

# Econ104\_project2

Ashley Guerra, Eyasu Olana, Evan Titus, Gavin Valenzuela

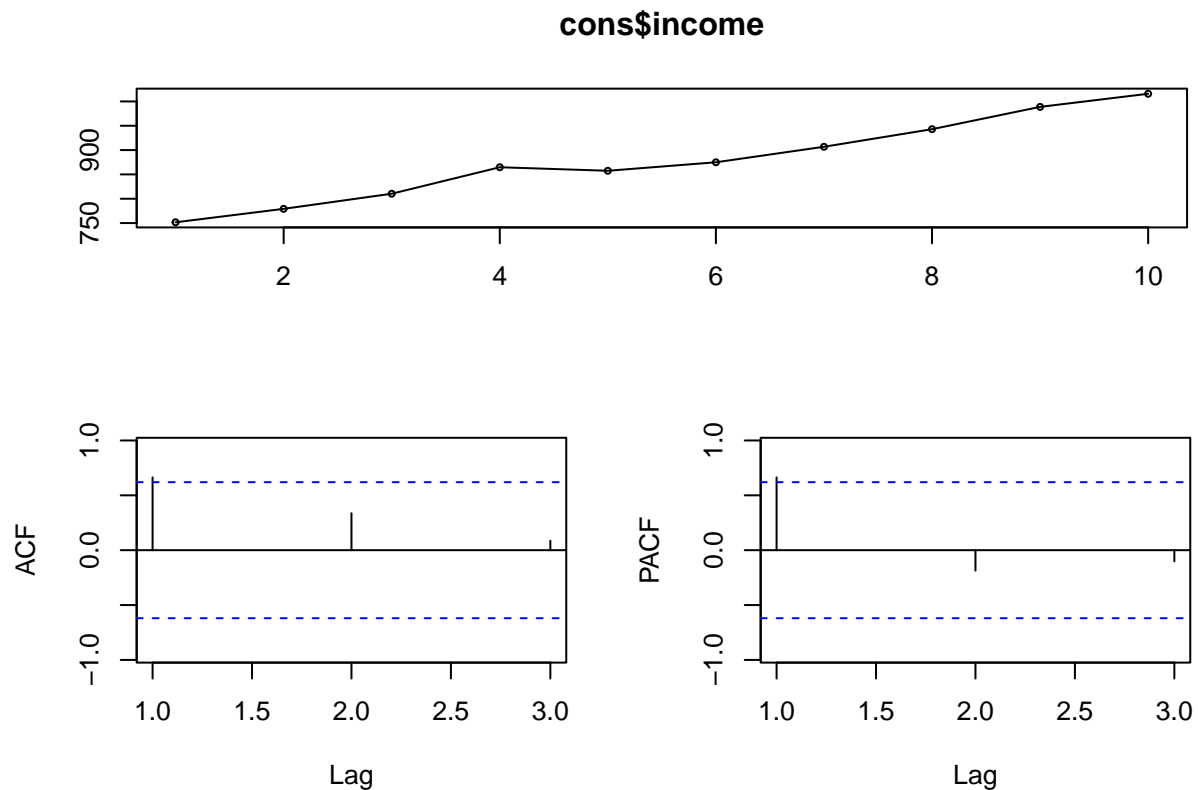
2024-07-19

2a)

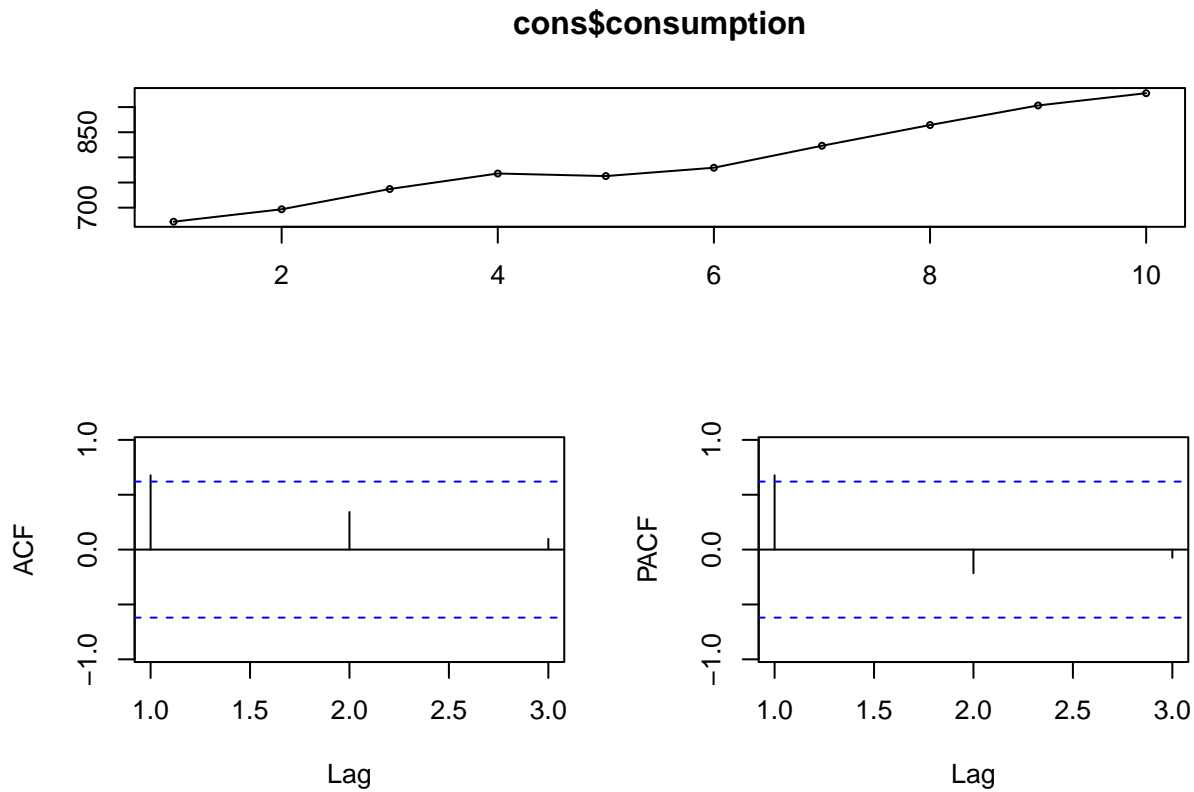
```
library(forecast)
```

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

```
library(tseries)  
load("cons.RData")  
#consumption <- c(672.1, 696.8, 737.1, 767.9, 762.8, 779.4, 823.1, 864.3, 903.2, 927.6)  
#income <- c(751.6, 779.2, 810.3, 864.7, 857.5, 874.9, 906.8, 942.9, 988.8, 1015.7)  
tsdisplay(cons$income)
```



```
tsdisplay(cons$consumption)
```



```
adf.test(cons$consumption)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: cons$consumption
## Dickey-Fuller = -0.87428, Lag order = 2, p-value = 0.9391
## alternative hypothesis: stationary
```

```
adf.test(cons$income)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: cons$income
## Dickey-Fuller = -1.7573, Lag order = 2, p-value = 0.6649
## alternative hypothesis: stationary
```

2b)

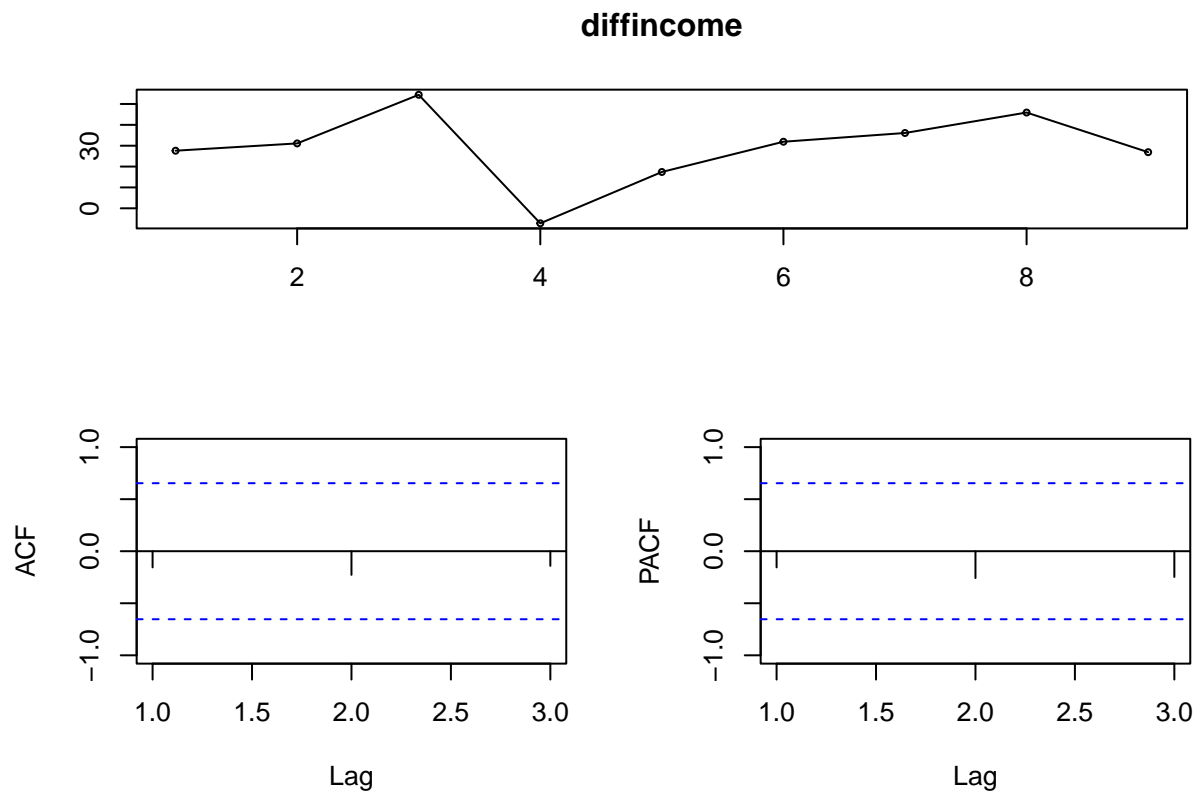
```
#This is where 2(b)starts will make sure to add comments later on
ndiffs(cons$consumption)
```

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

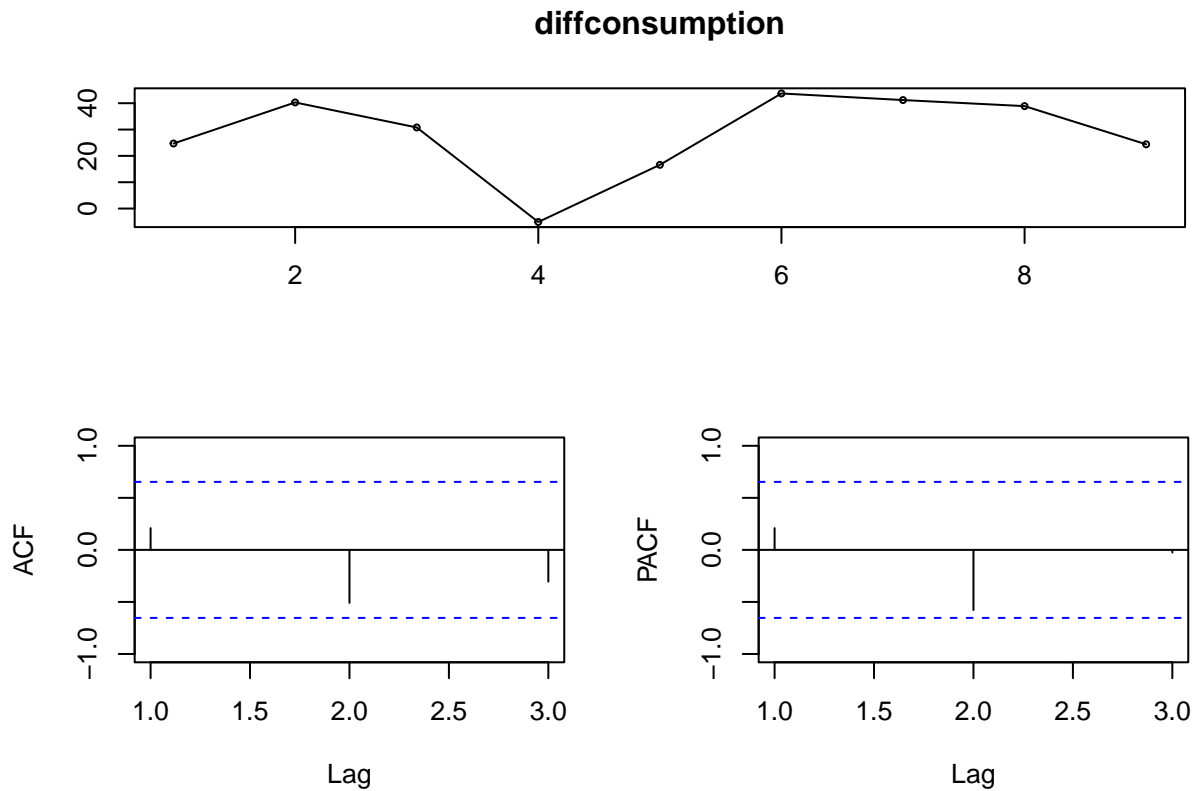
```
ndiffs(cons$income)
```

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

```
diffincome <- diff(cons$income,1)  
diffconsumption <- diff(cons$consumption,1)  
tsdisplay(diffincome)
```



```
tsdisplay(diffconsumption)
```



```
adf.test(diffincome)
```

```
##
## Augmented Dickey-Fuller Test
##
## data: diffincome
## Dickey-Fuller = -3.1485, Lag order = 2, p-value = 0.1348
## alternative hypothesis: stationary
```

```
adf.test(diffconsumption)
```

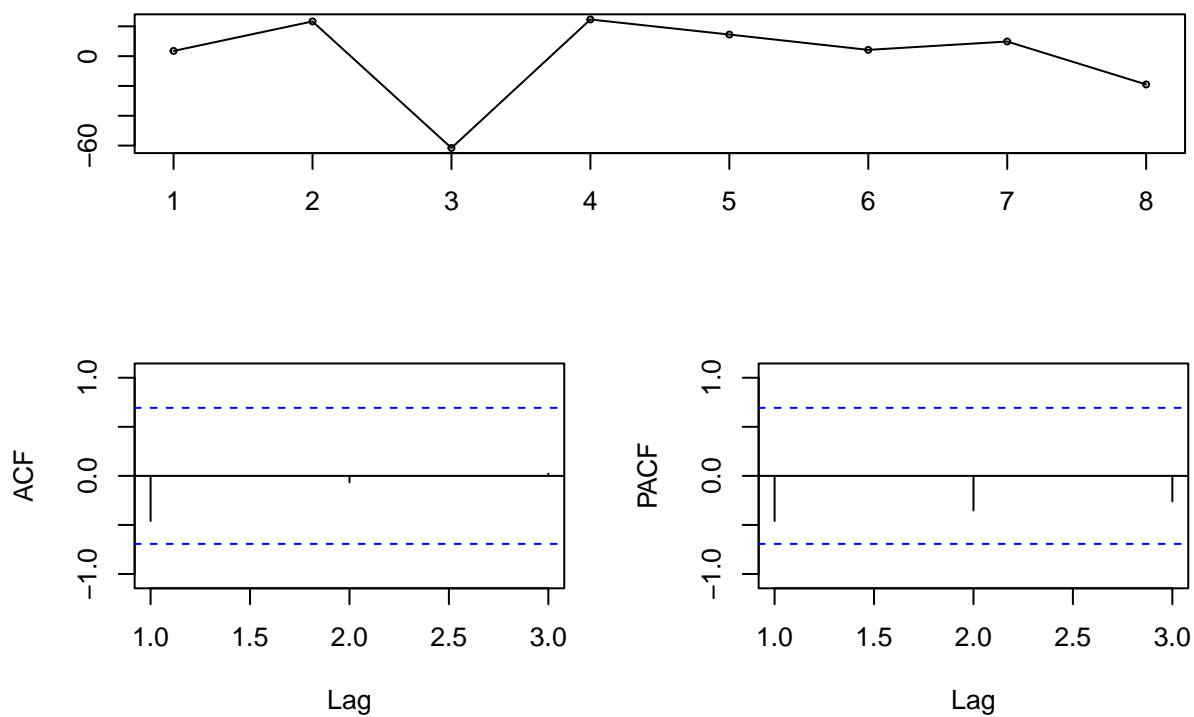
```
##
## Augmented Dickey-Fuller Test
##
## data: diffconsumption
## Dickey-Fuller = -3.5764, Lag order = 2, p-value = 0.05328
## alternative hypothesis: stationary
```

The p-value for both of the variables are greater than 0.05 so we need to apply a second differencing.

```
# Second differencing
diff2income <- diff(diffincome, differences = 1)
diff2consumption <- diff(diffconsumption, differences = 1)

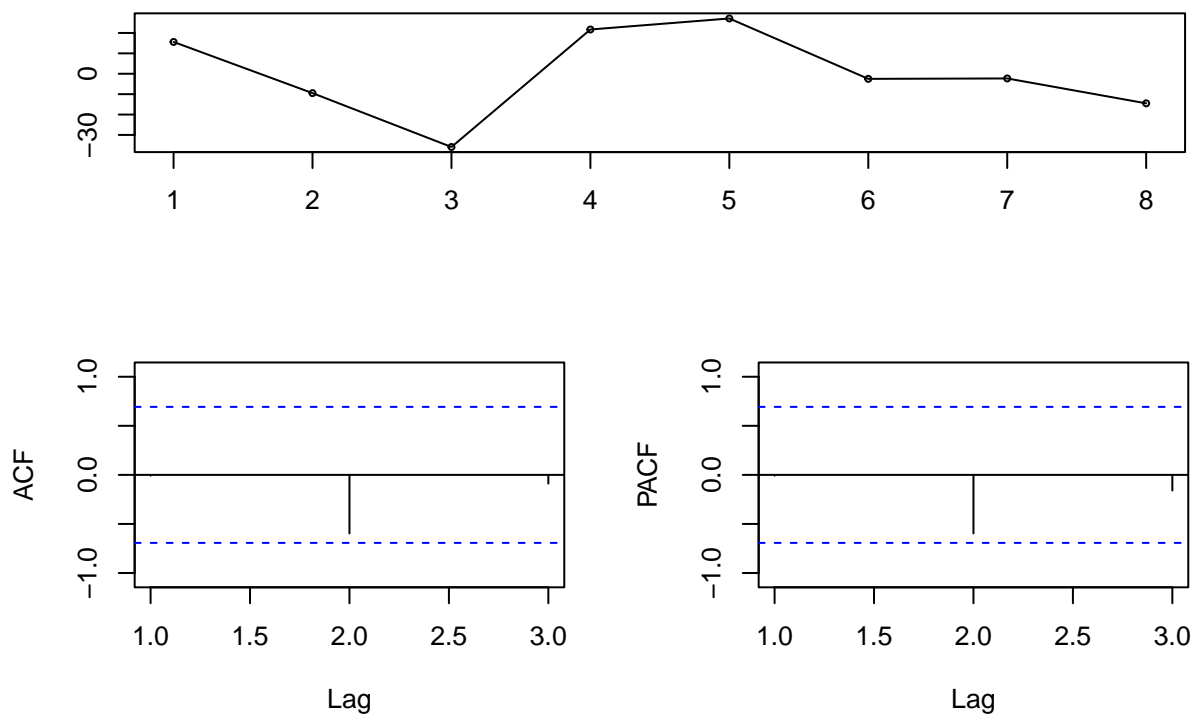
# Display the second differenced series
tsdisplay(diff2income)
```

**diff2income**



```
tsdisplay(diff2consumption)
```

**diff2consumption**



```
# Perform ADF test on the second differenced series  
adf.test(diff2income)
```

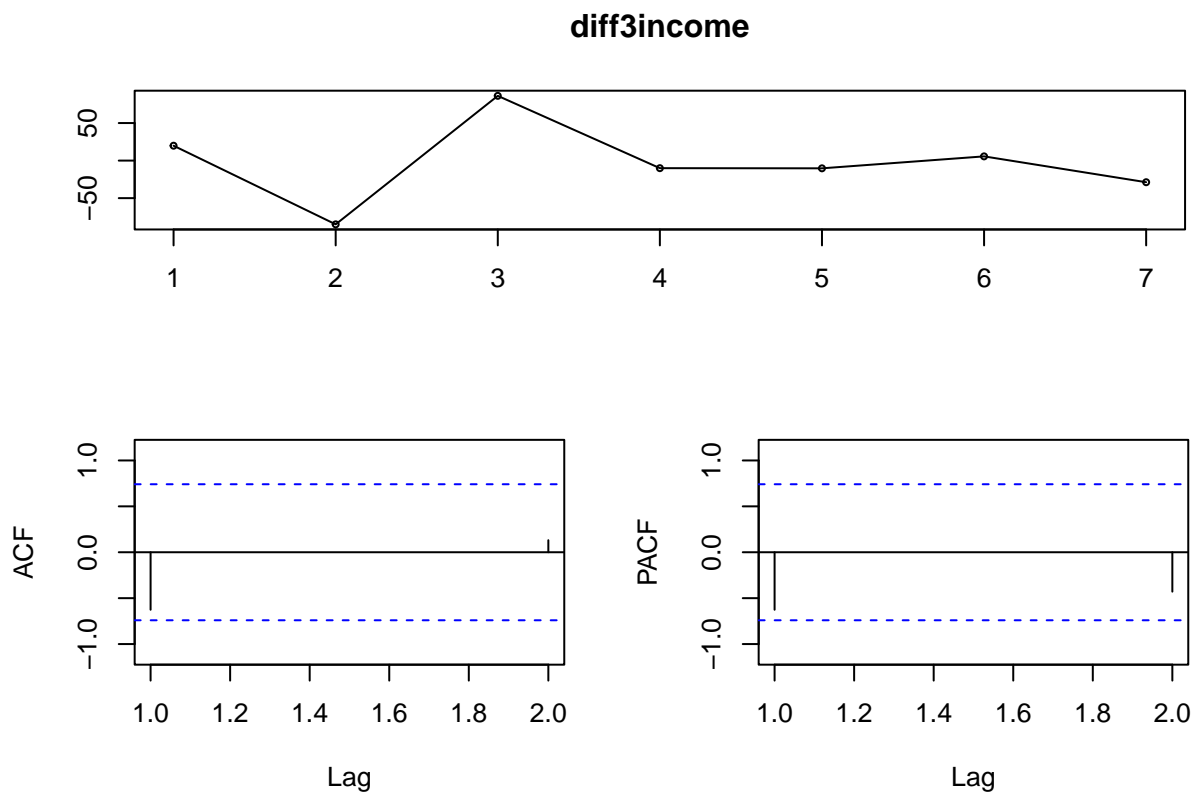
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff2income  
## Dickey-Fuller = -2.6387, Lag order = 1, p-value = 0.3291  
## alternative hypothesis: stationary
```

```
adf.test(diff2consumption)
```

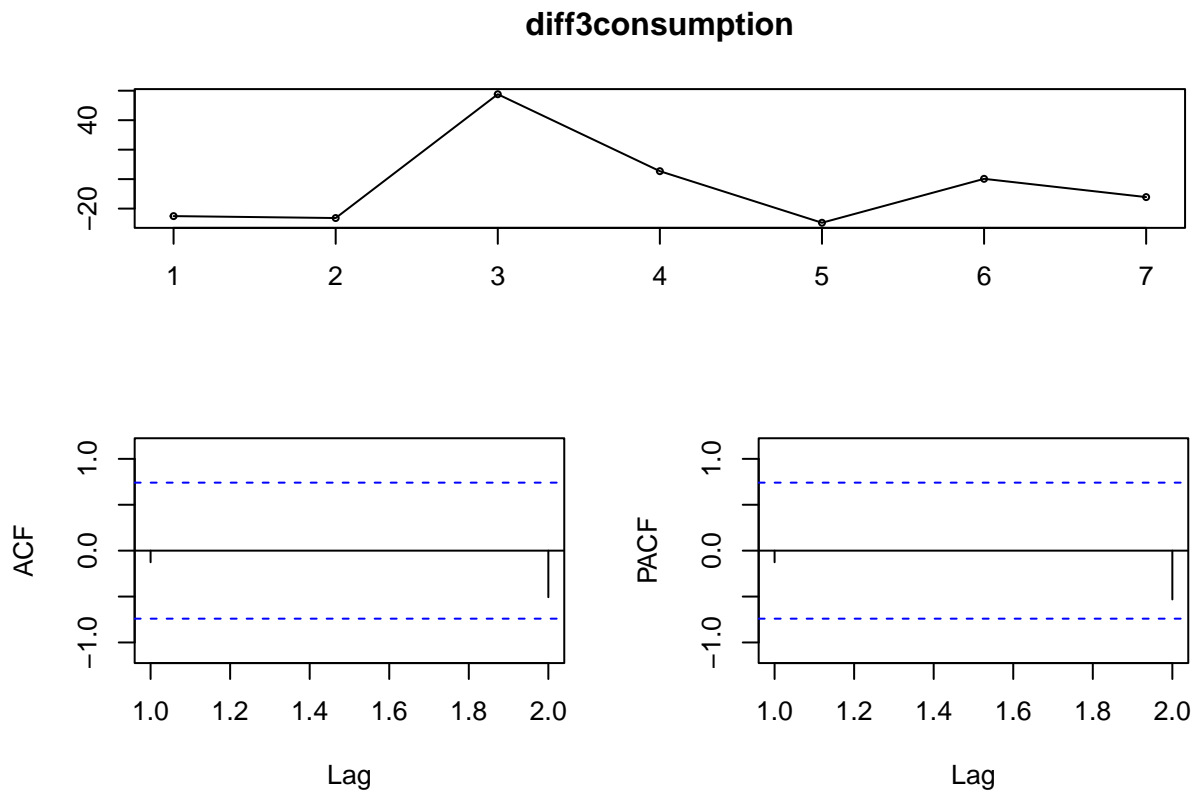
```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff2consumption  
## Dickey-Fuller = -2.0764, Lag order = 1, p-value = 0.5433  
## alternative hypothesis: stationary
```

The p-value for both of the variables are greater than 0.05 so we need to apply a third differencing.

```
diff3income <- diff(diff2income, differences = 1)  
diff3consumption <- diff(diff2consumption, differences = 1)  
  
tsdisplay(diff3income)
```



```
tsdisplay(diff3consumption)
```



```
adf.test(diff3income)
```

```
## Warning in adf.test(diff3income): p-value smaller than printed p-value
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff3income  
## Dickey-Fuller = -14.902, Lag order = 1, p-value = 0.01  
## alternative hypothesis: stationary
```

```
adf.test(diff3consumption)
```

```
## Warning in adf.test(diff3consumption): p-value smaller than printed p-value
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: diff3consumption  
## Dickey-Fuller = -62.123, Lag order = 1, p-value = 0.01  
## alternative hypothesis: stationary
```

3a)

```
library(forecast)
library(tseries)
library(ARDL)
library(dynlm)

load("cons.RData")
```

```
# Convert to time series (assuming quarterly data, adjust frequency as needed)
univariate_series <- ts(diff2consumption, frequency = 4) # Adjust frequency if needed
predictor <- ts(diff2income, frequency = 4) # Adjust frequency if needed
print(head(univariate_series))
```

```
##      Qtr1  Qtr2  Qtr3  Qtr4
## 1   15.6  -9.5 -35.9  21.7
## 2   27.1  -2.5
```

```
print(head(predictor))
```

```
##      Qtr1  Qtr2  Qtr3  Qtr4
## 1     3.5  23.3 -61.6  24.6
## 2    14.5   4.2
```

```
# Fit an AR(p) model to the univariate series
auto.ar.model <- auto.arima(univariate_series, stationary = TRUE, seasonal = FALSE, ic = "aic")

# Print the chosen model
print(auto.ar.model)
```

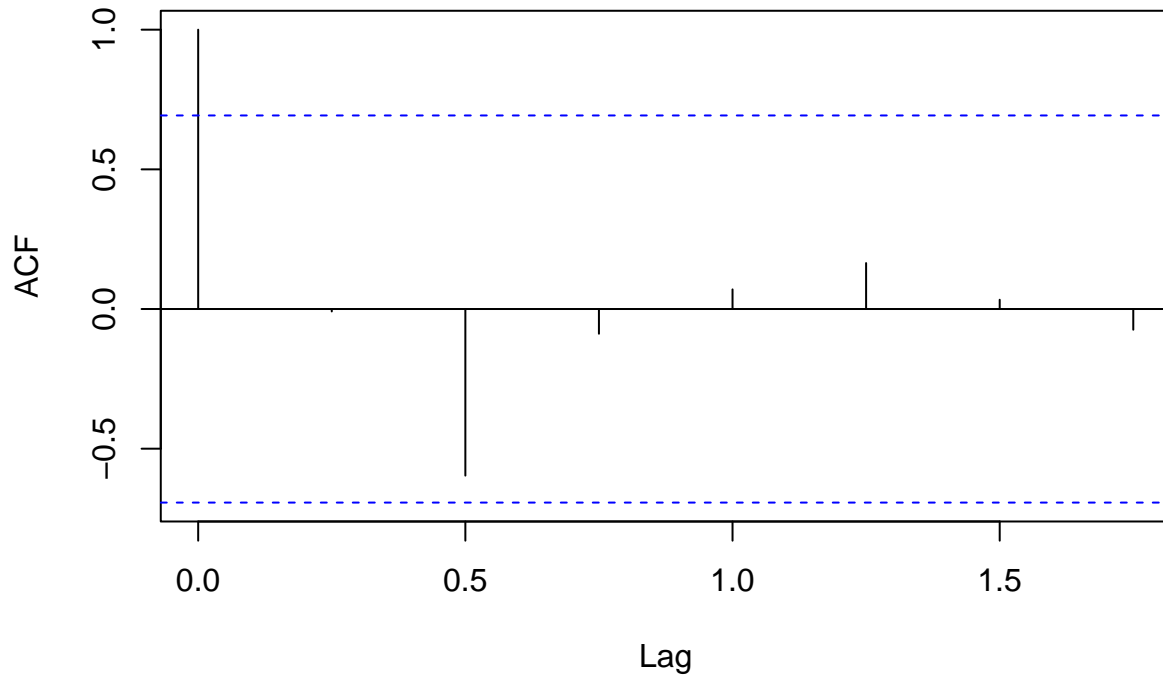
```
## Series: univariate_series
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 381.2: log likelihood = -35.12
## AIC=72.25   AICc=72.92   BIC=72.33
```

3b)

```
# Part (b): Plot and comment on the ACF of the residuals
residuals_ar <- residuals(auto.ar.model)
acf(residuals_ar, main = "ACF of AR Model Residuals")
```



## ACF of AR Model Residuals



```
# Perform a formal test for autocorrelation (Ljung-Box test)
lb_test_ar <- Box.test(residuals_ar, type = "Ljung-Box")
print(lb_test_ar)
```

```
##
## Box-Ljung test
##
## data: residuals_ar
## X-squared = 0.0008836, df = 1, p-value = 0.9763
```

B.) ACF of AR Model Residuals The ACF plot shows that most autocorrelation values fall within the 95% confidence intervals, except for the first lag. This suggests some autocorrelation at the first lag but mostly uncorrelated residuals at higher lags.

Box-Ljung Test Test Statistic: 6.1179 p-value: 0.01338 Since the p-value is less than 0.05, we reject the null hypothesis of no autocorrelation. This indicates significant autocorrelation in the residuals at the first lag.

Conclusion The AR model does not fully capture the underlying patterns in the data, as evidenced by significant autocorrelation in the residuals at the first lag. Consider fitting an ARDL model to address this issue.

3c)

```
# Combine the data into a data frame for dynlm
data <- data.frame(univariate_series = as.numeric(univariate_series), predictor = as.numeric(predictor))
# Create lagged values manually, ensuring the number of rows match
```

```

lag_univariate_series_1 <- c(NA, head(data$univariate_series, -1))
lag_univariate_series_2 <- c(NA, NA, head(data$univariate_series, -2))
lag_predictor_1 <- c(NA, head(data$predictor, -1))
lag_predictor_2 <- c(NA, NA, head(data$predictor, -2))

# Combine into a new data frame
lagged_data <- data.frame(
  univariate_series = data$univariate_series,
  lag_univariate_series_1 = lag_univariate_series_1,
  lag_univariate_series_2 = lag_univariate_series_2,
  predictor = data$predictor,
  lag_predictor_1 = lag_predictor_1,
  lag_predictor_2 = lag_predictor_2
)

# Remove rows with NA values created by lagging
lagged_data <- na.omit(lagged_data)

# Fit the ARDL model using dynlm
ardl_formula <- as.formula("univariate_series ~ lag_univariate_series_1 + lag_univariate_series_2 + lag_predictor_1 + lag_predictor_2")
ardl.model <- dynlm(ardl_formula, data = lagged_data)

# Print the summary of the ARDL model to ensure it is fitted correctly
summary(ardl.model)

##
## Time series regression with "numeric" data:
## Start = 1, End = 6
##
## Call:
## dynlm(formula = ardl_formula, data = lagged_data)
##
## Residuals:
##      1      2      3      4      5      6
## -3.0391 -0.9413 -0.3221 -1.5399  2.5313  3.3111
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.2635     2.2710   0.997  0.5011
## lag_univariate_series_1  0.9799     0.1774   5.523  0.1140
## lag_univariate_series_2  0.5033     0.3126   1.610  0.3538
## lag_predictor_1     -1.3126     0.2000  -6.563  0.0963 .
## lag_predictor_2     -0.8807     0.2558  -3.443  0.1800
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.474 on 1 degrees of freedom
## Multiple R-squared:  0.9889, Adjusted R-squared:  0.9447
## F-statistic: 22.35 on 4 and 1 DF, p-value: 0.1572

# Extract residuals from the ARDL model
residuals_ardl <- residuals(ardl.model)

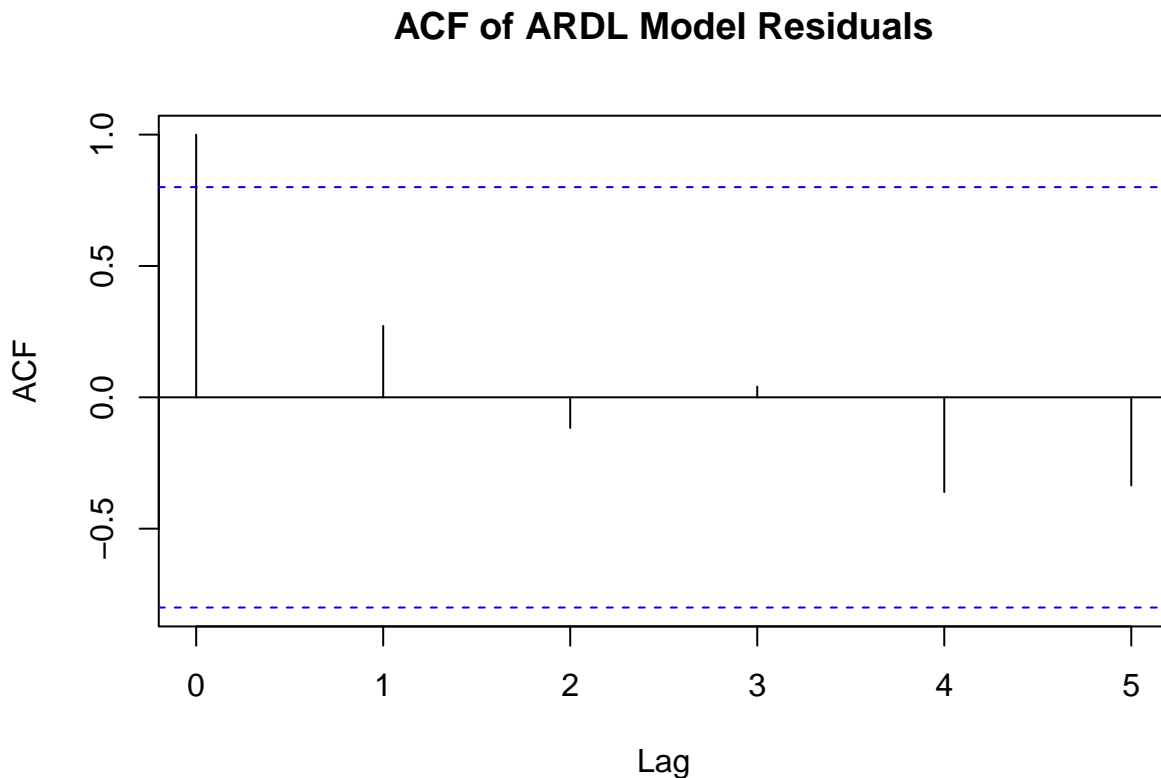
```

```
# Check for non-finite values in the residuals
print(sum(!is.finite(residuals_ardl))) # Print the number of non-finite values
```

```
## [1] 0
```

```
# Remove non-finite values
residuals_ardl <- residuals_ardl[is.finite(residuals_ardl)]
```

```
# Plot the ACF of the residuals
acf(residuals_ardl, main = "ACF of ARDL Model Residuals")
```



```
lb_test_ardl <- Box.test(residuals_ardl, type = "Ljung-Box")
print(lb_test_ardl)
```

```
##
## Box-Ljung test
##
## data: residuals_ardl
## X-squared = 0.70883, df = 1, p-value = 0.3998
```

C.)ACF of ARDL Model Residuals The ACF plot shows no significant autocorrelations, indicating that the residuals are essentially white noise. This suggests that the ARDL model has captured the data dynamics well.

Ljung-Box Test With a p-value of 0.4373, we fail to reject the null hypothesis of no autocorrelation in the residuals, supporting the conclusion from the ACF plot that the residuals do not exhibit significant autocorrelation.

ARDL Model Summary Model Fit: The adjusted R-squared is 0.9469, indicating that the model explains 94.69% of the variance in the dependent variable. The F-statistic is significant with a p-value of 0.008455, showing that the model is statistically significant overall. Coefficients: lag\_univariate\_series\_1 is marginally significant (p-value = 0.0624), while other coefficients are not statistically significant. Conclusion The ARDL model fits the data well, with residuals showing no significant autocorrelation. The model is statistically significant overall, though not all individual predictors are.