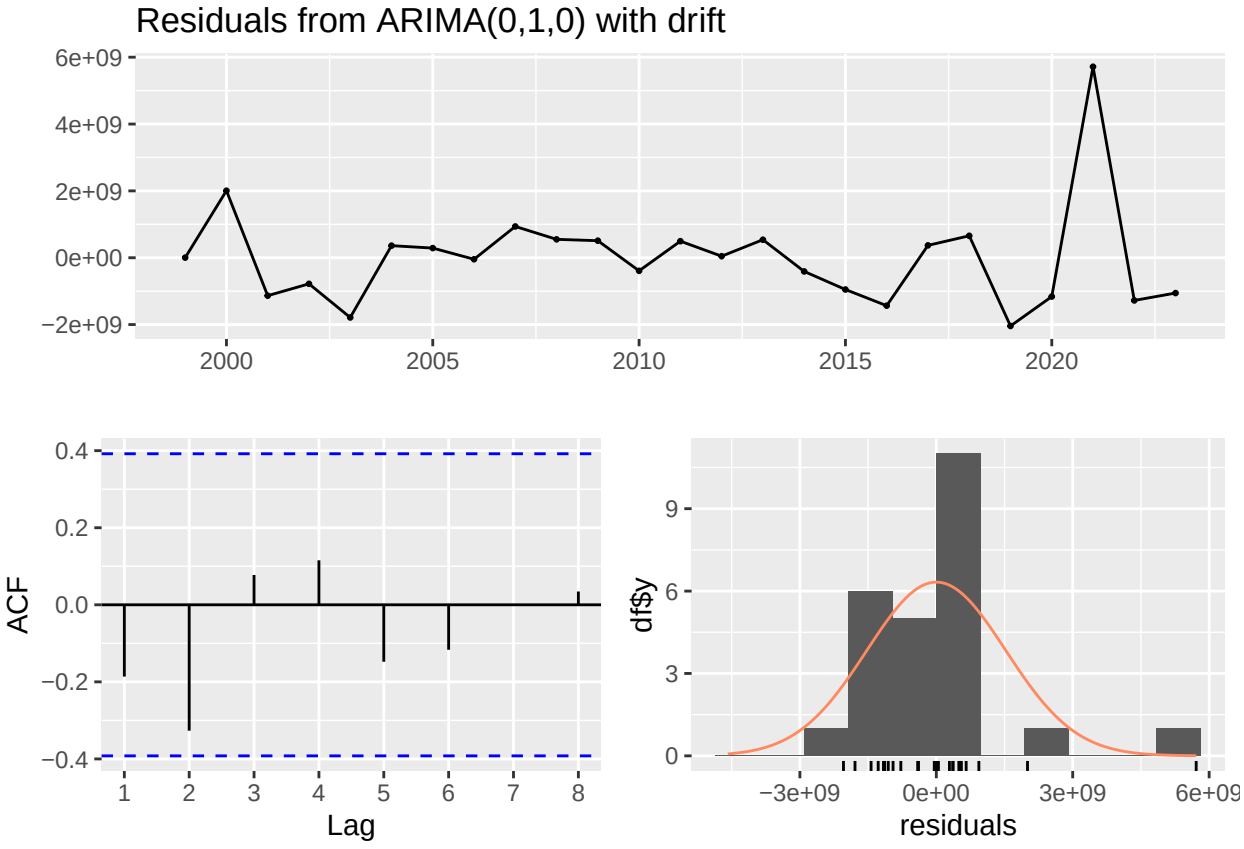



```r
summary(gdp_forecast)
```

```
##
## Forecast method: ARIMA(0,1,0) with drift
##
## Model Information:
## Series: gdp_ts
## ARIMA(0,1,0) with drift
##
## Coefficients:
##           drift
##       654045995
```

```
## s.e.  273448565
##
## sigma^2 = 2.439e+18:  log likelihood = -541.6
## AIC=1087.2   AICc=1087.77   BIC=1089.56
##
## Error measures:
##                     ME       RMSE       MAE        MPE     MAPE      MASE
## Training set 139987.1 1497932814 998116733 -0.8491878 8.926955 0.8813391
##                     ACF1
## Training set -0.1863638
##
## Forecasts:
##      Point Forecast       Lo 80       Hi 80       Lo 95       Hi 95
## 2024    20504875752 18503472936 22506278568 17443994284 23565757221
## 2025    21158921747 18328510741 23989332753 16830181662 25487661832
## 2026    21812967742 18346436378 25279499105 16511365523 27114569961
## 2027    22467013736 18464208105 26469819368 16345250800 28588776673
## 2028    23121059731 18645786985 27596332478 16276720697 29965398765
```

```
checkresiduals(gdp_forecast)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0) with drift
## Q* = 5.4584, df = 5, p-value = 0.3625
##
## Model df: 0.   Total lags used: 5
```

```r
arima_improved <- auto.arima(gdp_ts, d = 1, max.p = 3, max.q = 3)
summary(arima_improved)
```

```
## Series: gdp_ts
## ARIMA(0,1,0) with drift
##
## Coefficients:
##           drift
##       654045995
## s.e.  273448565
##
## sigma^2 = 2.439e+18:  log likelihood = -541.6
## AIC=1087.2   AICc=1087.77   BIC=1089.56
##
## Training set error measures:
##                     ME       RMSE       MAE        MPE     MAPE      MASE
## Training set 139987.1 1497932814 998116733 -0.8491878 8.926955 0.8813391
##                     ACF1
## Training set -0.1863638
```

```r
# Convert Inflation data to a time series
inflationdata1 <- predict_data[predict_data$Factor == "inflation", ]
head(inflationdata1)
```

```
##                              Series.Name    Series.Code Country.Name
## 26 Inflation, consumer prices (annual %) FP.CPI.TOTL.ZG        Haiti
## 27 Inflation, consumer prices (annual %) FP.CPI.TOTL.ZG        Haiti
## 28 Inflation, consumer prices (annual %) FP.CPI.TOTL.ZG        Haiti
## 29 Inflation, consumer prices (annual %) FP.CPI.TOTL.ZG        Haiti
## 30 Inflation, consumer prices (annual %) FP.CPI.TOTL.ZG        Haiti
## 31 Inflation, consumer prices (annual %) FP.CPI.TOTL.ZG        Haiti
##    Country.Code    Factor Year     Value
## 26          HTI inflation 1999  3.004394
## 27          HTI inflation 2000  9.333222
## 28          HTI inflation 2001 13.316722
## 29          HTI inflation 2002  7.032874
## 30          HTI inflation 2003 28.699578
## 31          HTI inflation 2004 21.031834
```

```r
inflation_ts <- ts(inflationdata1$Value, start = 1999, frequency = 1)

# Fit ARIMA model for Inflation
inflation_model <- auto.arima(inflation_ts)

# Forecast the next 5 years for Inflation
inflation_forecast <- forecast(inflation_model, h = 5)

# Combine forecast data into a data frame
inflation_combined <- data.frame(
  Year = c(time(inflation_ts), time(inflation_forecast$mean)),
  Inflation = c(as.numeric(inflation_ts), as.numeric(inflation_forecast$mean)),
  Type = c(rep("Historical", length(inflation_ts)), rep("Forecast", length(inflation_forecast$mean)))
)

# Extract confidence intervals for Inflation
```

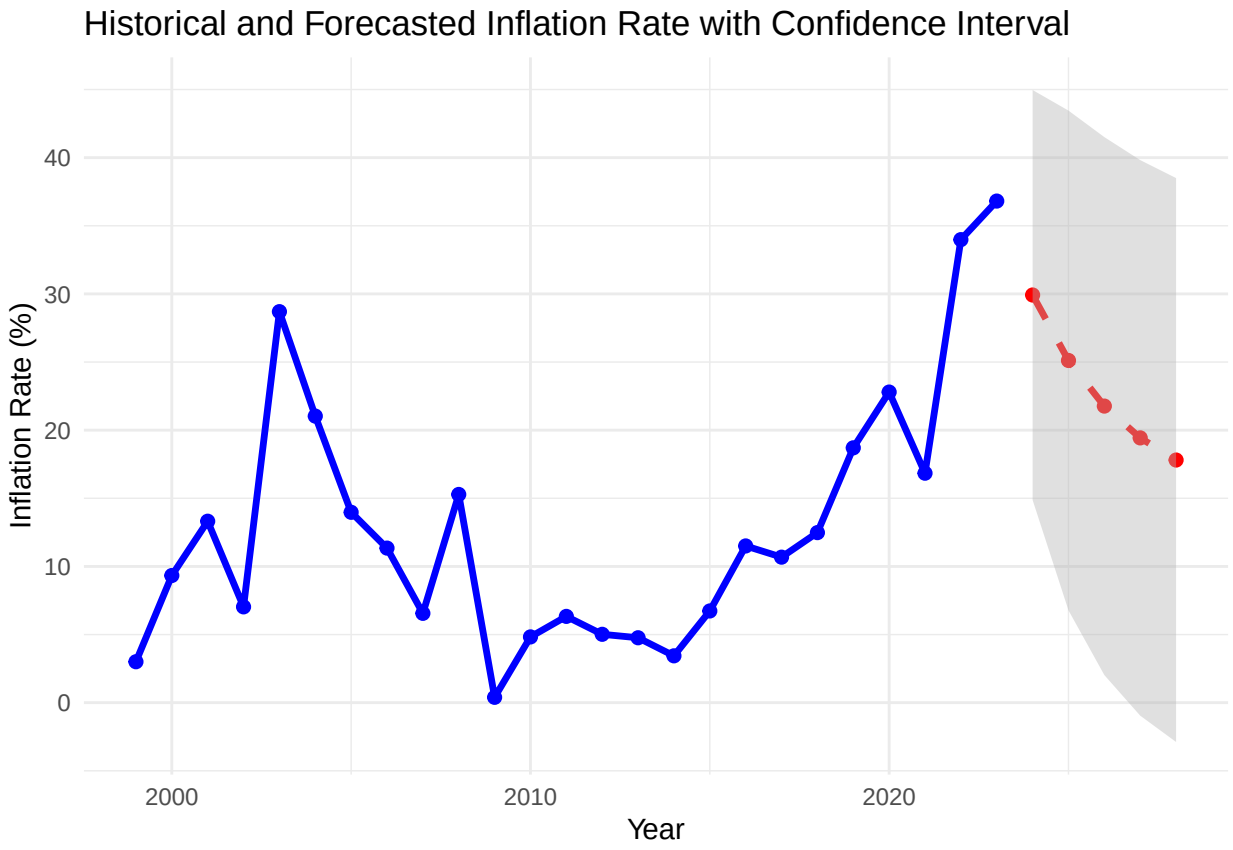
```r
inflation_conf <- data.frame(
  Year = time(inflation_forecast$mean),
  Lower = inflation_forecast$lower[, 2], # 95% lower bound
  Upper = inflation_forecast$upper[, 2]  # 95% upper bound
)

ggplot() +
    # Historical data
    geom_line(data = subset(inflation_combined, Type == "Historical"),
              aes(x = Year, y = Inflation), color = "blue", size = 1.2) +
    geom_point(data = subset(inflation_combined, Type == "Historical"),
               aes(x = Year, y = Inflation), color = "blue", size = 2) +

    # Forecasted data
    geom_line(data = subset(inflation_combined, Type == "Forecast"),
              aes(x = Year, y = Inflation), color = "red", size = 1.2, linetype = "dashed") +
    geom_point(data = subset(inflation_combined, Type == "Forecast"),
               aes(x = Year, y = Inflation), color = "red", size = 2) +

    # Confidence intervals
    geom_ribbon(data = inflation_conf, aes(x = Year, ymin = Lower, ymax = Upper),
                fill = "grey70", alpha = 0.4) +

    # Labels and theme
    labs(title = "Historical and Forecasted Inflation Rate with Confidence Interval",
         x = "Year", y = "Inflation Rate (%)") +
    theme_minimal()
```


Historical and Forecasted Inflation Rate with Confidence Interval
Inflation Rate (%)
40
30
20
10
0
2000
2010
2020
Year