HomeWork1

JAHNAVI GANGU

2024-02-19

```
# Set the CRAN mirror to a reliable source, e.g., USA or Global
options(repos = c(CRAN = "https://cran.rstudio.com/"))

# Now you can install packages or render the reprex




library(reprex)
```

```
## Warning: package 'reprex' was built under R version 4.2.3
```

```
library(fredr)
library(stats)
library(ggplot2)
library(forecast)
```

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

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(TSA)
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

```
##
## Attaching package: 'TSA'
```

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

```
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
library(urca)
```

reprex({

```
fredr_set_key("c387f7cbc3f36a5a52a03d391d17e253")
icnsa_data <- fredr(series_id = "ICNSA")
covariate_data <- fredr(series_id = "UNRATE")
```

```
str(icnsa_data)
```

```
## tibble [3,009 x 5] (S3: tbl_df/tbl/data.frame)
##  $ date          : Date[1:3009], format: "1967-01-07" "1967-01-14" ...
##  $ series_id     : chr [1:3009] "ICNSA" "ICNSA" "ICNSA" "ICNSA" ...
##  $ value         : num [1:3009] 346000 334000 277000 252000 274000 276000 247000 248000 326000 240000
##  $ realtime_start: Date[1:3009], format: "2024-09-09" "2024-09-09" ...
##  $ realtime_end  : Date[1:3009], format: "2024-09-09" "2024-09-09" ...
```

```
head(icnsa_data)
```

```
## # A tibble: 6 x 5
##   date       series_id  value realtime_start realtime_end
##   <date>     <chr>      <dbl> <date>         <date>
## 1 1967-01-07 ICNSA     346000 2024-09-09     2024-09-09
## 2 1967-01-14 ICNSA     334000 2024-09-09     2024-09-09
## 3 1967-01-21 ICNSA     277000 2024-09-09     2024-09-09
## 4 1967-01-28 ICNSA     252000 2024-09-09     2024-09-09
## 5 1967-02-04 ICNSA     274000 2024-09-09     2024-09-09
## 6 1967-02-11 ICNSA     276000 2024-09-09     2024-09-09
```

```
str(covariate_data)
```

```
## tibble [920 x 5] (S3: tbl_df/tbl/data.frame)
##  $ date          : Date[1:920], format: "1948-01-01" "1948-02-01" ...
##  $ series_id     : chr [1:920] "UNRATE" "UNRATE" "UNRATE" "UNRATE" ...
##  $ value         : num [1:920] 3.4 3.8 4 3.9 3.5 3.6 3.6 3.9 3.8 3.7 ...
##  $ realtime_start: Date[1:920], format: "2024-09-06" "2024-09-06" ...
##  $ realtime_end  : Date[1:920], format: "2024-09-06" "2024-09-06" ...
```

```
head(covariate_data)
```

```
## # A tibble: 6 x 5
##   date       series_id value realtime_start realtime_end
##   <date>     <chr>     <dbl> <date>         <date>
## 1 1948-01-01 UNRATE      3.4 2024-09-06     2024-09-06
## 2 1948-02-01 UNRATE      3.8 2024-09-06     2024-09-06
## 3 1948-03-01 UNRATE      4   2024-09-06     2024-09-06
## 4 1948-04-01 UNRATE      3.9 2024-09-06     2024-09-06
## 5 1948-05-01 UNRATE      3.5 2024-09-06     2024-09-06
## 6 1948-06-01 UNRATE      3.6 2024-09-06     2024-09-06
```

```
# Handle missing values (consider alternative methods if needed)
missing_icnsa <- colSums(is.na(icnsa_data))
missing_covariate <- colSums(is.na(covariate_data))
```

```r
cat("Missing values in ICNSA data:\n")
```

```
## Missing values in ICNSA data:
```

```r
print(missing_icnsa)
```

```
##           date      series_id          value realtime_start   realtime_end
##              0              0              0              0              0
```

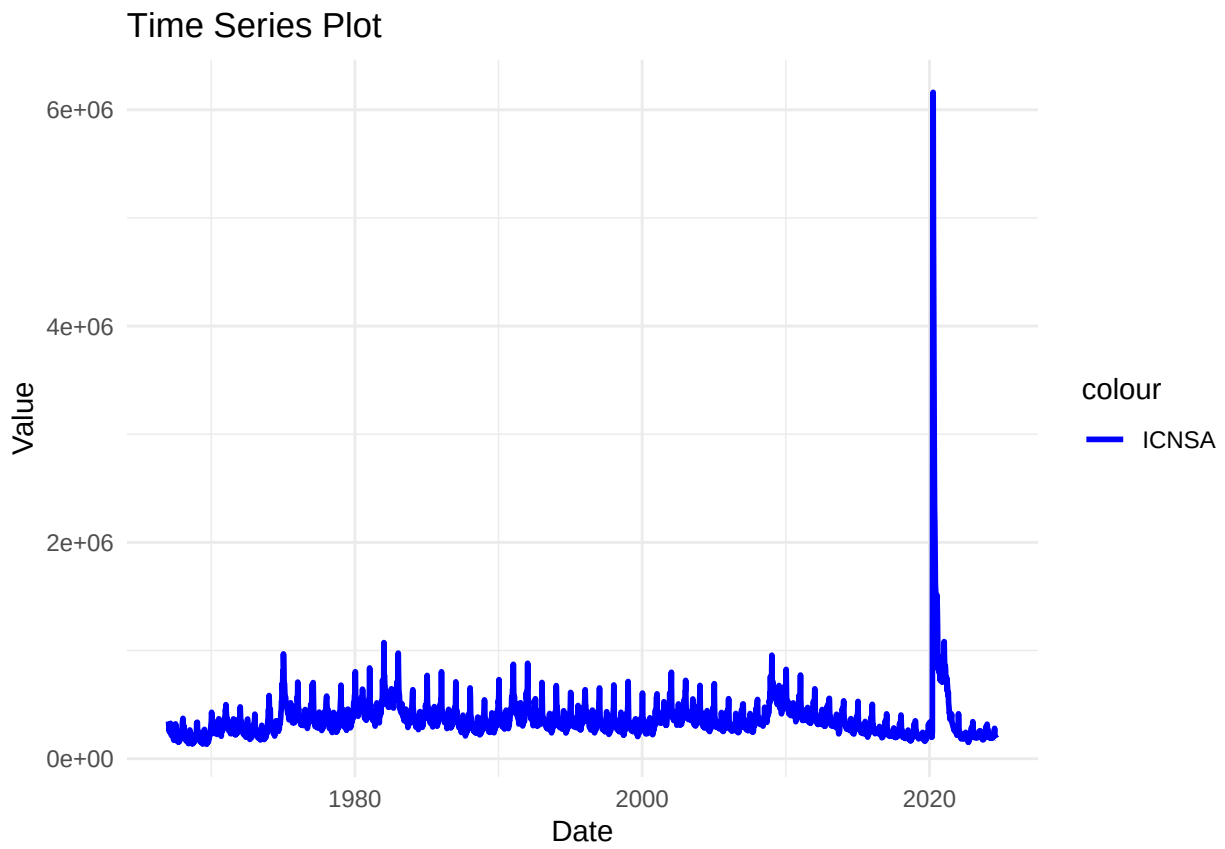
```r
cat("\nMissing values in Covariate data:\n")
```

```
##
## Missing values in Covariate data:
```

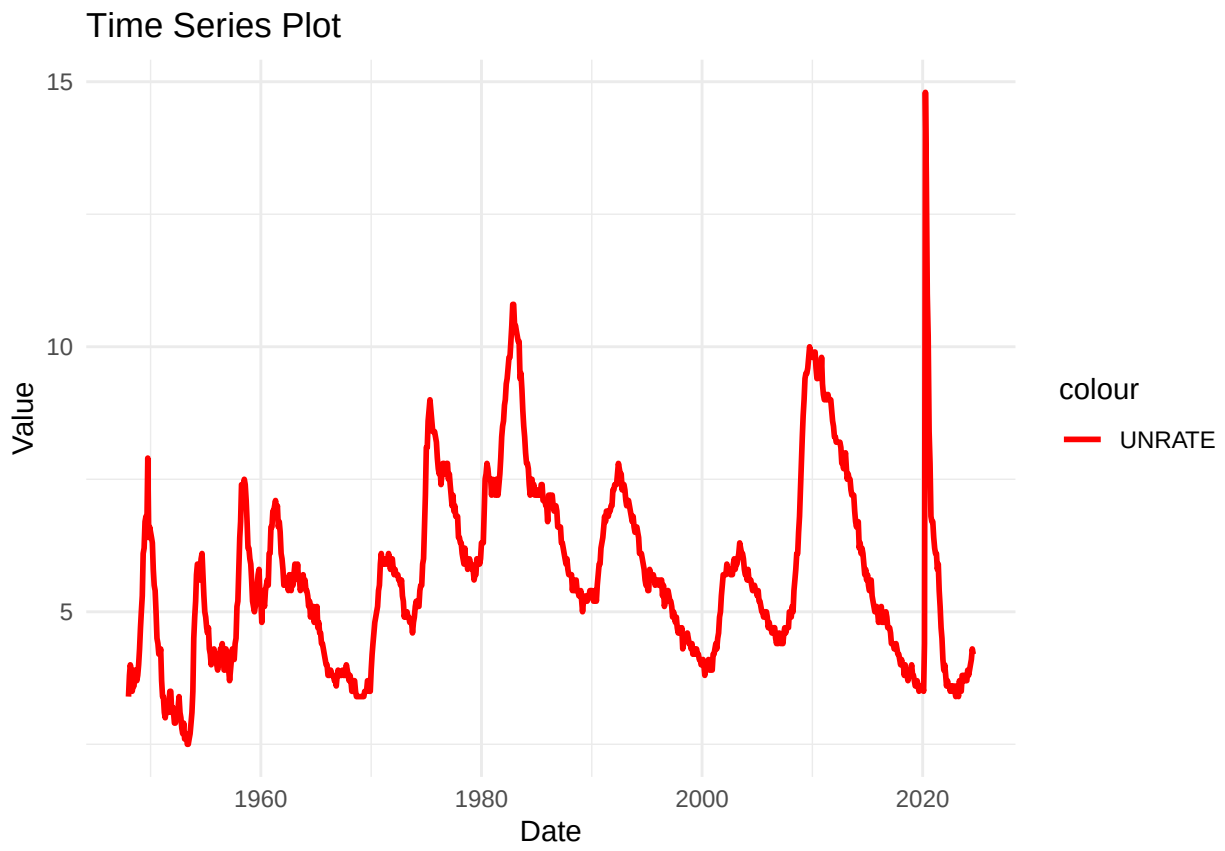
```r
print(missing_covariate)
```

```
##           date      series_id          value realtime_start   realtime_end
##              0              0              0              0              0
```

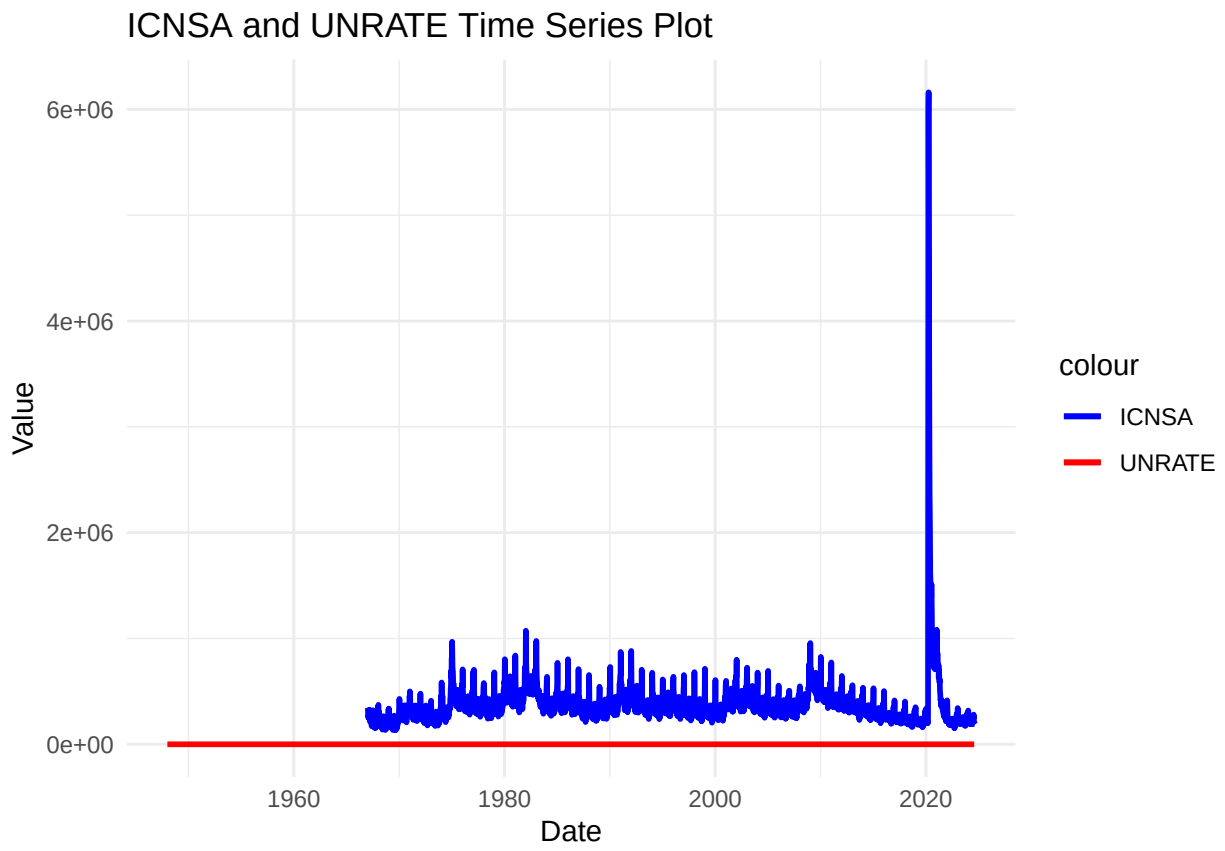
```r
library(ggplot2)

# Convert data frames to regular data frames (if necessary)
icnsa_data <- as.data.frame(icnsa_data)
covariate_data <- as.data.frame(covariate_data)

# Plotting the data
ggplot() +
  geom_line(data = icnsa_data, aes(x = date, y = value, color = "ICNSA"), linewidth = 1) +
  labs(title = "Time Series Plot", x = "Date", y = "Value") +
  scale_color_manual(values = c("ICNSA" = "blue")) +
  theme_minimal()
```




```
# Plotting the data
ggplot() +
  geom_line(data = covariate_data, aes(x = date, y = value, color = "UNRATE"), linewidth = 1) +
  labs(title = "Time Series Plot", x = "Date", y = "Value") +
  scale_color_manual(values = c("UNRATE" = "red")) +
  theme_minimal()
```




```
ggplot() +
  geom_line(data = icnsa_data, aes(x = date, y = value, color = "ICNSA"), linewidth = 1) +
  geom_line(data = covariate_data, aes(x = date, y = value, color = "UNRATE"), linewidth = 1) +
  labs(title = "ICNSA and UNRATE Time Series Plot", x = "Date", y = "Value") +
  scale_color_manual(values = c("ICNSA" = "blue", "UNRATE" = "red")) +
  theme_minimal()
```




```
# Correlation analysis
merged_data <- merge(icnsa_data, covariate_data, by = "date")
correlation <- cor(merged_data$value.x, merged_data$value.y)
cat("Correlation between ICNSA and UNRATE:", correlation)
```

```
## Correlation between ICNSA and UNRATE: 0.6367814
```

```
# Print correlation coefficient
print(correlation)
```

```
## [1] 0.6367814
```

```
# Define a function for ADF test
adf_test <- function(series) {
  result <- adf.test(series)
  cat("ADF Statistic:", result$statistic, "\n")
  cat("p-value:", result$p.value, "\n")
}

# Perform ADF test for ICNSA
adf_test(icnsa_data$value)
```

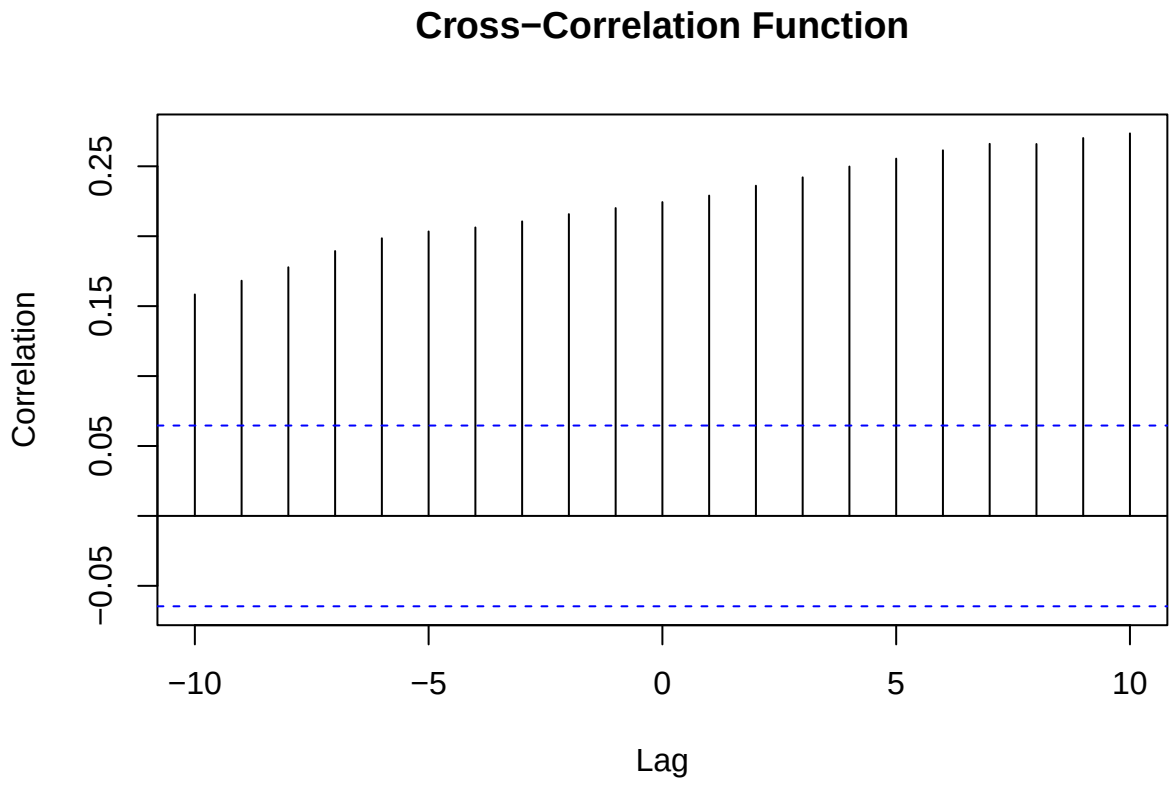
```
## Warning in adf.test(series): p-value smaller than printed p-value
```

```
## ADF Statistic: -8.860967
## p-value: 0.01
```

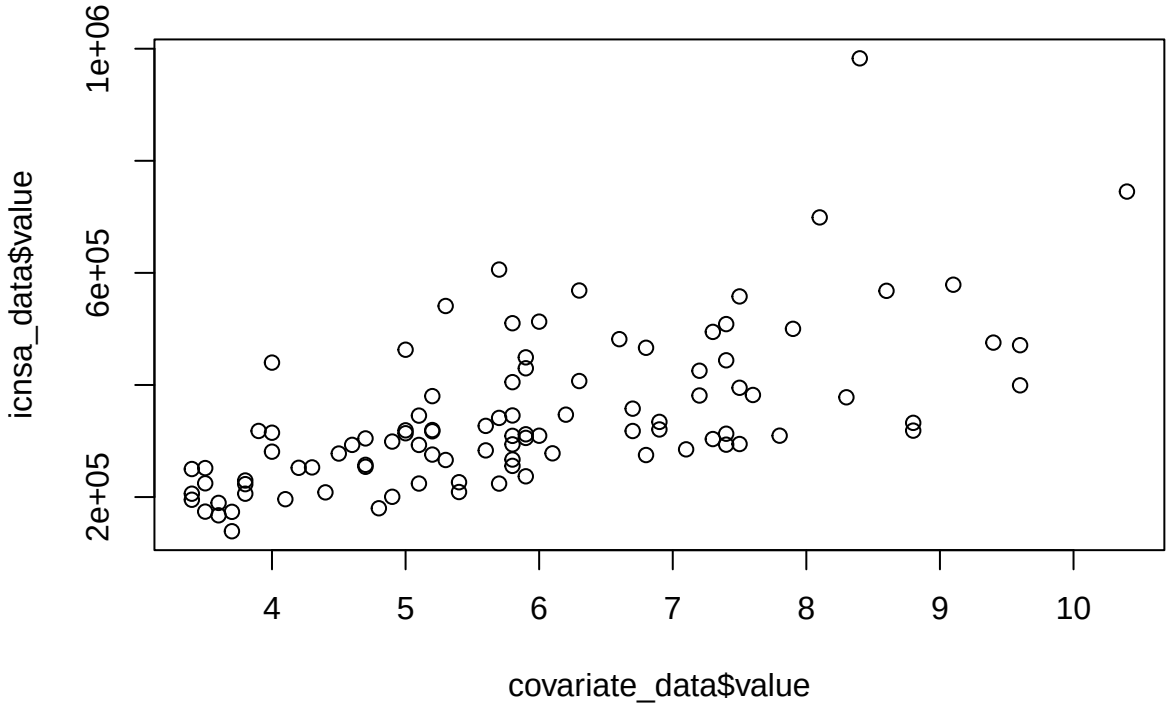
```
# Perform ADF test for UNRATE
adf_test(covariate_data$value)
```

```
## ADF Statistic: -3.967377
## p-value: 0.01062694
```

```r
# Cross-correlation function (CCF) plot
ccf_result <- ccf(icnsa_data$value, covariate_data$value, lag.max = 10, plot = TRUE,
                  xlab = "Lag", ylab = "Correlation", main = "Cross-Correlation Function")
```




```r
# Identify common time period with non-missing values
common_dates <- intersect(icnsa_data$date, covariate_data$date)

# Subset data based on common dates
icnsa_data <- icnsa_data[icnsa_data$date %in% common_dates, ]
covariate_data <- covariate_data[covariate_data$date %in% common_dates, ]

# Explore relationship between variables
plot(icnsa_data$value ~ covariate_data$value)
```




```
# Preprocess data (consider alternative methods if needed)
data <- complete.cases(icnsa_data, covariate_data)
icnsa_value <- icnsa_data$value[data]
covariate_value <- covariate_data$value[data]
```

```
# Install and load lubridate for date handling
install.packages("lubridate")
```

```
##
## The downloaded binary packages are in
##  /var/folders/8h/gp75lz8n0vqbmj3gkpfjd8qw0000gn/T//Rtmp6rizKz/downloaded_packages
```

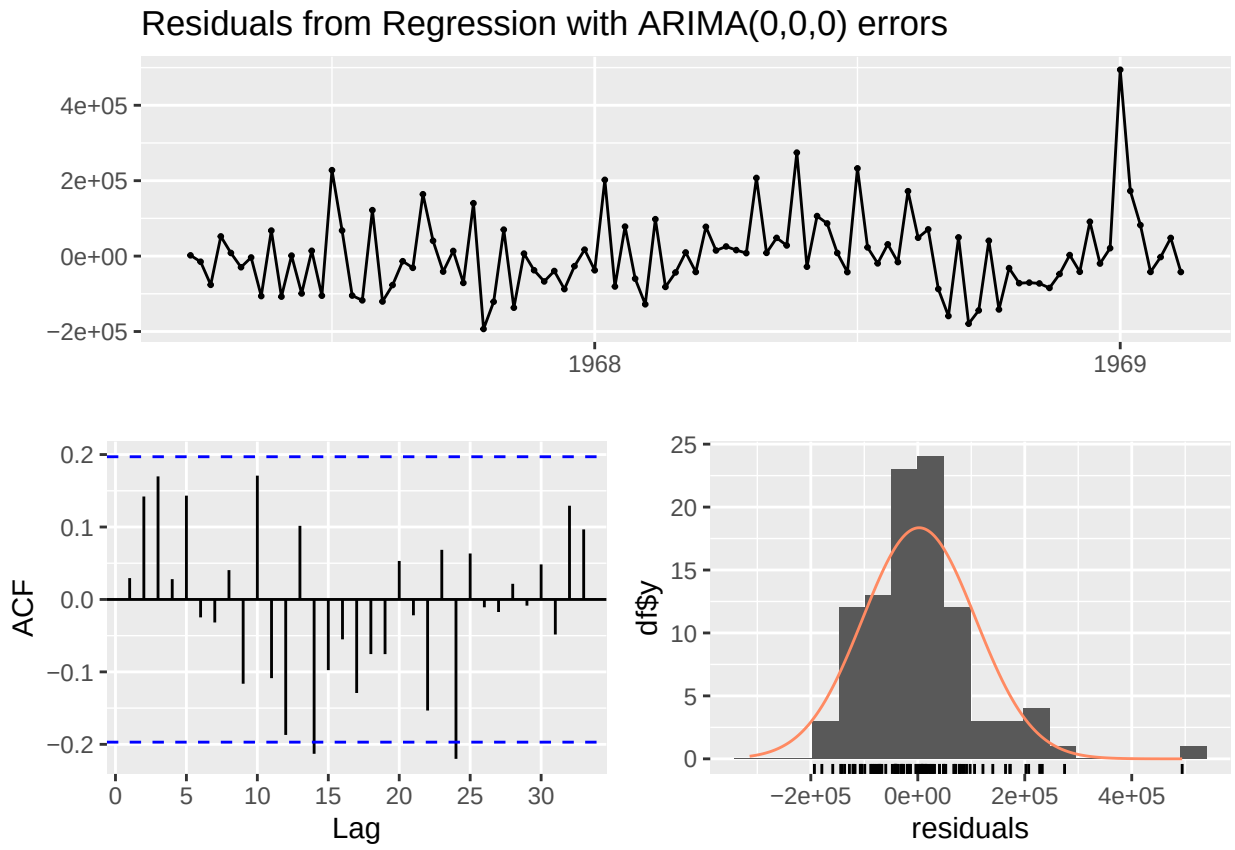
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
# For ICNSA weekly data
icnsa_start <- c(year(min(icnsa_data$date)), week(min(icnsa_data$date)))
icnsa_ts <- ts(icnsa_value, start = icnsa_start, frequency = 52)

# For UNRATE monthly data
covariate_start <- c(year(min(covariate_data$date)), month(min(covariate_data$date)))
covariate_ts <- ts(covariate_value, start = covariate_start, frequency = 12)
```

```
# Fit ARIMA model with exogenous variable

final_model <- auto.arima(icnsa_ts, xreg = covariate_ts)
#final_model <- Arima(icnsa_data$value, xreg = covariate_data$value)
checkresiduals(final_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,0) errors
## Q* = 29.787, df = 20, p-value = 0.07338
##
## Model df: 0.   Total lags used: 20
```

```
# Print model summary
summary(final_model)
```

```
## Series: icnsa_ts
## Regression with ARIMA(0,0,0) errors
##
## Coefficients:
##            xreg
##       58157.514
## s.e.   1728.086
##
## sigma^2 = 1.117e+10:  log likelihood = -1285.23
## AIC=2574.46   AICc=2574.59   BIC=2579.65
```

```
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE    MAPE     MASE       ACF1
## Training set 2524.453 105155.1 76164.59 -5.442832 21.5348 0.538466 0.02947259
```

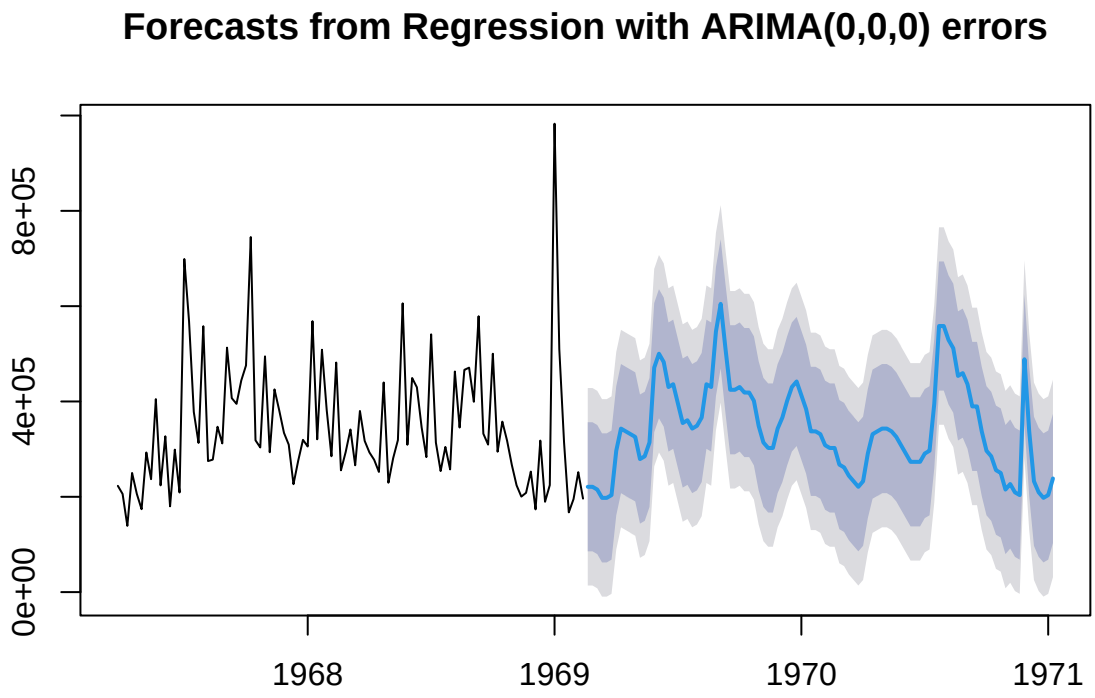
```
str(final_model)
```

```
## List of 19
##  $ coef     : Named num 58158
##   ..- attr(*, "names")= chr "xreg"
##  $ sigma2   : num 1.12e+10
##  $ var.coef : num [1, 1] 2986282
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "xreg"
##   .. ..$ : chr "xreg"
##  $ mask     : logi TRUE
##  $ loglik   : num -1285
##  $ aic      : num 2574
##  $ arma     : int [1:7] 0 0 0 0 52 0 0
##  $ residuals: Time-Series [1:99] from 1967 to 1969: 2001 -14999 -76183 52264 8264 ...
##  $ call     : language auto.arima(y = icnsa_ts, xreg = covariate_ts, x = list(x = c(223000, 206000,
##  $ series   : chr "icnsa_ts"
##  $ code     : int 0
##  $ n.cond   : int 0
##  $ nobs     : int 99
##  $ model    :List of 10
##   ..$ phi  : num(0)
##   ..$ theta: num(0)
##   ..$ Delta: num(0)
##   ..$ Z    : num 1
##   ..$ a    : num -42269
##   ..$ P    : num [1, 1] 0
##   ..$ T    : num [1, 1] 0
##   ..$ V    : num [1, 1] 1
##   ..$ h    : num 0
##   ..$ Pn   : num [1, 1] 1
##  $ bic      : num 2580
##  $ aicc     : num 2575
##  $ xreg     : num [1:99, 1] 3.8 3.8 3.7 3.4 3.4 3.5 5.1 5.9 5.8 5.7 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "xreg"
##  $ x        : Time-Series [1:99] from 1967 to 1969: 223000 206000 139000 250000 206000 174000 293000
##  $ fitted   : Time-Series [1:99] from 1967 to 1969: 220999 220999 215183 197736 197736 ...
##  - attr(*, "class")= chr [1:3] "forecast_ARIMA" "ARIMA" "Arima"
```

```
# Forecasting
forecast_values <- forecast(final_model, xreg = covariate_ts, h=1)
prediction <- forecast_values$mean[1]
prediction
```

```
## [1] 220998.6
```

```
plot(forecast_values)
```



})