

# 104\_project2

Krish Methi, Krithik Jatavallabhula, Brandon Chan

1/31/2024

```
# Load libraries
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.2
```

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

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.1.2
```

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.1.2
```

```
## -- Attaching packages ----- fpp3 0.5 --
```

```
## v tibble      3.2.1      v tsibble      1.1.3
## v dplyr       1.1.2      v tsibbledata 0.4.1
## v tidyr       1.2.1      v feasts       0.3.1
## v lubridate   1.8.0      v fable        0.3.3
## v ggplot2     3.4.4      v fabletools   0.3.4
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'tsibble' was built under R version 4.1.2
```

```
## Warning: package 'tsibbledata' was built under R version 4.1.2
```

```
## Warning: package 'feasts' was built under R version 4.1.2
```

```
## Warning: package 'fable' was built under R version 4.1.2

## -- Conflicts ----- fpp3_conflicts --
## x lubridate::date()      masks base::date()
## x dplyr::filter()        masks stats::filter()
## x tsibble::intersect()   masks base::intersect()
## x tsibble::interval()    masks lubridate::interval()
## x dplyr::lag()           masks stats::lag()
## x tsibble::setdiff()     masks base::setdiff()
## x tsibble::union()       masks base::union()
```

```
library(tseries)
library(seasonal)
```

```
## Warning: package 'seasonal' was built under R version 4.1.2
```

```
##
## Attaching package: 'seasonal'
```

```
## The following object is masked from 'package:tibble':
##
##      view
```

```
library(fable)
library(stats)
require(graphics)
library(dplyr)
library(tsibble)
library(tsibbledata)
library(dynlm)
```

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

```
## Warning: package 'zoo' was built under R version 4.1.2
```

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

```
## The following object is masked from 'package:tsibble':
##
##      index
```

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

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.1.2
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:tsibble':
##
##     key

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

library(zoo)
library(ARDL)

## To cite the ARDL package in publications:
##
## Use this reference to refer to the validity of the ARDL package.
##
## Natsiopoulos, Kleanthis, and Tzeremes, Nickolaos G. (2022). ARDL
## bounds test for cointegration: Replicating the Pesaran et al. (2001)
## results for the UK earnings equation using R. Journal of Applied
## Econometrics, 37(5), 1079-1090. https://doi.org/10.1002/jae.2919
##
## Use this reference to cite this specific version of the ARDL package.
##
## Kleanthis Natsiopoulos and Nickolaos Tzeremes (2023). ARDL: ARDL, ECM
## and Bounds-Test for Cointegration. R package version 0.2.4.
## https://CRAN.R-project.org/package=ARDL
```

## Introduction

The dataset we are working with for this project is the USMacroG dataset, a quarterly multiple time series on 12 US Macroeconomic variables from 1950-2000. We have removed variables so that the dataset we are operating on contains GDP, consumption, investment, government expenditures, and real disposable personal income. For our analysis, we will be using GDP and consumption as our response variables aiming to predict, and investment, government expenditures, and Rdpi as our 3 predictors. We calculated the growth rates of these variables over time, and operated on those growth rates.

```
rm(list=ls(all=TRUE))
library(AER)
```

```
## Warning: package 'AER' was built under R version 4.1.2
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 4.1.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.2

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##      recode

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.1.2

## Loading required package: sandwich

## Loading required package: survival
```

```
# Access the "USMacroG" dataset
data("USMacroG")

# Assign the dataset to the variable myData, and remove the columns we don't want
myData <- USMacroG
mydata_ts <- ts(myData, start=1950, freq=4)
mydata_ts <- mydata_ts[, -c(6:12)]
```

## Part 1: descriptive analysis of variables

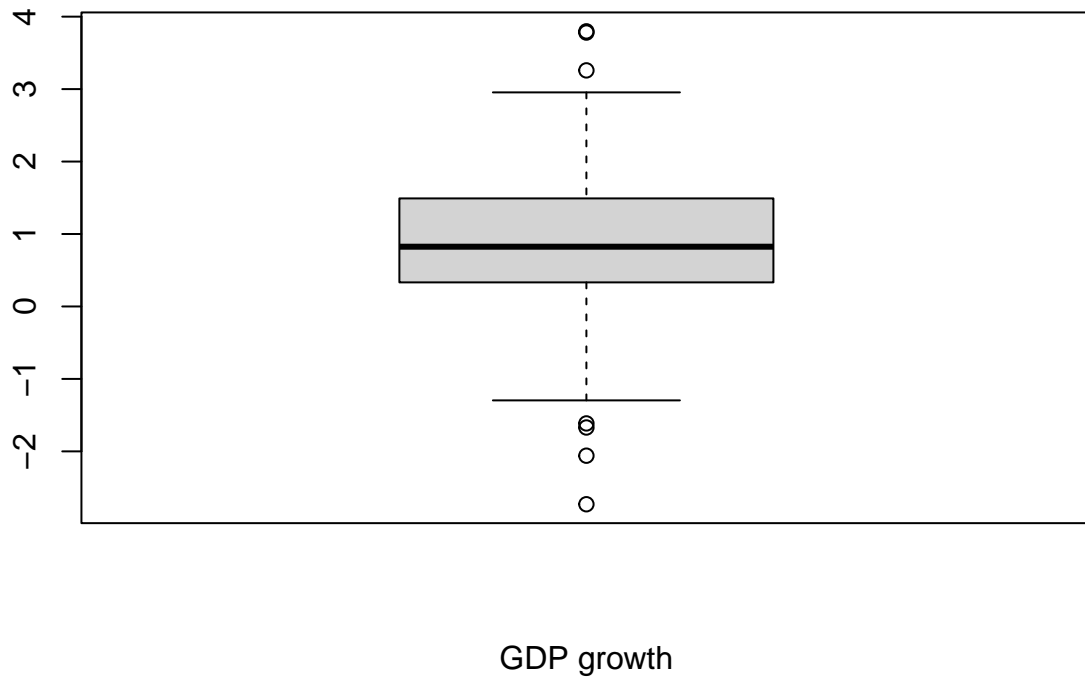
### GDP growth

```
GDP <- mydata_ts[, 1]
log_gdp <- log(GDP)
gdp_growth <- diff(log_gdp) * 100
summary(gdp_growth)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.7299  0.3322   0.8249   0.8640  1.4916   3.7973
```

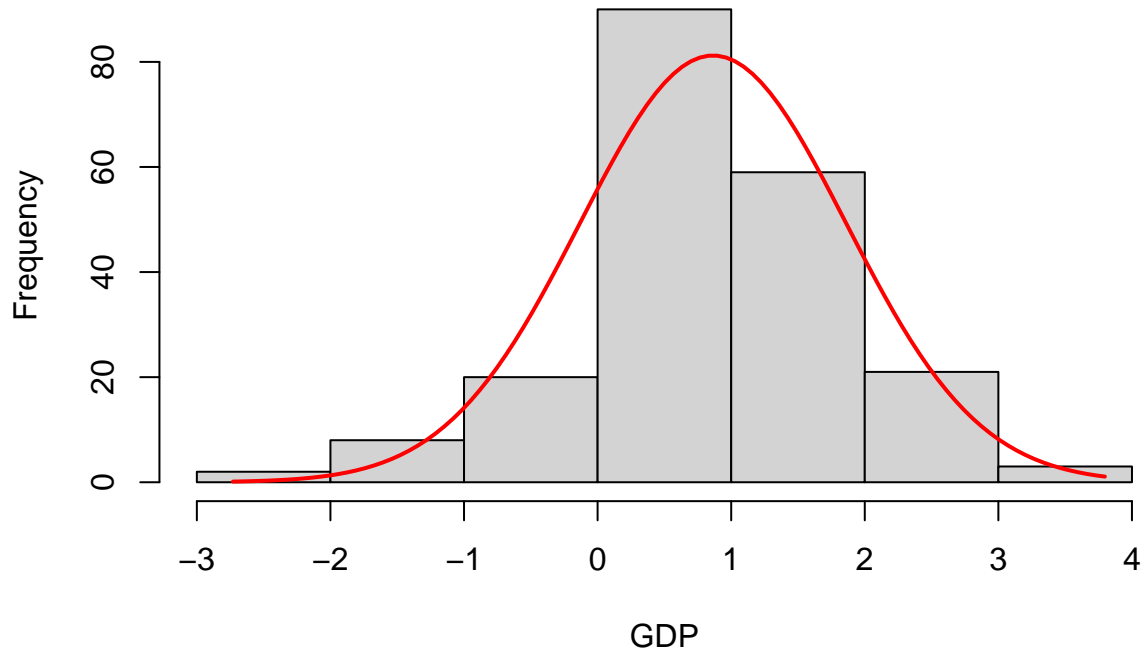
```
boxplot(gdp_growth, main = "Boxplot of GDP growth from 1950-2000", xlab = "GDP growth")
```

## Boxplot of GDP growth from 1950–2000



```
GDP_hist <- hist(gdp_growth, main = "Histogram of GDP growth", xlab = "GDP", ylab = "Frequency")
xfit <- seq(min(gdp_growth), max(gdp_growth), length = 100)
yfit <- dnorm(xfit, mean = mean(gdp_growth), sd = sd(gdp_growth))
yfit <- yfit * diff(GDP_hist$mids[1:2]) * length(gdp_growth)
lines(xfit, yfit, col = "red", lwd = 2)
```

## Histogram of GDP growth



Comments: The GDP variable refers to the the real gross domestic product growth in percent, one of the response variables in our dataset. From the histogram, the data looks relatively normal, or potentially right skewed slightly, with a peak in between 0 and 1% gdp growth. There seems to be potential outliers on the left side and one on the right side from the histogram. With the fitted red line normal curve, it shows us that the GDP variable does somewhat resemble a normal distribution. Looking at the box plot, we can see the Interquartile range box, with the median slightly under the center point, suggesting the right skewness of the data, and we can see that there are outliers in the lower and higher half of the data. Looking at the 5 number summary, we have the median and mean at 0.825 and 0.864 % growth, and we have the IQR from 0.332 to 1.49 %

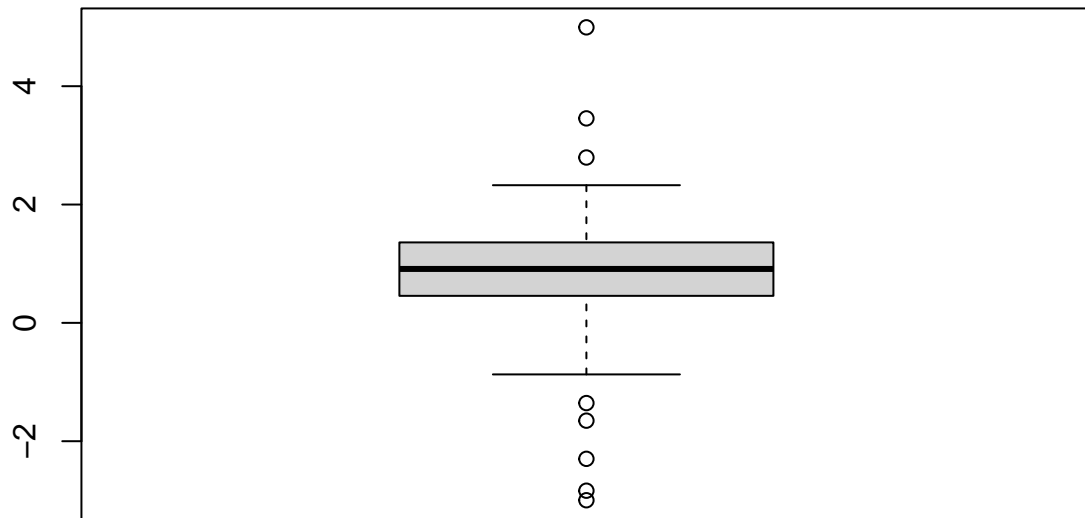
## Consumption

```
Consumption <- mydata_ts[, 2]
log_cons <- log(Consumption)
cons_growth <- diff(log_cons) * 100
summary(cons_growth)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.9976  0.4565   0.9113   0.8817  1.3597   4.9945
```

```
boxplot(cons_growth, main = "Boxplot of Consumption Growth from 1950-2000", xlab = "Consumption Growth")
```

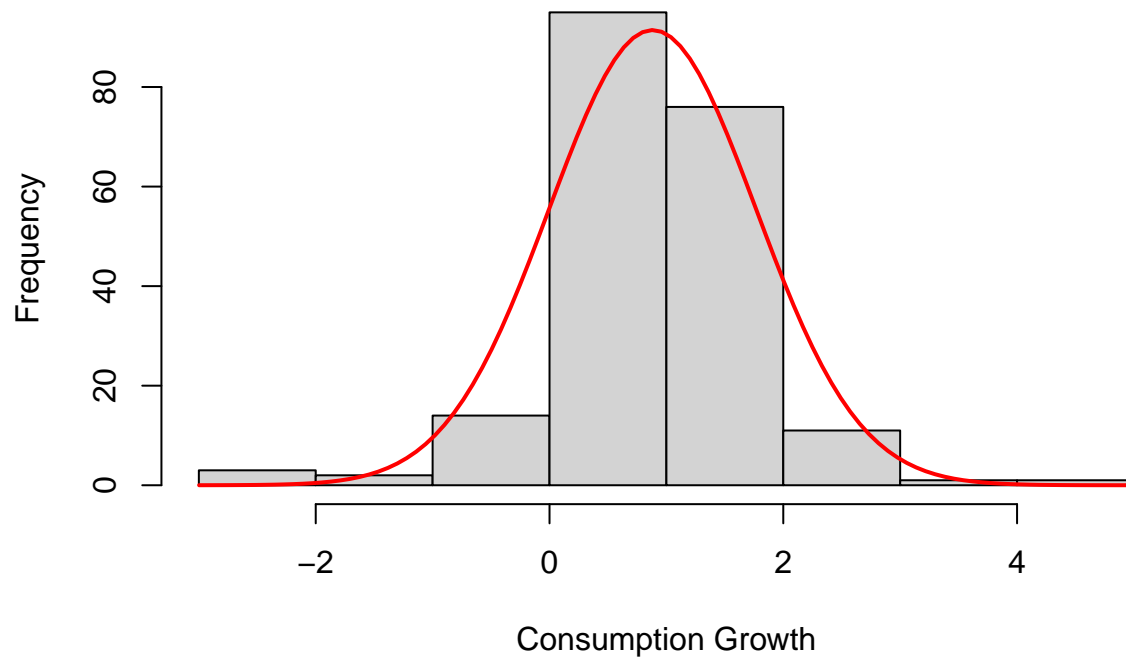
## Boxplot of Consumption Growth from 1950–2000



Consumption Growth

```
Consumption_hist <- hist(cons_growth, main = "Histogram of Consumption Growth", xlab = "Consumption Growth",  
xfit <- seq(min(cons_growth), max(cons_growth), length = 100)  
yfit <- dnorm(xfit, mean = mean(cons_growth), sd = sd(cons_growth))  
yfit <- yfit * diff(Consumption_hist$mids[1:2]) * length(cons_growth)  
lines(xfit, yfit, col = "red", lwd = 2)
```

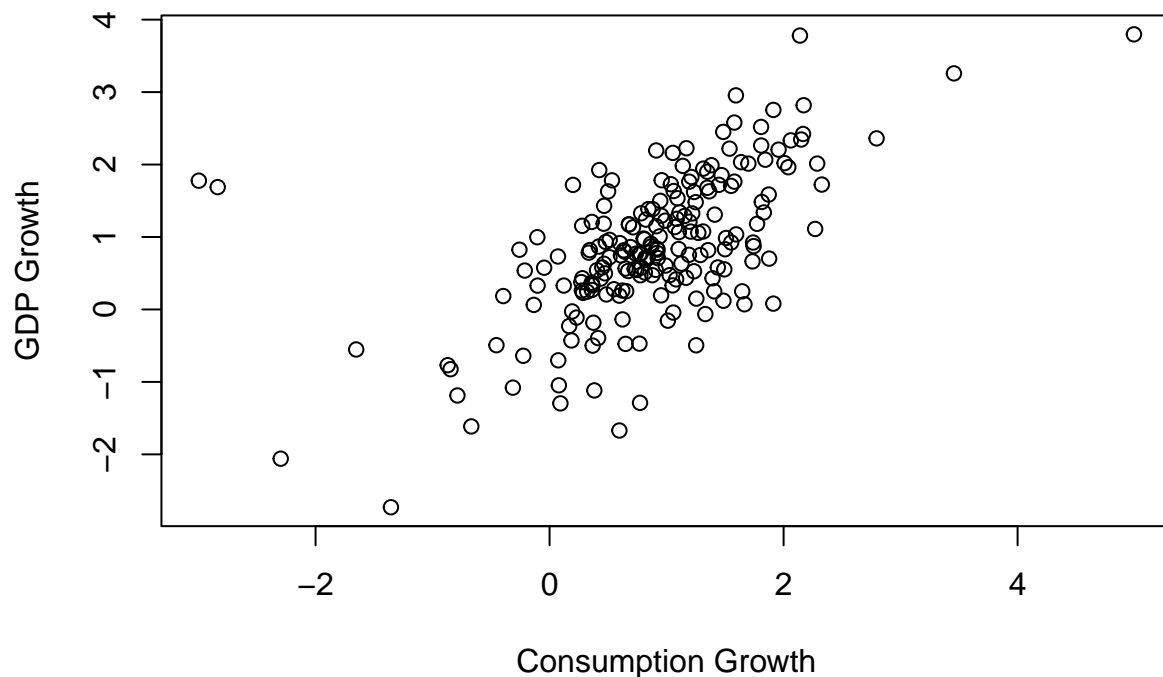
## Histogram of Consumption Growth



```
plot(cons_growth, gdp_growth, main = "Scatterplot of Consumption Growth compared to GDP Growth", xlab =
```



## Scatterplot of Consumption Growth compared to GDP Growth



Comments: The Consumption variable refers to the growth of real consumption expenditures, one of the response variables in our dataset. From the histogram, the data appears to be relatively normally distributed, with peaks in between 0-2 percent growth. There seems to be potential outliers on both sides from the histogram. the fitted red normal distribution line also looks like a pretty good fit on the histogram. Looking at the box plot, we can see the Interquartile range box, and the median looks to be right around the center, suggesting symmetrical data, and we can see that there are outliers on both sides present in the data from the boxplot. Looking at the 5 number summary, we have the median and mean at 0.911 and 0.882 percent growth, and we have the IQR from 0.457 to 1.36 percent. Looking at the scatterplot, we can see a pretty clear trend, as consumption growth increases, GDP growth increases.

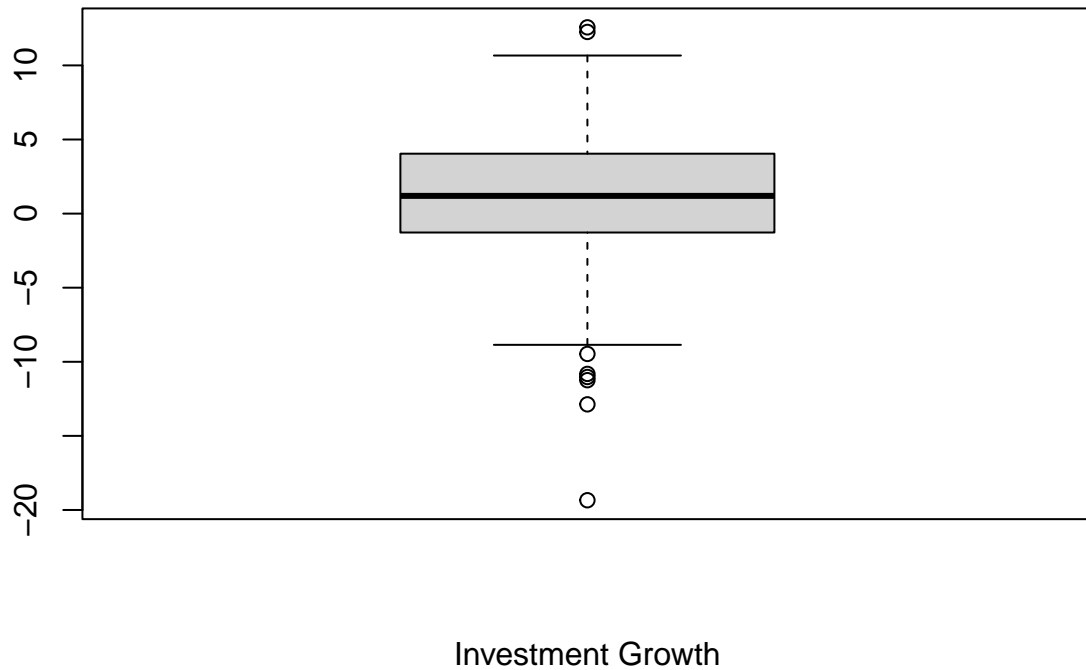
## Investment

```
Investment <- mydata_ts[, 3]
log_investment <- log(Investment)
inv_growth <- diff(log_investment) * 100
summary(inv_growth)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -19.343  -1.276    1.196    1.081   4.043   12.568
```

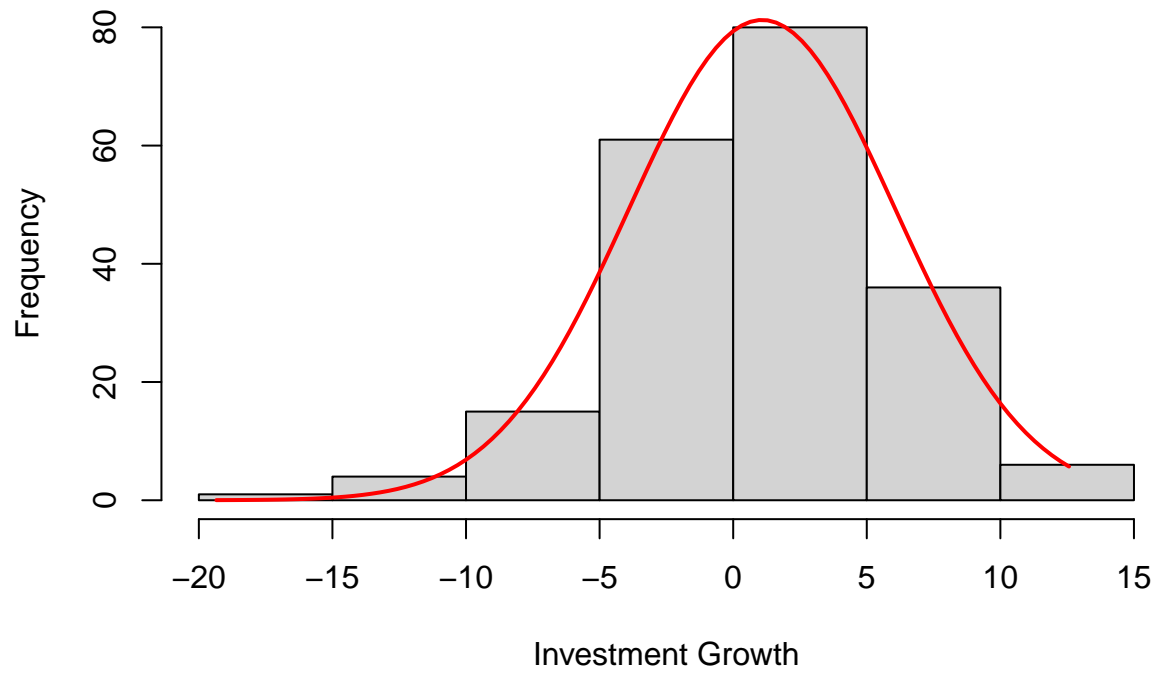
```
boxplot(inv_growth, main = "Boxplot of Investment Growth from 1950-2000", xlab = "Investment Growth")
```

## Boxplot of Investment Growth from 1950–2000



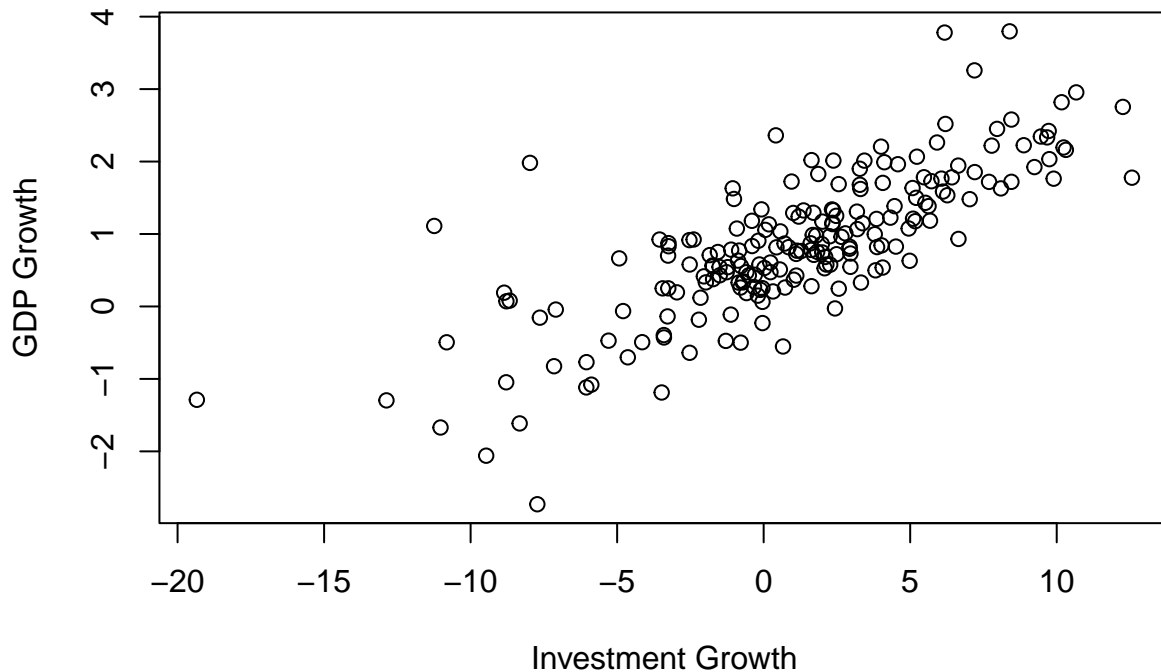
```
Investment_hist <- hist(inv_growth, main = "Histogram of Investment Growth", xlab = "Investment Growth")
xfit <- seq(min(inv_growth), max(inv_growth), length = 100)
yfit <- dnorm(xfit, mean = mean(inv_growth), sd = sd(inv_growth))
yfit <- yfit * diff(Investment_hist$mids[1:2]) * length(inv_growth)
lines(xfit, yfit, col = "red", lwd = 2)
```

### Histogram of Investment Growth



```
plot(inv_growth, gdp_growth, main = "Scatterplot of Investment Growth compared to GDP Growth", xlab = "Investment Growth", ylab = "GDP Growth")
```

## Scatterplot of Investment Growth compared to GDP Growth



Comments: The Investment variable refers to the growth the real investment by private sector, one of the predictor variables in our dataset. From the histogram, it seems that the data is slightly left skewed, with a main peak in between 0 and 5 percent growth. There seems to be a couple potential outliers on the left side from the histogram. The fitted red normal distribution line also looks like a pretty good fit on the histogram here. Looking at the box plot, we can see the Interquartile range box, with the median right around the center point, suggesting relatively symmetrical data, and we can see that there are outliers identified in the data towards the top and bottom side, more on the bottom side. Looking at the 5 number summary, we have the median and mean at 1.2 and 1.08 percent, and we have the IQR from -1.28 to 4.04 percent. Looking at the scatterplot, we can see a massively clear trend as well, as investment increases, GDP increases. In this variable, we see the range is much wider than the previous variables, and the IQR goes into negative growth as well, suggesting more volatility.

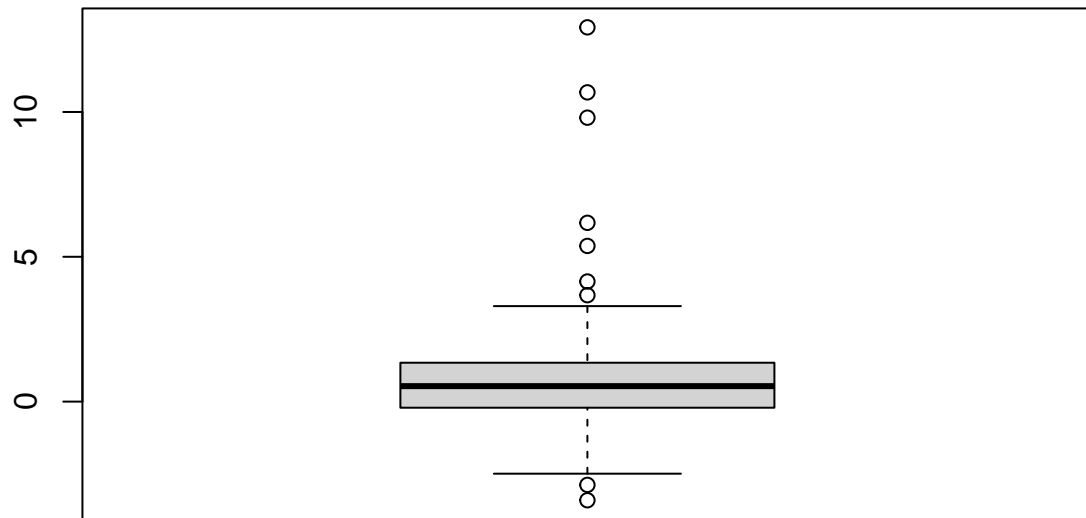
## Government Expenditures

```
Government <- mydata_ts[, 4]
log_gov <- log(Government)
gov_growth <- diff(log_gov) * 100
summary(gov_growth)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.4061 -0.2112   0.5339   0.7281  1.3425 12.9226
```

```
boxplot(gov_growth, main = "Boxplot of Government Expenditure growth from 1950-2000", xlab = "Government")
```

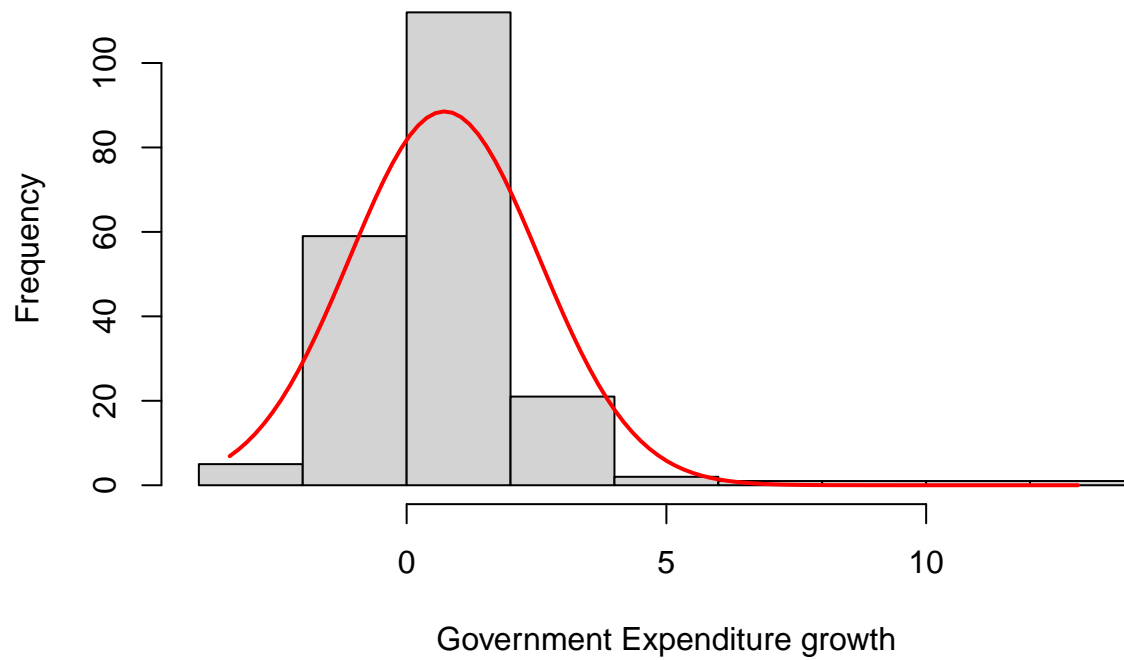
## Boxplot of Government Expenditure growth from 1950–2000



Government Expenditure growth

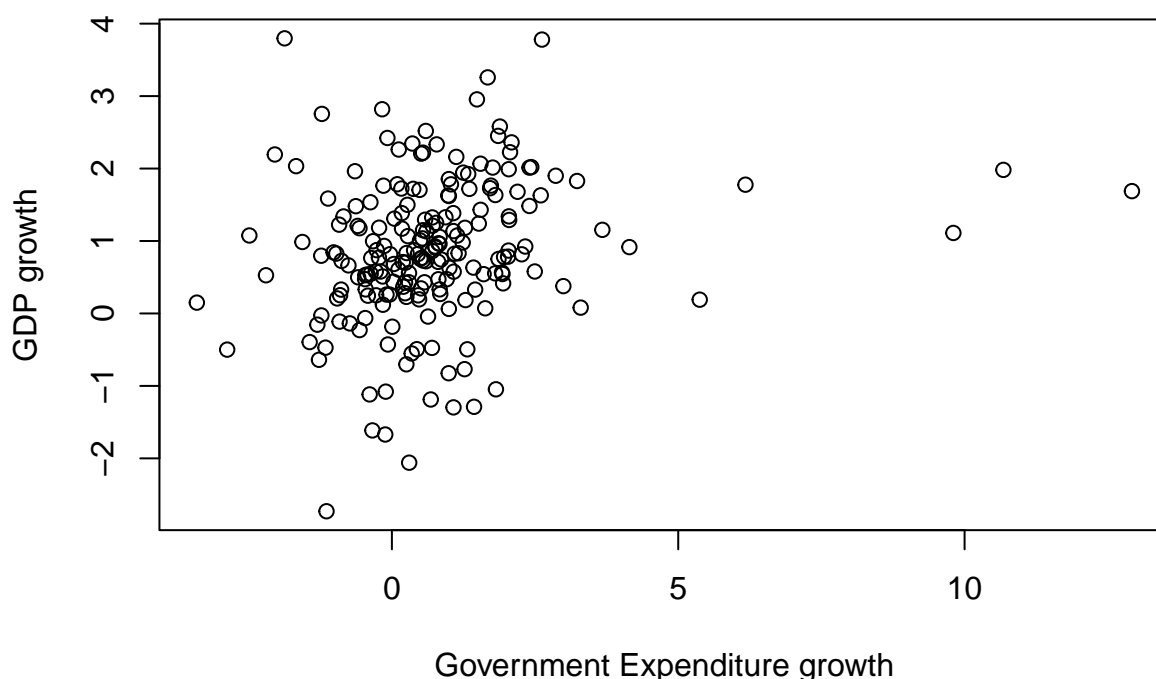
```
Government_hist <- hist(gov_growth, main = "Histogram of Government Expenditure growth", xlab = "Government Expenditure growth")
xfit <- seq(min(gov_growth), max(gov_growth), length = 100)
yfit <- dnorm(xfit, mean = mean(gov_growth), sd = sd(gov_growth))
yfit <- yfit * diff(Government_hist$mids[1:2]) * length(gov_growth)
lines(xfit, yfit, col = "red", lwd = 2)
```

## Histogram of Government Expenditure growth



```
plot(gov_growth, gdp_growth, main = "Scatterplot of Government Expenditure growth compared to GDP growth")
```

## Scatterplot of Government Expenditure growth compared to GDP growth



Comments: The Government variable refers to the the real government expenditures, one of the predictor variables in our dataset. From the histogram, the data clearly seems to be right skewed, with one main peak from 0 to 2.5 percent growth. There seems to be many outliers present on the top side from the histogram. With the fitted red line normal curve, the line doesn't appear to work too well with the data. Looking at the box plot, we can see the Interquartile range box, with the median right around the center point, and we can see that there are outliers in the data identified, more so on the top side than on the bottom side. Looking at the 5 number summary, we have the median and mean at 0.534 and 0.728 percent, and we have the IQR from -0.211 to 1.342 percent. Looking at the scatterplot, we see an overall trend of as government expenditures increase, GDP increases; however, the points are very clustered, and after a certain point of around 5% growth, doesn't seem to be a positive trend. Again, this variable includes negative growth in the IQR, however, the range is less than investment.

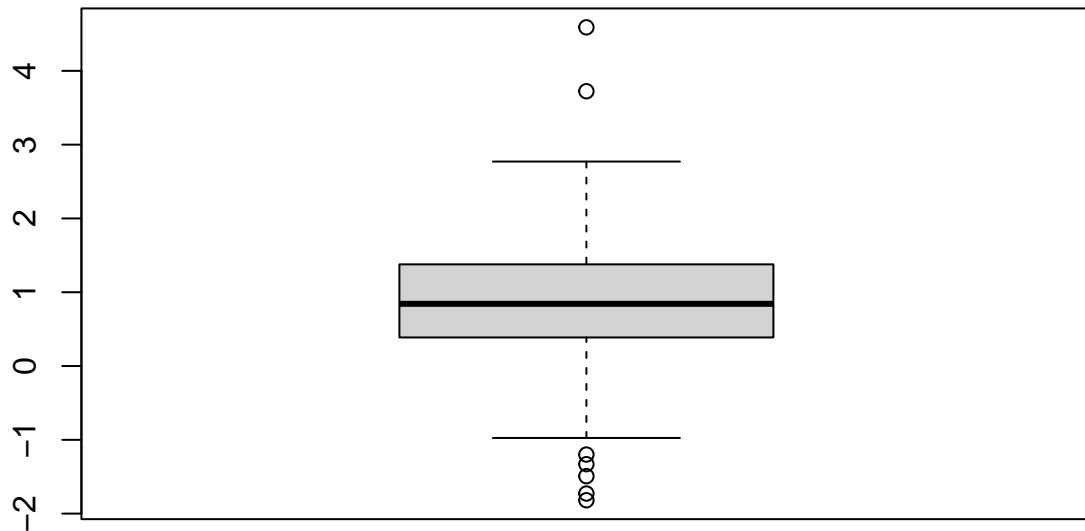
## Real disposable personal income

```
Rdpi <- mydata_ts[, 5]
log_rdpi <- log(Rdpi)
rdpi_growth <- diff(log_rdpi) * 100
summary(rdpi_growth)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.8188  0.3873   0.8428   0.8481  1.3778   4.5896
```

```
boxplot(rdpi_growth, main = "Boxplot of Rdpi growth from 1950-2000", xlab = "Disposable personal income")
```

## Boxplot of Rdpi growth from 1950–2000

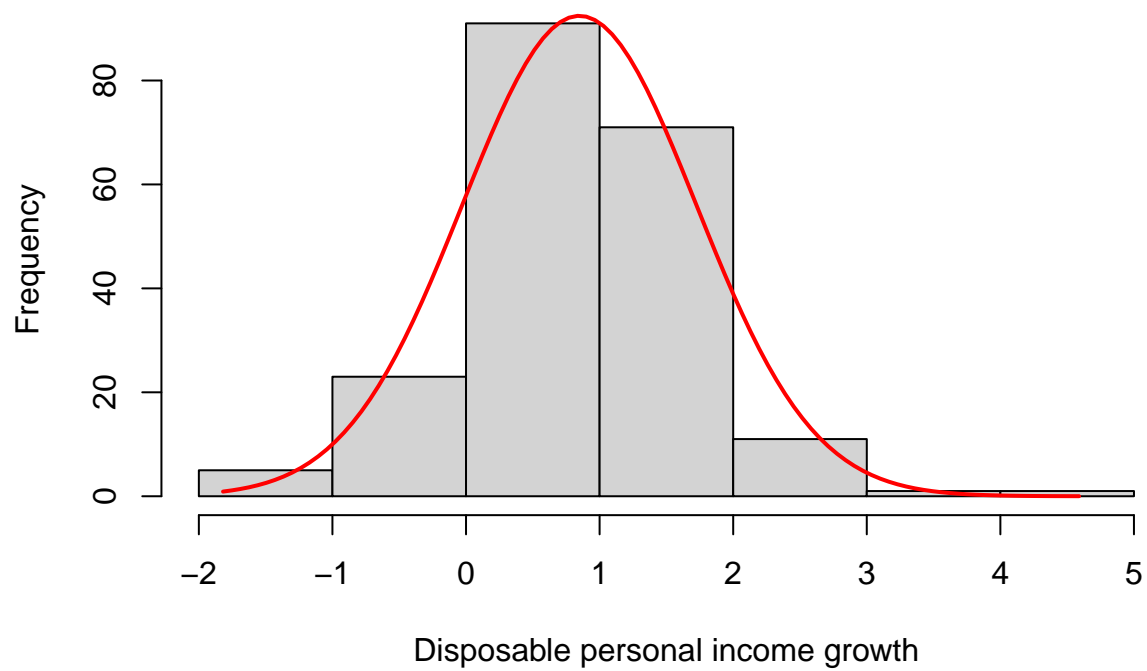


Disposable personal income growth

```
Rdpi_hist <- hist(rdpi_growth, main = "Histogram of Disposable personal income growth", xlab = "Disposable personal income growth")
xfit <- seq(min(rdpi_growth), max(rdpi_growth), length = 100)
yfit <- dnorm(xfit, mean = mean(rdpi_growth), sd = sd(rdpi_growth))
yfit <- yfit * diff(Rdpi_hist$mids[1:2]) * length(rdpi_growth)
lines(xfit, yfit, col = "red", lwd = 2)
```

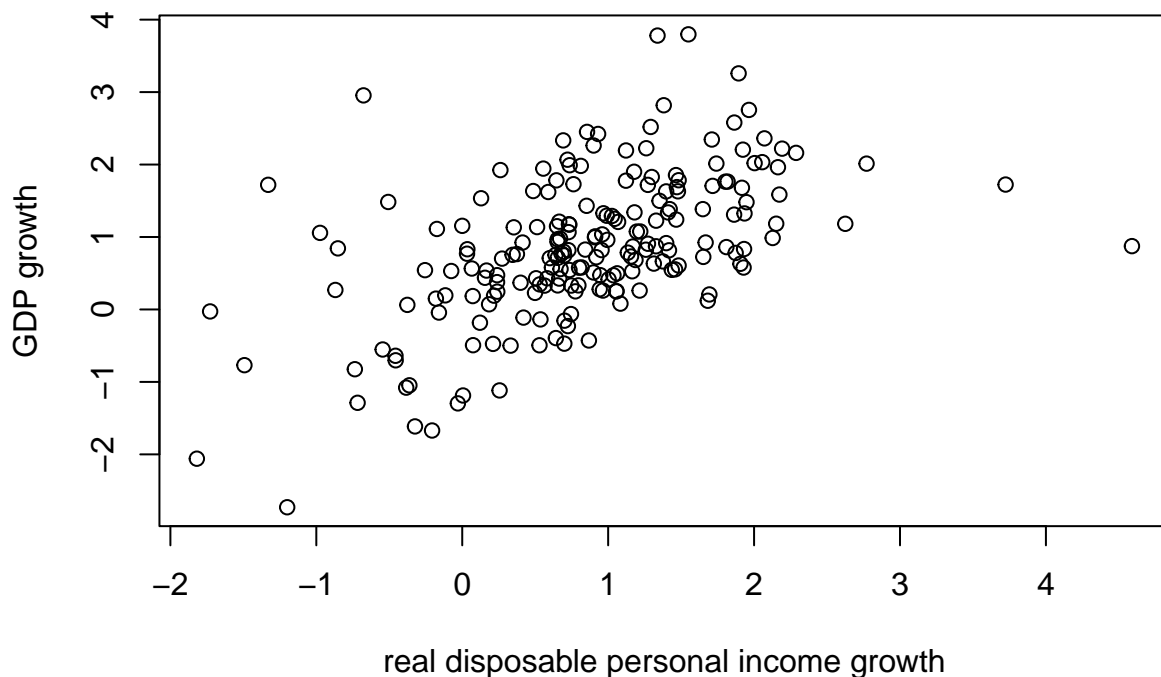


## Histogram of Disposable personal income growth



```
plot(rdpi_growth, gdp_growth, main = "Scatterplot of real disposable personal income growth compared to
```

## atterplot of real disposable personal income growth compared to GDP



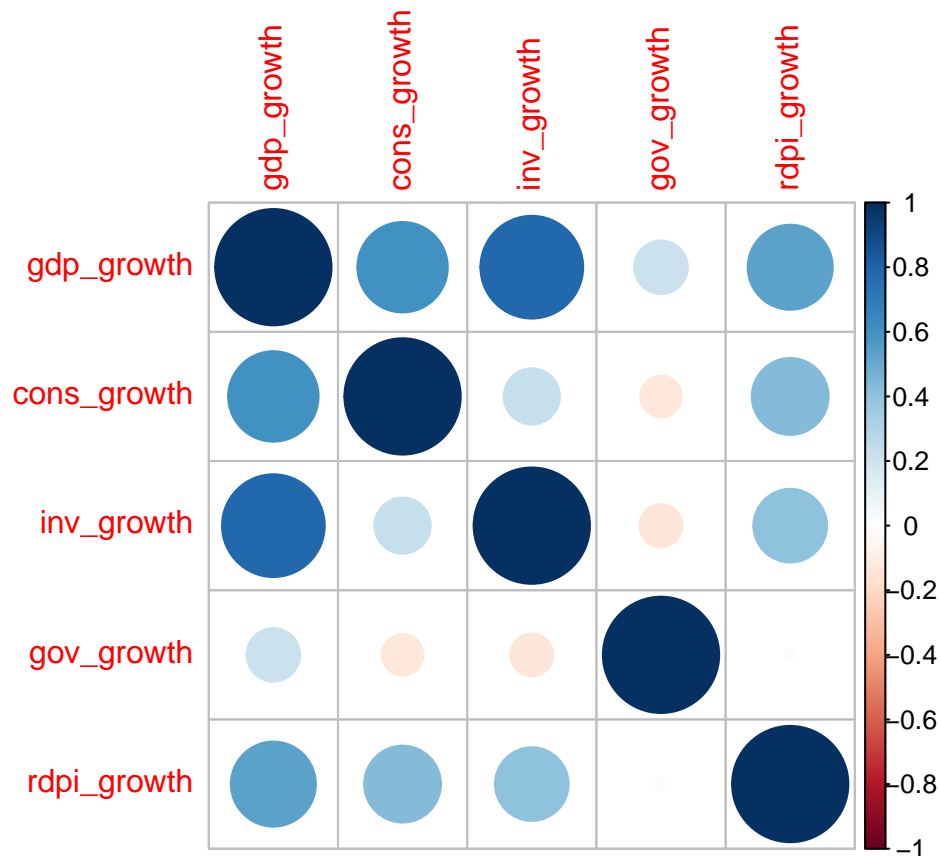
Comments: The Rdpi variable refers to the growth of real disposable personal income, one of the predictor variables in our dataset. From the histogram, the data looks to be relatively normally distributed, with a couple main peaks in between 0 and 2 percent growth. There seems to be some potential outliers towards the top half of the data. The fitted red normal distribution line also looks like a pretty good fit on the histogram here. Looking at the box plot, we can see the Interquartile range box, with the median pretty much right at the center point, suggesting symmetrical data, and we can see that there are outliers in the data identified both on the top and bottom side of the data. Looking at the 5 number summary, we have the median and mean at 0.843 and 0.848 percent, and we have the IQR from 0.387 to 1.378 percent. Looking at the scatterplot, we can also see a clear trend, as consumption increases, GDP increases. However, similar to government expenditures, past a certain point, around 2.5 percent, it seems like the positive trend becomes less evident although this could be due to a lack of data points as well.

## Correlation matrix

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
my_dataframe <- cbind(gdp_growth, cons_growth, inv_growth, gov_growth, rdpi_growth)
cor_matrix <- cor(my_dataframe)
corrplot(cor_matrix)
```

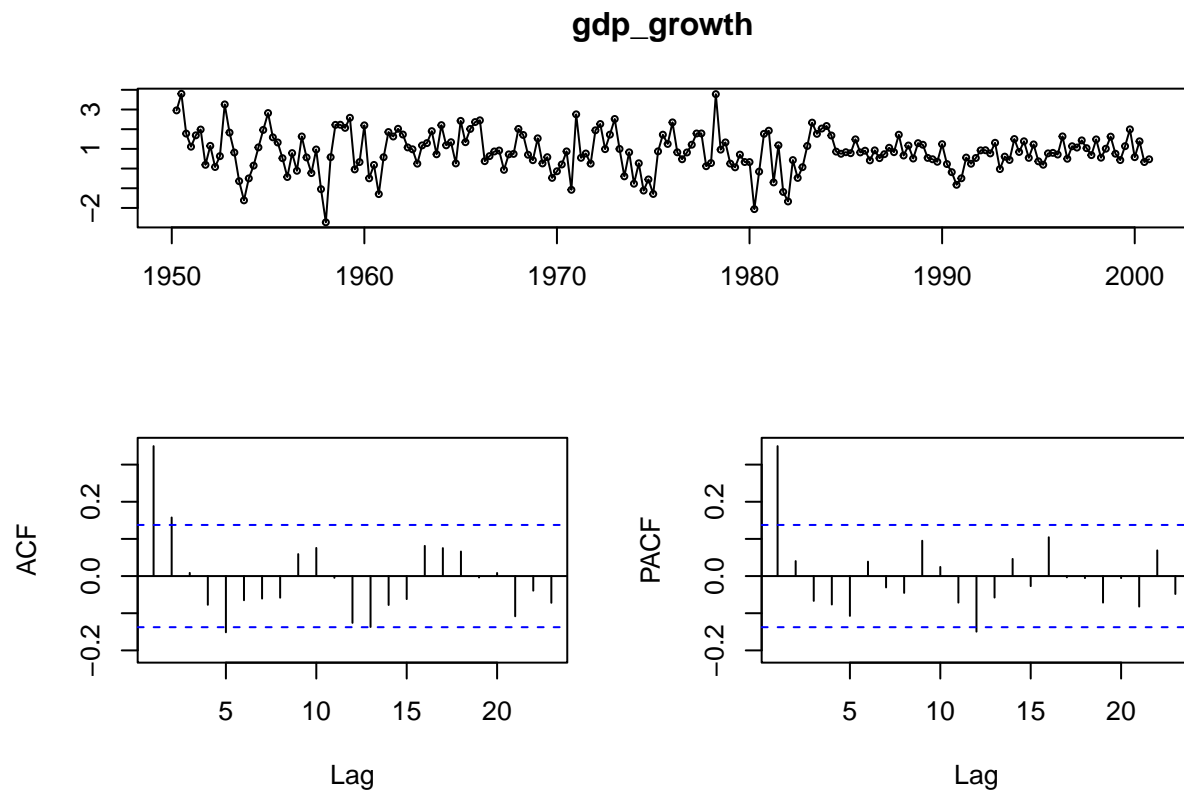


Comments: From this correlation matrix, we can see different correlation values for different pairs, but we can see trends in the correlation matrix. For gdp growth, we can see that it has relatively high correlations with most of the variables except for government expenditure growth. For consumption growth, it's pretty similar however it doesn't have as high of a correlation with investment growth on top of government expenditure growth. This same trend in the correlations can be seen with real disposable income growth as well. For government expenditure growth, it has relatively low correlations with all the other variables which could be significant. So overall, this might show that government expenditure growth may not be as significant in gdp and consumption growth than the other two predictors of rdpi and investment growth

## Part 2: tsdisplay, ACF and PACF of each variable

### GDP

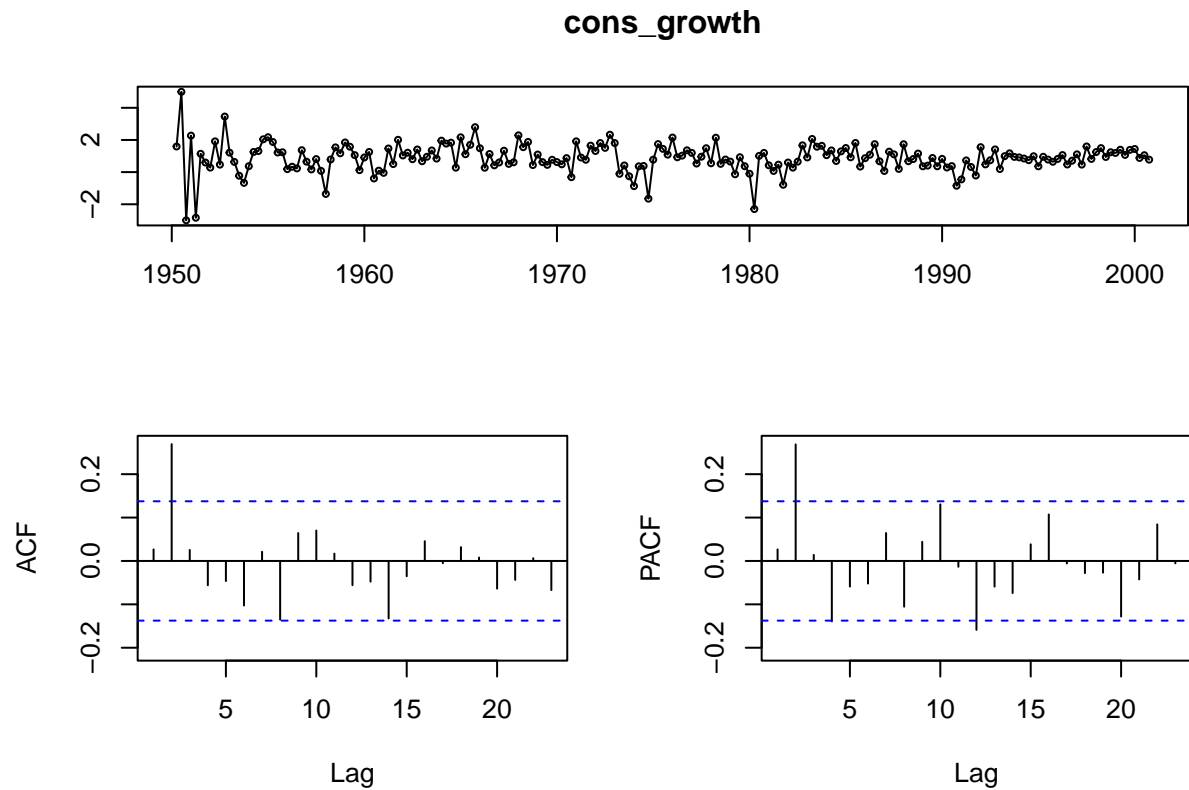
```
tsdisplay(gdp_growth)
```



Based on the `tsdisplay`, the series looks relatively stationary, with potentially a slight downward trend but mean reverting mostly. Looking at the ACF and PACF plots, there aren't too much dynamics to see, however the first 2 lags in the ACF and the first lag in the PACF cross the bands, suggesting up to 2 lags of gdp growth may be significant.

## Consumption

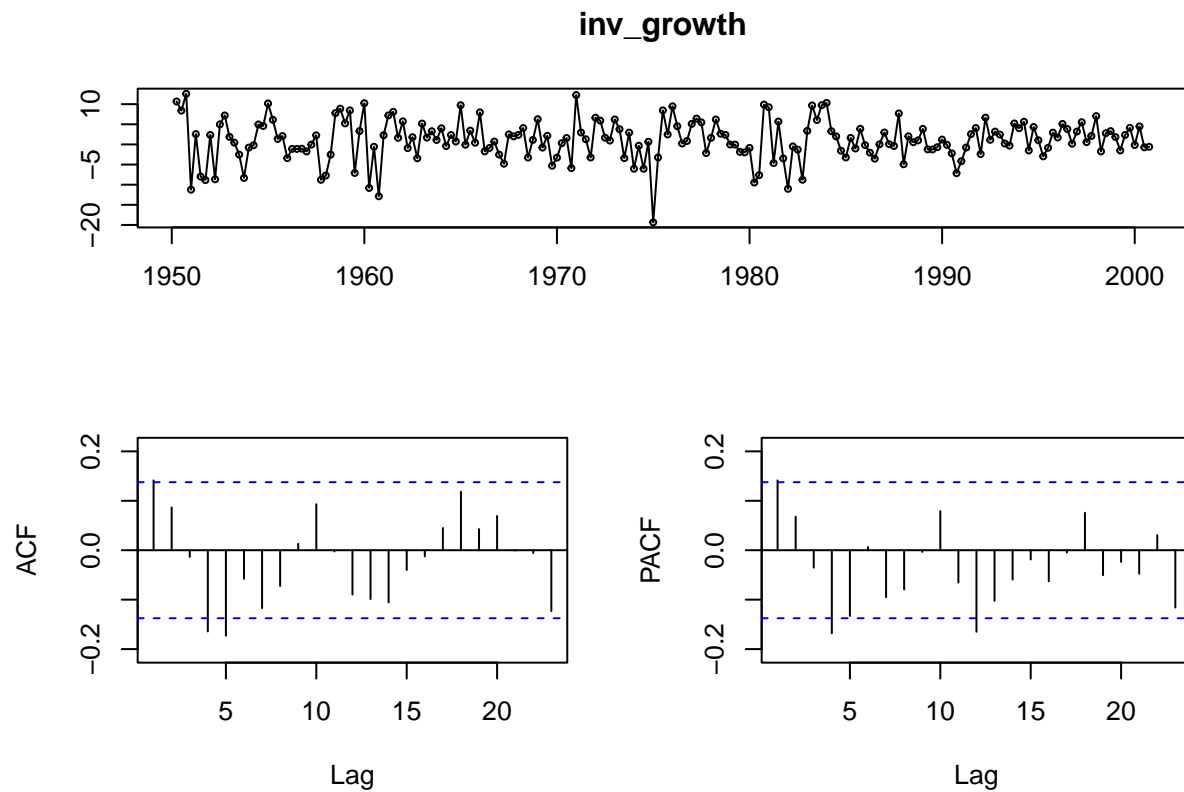
```
tsdisplay(cons_growth)
```



Based on the `tsdisplay`, the series looks mostly stationary except for the first couple years where there is an increased volatility. There are some downward spikes however throughout the data with no real big upward jumps. Looking at the ACF and PACF plots the 2nd lag is the only lag that crosses the threshold in both the ACF and PACF plot, suggesting that lag is significant.

## Investment

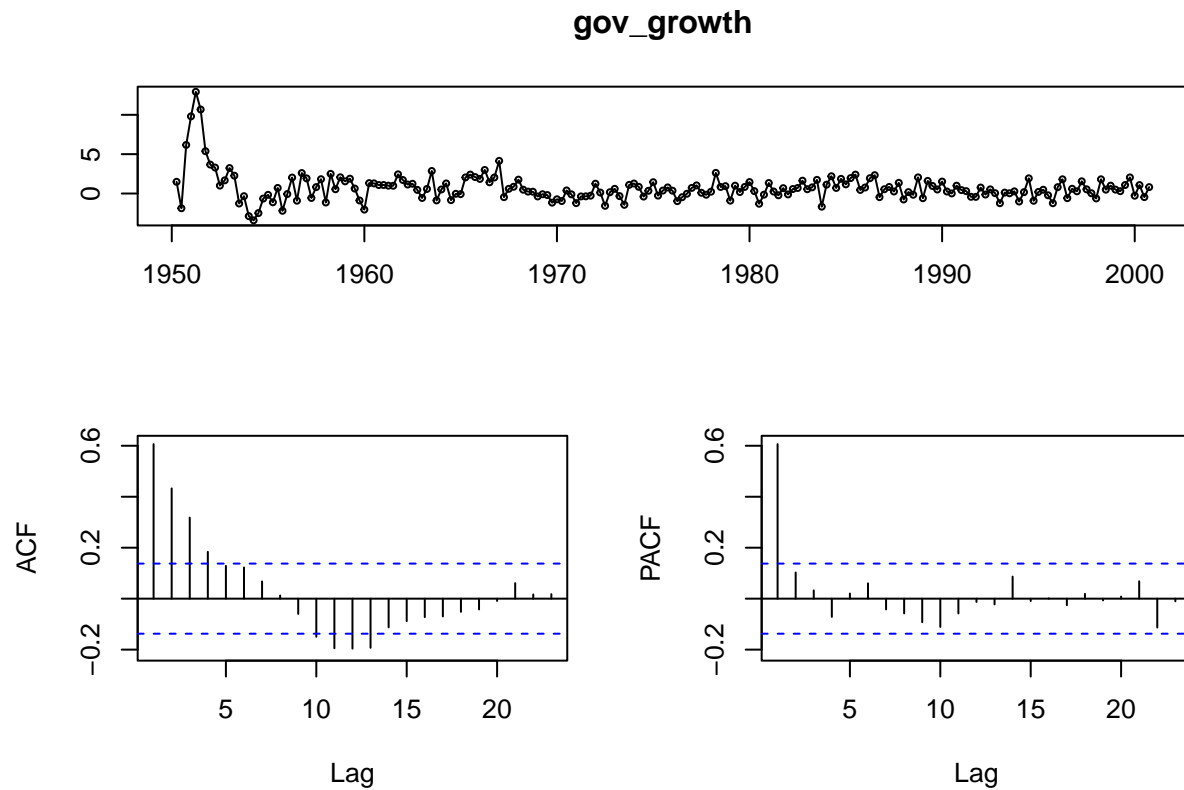
```
tsdisplay(inv_growth)
```



Based on the `tsdisplay`, the series also looks mostly stationary however there is a lot more volatility in the data, but the series still reverts. Looking at the ACF and PACF plots there are no lags that significantly cross the threshold, however there are a good amount of lags that barely cross the threshold, suggesting there may not be too much important dynamics.

## Government spending

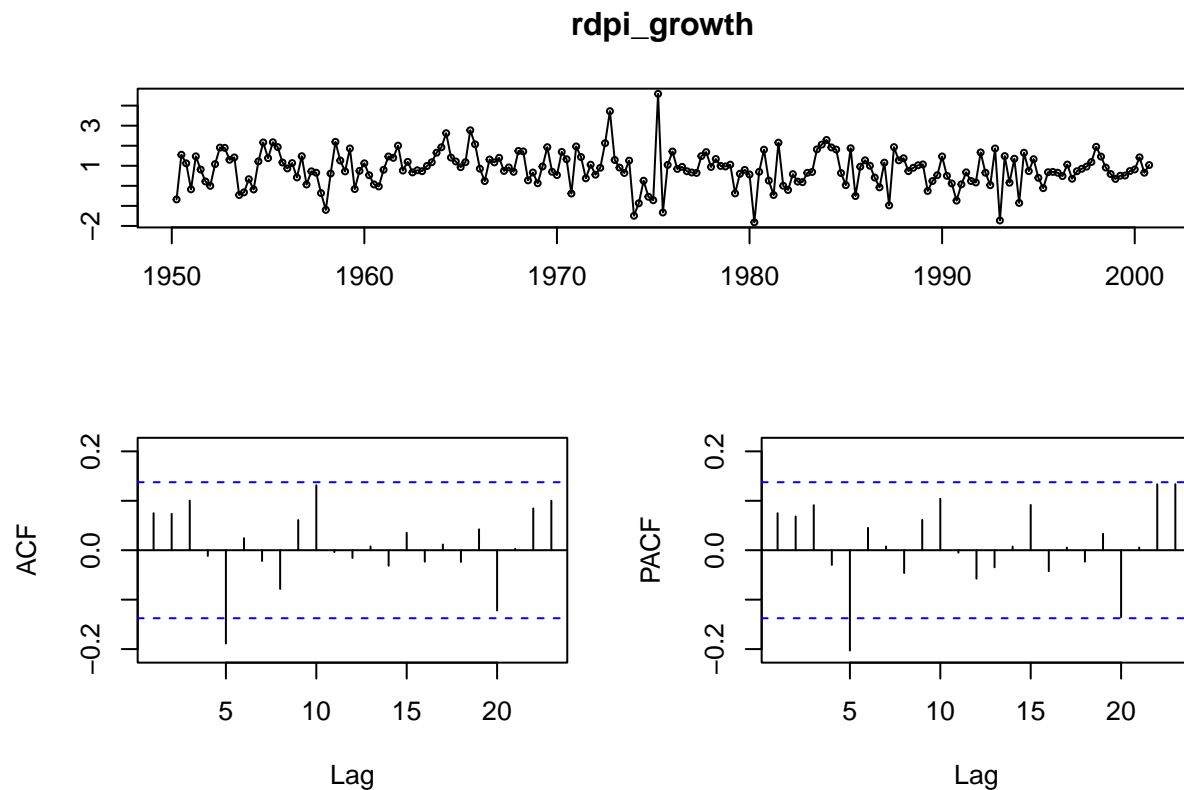
```
tsdisplay(gov_growth)
```



Based on the `tsdisplay`, except for the beginning years the series also looks mostly stationary and mean reverting. Looking at the ACF and PACF plots, we see that the PACF has one big spike at lag one, and the ACF has decay in the lags with the first 3-4 lags crossing the threshold. This suggests that the first lag is statistically significant, and potentially the next couple as well but mainly the first lag.

## Disposable personal income

```
tsdisplay(rdpi_growth)
```



Based on the tsdisplay, the series also looks mostly stationary however there is a lot more volatility in the data, but the series still reverts but not too quickly. Looking at the ACF and PACF plots, the 5th lag crosses the threshold pretty significantly in both cases, suggesting significance, with no other lags crossing the threshold bands. This seems a bit unusual as economically it doesn't make too much sense that the 5th lag would be more significant than lags 1-4

```
#Part 3
#This is for variable gdp growth.
```

```
library(stats)
library(forecast)
library(dLagM)
```

```
## Loading required package: nardl
```

```
##
## Attaching package: 'dLagM'
```

```
## The following objects are masked from 'package:fabletools':
##
##   forecast, MAPE, MASE
```

```
## The following object is masked from 'package:forecast':
##
##   forecast
```



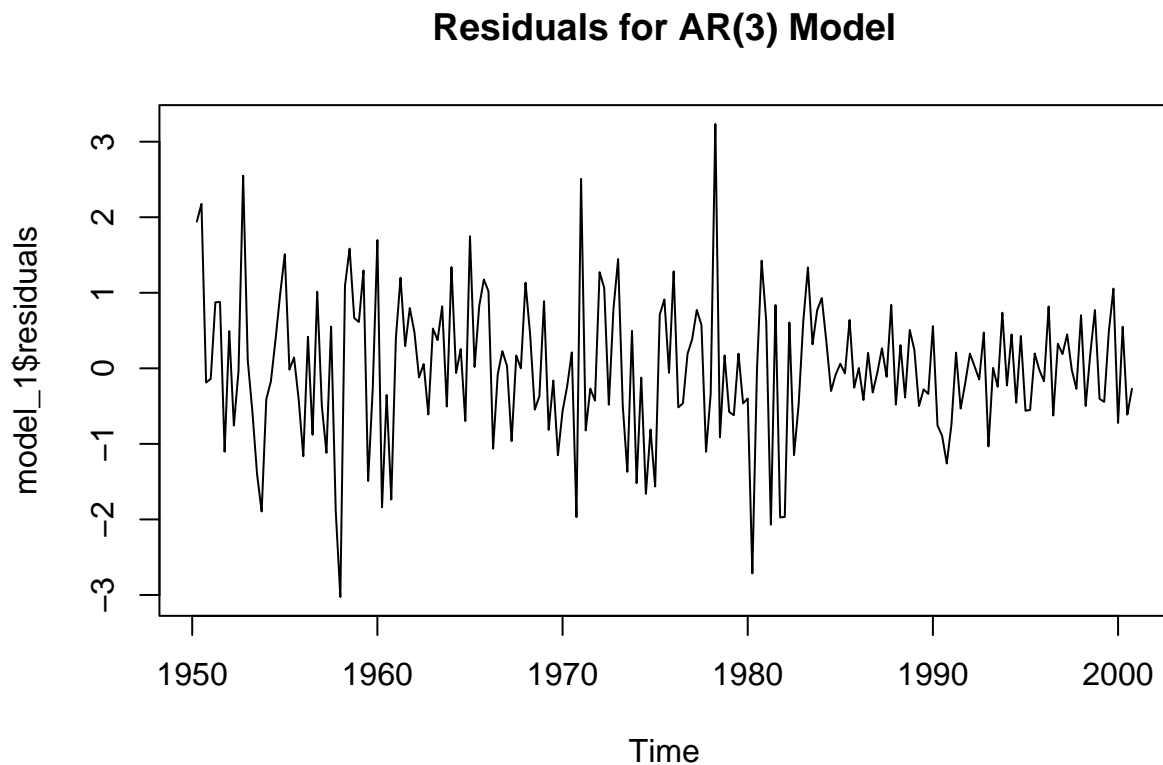
```

# Fit AR(p) models
order_1 <- c(3, 0, 0) # AR(3) model
order_2 <- c(5, 0, 0) # AR(5) model

# Fit ARIMA models to the 'cons_growth' variable
model_1 <- arima(my_dataframe[, 1], order=order_1)
model_2 <- arima(my_dataframe[, 1], order=order_2)

# Plot and comment on ACF and PACF of the residuals
plot(model_1$residuals, main="Residuals for AR(3) Model")

```

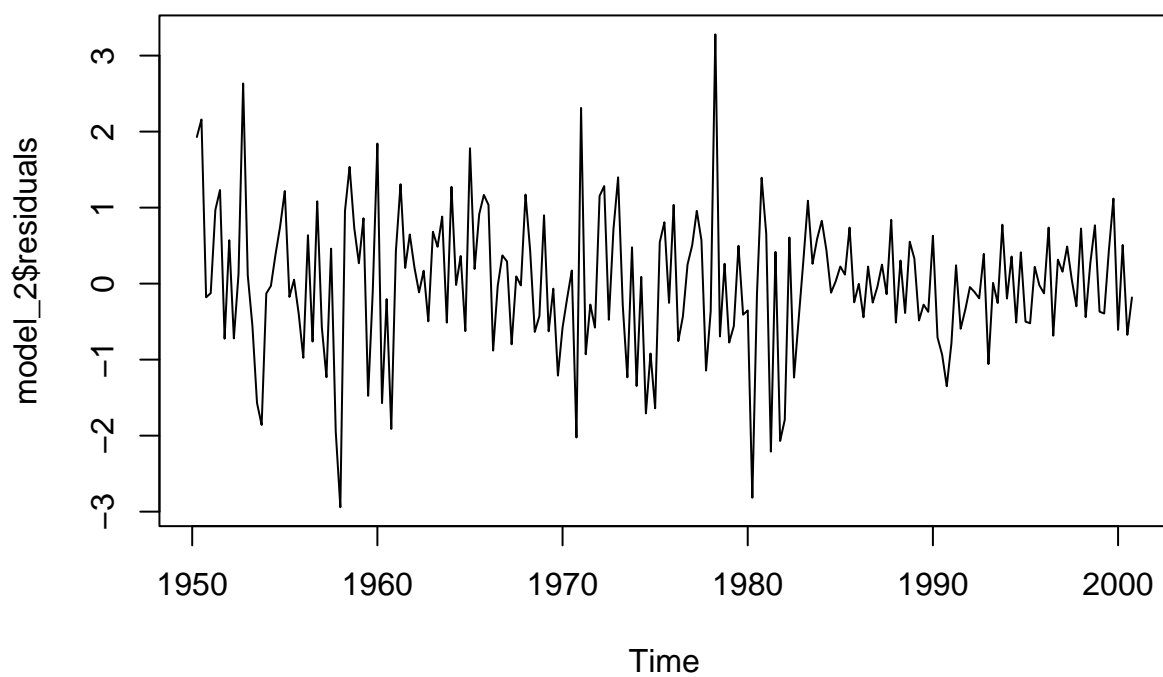


```

plot(model_2$residuals, main="Residuals for AR(5) Model")

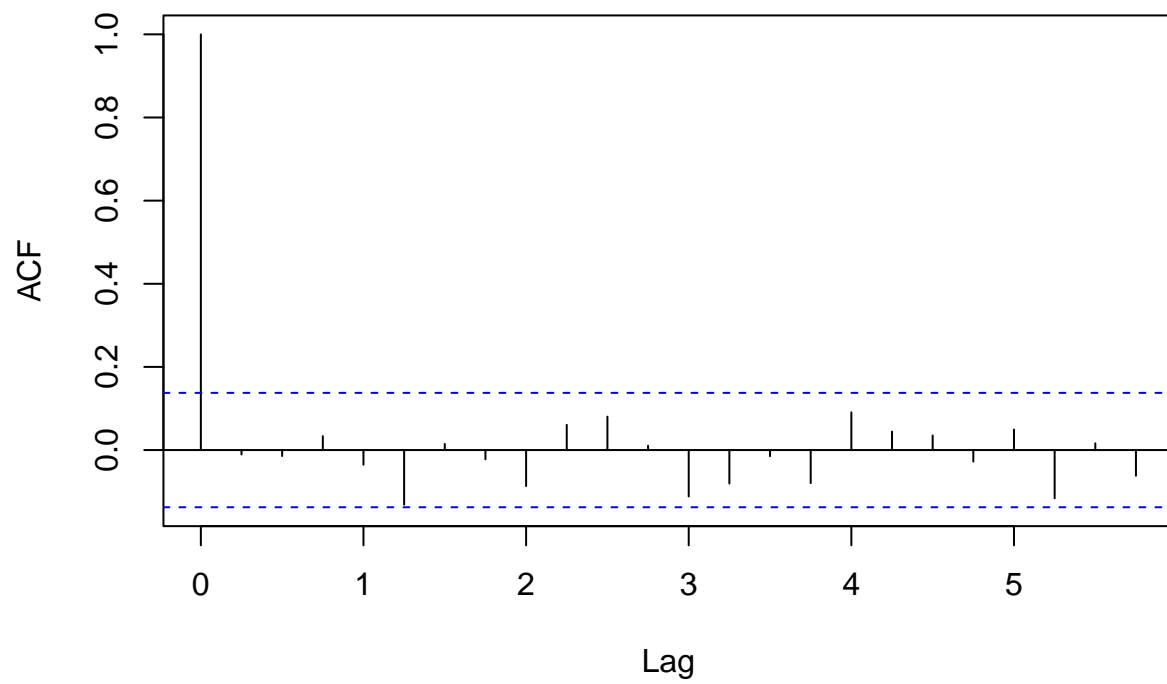
```

## Residuals for AR(5) Model



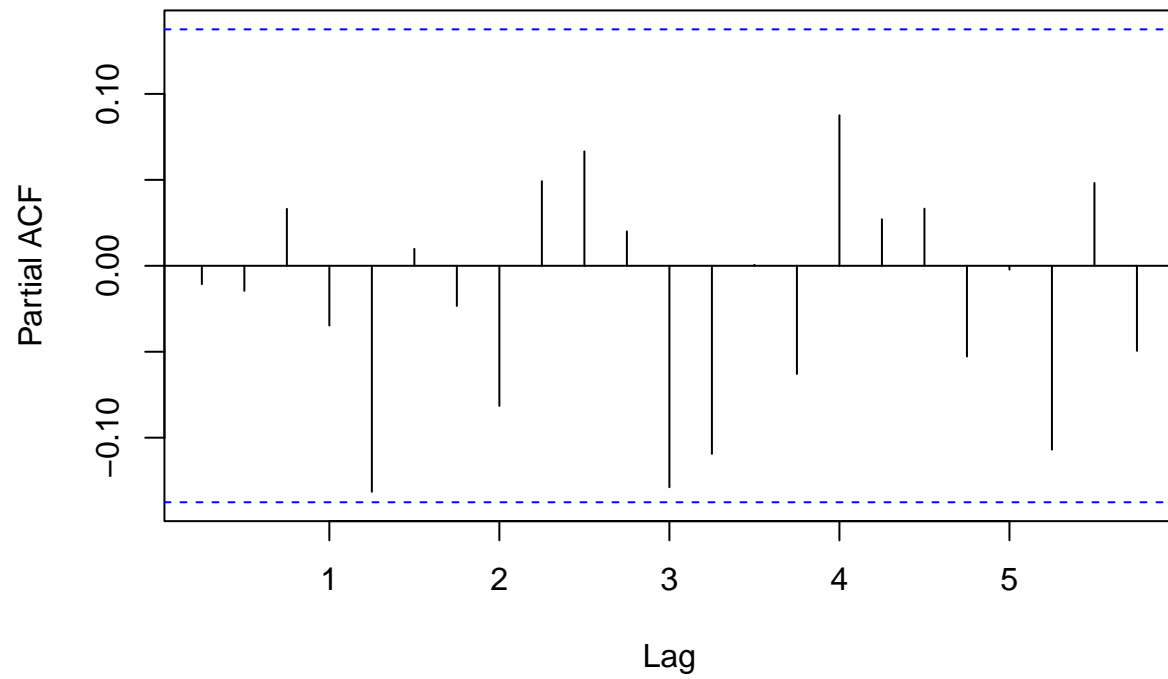
```
acf(model_1$residuals, main="ACF of Residuals for AR(3) Model")
```

### ACF of Residuals for AR(3) Model



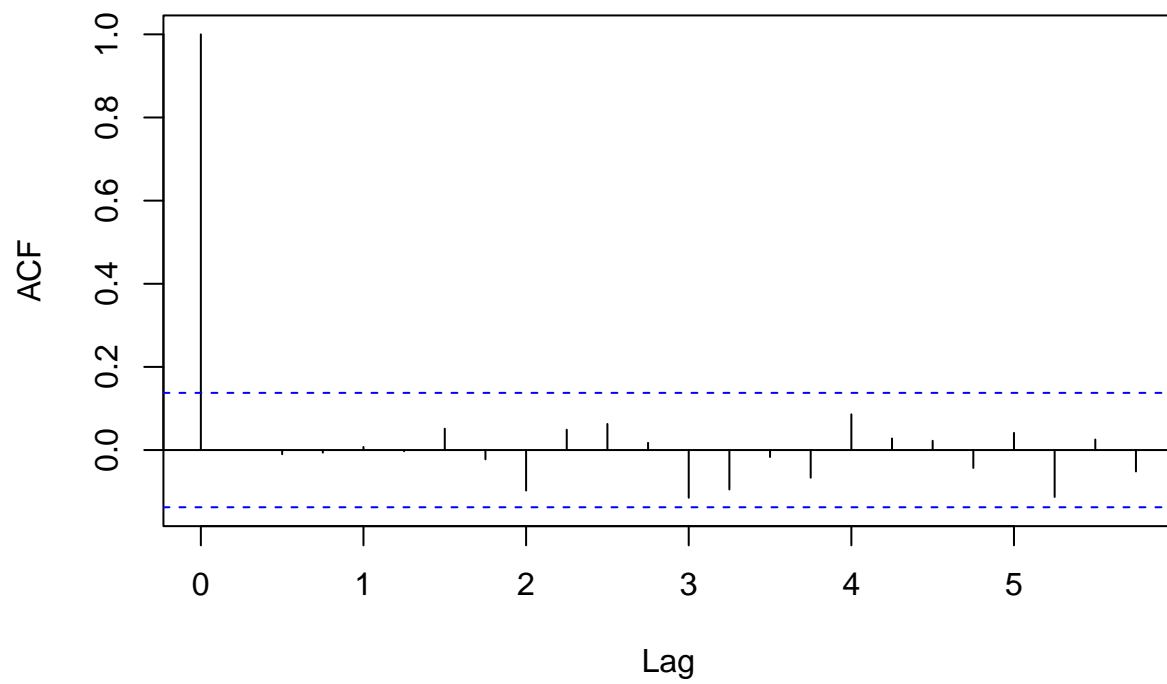
```
pacf(model_1$residuals, main="PACF of Residuals for AR(3) Model")
```

### PACF of Residuals for AR(3) Model



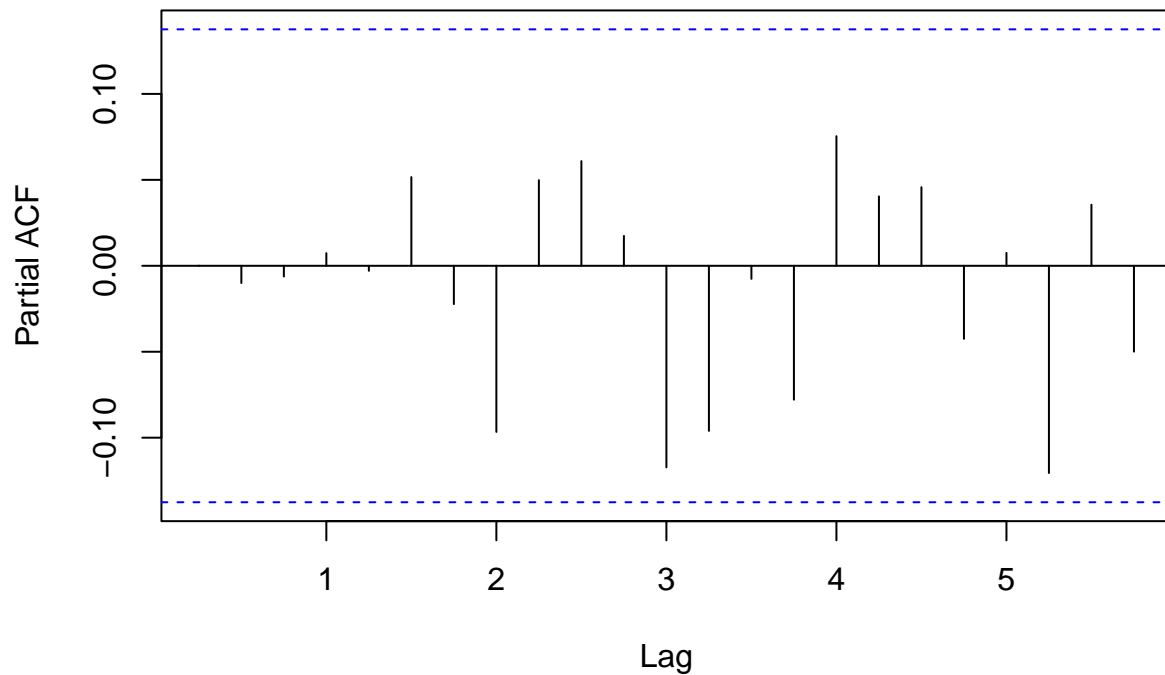
```
acf(model_2$residuals, main="ACF of Residuals for AR(5) Model")
```

### ACF of Residuals for AR(5) Model



```
pacf(model_2$residuals, main="PACF of Residuals for AR(5) Model")
```

## PACF of Residuals for AR(5) Model



*#The ACFs of model 1 and model 2 for the AR(3) and AR(5) models respectively both only had one dynamic*

```
train_size <- round(2/3 * length(my_dataframe[, 1]))
```

*# Split the data*

```
train_data <- my_dataframe[, 1][1:train_size]
```

```
test_data <- ts(my_dataframe[, 1][(train_size + 1):length(my_dataframe[, 1])], )
```

```
library(forecast)
```

*# Forecasting*

```
forecast_1 <- predict(model_1, n.ahead = length(test_data))
```

```
forecast_2 <- predict(model_2, n.ahead=length(test_data))
```

*# Compute MSE/RMSE*

```
mse_1 <- mean((forecast_1$pred - test_data)^2)
```

```
## Error in .cbind.ts(list(e1, e2), c(deparse(substitute(e1))[1L], deparse(substitute(e2))[1L]), : not a
```

```
mse_2 <- mean((forecast_2$pred - test_data)^2)
```

```
## Error in .cbind.ts(list(e1, e2), c(deparse(substitute(e1))[1L], deparse(substitute(e2))[1L]), : not a
```

```
cat("MSE for Model 1:", mse_1, "\n")
```

```
## Error in cat("MSE for Model 1:", mse_1, "\n"): object 'mse_1' not found
```

```
cat("MSE for Model 2:", mse_2, "\n")
```

```
## Error in cat("MSE for Model 2:", mse_2, "\n"): object 'mse_2' not found
```

```
# Calculate AIC and BIC for both models
```

```
aic_1 <- AIC(model_1)
```

```
bic_1 <- BIC(model_1)
```

```
aic_2 <- AIC(model_2)
```

```
bic_2 <- BIC(model_2)
```

```
cat("AIC for Model 1:", aic_1, "\n")
```

```
## AIC for Model 1: 555.4239
```

```
cat("BIC for Model 1:", bic_1, "\n")
```

```
## BIC for Model 1: 571.9899
```

```
cat("AIC for Model 2:", aic_2, "\n")
```

```
## AIC for Model 2: 555.7423
```

```
cat("BIC for Model 2:", bic_2, "\n")
```

```
## BIC for Model 2: 578.9348
```

```
#According to the MSE, model 2 is better due to it being a lower value; however according to the AIC/BIC
```

```
# 10-step ahead forecast
```

```
forecast_10_1 <- predict(model_1, n.ahead=10)
```

```
forecast_10_2 <- predict(model_2, n.ahead=10)
```

```
print("10-step ahead forecast for Model 1:")
```

```
## [1] "10-step ahead forecast for Model 1:"
```

```
print(forecast_10_1)
```

```
## $pred
```

```
##           Qtr1           Qtr2           Qtr3           Qtr4
```

```
## 2001 0.6564012 0.8033842 0.8576489 0.8742562
```

```
## 2002 0.8737119 0.8709707 0.8688329 0.8679355
```

```
## 2003 0.8676609 0.8676495
```

```
##
```

```
## $se
```

```
##           Qtr1           Qtr2           Qtr3           Qtr4
```

```
## 2001 0.9268957 0.9797505 0.9956874 0.9958810
```

```
## 2002 0.9958836 0.9959523 0.9959678 0.9959712
```

```
## 2003 0.9959713 0.9959713
```

```
print("10-step ahead forecast for Model 2:")
```

```
## [1] "10-step ahead forecast for Model 2:"
```

```
print(forecast_10_2)
```

```
## $pred
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.5642151 0.7603694 0.7869866 0.9166514
## 2002 0.9344062 0.9296039 0.9018642 0.8827846
## 2003 0.8594960 0.8496924
##
## $se
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.9183240 0.9663474 0.9807024 0.9817169
## 2002 0.9820756 0.9902132 0.9939479 0.9955957
## 2003 0.9958939 0.9958940
```

```
#Part 3
```

```
#This is for variable consumption growth
```

```
library(stats)
```

```
library(forecast)
```

```
# Fit AR(p) models
```

```
order_1 <- c(3, 0, 0) # AR(3) model
```

```
order_2 <- c(5, 0, 0) # AR(5) model
```

```
# Fit ARIMA models to the 'cons_growth' variable
```

```
model_1 <- Arima(my_dataframe[, 2], order=order_1)
```

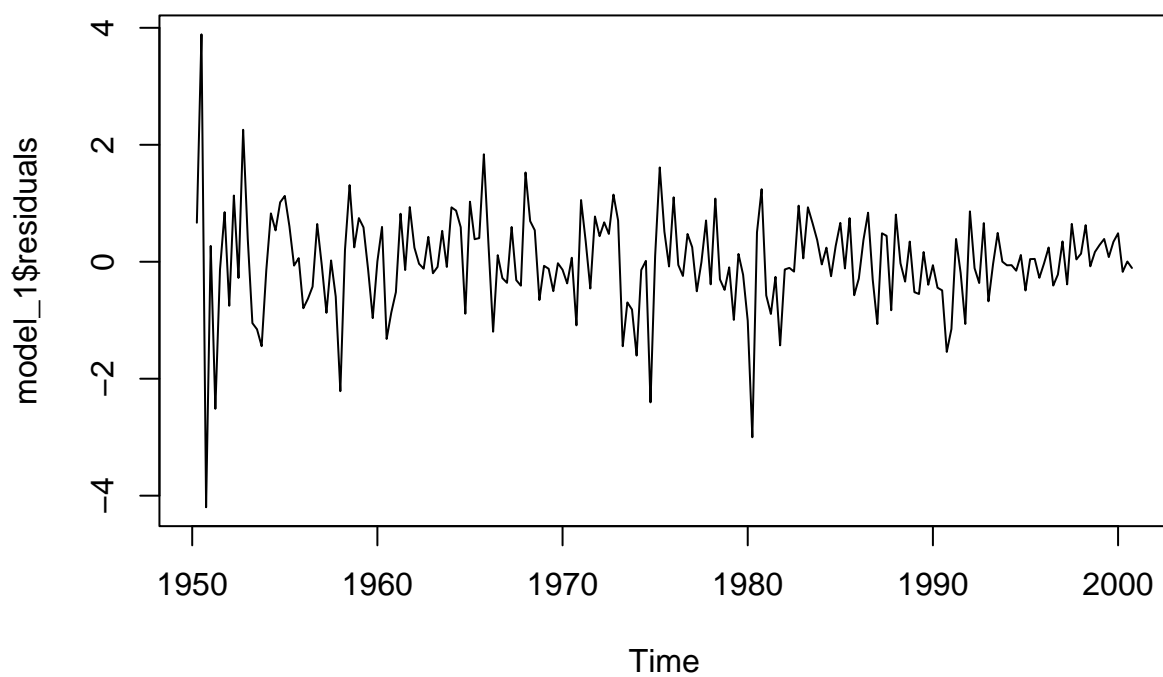
```
model_2 <- Arima(my_dataframe[, 2], order=order_2)
```

```
# Plot and comment on ACF and PACF of the residuals
```

```
plot(model_1$residuals, main="Residuals for AR(3) Model")
```

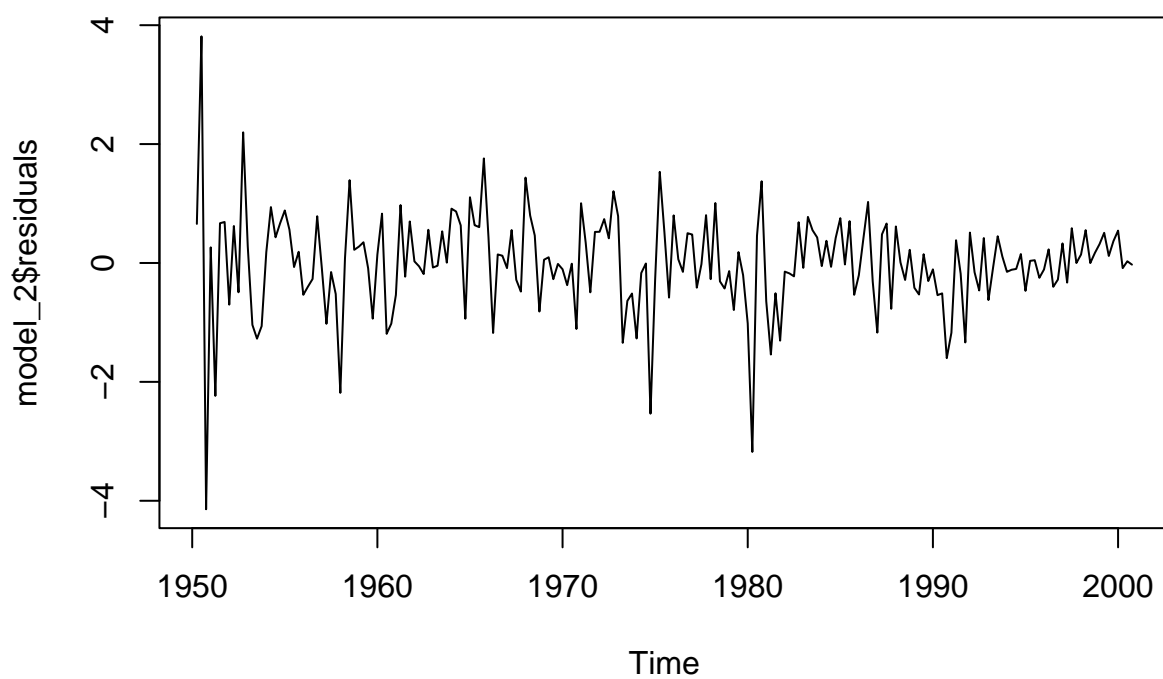


### Residuals for AR(3) Model



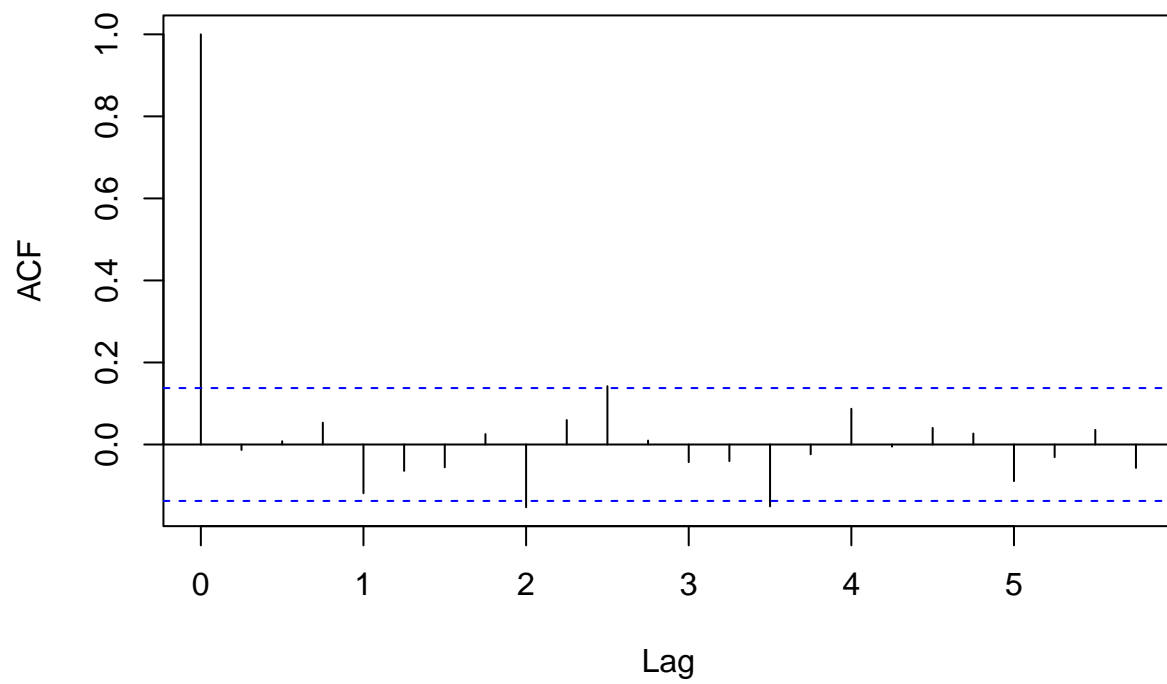
```
plot(model_2$residuals, main="Residuals for AR(5) Model")
```

## Residuals for AR(5) Model



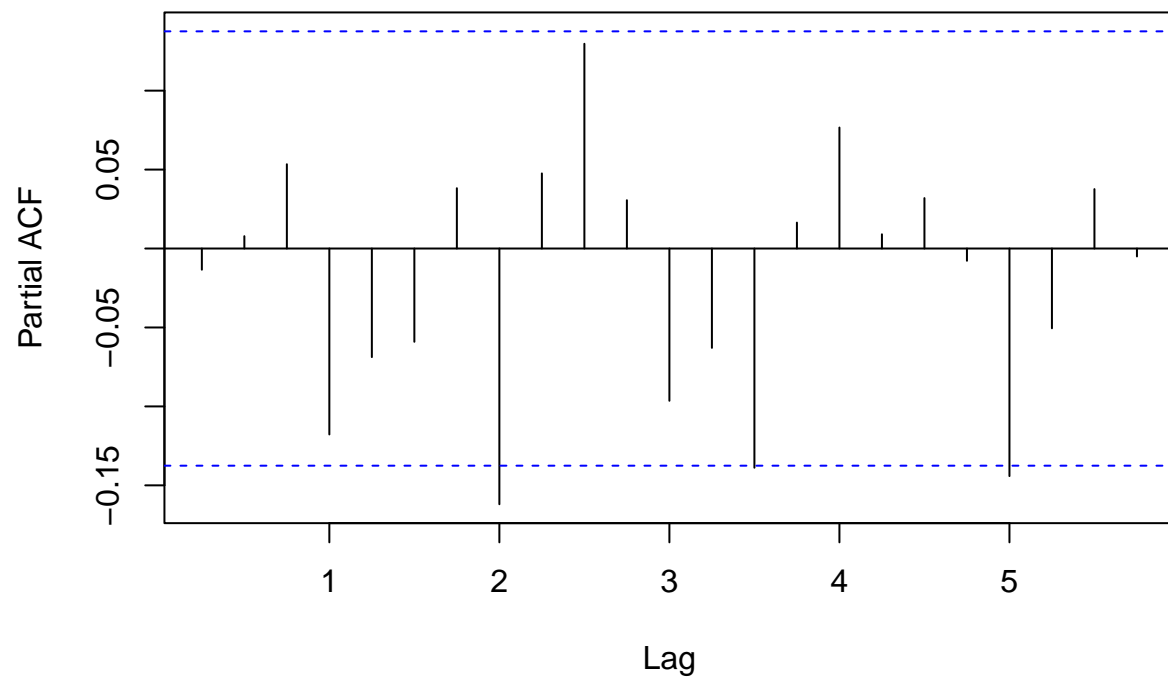
```
acf(model_1$residuals, main="ACF of Residuals for AR(3) Model")
```

### ACF of Residuals for AR(3) Model



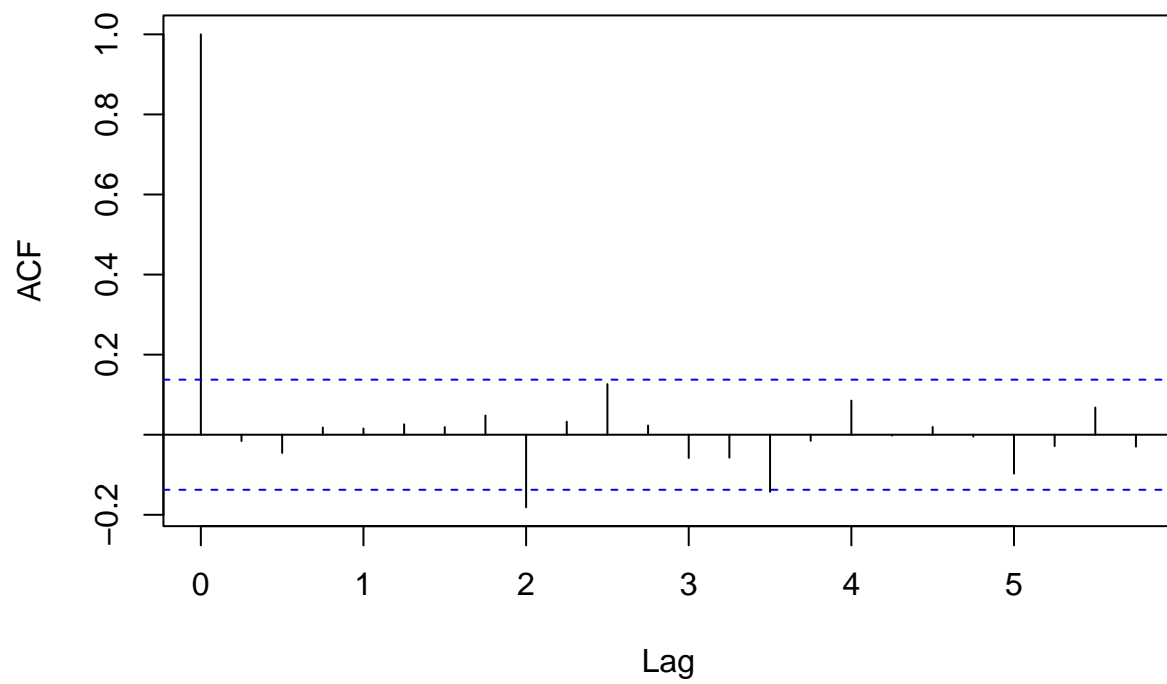
```
pacf(model_1$residuals, main="PACF of Residuals for AR(3) Model")
```

### PACF of Residuals for AR(3) Model



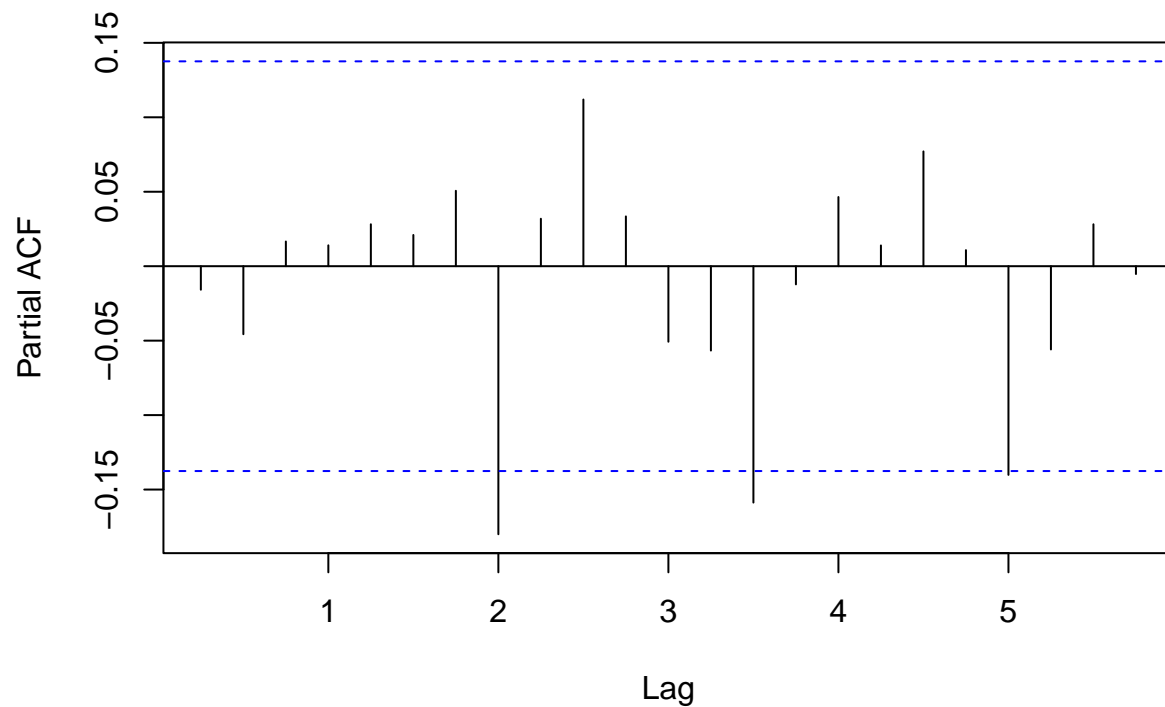
```
acf(model_2$residuals, main="ACF of Residuals for AR(5) Model")
```

### ACF of Residuals for AR(5) Model



```
pacf(model_2$residuals, main="PACF of Residuals for AR(5) Model")
```

## PACF of Residuals for AR(5) Model



*#The ACFs of model 1 and model 2 for the AR(3) and AR(5) models respectively both had 3 dynamics and th*

```
train_size <- round(2/3 * length(my_dataframe[, 2]))
```

*# Split the data*

```
train_data <- my_dataframe[, 2][1:train_size]
```

```
test_data <- my_dataframe[, 2][(train_size + 1):length(my_dataframe[, 2])]
```

*# Forecasting*

```
forecast_1 <- predict(model_1, n.ahead=length(test_data))
```

```
forecast_2 <- predict(model_2, n.ahead=length(test_data))
```

*# Compute MSE/RMSE*

```
mse_1 <- mean((forecast_1$pred - test_data)^2)
```

```
mse_2 <- mean((forecast_2$pred - test_data)^2)
```

```
cat("MSE for Model 1:", mse_1, "\n")
```

```
## MSE for Model 1: 0.2528981
```

```
cat("MSE for Model 2:", mse_2, "\n")
```

```
## MSE for Model 2: 0.2537305
```

```
# Calculate AIC and BIC for both models
```

```
aic_1 <- AIC(model_1)
```

```
bic_1 <- BIC(model_1)
```

```
aic_2 <- AIC(model_2)
```

```
bic_2 <- BIC(model_2)
```

```
cat("AIC for Model 1:", aic_1, "\n")
```

```
## AIC for Model 1: 518.5008
```

```
cat("BIC for Model 1:", bic_1, "\n")
```

```
## BIC for Model 1: 535.0668
```

```
cat("AIC for Model 2:", aic_2, "\n")
```

```
## AIC for Model 2: 516.1024
```

```
cat("BIC for Model 2:", bic_2, "\n")
```

```
## BIC for Model 2: 539.2949
```

```
#Model 1 is the better model based on the MSE and the AIC/BIC because all three metrics have lower values
```

```
# 10-step ahead forecast
```

```
forecast_10_1 <- predict(model_1, n.ahead = 10)
```

```
forecast_10_2 <- predict(model_2, n.ahead = 10)
```

```
print("10-step ahead forecast for Model 1:")
```

```
## [1] "10-step ahead forecast for Model 1:"
```

```
print(forecast_10_1)
```

```
## $pred
```

```
##           Qtr1           Qtr2           Qtr3           Qtr4
```

```
## 2001 0.9368957 0.8552046 0.9068471 0.8805481
```

```
## 2002 0.8969623 0.8884943 0.8937148 0.8909905
```

```
## 2003 0.8926520 0.8917762
```

```
##
```

```
## $se
```

```
##           Qtr1           Qtr2           Qtr3           Qtr4
```

```
## 2001 0.8547065 0.8550856 0.8933282 0.8933296
```

```
## 2002 0.8966807 0.8966867 0.8969893 0.8969911
```

```
## 2003 0.8970188 0.8970191
```

```
print("10-step ahead forecast for Model 2:")
```

```
## [1] "10-step ahead forecast for Model 2:"
```

```
print(forecast_10_2)
```

```
## $pred
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.8099630 0.8060580 0.8302742 0.8618783
## 2002 0.8887693 0.8987592 0.9055317 0.9024740
## 2003 0.8978468 0.8928699
##
## $se
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.8452728 0.8453812 0.8971806 0.8977176
## 2002 0.8983875 0.9001762 0.9024597 0.9040180
## 2003 0.9042833 0.9043213
```

```
#Part 3
#This is for variable investment growth.
library(dLagM)
library(stats)
library(forecast)

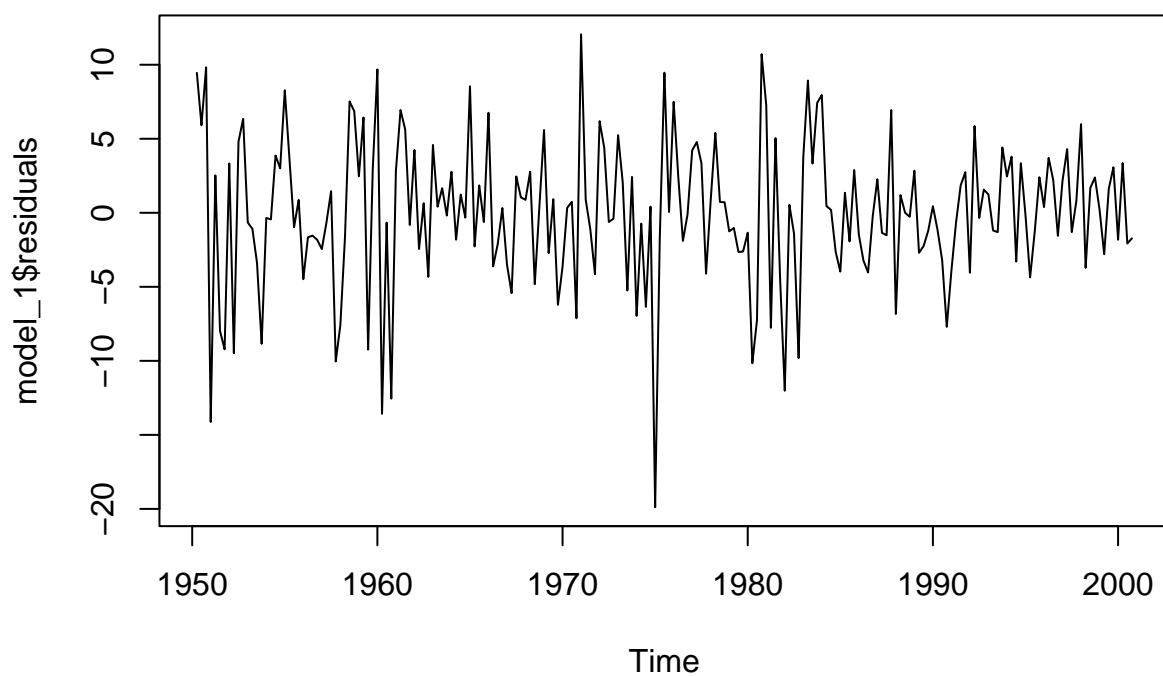
# Fit AR(p) models
order_1 <- c(3, 0, 0) # AR(3) model
order_2 <- c(5, 0, 0) # AR(5) model

# Fit ARIMA models to the 'inv_growth' variable
model_1 <- Arima(my_dataframe[, 3], order=order_1)
model_2 <- Arima(my_dataframe[, 3], order=order_2)

# Plot and comment on ACF and PACF of the residuals
plot(model_1$residuals, main="Residuals for AR(3) Model")
```

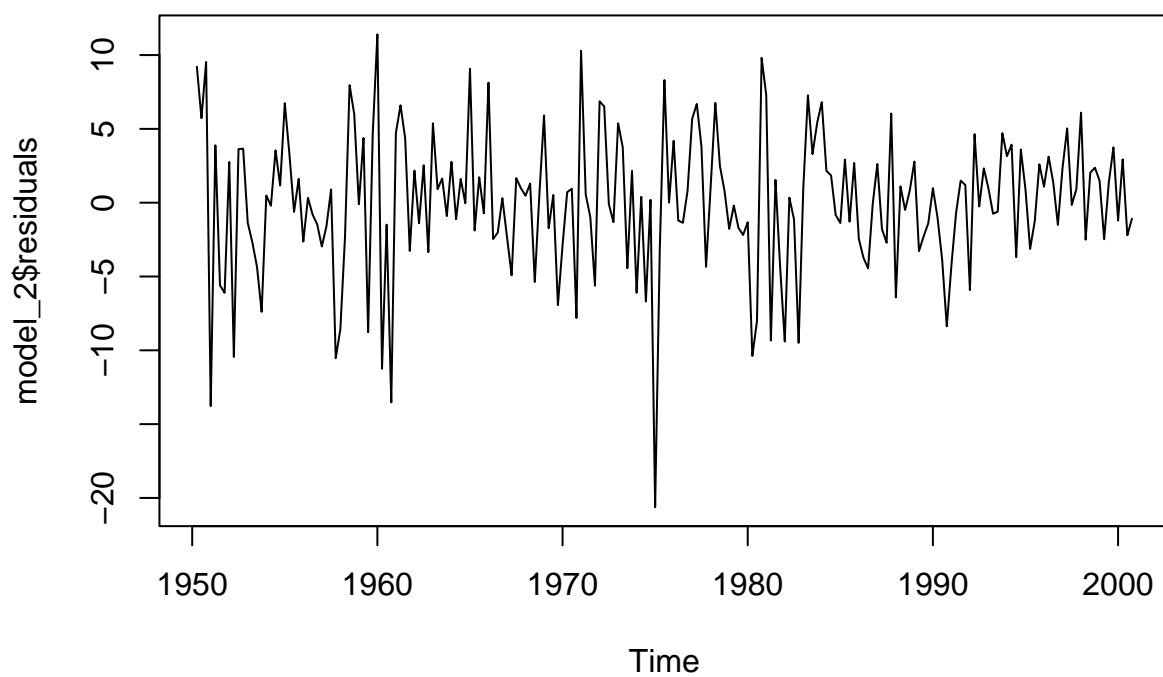


### Residuals for AR(3) Model



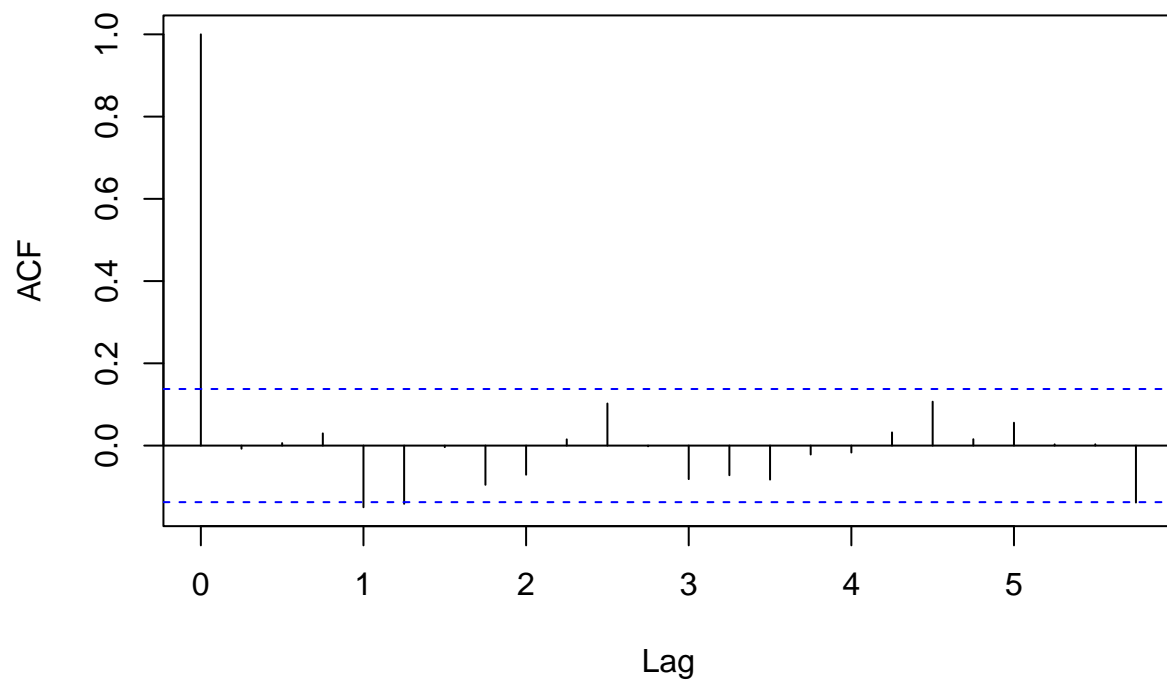
```
plot(model_2$residuals, main="Residuals for AR(5) Model")
```

### Residuals for AR(5) Model



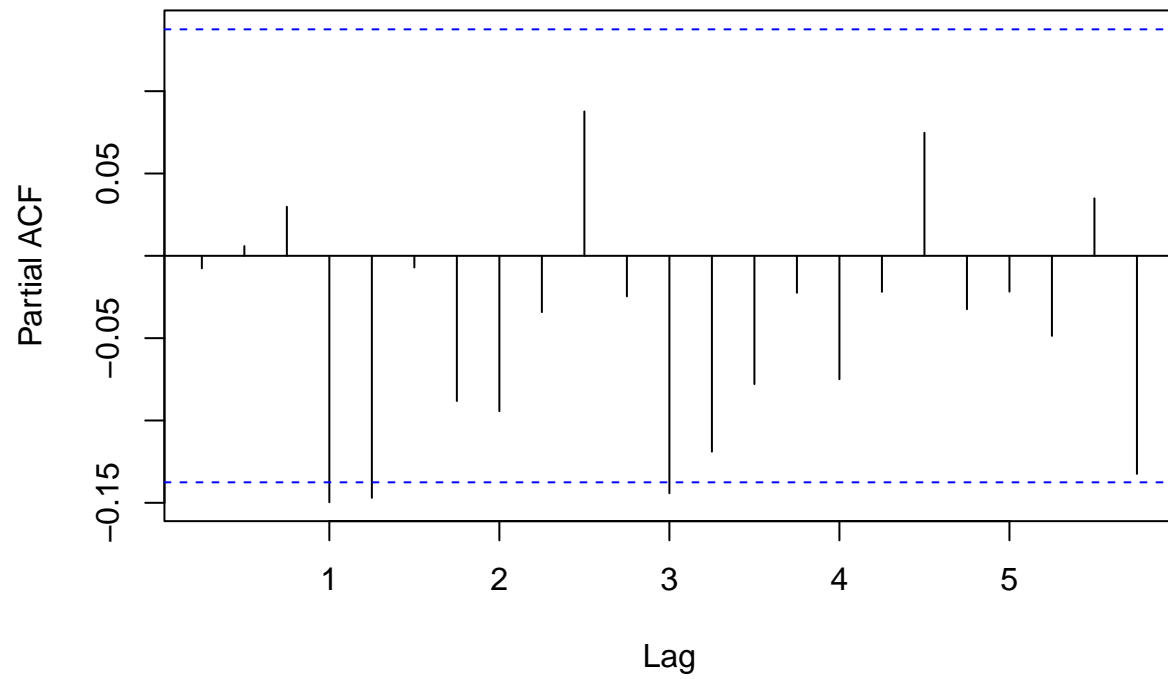
```
acf(model_1$residuals, main="ACF of Residuals for AR(3) Model")
```

### ACF of Residuals for AR(3) Model



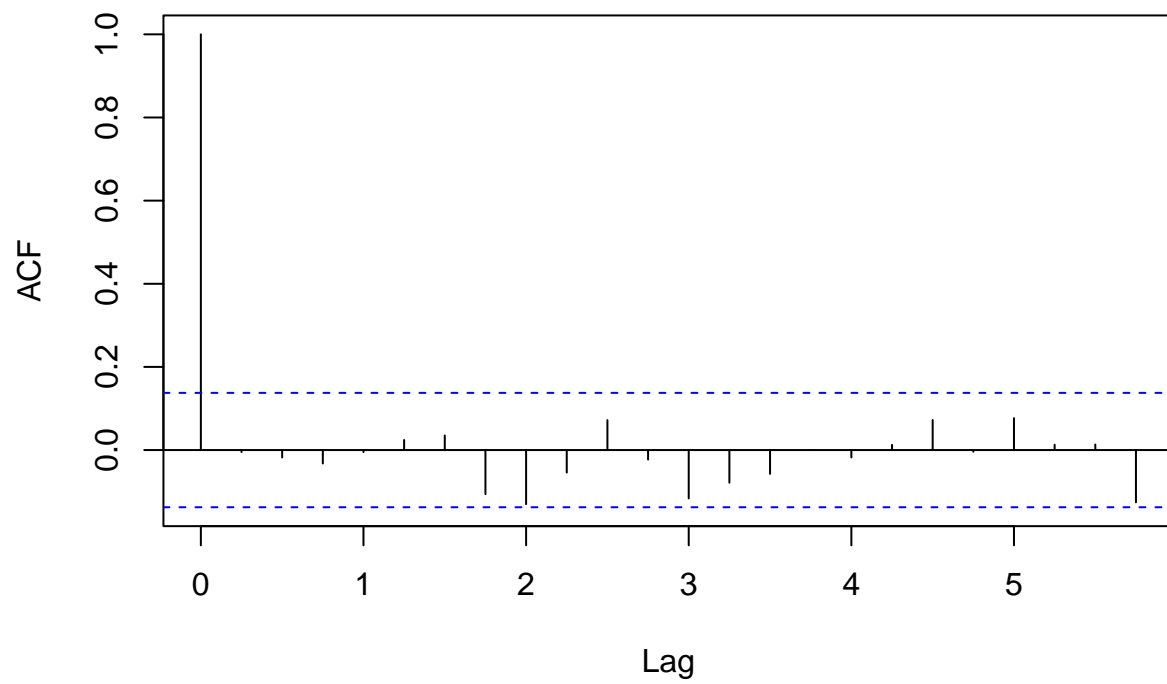
```
pacf(model_1$residuals, main="PACF of Residuals for AR(3) Model")
```

### PACF of Residuals for AR(3) Model



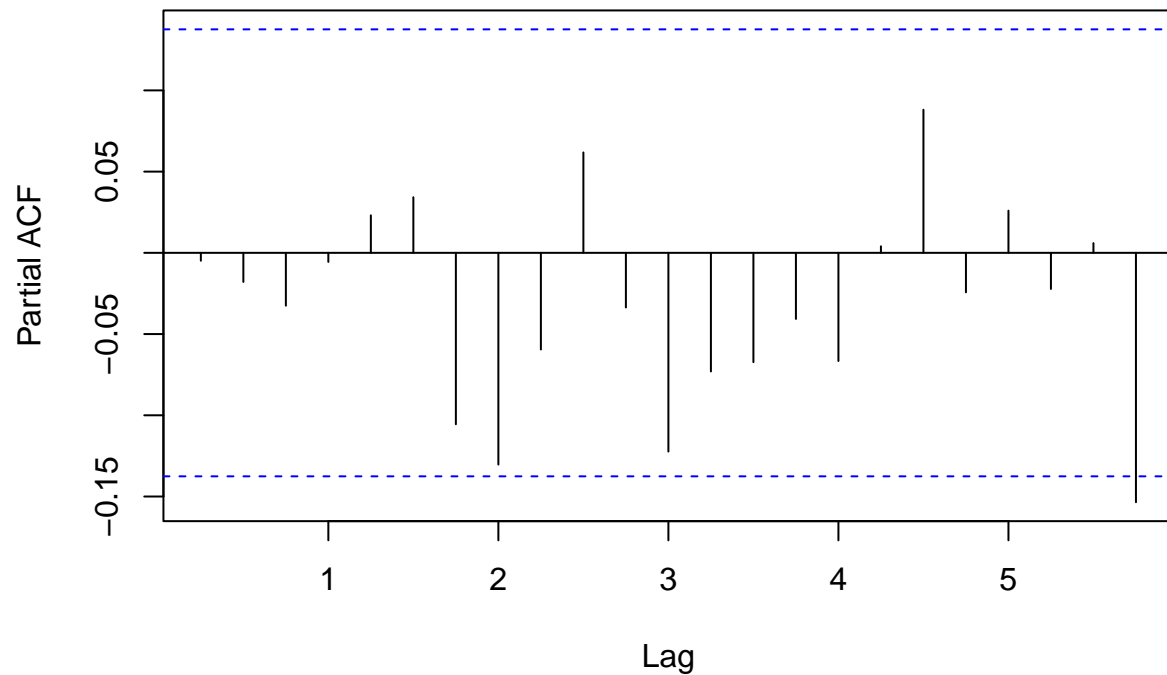
```
acf(model_2$residuals, main="ACF of Residuals for AR(5) Model")
```

### ACF of Residuals for AR(5) Model



```
pacf(model_2$residuals, main="PACF of Residuals for AR(5) Model")
```

## PACF of Residuals for AR(5) Model



*#The ACF of model 1 for the AR(3) model has 4 dynamics whereas the ACF of model 2 for the AR(5) model has 5 dynamics*

```
train_size <- round(2/3 * length(my_dataframe[, 3]))
```

*# Split the data*

```
train_data <- my_dataframe[, 3][1:train_size]
```

```
test_data <- my_dataframe[, 3][(train_size + 1):length(my_dataframe[, 3])]
```

*# Forecasting*

```
forecast_1 <- predict(model_1, n.ahead=length(test_data))
```

```
forecast_2 <- predict(model_2, n.ahead=length(test_data))
```

*# Compute MSE/RMSE*

```
mse_1 <- mean((forecast_1$pred - test_data)^2)
```

```
mse_2 <- mean((forecast_2$pred - test_data)^2)
```

```
cat("MSE for Model 1:", mse_1, "\n")
```

```
## MSE for Model 1: 10.0129
```

```
cat("MSE for Model 2:", mse_2, "\n")
```

```
## MSE for Model 2: 10.15255
```

```
# Calculate AIC and BIC for both models
```

```
aic_1 <- AIC(model_1)
```

```
bic_1 <- BIC(model_1)
```

```
aic_2 <- AIC(model_2)
```

```
bic_2 <- BIC(model_2)
```

```
cat("AIC for Model 1:", aic_1, "\n")
```

```
## AIC for Model 1: 1231.804
```

```
cat("BIC for Model 1:", bic_1, "\n")
```

```
## BIC for Model 1: 1248.37
```

```
cat("AIC for Model 2:", aic_2, "\n")
```

```
## AIC for Model 2: 1225.448
```

```
cat("BIC for Model 2:", bic_2, "\n")
```

```
## BIC for Model 2: 1248.641
```

```
#Model 1 is the better model based on the MSE but model 2 is the better model based on the AIC/BIC.
```

```
# 10-step ahead forecast
```

```
forecast_10_1 <- predict(model_1, n.ahead=10)
```

```
forecast_10_2 <- predict(model_2, n.ahead=10)
```

```
print("10-step ahead forecast for Model 1:")
```

```
## [1] "10-step ahead forecast for Model 1:"
```

```
print(forecast_10_1)
```

```
## $pred
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
```

```
## 2001 0.6174862 0.9574895 1.0893764 1.0936929
```

```
## 2002 1.0931292 1.0891222 1.0883970 1.0880186
```

```
## 2003 1.0880425 1.0880410
```

```
##
```

```
## $se
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
```

```
## 2001 4.954510 4.999975 5.021082 5.021304
```

```
## 2002 5.021308 5.021338 5.021338 5.021338
```

```
## 2003 5.021338 5.021338
```

```
print("10-step ahead forecast for Model 2:")
```

```
## [1] "10-step ahead forecast for Model 2:"
```

```
print(forecast_10_2)
```

```
## $pred
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.5120400 0.4662224 0.7583233 1.5014260
## 2002 1.4097770 1.3193771 1.2630764 1.0854088
## 2003 0.9708597 0.9701341
##
## $se
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2001 4.851330 4.879790 4.904264 4.905739
## 2002 4.960768 5.031117 5.040699 5.044678
## 2003 5.044935 5.050081
```

```
#Part 3
```

```
#This is for variable government growth.
```

```
library(stats)
library(forecast)
```

```
# Fit AR(p) models
```

```
order_1 <- c(3, 0, 0) # AR(3) model
```

```
order_2 <- c(5, 0, 0) # AR(5) model
```

```
# Fit ARIMA models to the 'gov_growth' variable
```

```
model_1 <- Arima(my_dataframe[, 4], order=order_1)
```

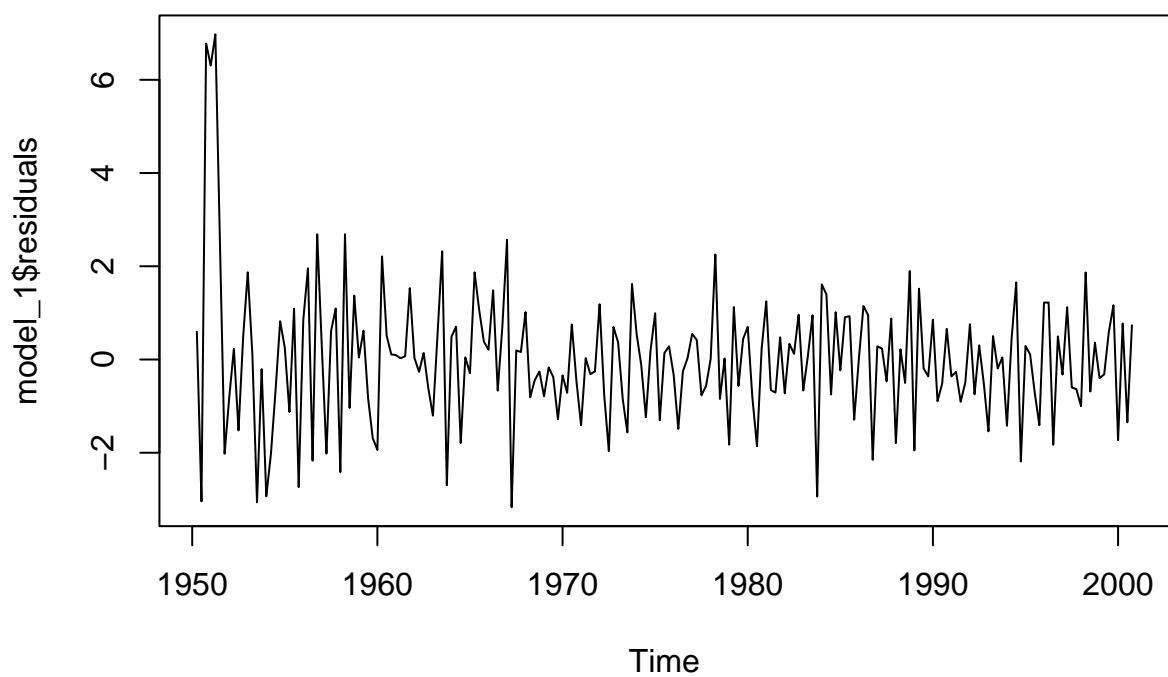
```
model_2 <- Arima(my_dataframe[, 4], order=order_2)
```

```
# Plot and comment on ACF and PACF of the residuals
```

```
plot(model_1$residuals, main="Residuals for AR(3) Model")
```

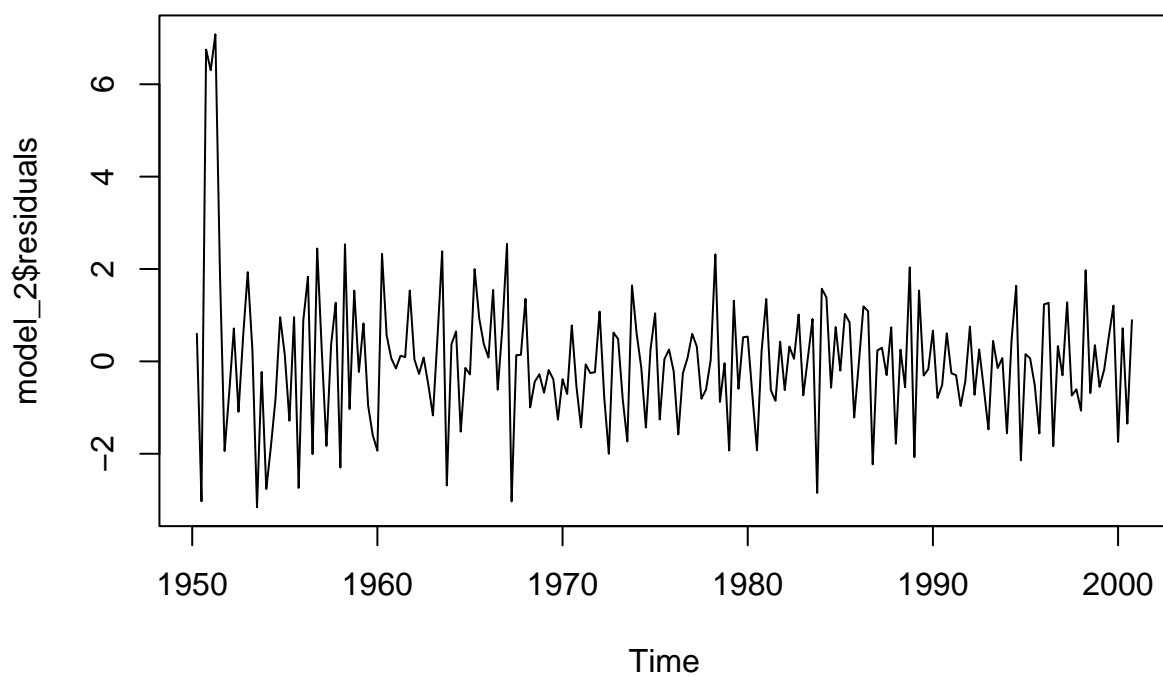


### Residuals for AR(3) Model



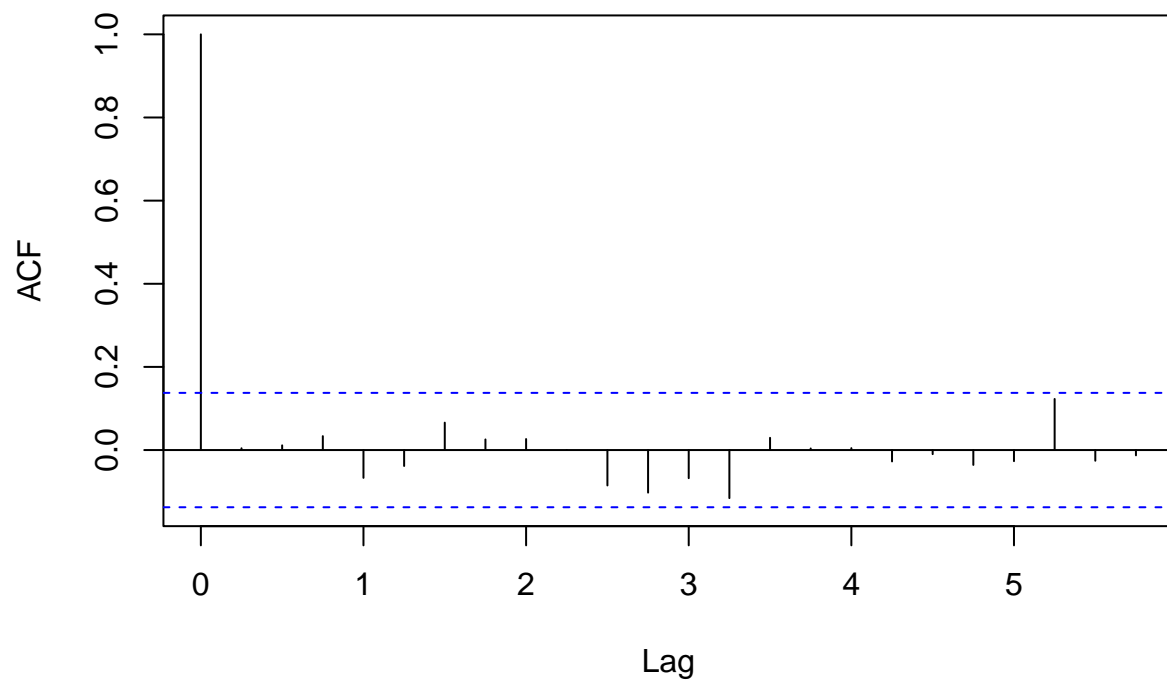
```
plot(model_2$residuals, main="Residuals for AR(5) Model")
```

## Residuals for AR(5) Model



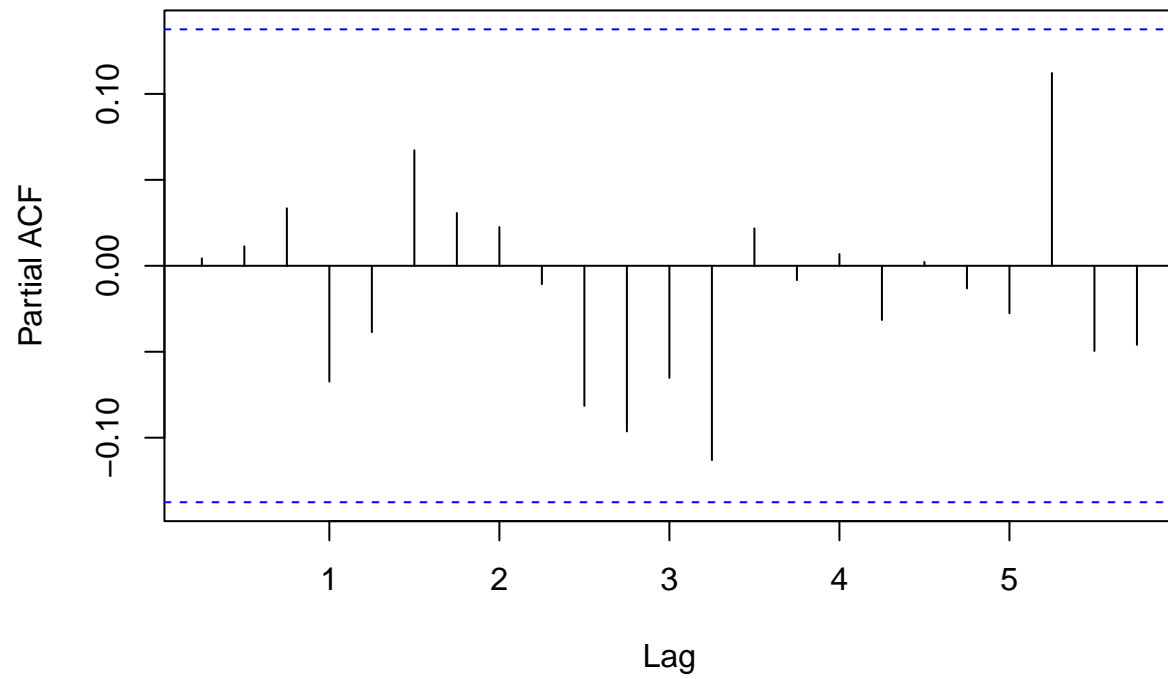
```
acf(model_1$residuals, main="ACF of Residuals for AR(3) Model")
```

### ACF of Residuals for AR(3) Model



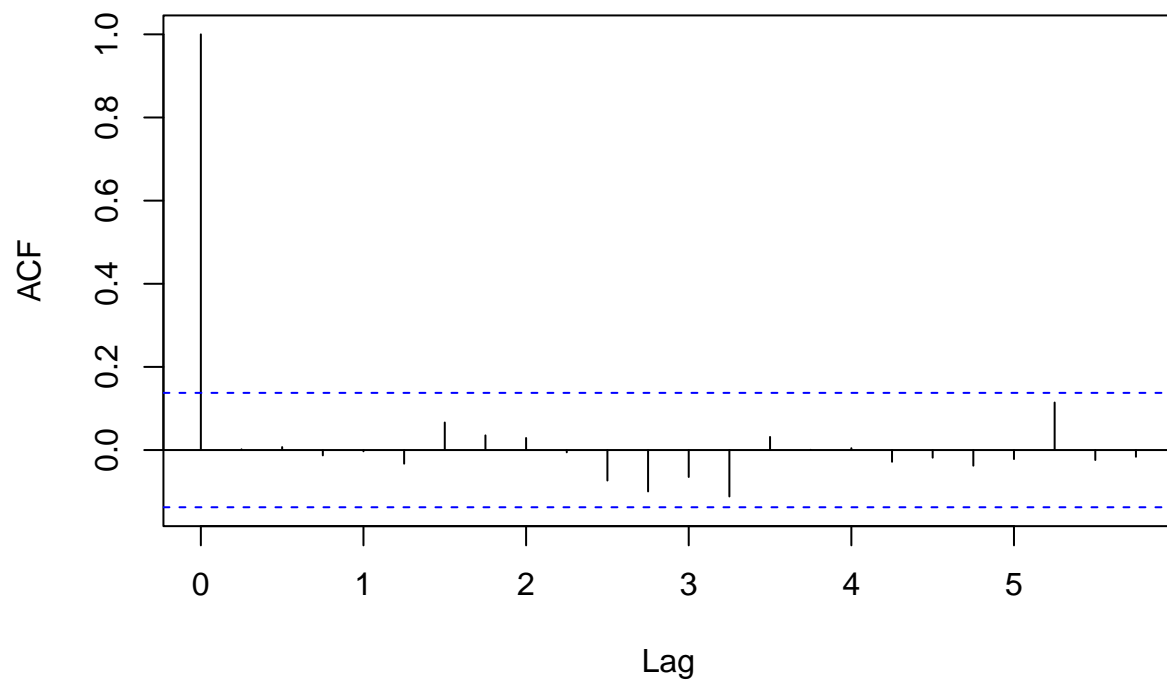
```
pacf(model_1$residuals, main="PACF of Residuals for AR(3) Model")
```

### PACF of Residuals for AR(3) Model



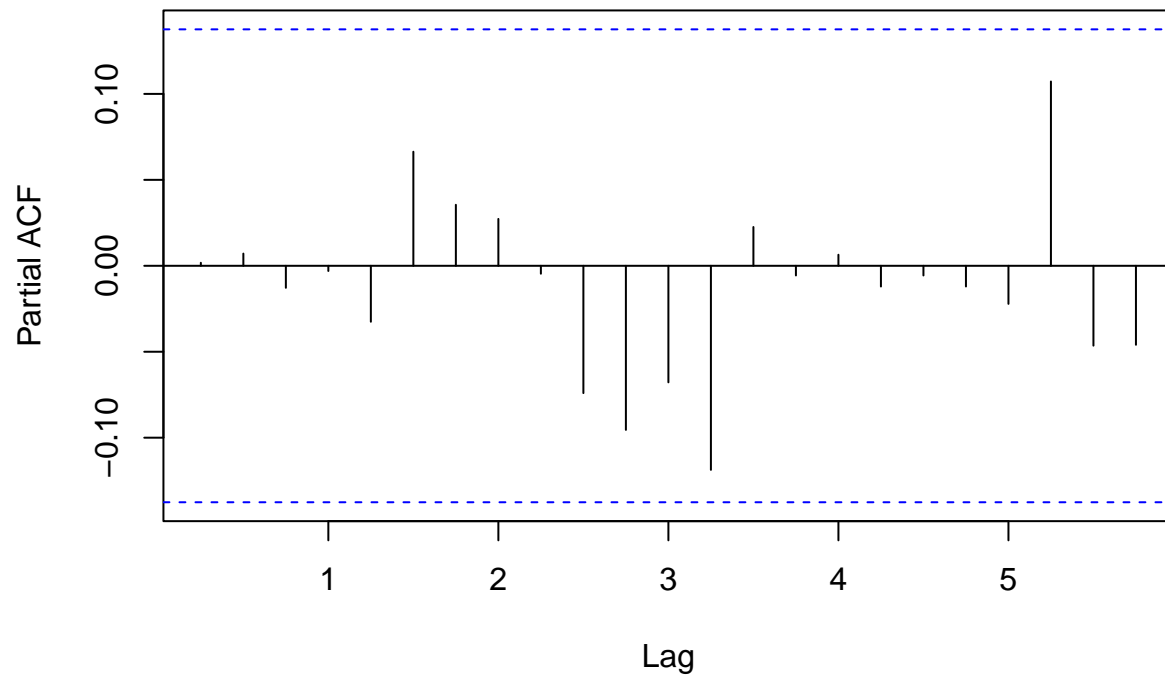
```
acf(model_2$residuals, main="ACF of Residuals for AR(5) Model")
```

### ACF of Residuals for AR(5) Model



```
pacf(model_2$residuals, main="PACF of Residuals for AR(5) Model")
```

## PACF of Residuals for AR(5) Model



*#The ACFs of model 1 and model 2 for the AR(3) and AR(5) models respectively both had 1 dynamic at lag*

```
train_size <- round(2/3 * length(my_dataframe[, 4]))
```

*# Split the data*

```
train_data <- my_dataframe[, 4][1:train_size]
```

```
test_data <- my_dataframe[, 4][(train_size + 1):length(my_dataframe[, 4])]
```

*# Forecasting*

```
forecast_1 <- predict(model_1, n.ahead=length(test_data))
```

```
forecast_2 <- predict(model_2, n.ahead=length(test_data))
```

*# Compute MSE/RMSE*

```
mse_1 <- mean((forecast_1$pred - test_data)^2)
```

```
mse_2 <- mean((forecast_2$pred - test_data)^2)
```

```
cat("MSE for Model 1:", mse_1, "\n")
```

```
## MSE for Model 1: 0.8476103
```

```
cat("MSE for Model 2:", mse_2, "\n")
```

```
## MSE for Model 2: 0.8430314
```

```

# Calculate AIC and BIC for both models
aic_1 <- AIC(model_1)
bic_1 <- BIC(model_1)
aic_2 <- AIC(model_2)
bic_2 <- BIC(model_2)

#Model 2 is a better model based on the MSE but model 1 is a better model based on the AIC/BIC.

cat("AIC for Model 1:", aic_1, "\n")

## AIC for Model 1: 735.2793

cat("BIC for Model 1:", bic_1, "\n")

## BIC for Model 1: 751.8453

cat("AIC for Model 2:", aic_2, "\n")

## AIC for Model 2: 738.0672

cat("BIC for Model 2:", bic_2, "\n")

## BIC for Model 2: 761.2597

# 10-step ahead forecast
forecast_10_1 <- predict(model_1, n.ahead=10)
forecast_10_2 <- predict(model_2, n.ahead=10)

print("10-step ahead forecast for Model 1:")

## [1] "10-step ahead forecast for Model 1:"

print(forecast_10_1)

## $pred
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.6967244 0.6729173 0.7007984 0.7094991
## 2002 0.7154202 0.7203791 0.7238537 0.7263441
## 2003 0.7281509 0.7294545
##
## $se
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2001 1.456799 1.655452 1.740384 1.787485
## 2002 1.811225 1.823391 1.829719 1.833007
## 2003 1.834718 1.835609

print("10-step ahead forecast for Model 2:")

## [1] "10-step ahead forecast for Model 2:"

```

```
print(forecast_10_2)
```

```
## $pred
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.8242339 0.6383582 0.8123628 0.7384908
## 2002 0.7266994 0.7453579 0.7281524 0.7303650
## 2003 0.7309680 0.7281141
##
## $se
##      Qtr1      Qtr2      Qtr3      Qtr4
## 2001 1.459708 1.660998 1.751063 1.816639
## 2002 1.831967 1.840146 1.843999 1.845176
## 2003 1.845821 1.846078
```

```
#Part 3
```

```
#This is for variable rdpi growth.
```

```
library(stats)
library(forecast)
```

```
# Fit AR(p) models
```

```
order_1 <- c(3, 0, 0) # AR(3) model
```

```
order_2 <- c(5, 0, 0) # AR(5) model
```

```
# Fit ARIMA models to the 'rdpi_growth' variable
```

```
model_1 <- Arima(my_dataframe[, 5], order=order_1)
```

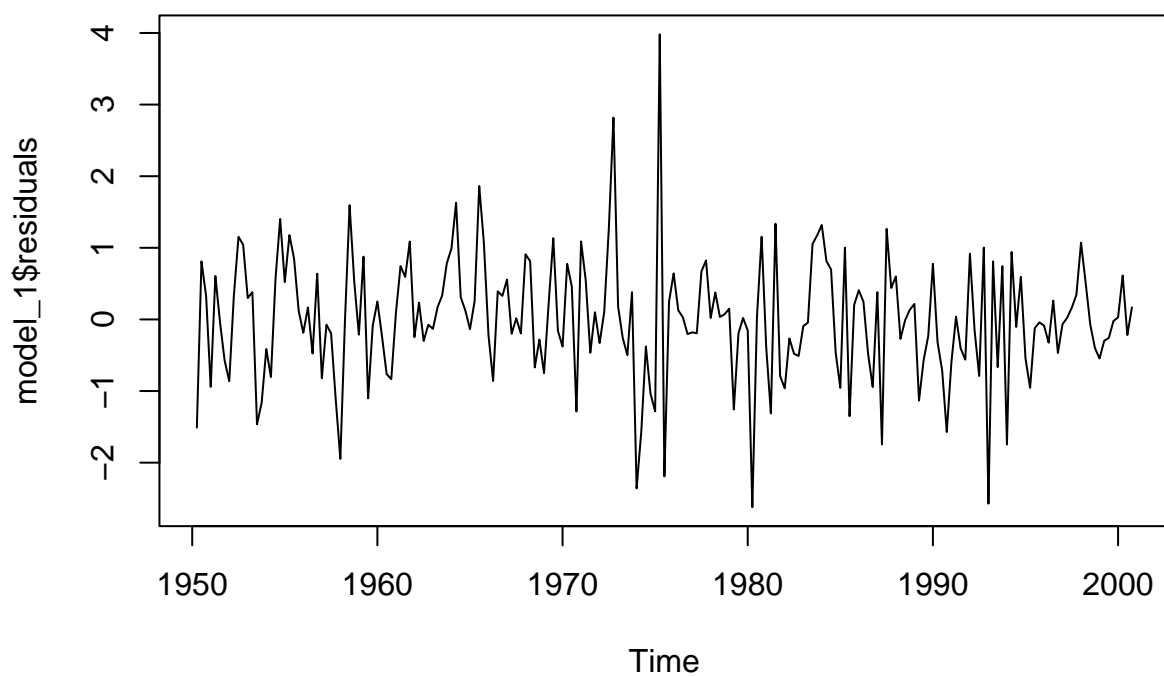
```
model_2 <- Arima(my_dataframe[, 5], order=order_2)
```

```
# Plot and comment on ACF and PACF of the residuals
```

```
plot(model_1$residuals, main="Residuals for AR(3) Model")
```

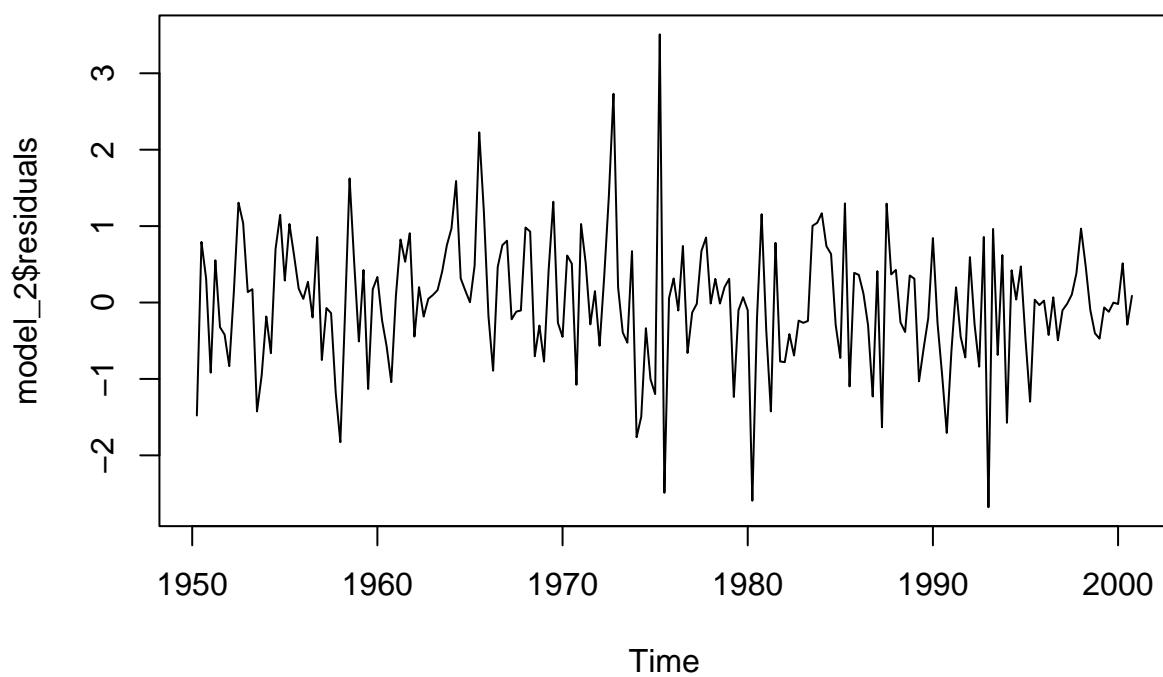


## Residuals for AR(3) Model



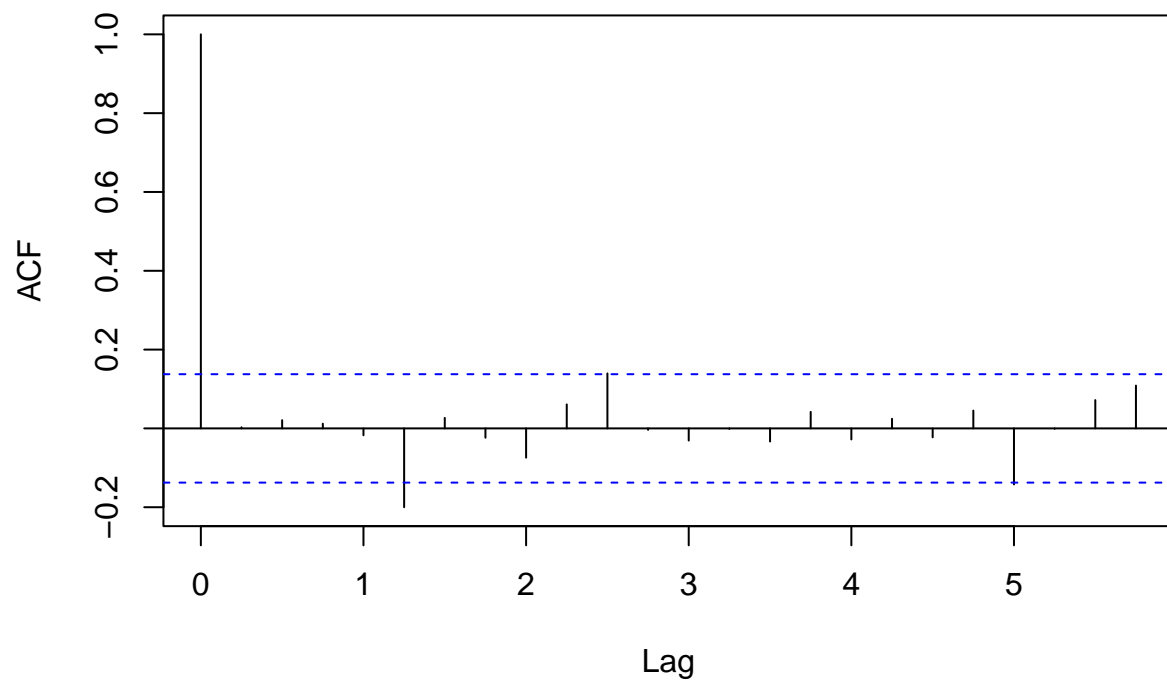
```
plot(model_2$residuals, main="Residuals for AR(5) Model")
```

## Residuals for AR(5) Model



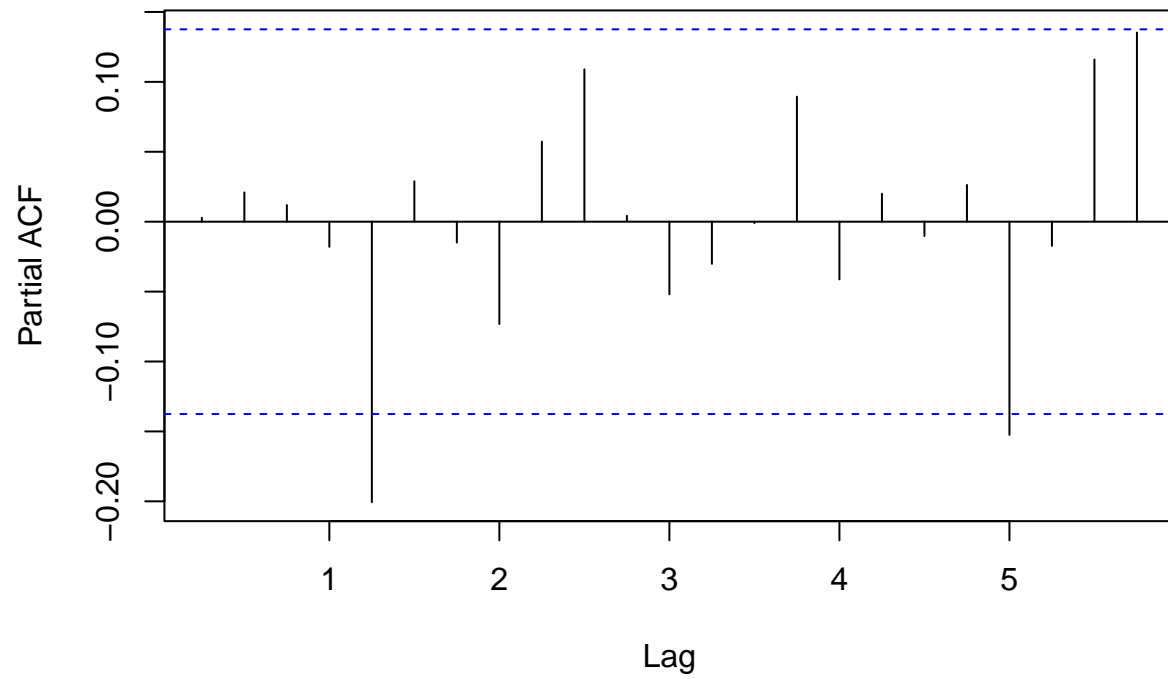
```
acf(model_1$residuals, main="ACF of Residuals for AR(3) Model")
```

### ACF of Residuals for AR(3) Model



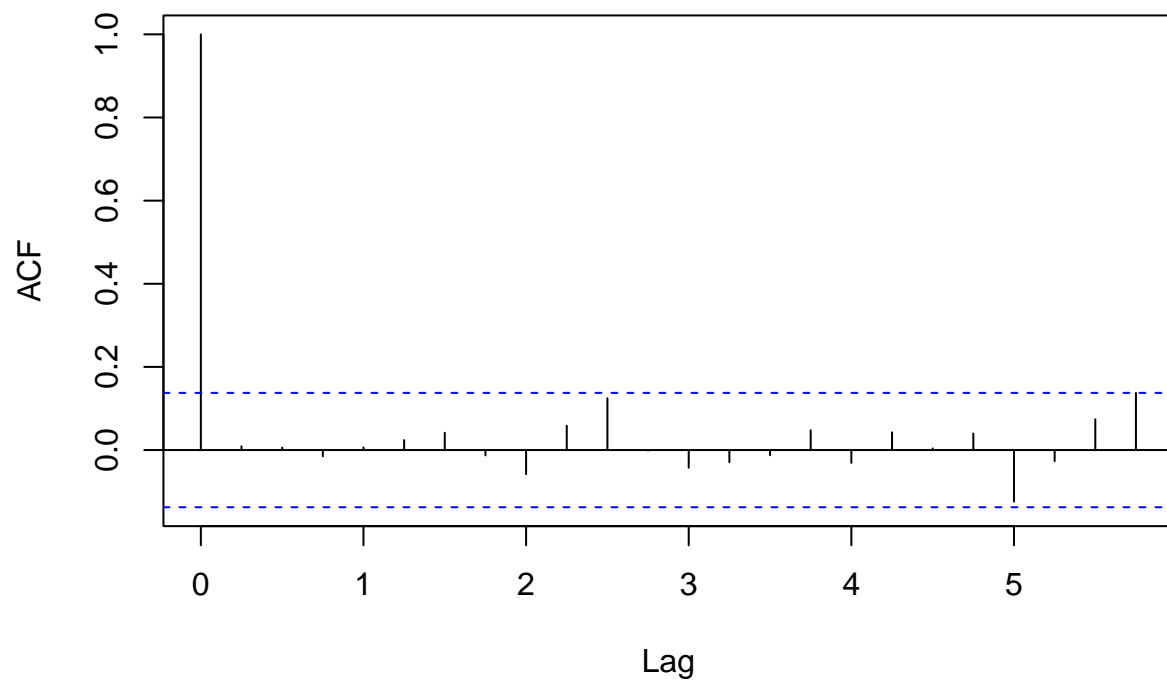
```
pacf(model_1$residuals, main="PACF of Residuals for AR(3) Model")
```

### PACF of Residuals for AR(3) Model



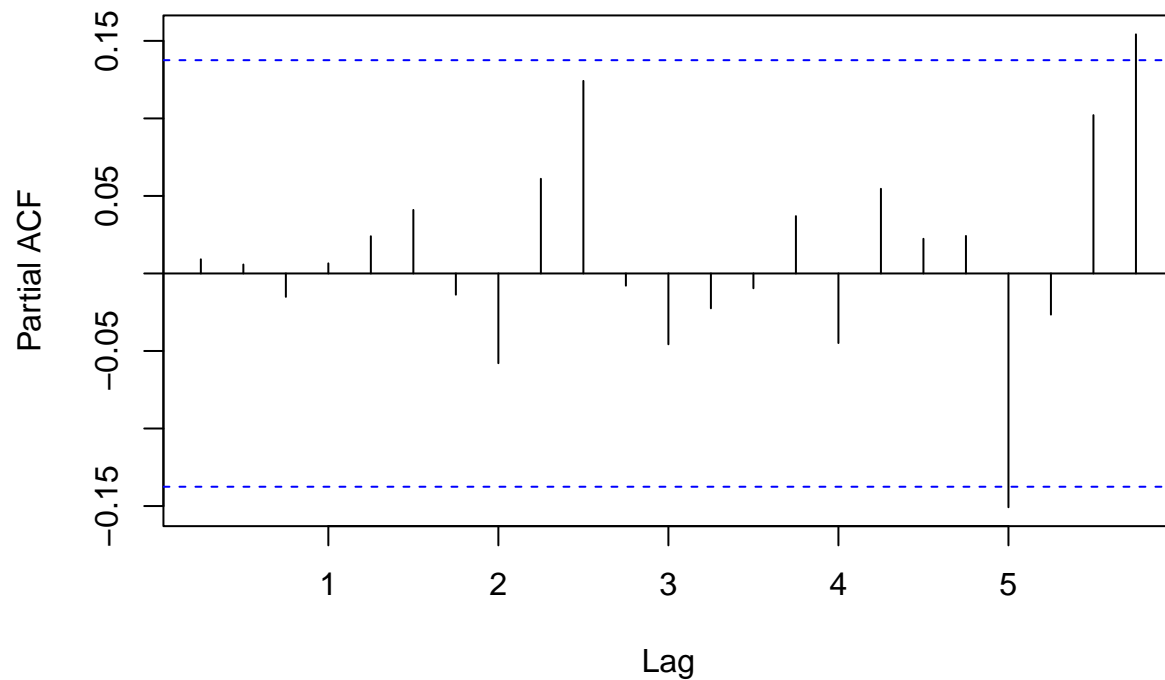
```
acf(model_2$residuals, main="ACF of Residuals for AR(5) Model")
```

### ACF of Residuals for AR(5) Model



```
pacf(model_2$residuals, main="PACF of Residuals for AR(5) Model")
```

## PACF of Residuals for AR(5) Model



*#The ACF of model 1 for the AR(3) model has 4 dynamics whereas the ACF of model 2 for the AR(5) model has 5 dynamics*

```
train_size <- round(2/3 * length(my_dataframe[, 5]))
```

*# Split the data*

```
train_data <- my_dataframe[, 5][1:train_size]
```

```
test_data <- my_dataframe[, 5][(train_size + 1):length(my_dataframe[, 5])]
```

*# Forecasting*

```
forecast_1 <- predict(model_1, n.ahead=length(test_data))
```

```
forecast_2 <- predict(model_2, n.ahead=length(test_data))
```

*# Compute MSE/RMSE*

```
mse_1 <- mean((forecast_1$pred - test_data)^2)
```

```
mse_2 <- mean((forecast_2$pred - test_data)^2)
```

```
cat("MSE for Model 1:", mse_1, "\n")
```

```
## MSE for Model 1: 0.5724188
```

```
cat("MSE for Model 2:", mse_2, "\n")
```

```
## MSE for Model 2: 0.5749573
```

```
# Calculate AIC and BIC for both models
```

```
aic_1 <- AIC(model_1)
```

```
bic_1 <- BIC(model_1)
```

```
aic_2 <- AIC(model_2)
```

```
bic_2 <- BIC(model_2)
```

```
cat("AIC for Model 1:", aic_1, "\n")
```

```
## AIC for Model 1: 527.5424
```

```
cat("BIC for Model 1:", bic_1, "\n")
```

```
## BIC for Model 1: 544.1084
```

```
cat("AIC for Model 2:", aic_2, "\n")
```

```
## AIC for Model 2: 522.7764
```

```
cat("BIC for Model 2:", bic_2, "\n")
```

```
## BIC for Model 2: 545.9688
```

```
#Model 2 is a better model based on the MSE but model 1 is a better model based on the AIC/BIC.
```

```
# 10-step ahead forecast
```

```
forecast_10_1 <- predict(model_1, n.ahead=10)
```

```
forecast_10_2 <- predict(model_2, n.ahead=10)
```

```
print("10-step ahead forecast for Model 1:")
```

```
## [1] "10-step ahead forecast for Model 1:"
```

```
print(forecast_10_1)
```

```
## $pred
```

```
##           Qtr1           Qtr2           Qtr3           Qtr4
```

```
## 2001 0.8997751 0.8445683 0.8675333 0.8531797
```

```
## 2002 0.8486457 0.8495709 0.8480447 0.8475914
```

```
## 2003 0.8475535 0.8473843
```

```
##
```

```
## $se
```

```
##           Qtr1           Qtr2           Qtr3           Qtr4
```

```
## 2001 0.8742833 0.8760438 0.8778991 0.8821655
```

```
## 2002 0.8822774 0.8823509 0.8824018 0.8824056
```

```
## 2003 0.8824074 0.8824081
```

```
print("10-step ahead forecast for Model 2:")
```

```
## [1] "10-step ahead forecast for Model 2:"
```

```
print(forecast_10_2)
```

```
## $pred
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.9289566 0.8448897 0.7602501 0.8879985
## 2002 0.8035261 0.8225607 0.8491034 0.8585730
## 2003 0.8384693 0.8578160
##
## $se
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2001 0.8594826 0.8610744 0.8640556 0.8699436
## 2002 0.8699587 0.8845513 0.8846098 0.8850821
## 2003 0.8860073 0.8860193
```

```
#Part 4
```

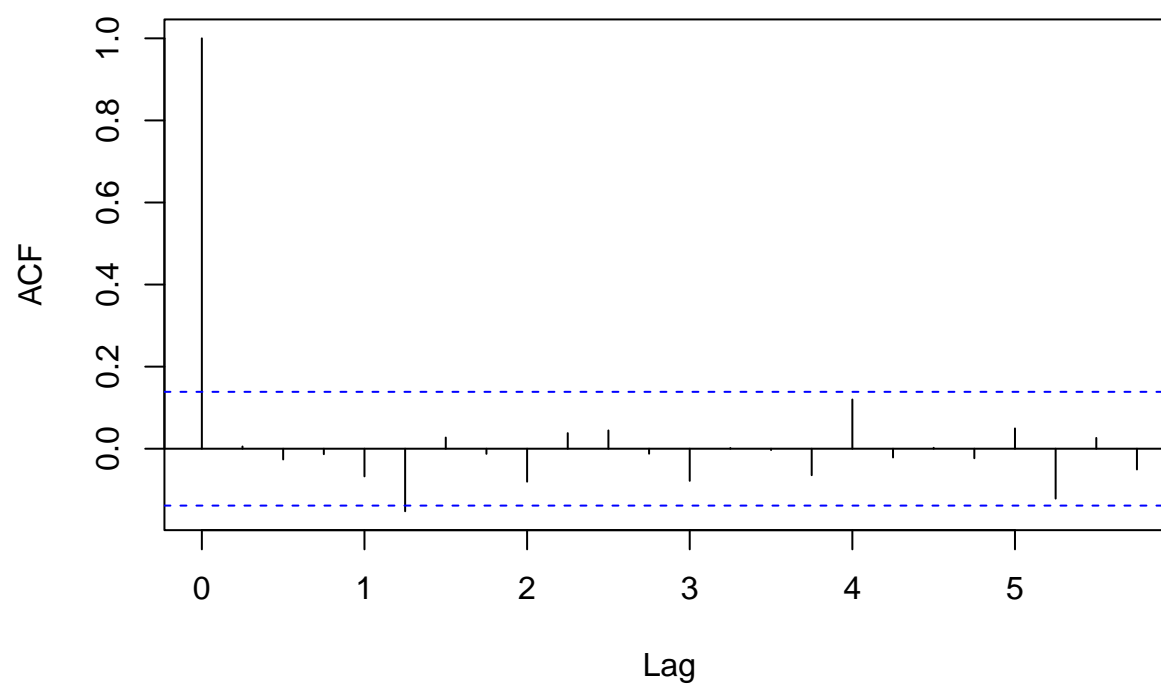
```
train_data <- my_dataframe[1:136,]
test_data <- my_dataframe[137:203,]
train_data_df <- data.frame(train_data)
test_data_df <- data.frame(test_data)
```

```
#Step 1. Get the residuals of order 3 and 5
```

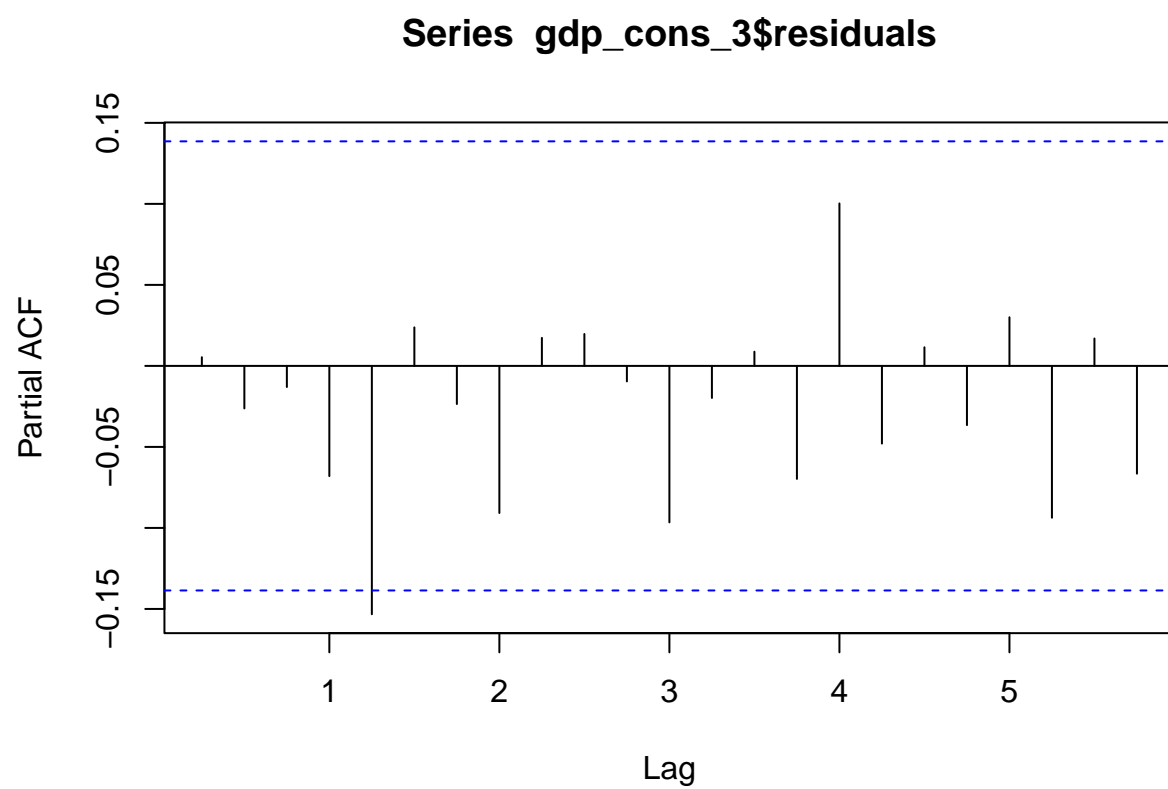
```
gdp_cons_3 <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(cons_growth, 1:3), data = my_dataframe)
acf(gdp_cons_3$residuals)
```



### Series gdp\_cons\_3\$residuals

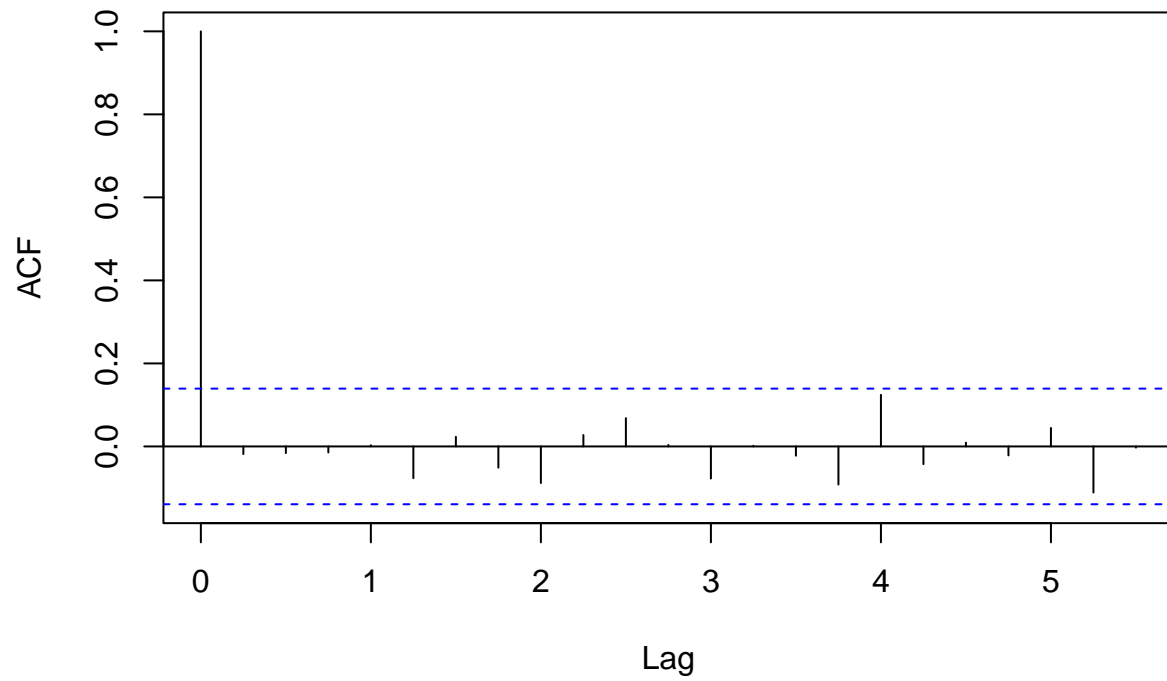


```
pacf(gdp_cons_3$residuals)
```



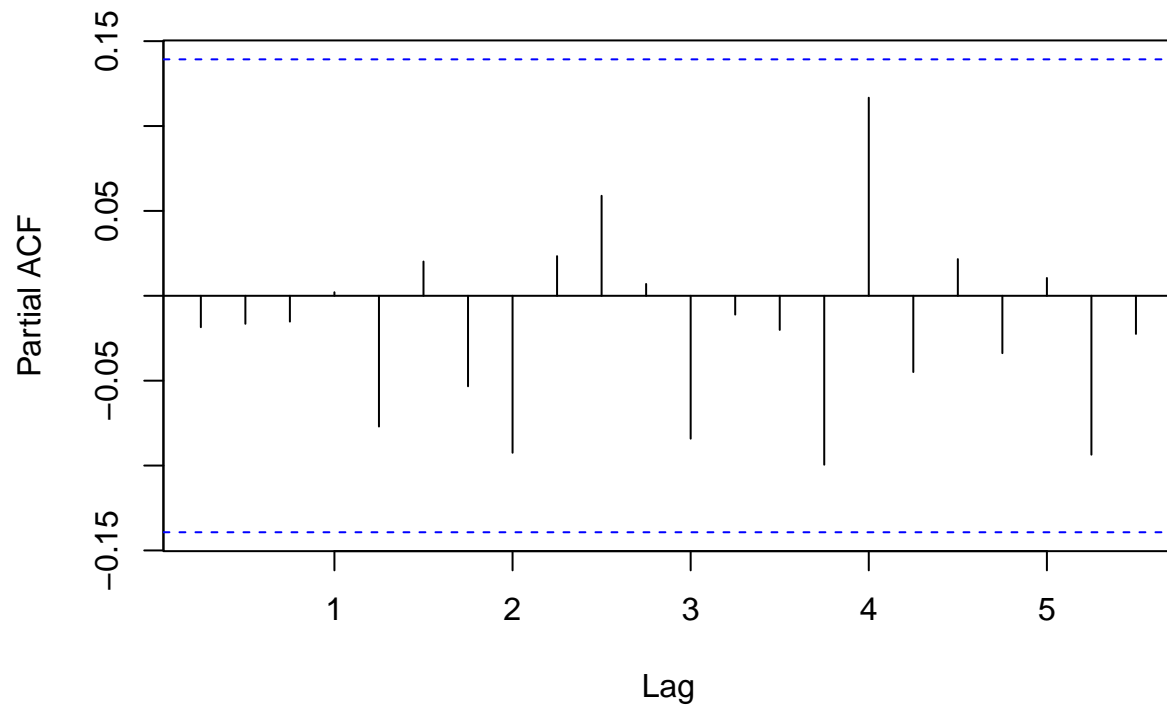
```
gdp_cons_5 <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(cons_growth, 1:5), data=train_data)
acf(gdp_cons_5$residuals)
```

### Series gdp\_cons\_5\$residuals



```
pacf(gdp_cons_5$residuals)
```

## Series gdp\_cons\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
gdp_cons_3_train <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(cons_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_cons_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.040775
```

```
predictions <- predict(gdp_cons_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2558521
```

```
gdp_cons_5_train <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(cons_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_cons_5_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 0.9944017
```

```
predictions <- predict(gdp_cons_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2763458
```

```
#AIC and BIC test  
AIC(gdp_cons_3_train)
```

```
## [1] 530.7834
```

```
AIC(gdp_cons_5_train)
```

```
## [1] 525.6873
```

```
BIC(gdp_cons_3_train)
```

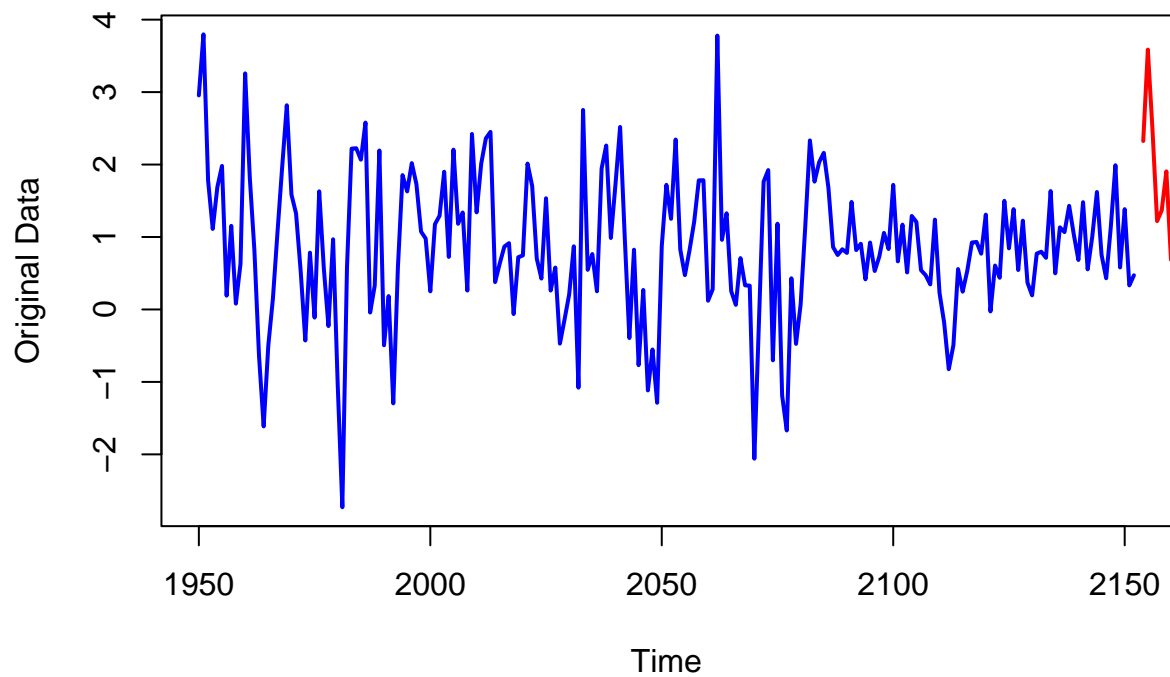
```
## [1] 557.1699
```

```
BIC(gdp_cons_5_train)
```

```
## [1] 565.1465
```

```
#Step 3: forecast 10 steps ahead (for the forecasts, it didn't work properly using the model in the not  
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + cons_growth, data = data.frame(my_dataframe), p = 3  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)  
  
original_data <- ts(my_dataframe[, 1], start = 1950)  
  
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",  
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#x axis is off because it is going by year instead of by quarter. don't know  
#how to fix it
```

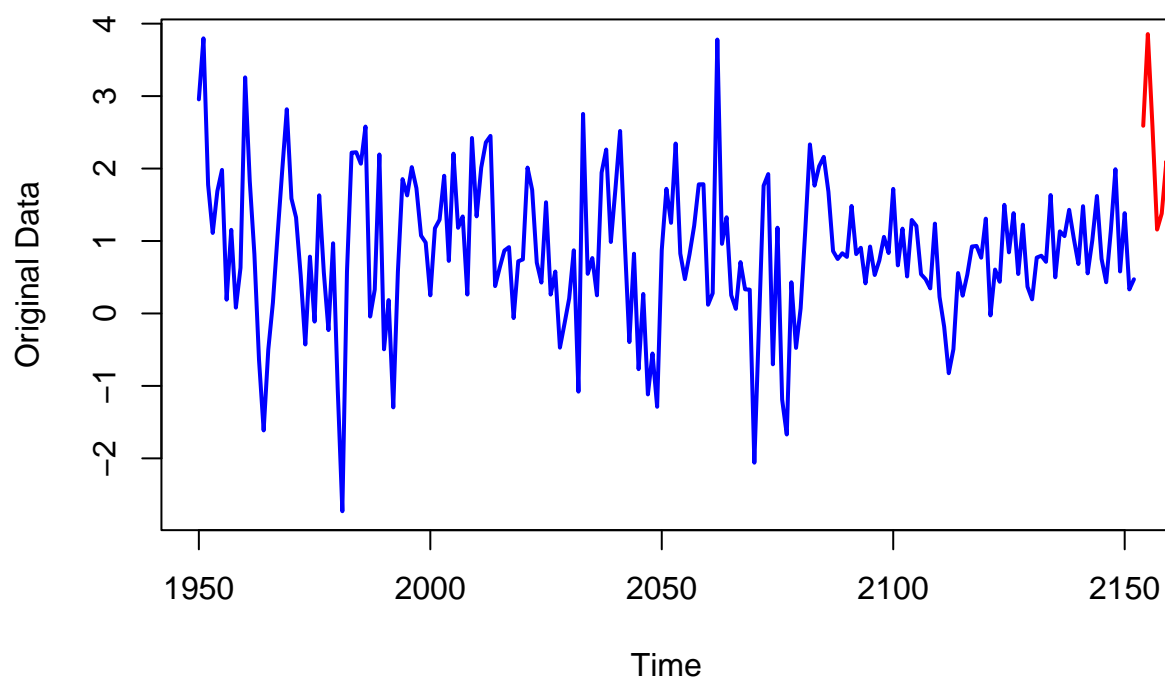
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + cons_growth, data = data.frame(my_dataframe), p = 5  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(gdp_cons_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:3) + L(cons_growth,
##   1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.13422 -0.53206 -0.04266  0.57753  2.96298
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.34590   0.11975   2.889  0.00431 **
## L(gdp_growth, 1:3)1  0.11055   0.08813   1.254  0.21124
## L(gdp_growth, 1:3)2 -0.04201   0.08883  -0.473  0.63682
## L(gdp_growth, 1:3)3 -0.02203   0.08375  -0.263  0.79276
## L(cons_growth, 1:3)1  0.34283   0.09890   3.466  0.00065 ***
## L(cons_growth, 1:3)2  0.23752   0.09867   2.407  0.01701 *
## L(cons_growth, 1:3)3 -0.05911   0.09892  -0.598  0.55080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.8921 on 193 degrees of freedom
## Multiple R-squared: 0.1779, Adjusted R-squared: 0.1524
## F-statistic: 6.961 on 6 and 193 DF, p-value: 1.029e-06

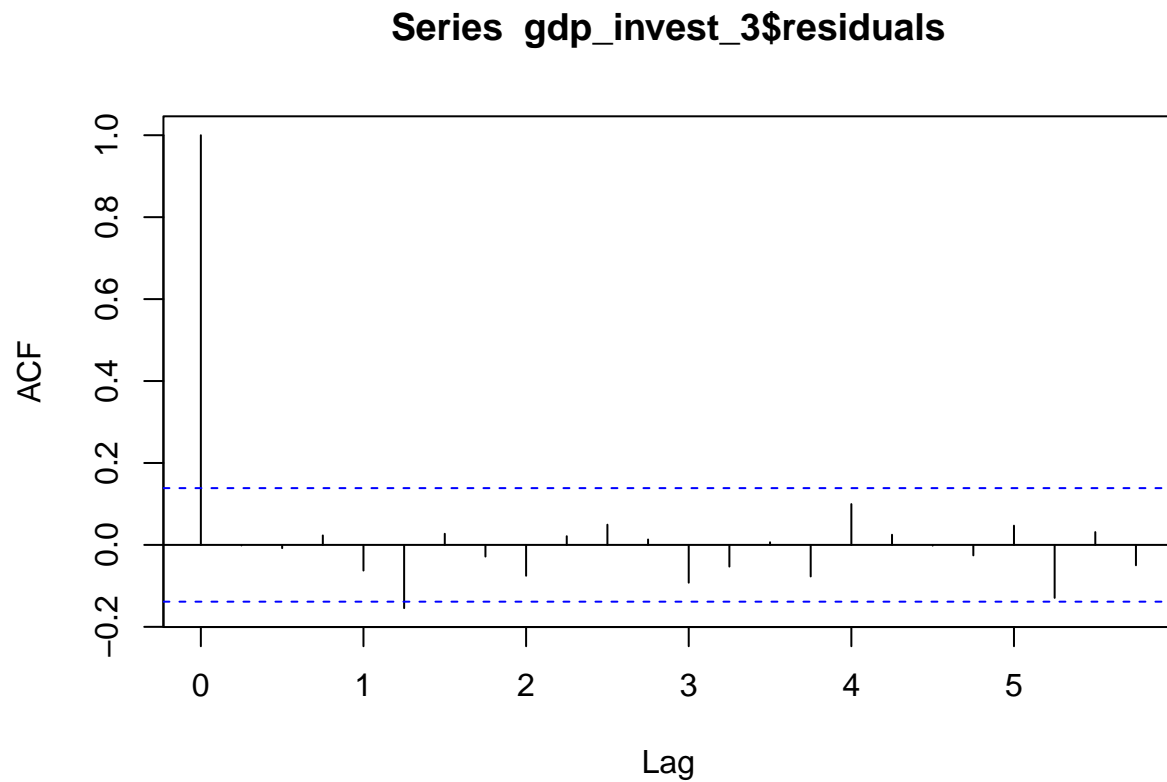
summary(gdp_cons_5)

##
## Time series regression with "zooreg" data:
## Start = 1951 Q3, End = 2000 Q4
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:5) + L(cons_growth,
## 1:5), data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08754 -0.51317 -0.00583  0.58151  3.14029
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.40780    0.13960   2.921  0.00392 **
## L(gdp_growth, 1:5)1  0.11024    0.09154   1.204  0.23001
## L(gdp_growth, 1:5)2 -0.07793    0.09101  -0.856  0.39294
## L(gdp_growth, 1:5)3 -0.04686    0.08997  -0.521  0.60308
## L(gdp_growth, 1:5)4 -0.11891    0.09015  -1.319  0.18874
## L(gdp_growth, 1:5)5 -0.05943    0.08479  -0.701  0.48418
## L(cons_growth, 1:5)1  0.32922    0.11157   2.951  0.00357 **
## L(cons_growth, 1:5)2  0.26127    0.11327   2.307  0.02217 *
## L(cons_growth, 1:5)3 -0.04321    0.10719  -0.403  0.68734
## L(cons_growth, 1:5)4  0.14703    0.10385   1.416  0.15850
## L(cons_growth, 1:5)5 -0.02350    0.10082  -0.233  0.81595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8838 on 187 degrees of freedom
## Multiple R-squared: 0.2147, Adjusted R-squared: 0.1727
## F-statistic: 5.114 on 10 and 187 DF, p-value: 1.349e-06
```

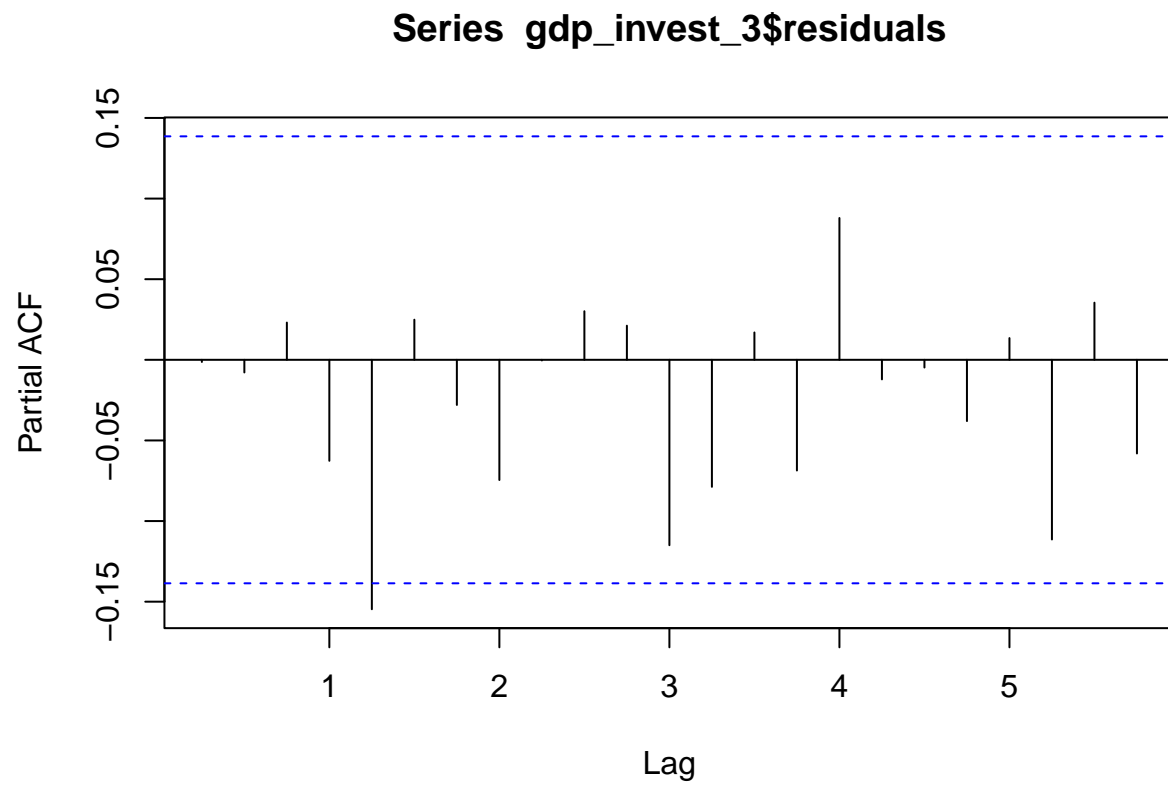
When looking at the ACF model for ARDL order 3, we see that there is little to no autocorrelation. This can be seen by the lack of significant spikes in the ACF (0 doesn't count, lag at 1.25 just barely crosses the line) and 1 line that barely crosses in the PACF. In the PACF, there are 0 significant lags. While this doesn't definitively prove anything, it does imply that there's no external factors that haven't been accounted for, or else there would be a pattern. For the MSEs, they're both around the same, with order 3 having a better test MSE and order 5 having a better train MSE. The final comparison factor is AIC and BIC. We have yet another conflict here because the AIC suggests that order 5 is better and BIC suggests that order 3 is better. Overall, when looking at this model with these two orders, we see that neither model is good. When running summaries of the models, we see that in both cases, the only significant components of this model are the first and second lag of `cons_growth`, implying that we should have never been using an ARDL model in the first place for this situation (if it was good, `gdp` would have been significant in at least one lag).



```
#GDP-Invest  
gdp_invest_3 <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(inv_growth, 1:3), data = my_dataframe)  
acf(gdp_invest_3$residuals)
```

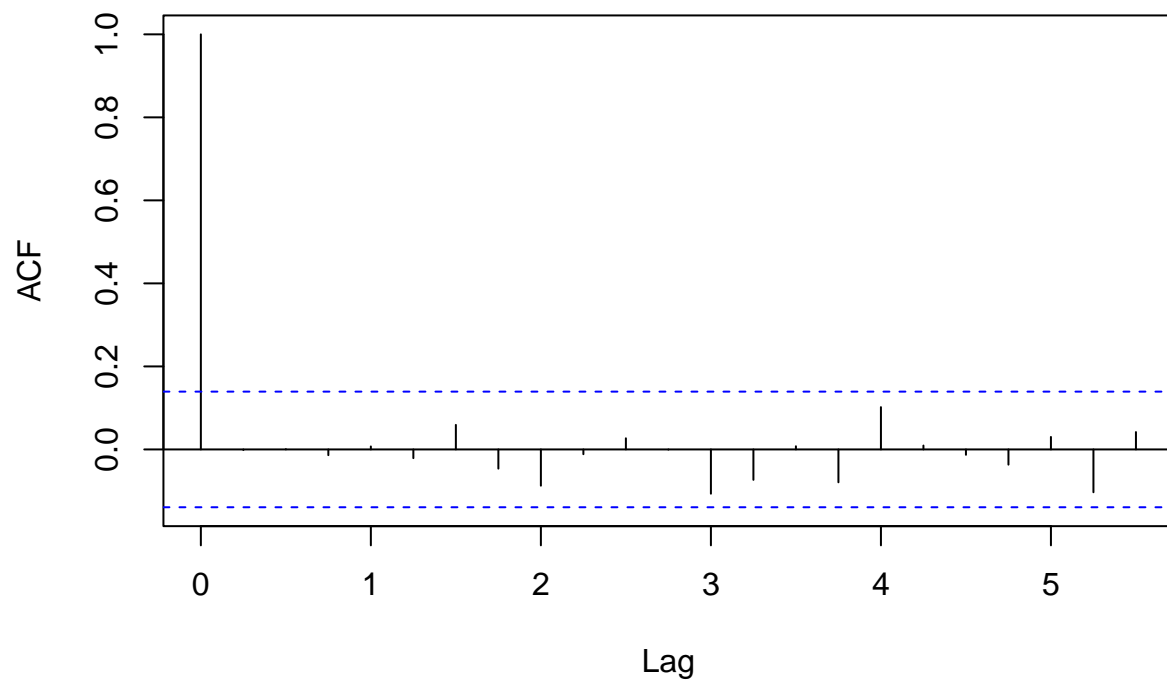


```
pacf(gdp_invest_3$residuals)
```



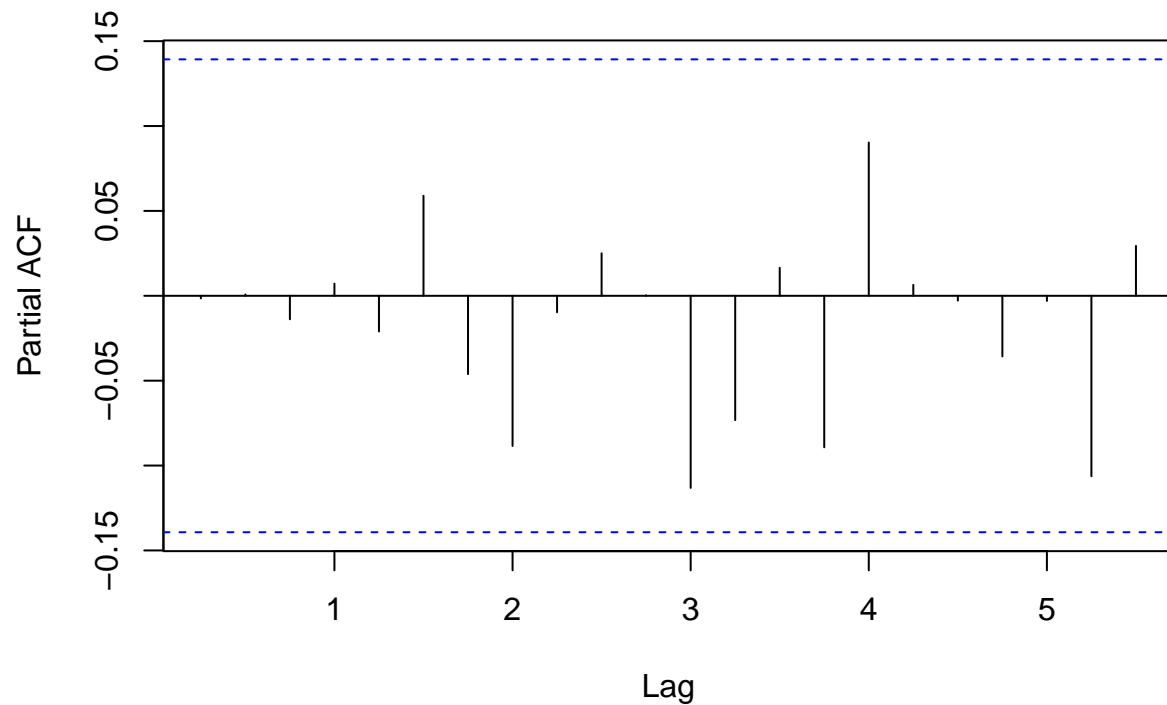
```
gdp_invest_5 <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(inv_growth, 1:5), data=my_dataframe)
acf(gdp_invest_5$residuals)
```

### Series gdp\_invest\_5\$residuals



```
pacf(gdp_invest_5$residuals)
```

## Series gdp\_invest\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
gdp_invest_3_train <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(inv_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_invest_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.108356
```

```
predictions <- predict(gdp_invest_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2832338
```

```
gdp_invest_5_train <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(inv_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_invest_5_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.069694
```

```
predictions <- predict(gdp_invest_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.3197597
```

```
AIC(gdp_invest_3_train)
```

```
## [1] 543.548
```

```
AIC(gdp_invest_5_train)
```

```
## [1] 541.0019
```

```
BIC(gdp_invest_3_train)
```

```
## [1] 569.9345
```

```
BIC(gdp_invest_5_train)
```

```
## [1] 580.4611
```

```
#Step 3: forecast 10 steps ahead
```

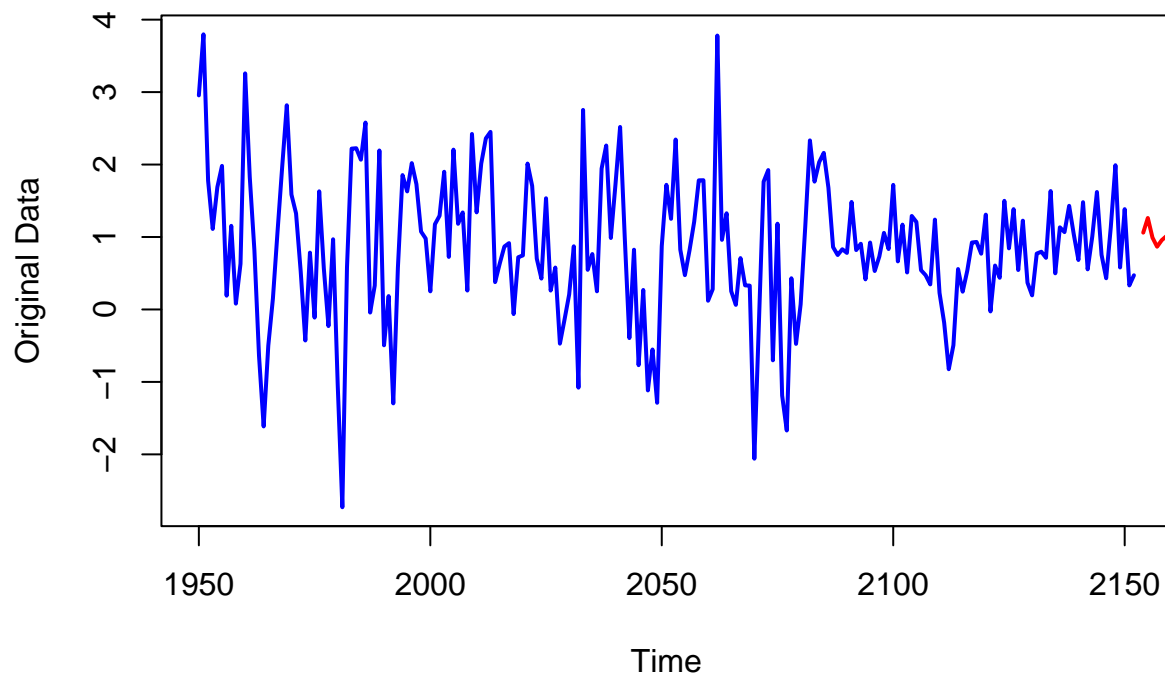
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + inv_growth, data = data.frame(my_dataframe), p = 3,  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

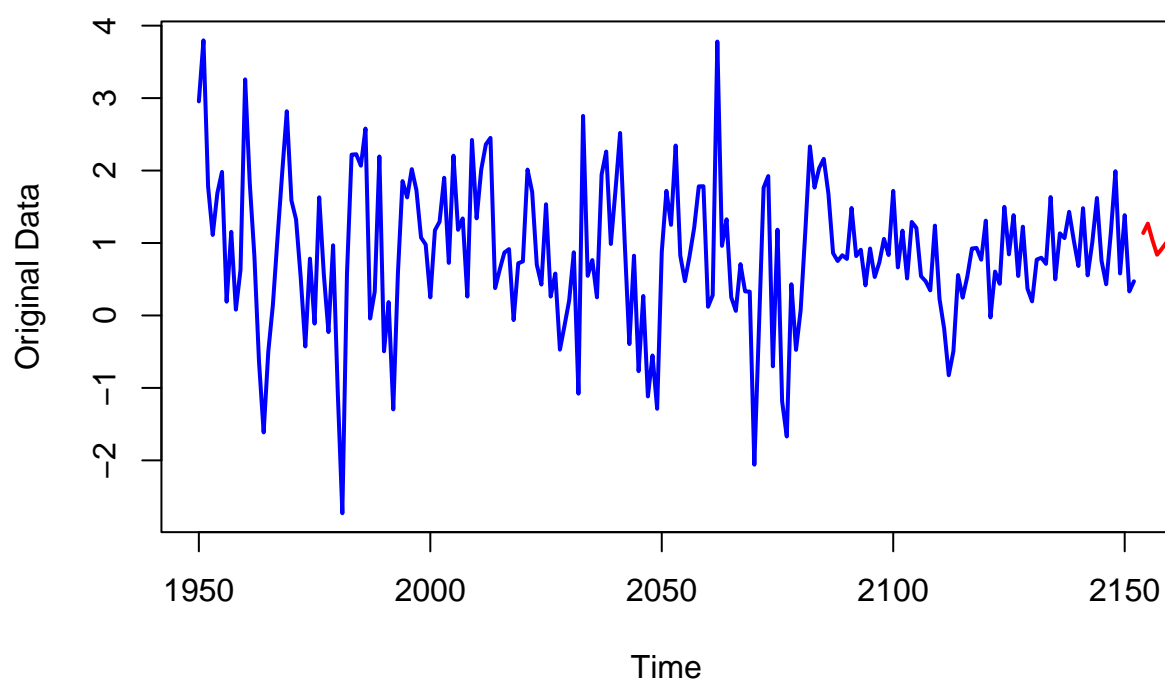
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + inv_growth, data = data.frame(my_dataframe), p = 5,  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(gdp_invest_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:3) + L(inv_growth,
##   1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1504 -0.4969  0.0073  0.5630  3.2704
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.54037    0.12943   4.175 4.51e-05 ***
## L(gdp_growth, 1:3)1  0.38097    0.11144   3.419 0.000768 ***
## L(gdp_growth, 1:3)2  0.14234    0.11131   1.279 0.202538
## L(gdp_growth, 1:3)3 -0.15879    0.10922  -1.454 0.147621
## L(inv_growth, 1:3)1 -0.01642    0.02128  -0.772 0.441266
## L(inv_growth, 1:3)2 -0.01720    0.02105  -0.817 0.414751
## L(inv_growth, 1:3)3  0.02176    0.02117   1.028 0.305287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.921 on 193 degrees of freedom
## Multiple R-squared:  0.1237, Adjusted R-squared:  0.09649
## F-statistic: 4.542 on 6 and 193 DF,  p-value: 0.0002476

summary(gdp_invest_5)

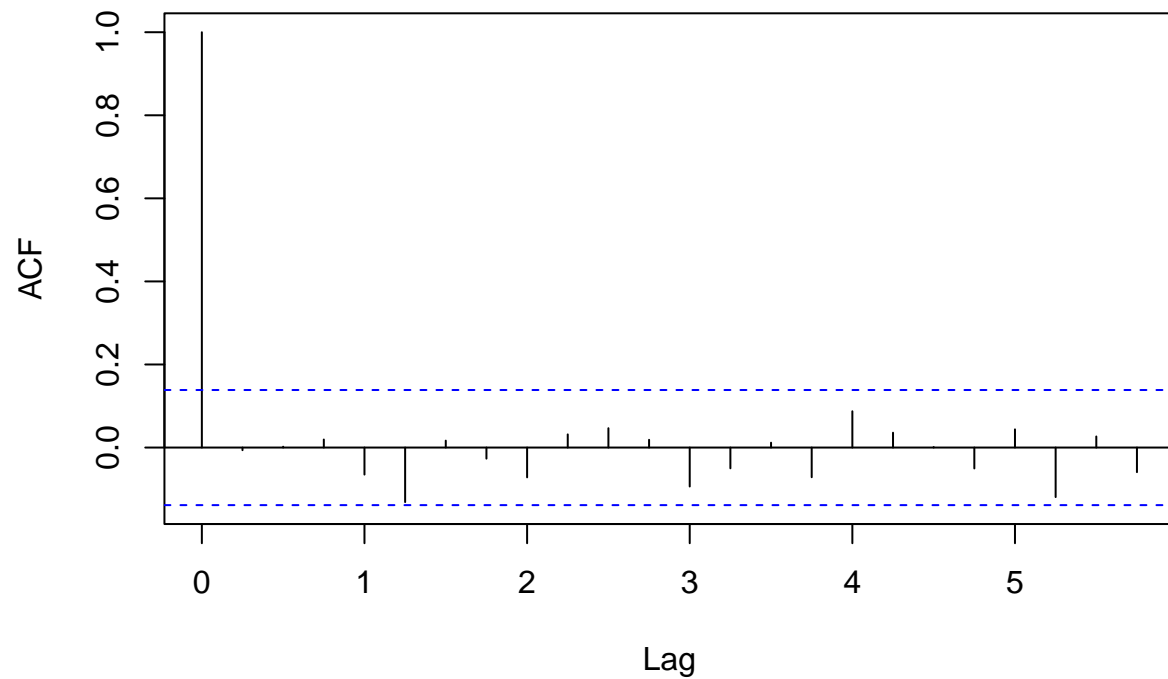
##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:5) + L(inv_growth,
## 1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.0460 -0.4800 -0.0137  0.5360  3.3538
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.607696   0.148297    4.098 6.21e-05 ***
## L(gdp_growth, 1:5)1  0.350030   0.115443    3.032 0.00277 **
## L(gdp_growth, 1:5)2  0.177071   0.116358    1.522 0.12975
## L(gdp_growth, 1:5)3 -0.111984   0.115735   -0.968 0.33450
## L(gdp_growth, 1:5)4  0.036311   0.114475    0.317 0.75145
## L(gdp_growth, 1:5)5 -0.133780   0.112295   -1.191 0.23504
## L(inv_growth, 1:5)1 -0.011395   0.023038   -0.495 0.62144
## L(inv_growth, 1:5)2 -0.026928   0.022120   -1.217 0.22499
## L(inv_growth, 1:5)3  0.015975   0.021615    0.739 0.46079
## L(inv_growth, 1:5)4 -0.022890   0.021479   -1.066 0.28793
## L(inv_growth, 1:5)5  0.003363   0.021550    0.156 0.87616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9187 on 187 degrees of freedom
## Multiple R-squared:  0.1516, Adjusted R-squared:  0.1062
## F-statistic: 3.341 on 10 and 187 DF,  p-value: 0.0005064
```

In the ACF of the residuals for order 3, we see that there no extremely significant spikes. However, in the PACF, we see that there are significant spikes at lag = 2 and 3. This shows that there potential flaws in the model because if the residuals can be modeled, then there is a chance that there is another component we're missing in this model. No significant spikes in the ACF for order 5. However, we see significant spikes in lag 3 and 3.75 for the PACF. Once again, it shows a flaw in our model. When looking at the training and testing mse, once again, we get an issue where the training MSE is better in order 5 model but the testing MSE is better in the order 3 model. Since the training set contains more data points, that's probably a better one to go off of, but we still have the AIC and BIC to make final conclusions. The AIC and BIC has yet another conflict of interest, where order 5 has the better AIC, and order 3 has the better BIC. The final thing is to look at the summaries of the models and see which one is better. In the order 3 model, we see that nothing in the inv\_growth is useful and the only successful component is the gdp\_growth of Lag 1 and the adjusted R squared is 9.649%. In the 5 order model, we see only gdp\_growth of lag 1 is useful. Based on an order 3 and order 5 model, we see that an ARDL model is not useful since in neither model, inv\_growth is not useful. We are better off just running an AR(1) model. However, if I had to choose between the two, I would give the slight edge to the order 3 model despite having a lower adjusted R squared due to simplicity.



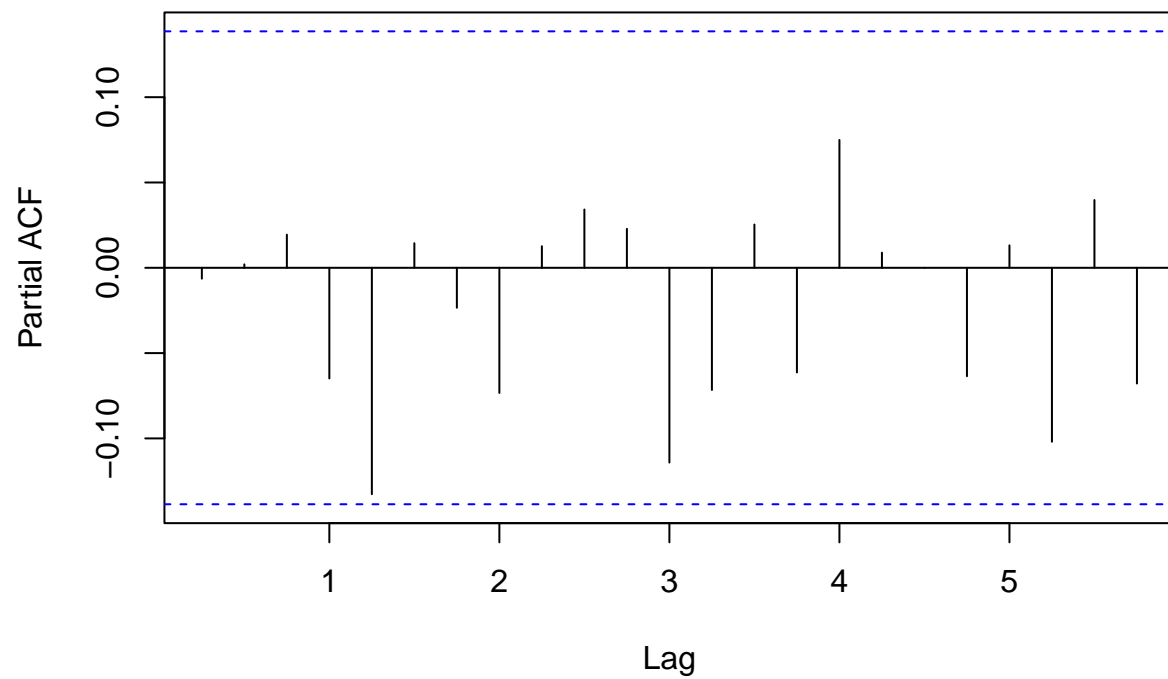
```
#GDP-Gov  
gdp_gov_3 <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(gov_growth, 1:3), data = my_dataframe)  
acf(gdp_gov_3$residuals)
```

### Series gdp\_gov\_3\$residuals



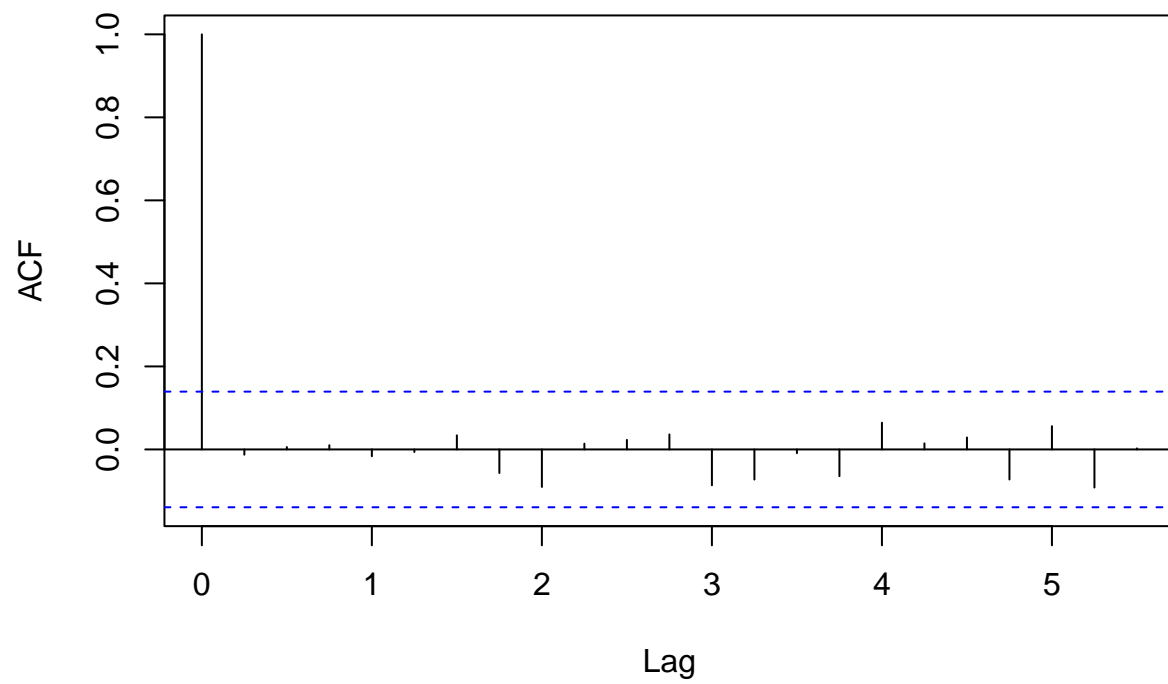
```
pacf(gdp_gov_3$residuals)
```

### Series gdp\_gov\_3\$residuals



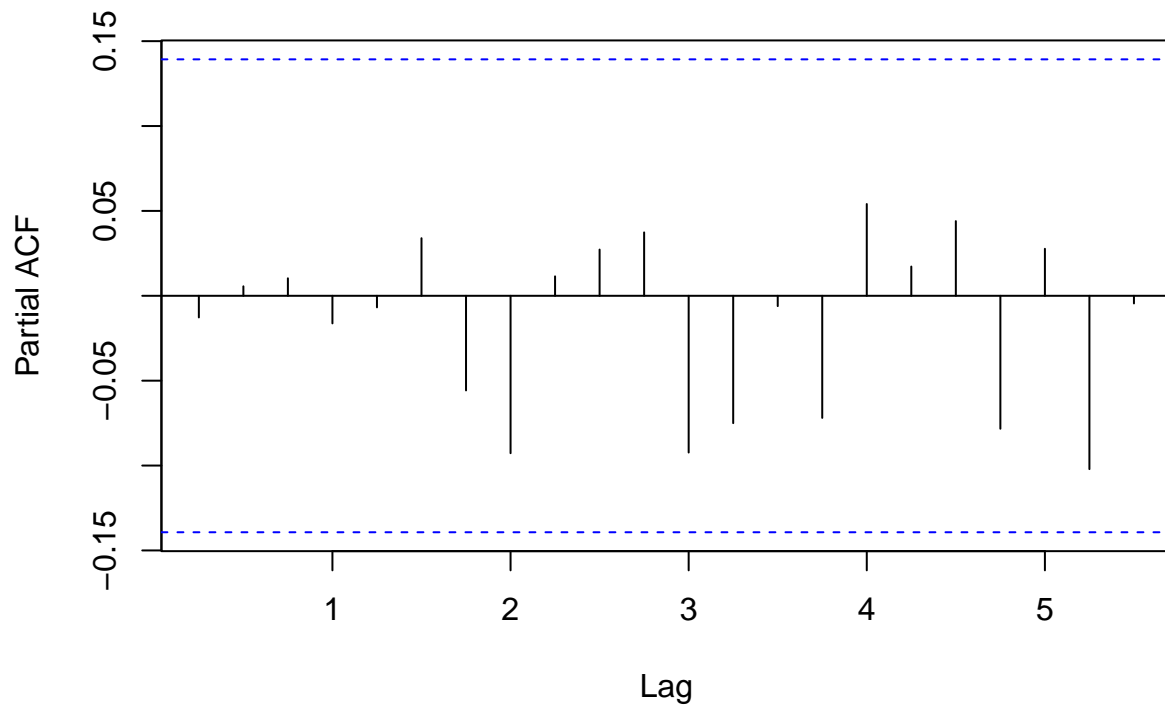
```
gdp_gov_5 <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(gov_growth, 1:5), data=my_dataframe)
acf(gdp_gov_5$residuals)
```

### Series gdp\_gov\_5\$residuals



```
pacf(gdp_gov_5$residuals)
```

## Series gdp\_gov\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
gdp_gov_3_train <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(gov_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_gov_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.120493
```

```
predictions <- predict(gdp_gov_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2798117
```

```
gdp_gov_5_train <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(gov_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_gov_3_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.44783
```

```
predictions <- predict(gdp_gov_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.3040825
```

```
AIC(gdp_gov_3_train)
```

```
## [1] 545.4115
```

```
AIC(gdp_gov_5_train)
```

```
## [1] 540.7134
```

```
BIC(gdp_gov_3_train)
```

```
## [1] 571.7981
```

```
BIC(gdp_gov_5_train)
```

```
## [1] 580.1726
```

```
#Step 3: forecast 10 steps ahead
```

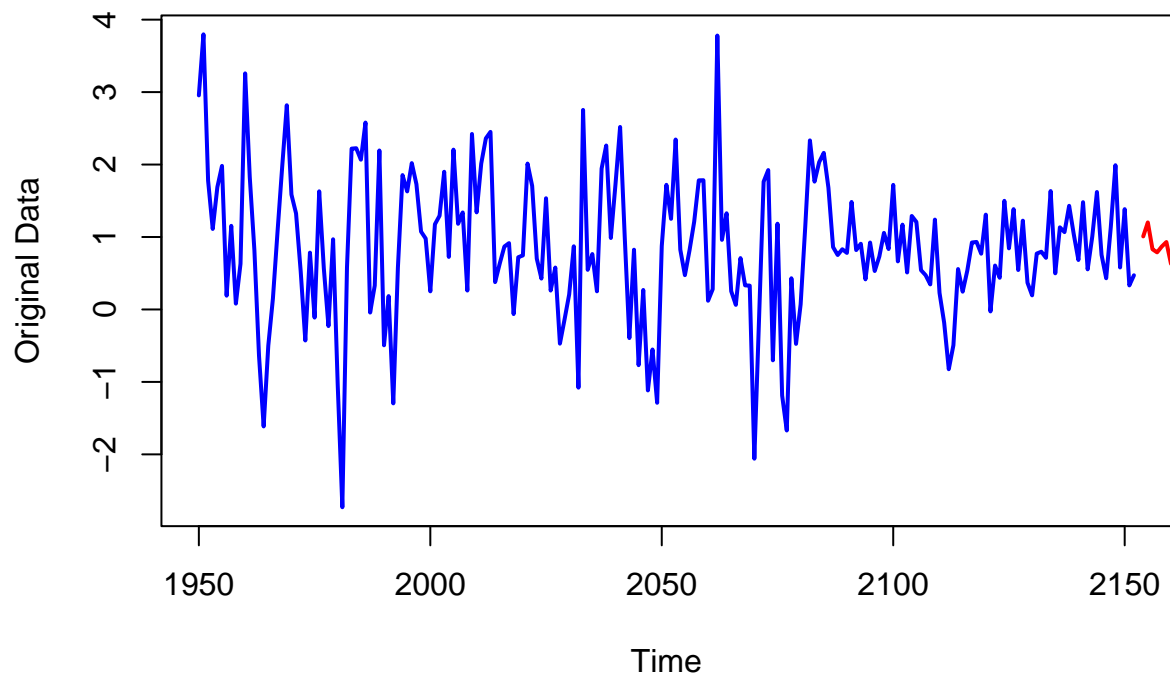
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + gov_growth, data = data.frame(my_dataframe), p = 3,  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

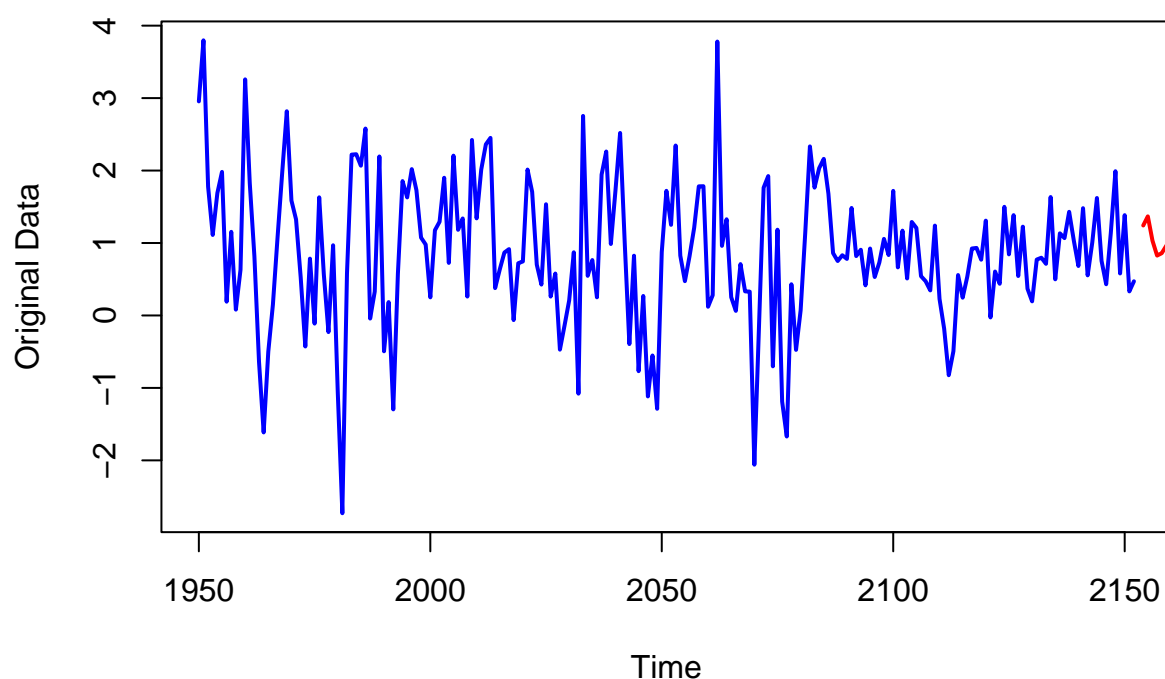
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + gov_growth, data = data.frame(my_dataframe), p = 5,  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(gdp_gov_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:3) + L(gov_growth,
## 1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.0314 -0.4813  0.0205  0.5388  3.2348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.56727    0.10380   5.465 1.42e-07 ***
## L(gdp_growth, 1:3)1  0.32193    0.07399   4.351 2.19e-05 ***
## L(gdp_growth, 1:3)2  0.06932    0.07577   0.915  0.361
## L(gdp_growth, 1:3)3 -0.06133    0.07173  -0.855  0.394
## L(gov_growth, 1:3)1 -0.02194    0.04740  -0.463  0.644
## L(gov_growth, 1:3)2  0.02512    0.05259   0.478  0.633
## L(gov_growth, 1:3)3 -0.01667    0.04593  -0.363  0.717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.9253 on 193 degrees of freedom
## Multiple R-squared: 0.1155, Adjusted R-squared: 0.08803
## F-statistic: 4.202 on 6 and 193 DF, p-value: 0.0005366
```

```
summary(gdp_gov_5)
```

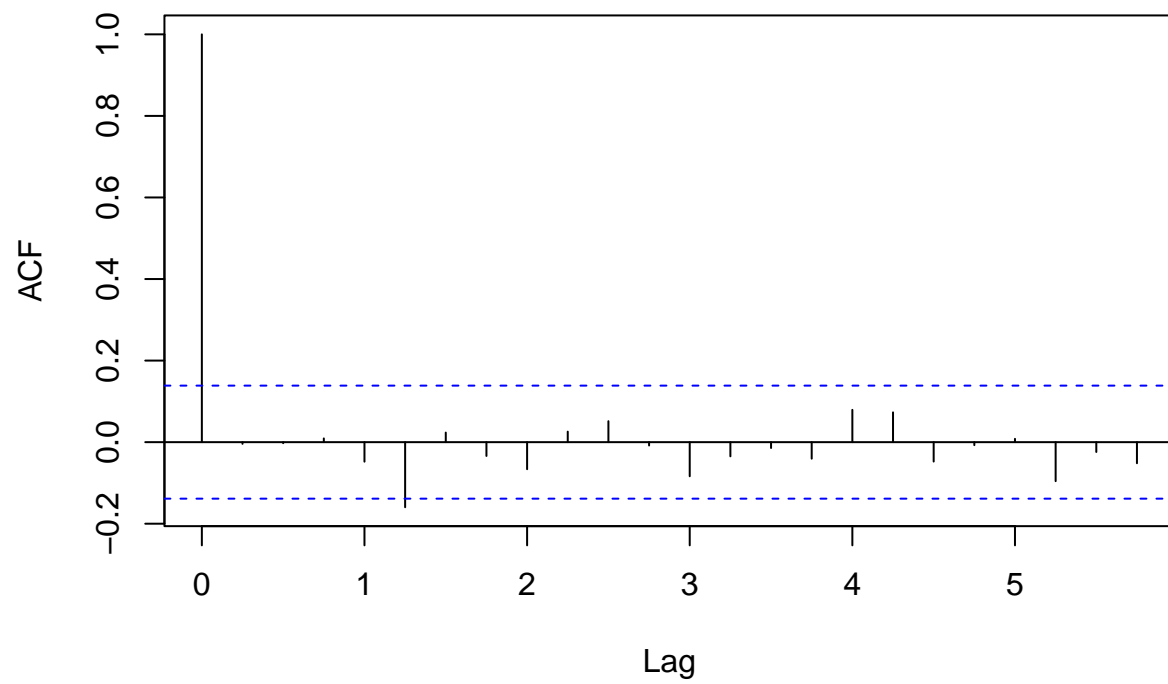
```
##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:5) + L(gov_growth,
##      1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.9712 -0.4407 -0.0629  0.5524  3.3461
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.65863    0.12001   5.488 1.31e-07 ***
## L(gdp_growth, 1:5)1  0.31351    0.07492   4.185 4.39e-05 ***
## L(gdp_growth, 1:5)2  0.07901    0.07744   1.020  0.3089
## L(gdp_growth, 1:5)3 -0.05986    0.07751  -0.772  0.4409
## L(gdp_growth, 1:5)4 -0.02412    0.07562  -0.319  0.7501
## L(gdp_growth, 1:5)5 -0.11961    0.07305  -1.637  0.1032
## L(gov_growth, 1:5)1 -0.02100    0.05420  -0.387  0.6989
## L(gov_growth, 1:5)2  0.01142    0.05627   0.203  0.8393
## L(gov_growth, 1:5)3  0.02396    0.05643   0.425  0.6717
## L(gov_growth, 1:5)4 -0.08316    0.05255  -1.583  0.1152
## L(gov_growth, 1:5)5  0.08905    0.04605   1.934  0.0546 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.918 on 187 degrees of freedom
## Multiple R-squared: 0.1528, Adjusted R-squared: 0.1075
## F-statistic: 3.373 on 10 and 187 DF, p-value: 0.0004554
```

When looking at the residual plots, we see no signs of autocorrelation in the residuals. This is good because we don't want our residuals to have any relation with each other. When looking at the training and testing MSE, they finally come to an agreement and we see that in both instances, the order 3 model is better than the order 5 model. We unfortunately get discrepancies in the AIC and BIC, but we can use MSE here and say that for now, order 3 is the better model. In the summaries, we see that once again, ARDL may not be the best way to go about this. In the order 3 and the 5 model, we see that gov\_growth is not significant in either lag and within gdp\_growth, only the first lag is significant. This could probably be modelled with AR(1), but for the sake of simplicity, I would prefer the order 3 model over the order 5.

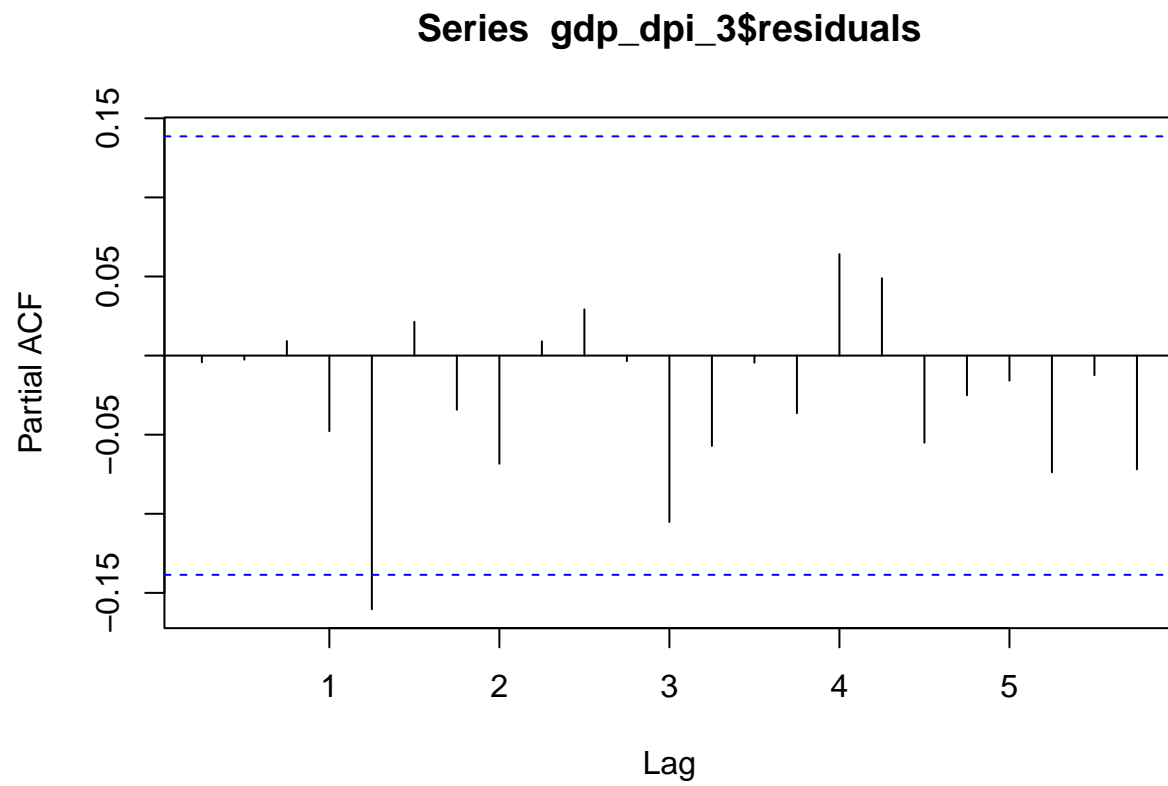
```
#GDP-RDPI
gdp_dpi_3 <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(rdpi_growth, 1:3), data = my_dataframe)
acf(gdp_dpi_3$residuals)
```



### Series gdp\_dpi\_3\$residuals

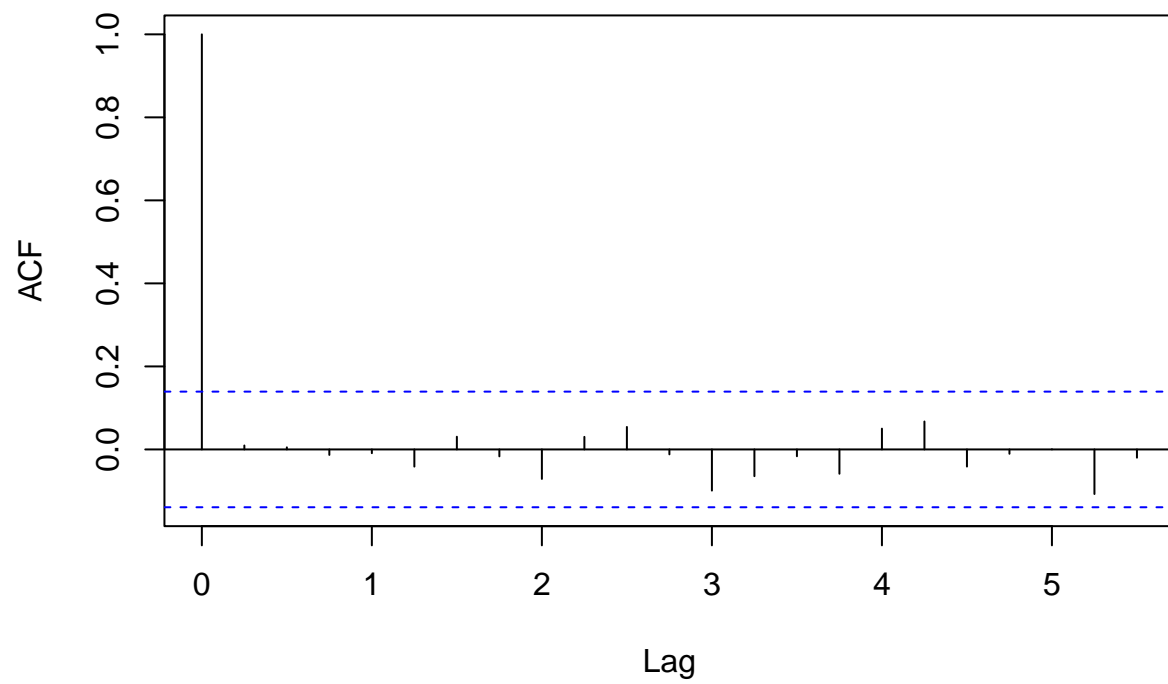


```
pacf(gdp_dpi_3$residuals)
```



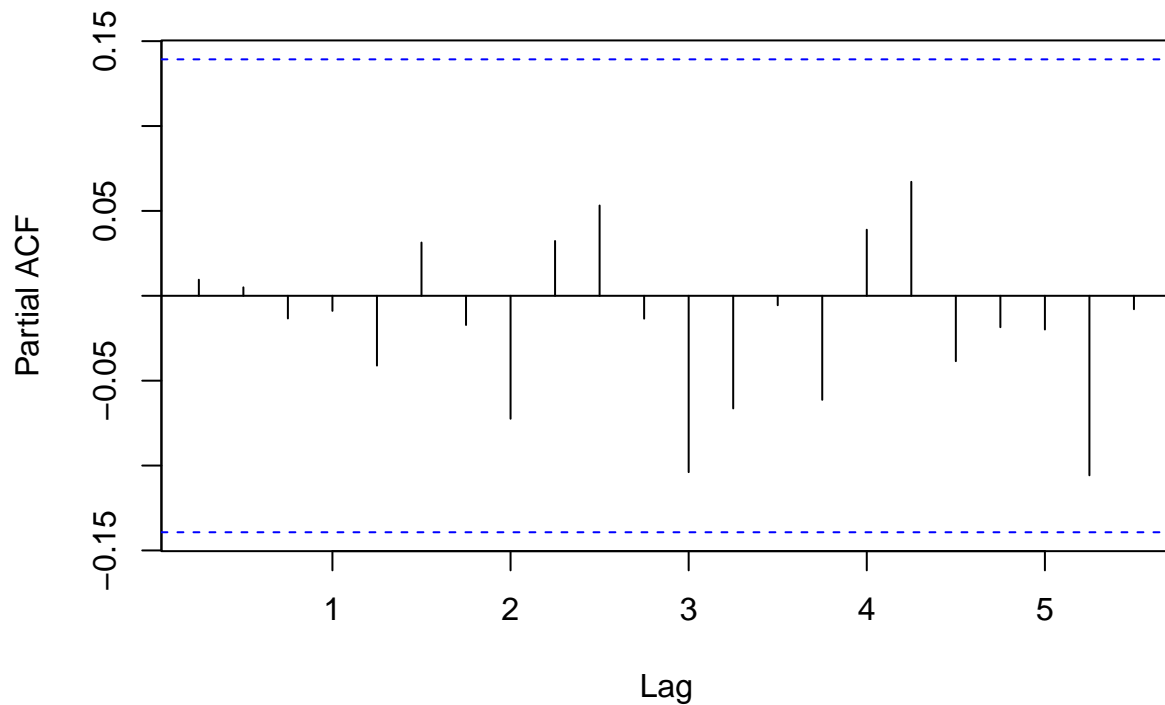
```
gdp_dpi_5 <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(rdpi_growth, 1:5), data=my_dataframe)
acf(gdp_dpi_5$residuals)
```

### Series gdp\_dpi\_5\$residuals



```
pacf(gdp_dpi_5$residuals)
```

## Series gdp\_dpi\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
gdp_rdpi_3_train <- dynlm(gdp_growth~L(gdp_growth, 1:3) + L(rdpi_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_rdpi_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.060069
```

```
predictions <- predict(gdp_rdpi_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2913027
```

```
gdp_rdpi_5_train <- dynlm(gdp_growth~L(gdp_growth, 1:5) + L(rdpi_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(gdp_rdpi_3_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.438254
```

```
predictions <- predict(gdp_rdpi_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.3484835
```

```
AIC(gdp_rdpi_3_train)
```

```
## [1] 537.6379
```

```
AIC(gdp_rdpi_5_train)
```

```
## [1] 533.5333
```

```
BIC(gdp_rdpi_3_train)
```

```
## [1] 564.0244
```

```
BIC(gdp_rdpi_5_train)
```

```
## [1] 572.9925
```

```
#Step 3: forecast 10 steps ahead
```

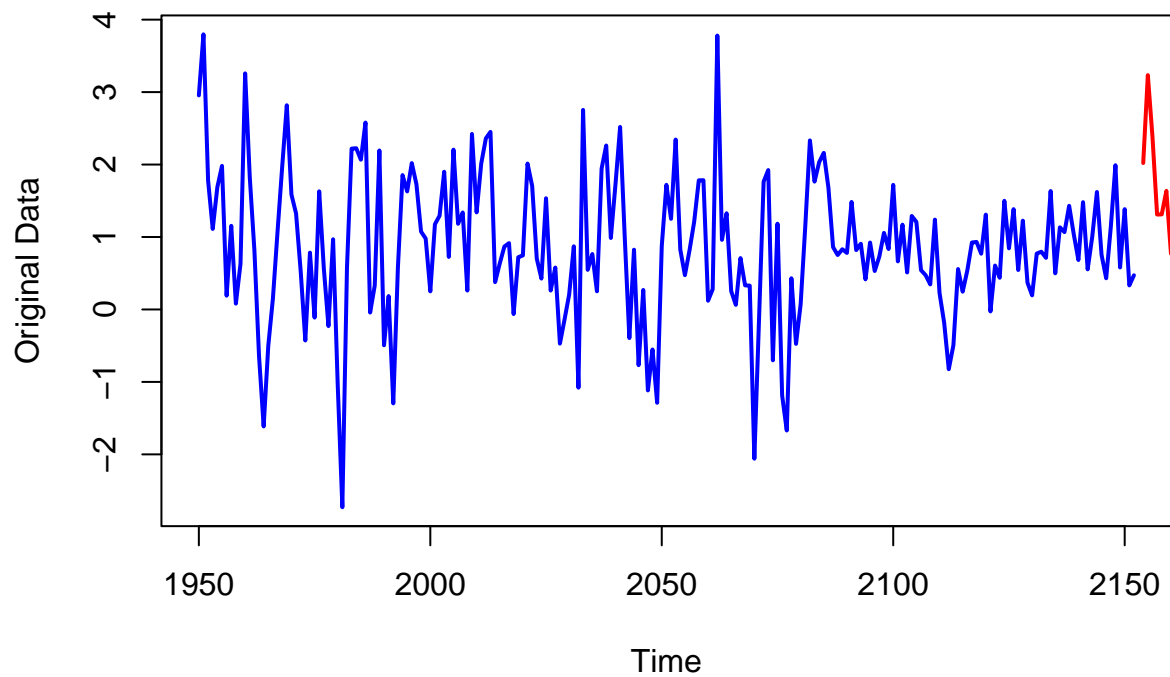
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + rdpi_growth, data = data.frame(my_dataframe), p = 3  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

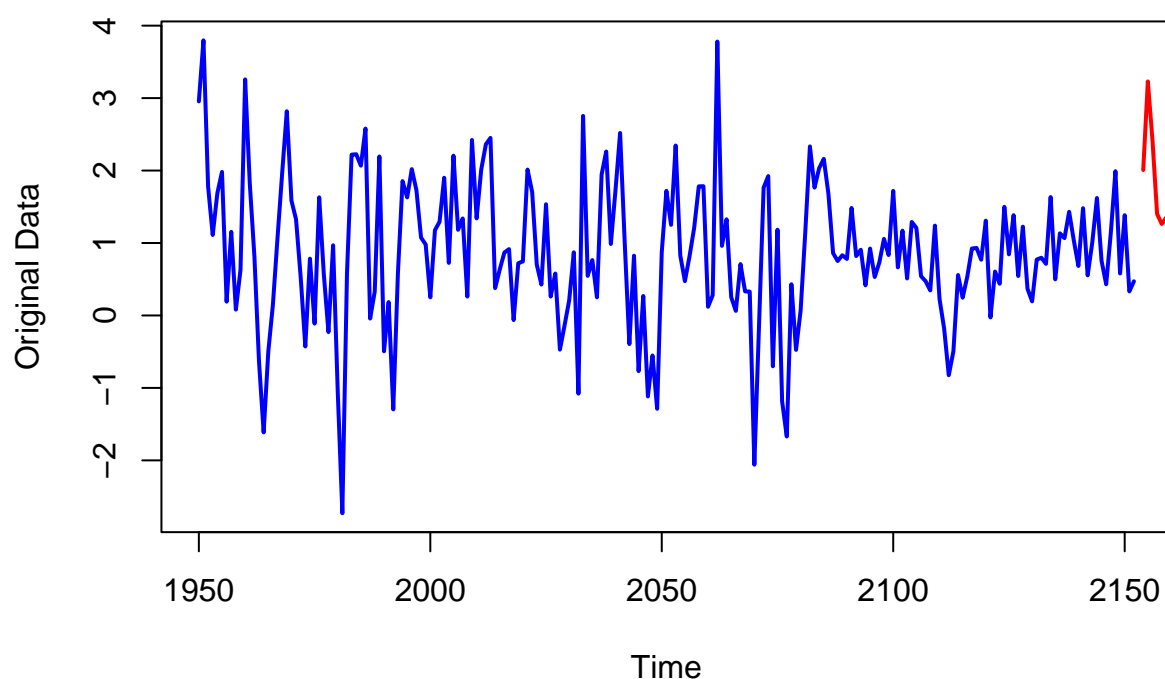
```
ardl <- ardlDlm(formula = gdp_growth ~ gdp_growth + rdpi_growth, data = data.frame(my_dataframe), p = 5  
forecast_data <- forecast(model = ardl, x = gdp_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(gdp_dpi_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:3) + L(rdpi_growth,
## 1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.0358 -0.4995  0.0477  0.5524  3.2419
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.47574    0.11974   3.973  0.0001 ***
## L(gdp_growth, 1:3)1  0.21785    0.08586   2.537  0.0120 *
## L(gdp_growth, 1:3)2  0.10181    0.08495   1.198  0.2322
## L(gdp_growth, 1:3)3 -0.11995    0.07910  -1.516  0.1311
## L(rdpi_growth, 1:3)1  0.21798    0.09071   2.403  0.0172 *
## L(rdpi_growth, 1:3)2 -0.07040    0.09185  -0.767  0.4443
## L(rdpi_growth, 1:3)3  0.07876    0.08785   0.896  0.3711
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.9075 on 193 degrees of freedom
## Multiple R-squared: 0.1492, Adjusted R-squared: 0.1228
## F-statistic: 5.643 on 6 and 193 DF, p-value: 2.023e-05

summary(gdp_dpi_5)

##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = gdp_growth ~ L(gdp_growth, 1:5) + L(rdpi_growth,
## 1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.9606 -0.4812  0.0112  0.5549  3.1743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.63505    0.13960   4.549 9.68e-06 ***
## L(gdp_growth, 1:5)1  0.18185    0.08651   2.102 0.03689 *
## L(gdp_growth, 1:5)2  0.08529    0.08738   0.976 0.33029
## L(gdp_growth, 1:5)3 -0.12446    0.08787  -1.416 0.15831
## L(gdp_growth, 1:5)4 -0.02426    0.08548  -0.284 0.77684
## L(gdp_growth, 1:5)5 -0.04779    0.07967  -0.600 0.54931
## L(rdpi_growth, 1:5)1  0.24917    0.09159   2.721 0.00713 **
## L(rdpi_growth, 1:5)2 -0.02952    0.09321  -0.317 0.75180
## L(rdpi_growth, 1:5)3  0.10721    0.09282   1.155 0.24955
## L(rdpi_growth, 1:5)4 -0.03290    0.09164  -0.359 0.72001
## L(rdpi_growth, 1:5)5 -0.13492    0.08780  -1.537 0.12606
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9015 on 187 degrees of freedom
## Multiple R-squared: 0.183, Adjusted R-squared: 0.1393
## F-statistic: 4.189 on 10 and 187 DF, p-value: 3e-05
```

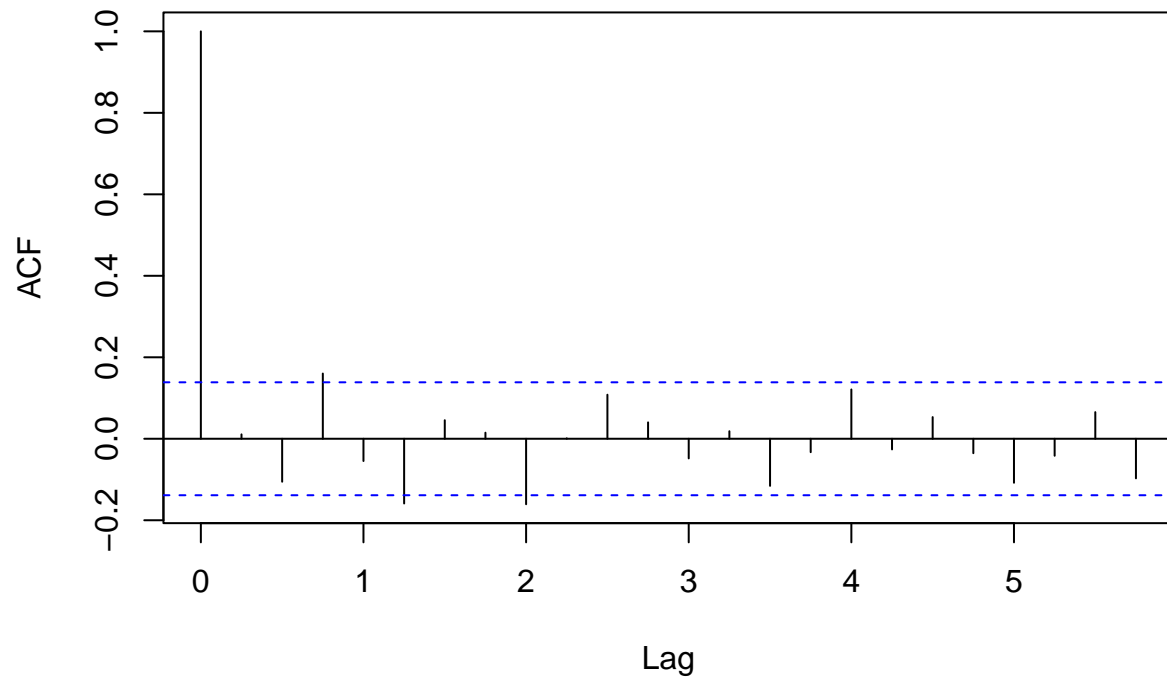
When looking at the ACF and PACF of the residuals, we see no significant spikes (order 3 has some spikes that cross the line in the ACF and PACF, but it is nothing overly concerning). When looking at the MSEs, order 3 does better in both the training and testing. We get another discrepancy in the AIC and BIC, but we can use MSE as the tiebreaker and say that order 3 is better for now. When looking at the summary of both models, we see a positive change: `gdp_growth` and `rdpi_growth` are BOTH significant in lag 1. This means ARDL does actually work here, albeit we should probably only use ARDL order 1.

```
#Cons-GDP
cons_gdp_3 <- dynlm(cons_growth~L(cons_growth, 1:3) + L(gdp_growth, 1:3), data = my_dataframe)

acf(cons_gdp_3$residuals)
```

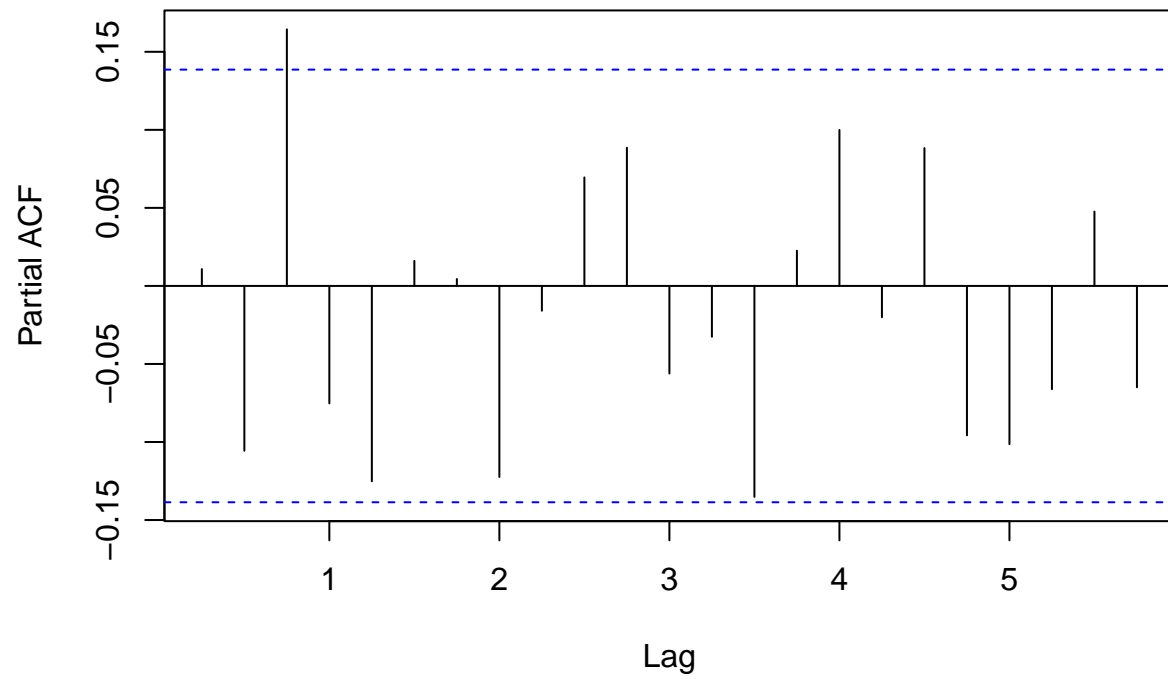


### Series cons\_gdp\_3\$residuals



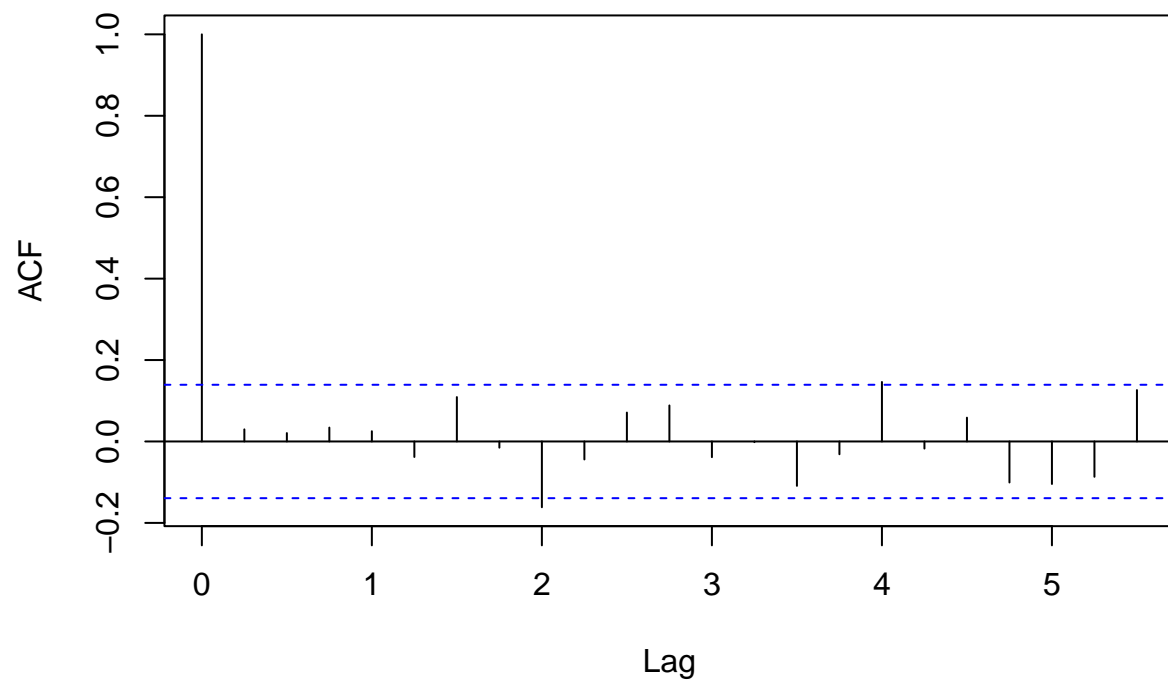
```
pacf(cons_gdp_3$residuals)
```

### Series cons\_gdp\_3\$residuals



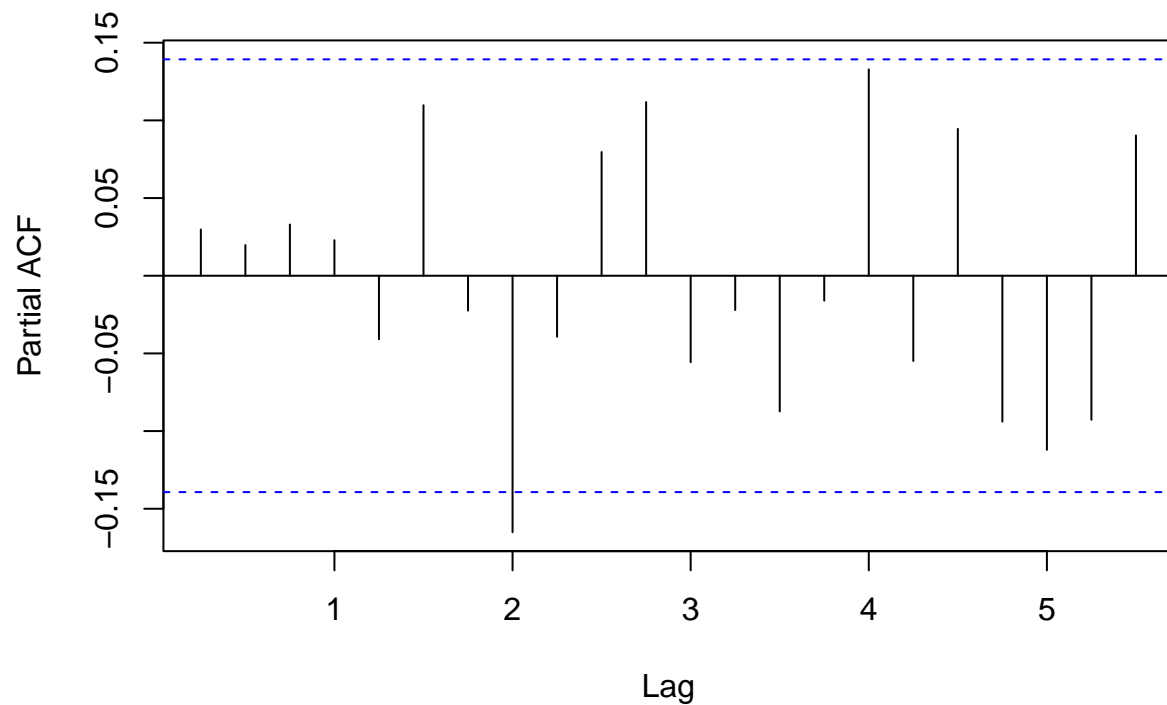
```
cons_gdp_5 <- dynlm(cons_growth~L(cons_growth, 1:5) + L(gdp_growth, 1:5), data=my_dataframe)
acf(cons_gdp_5$residuals)
```

### Series cons\_gdp\_5\$residuals



```
pacf(cons_gdp_5$residuals)
```

## Series cons\_gdp\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
cons_gdp_3_train <- dynlm(cons_growth~L(cons_growth, 1:3) + L(gdp_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(cons_gdp_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.1413
```

```
predictions <- predict(cons_gdp_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2629871
```

```
cons_gdp_5_train <- dynlm(cons_growth~L(cons_growth, 1:5) + L(gdp_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(cons_gdp_5_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.109662
```

```
predictions <- predict(cons_gdp_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2430077
```

```
AIC(cons_gdp_3_train)
```

```
## [1] 452.9726
```

```
AIC(cons_gdp_5_train)
```

```
## [1] 436.7922
```

```
BIC(cons_gdp_3_train)
```

```
## [1] 479.3592
```

```
BIC(cons_gdp_5_train)
```

```
## [1] 476.2514
```

```
#Step 3: forecast 10 steps ahead
```

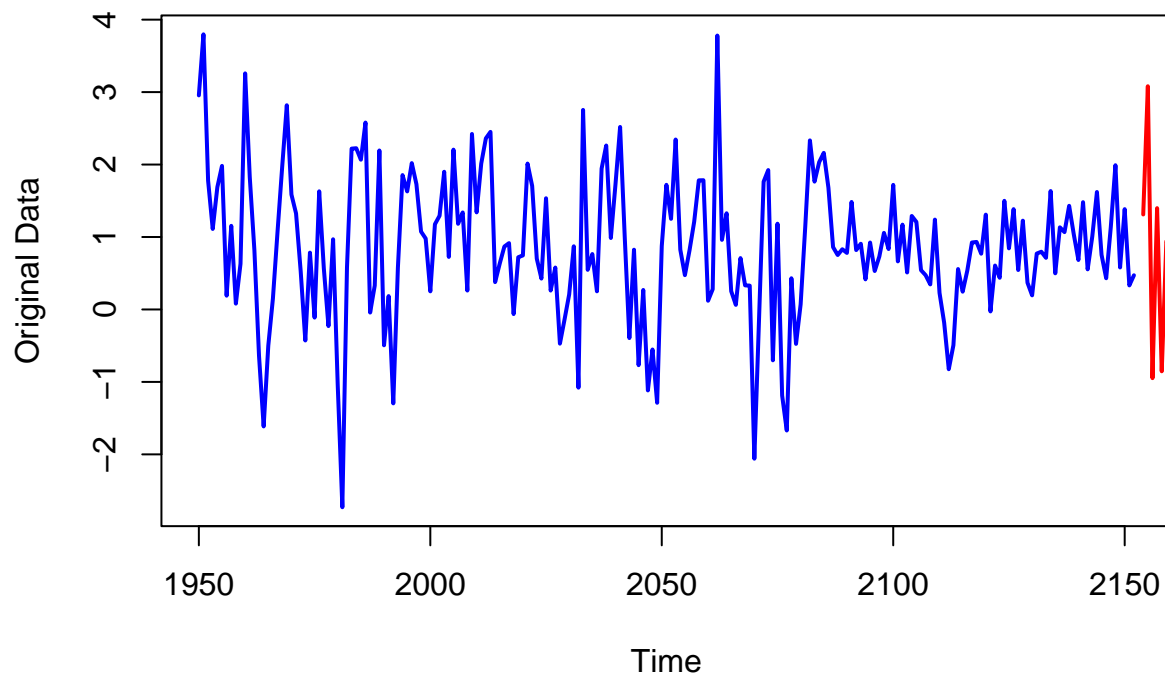
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + gdp_growth, data = data.frame(my_dataframe), p = 1, d = 1, q = 0, h = 10)
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data", main = "Original Data and 10-step Forecast")
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data)), col = "red", lty = 1, lwd = 2)
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

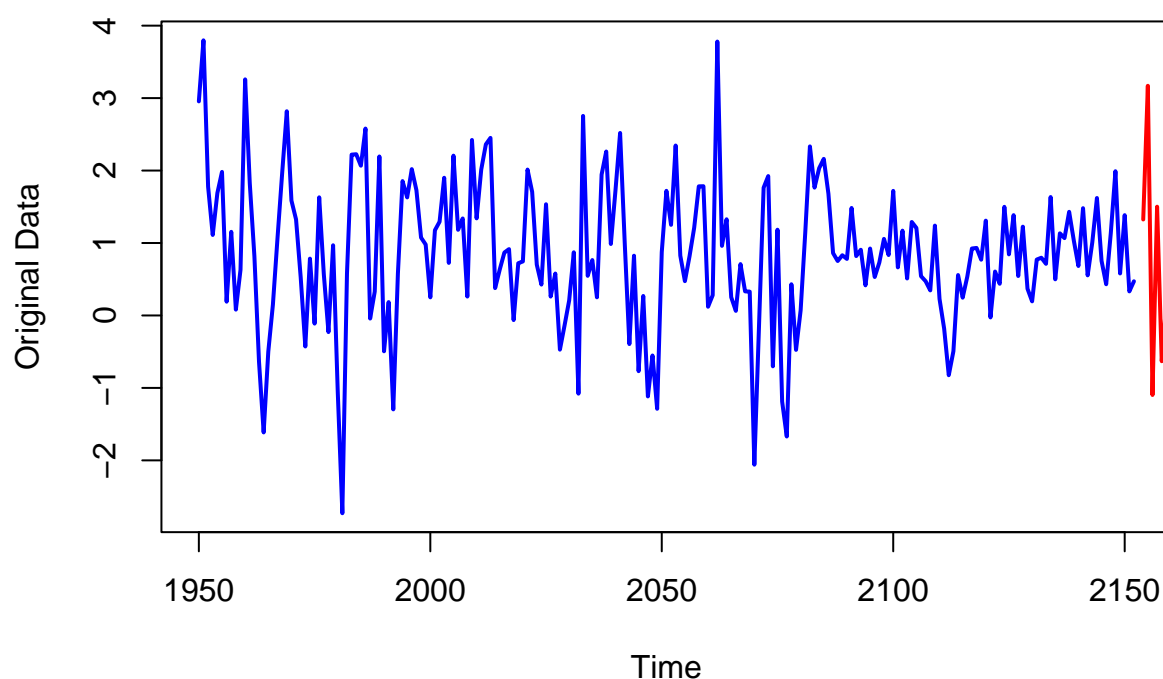
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + gdp_growth, data = data.frame(my_dataframe), p = 1, d = 1, q = 1)
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data", main = "Original Data")
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data)), col = "red", lty = 1, lwd = 2)
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(cons_gdp_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:3) + L(gdp_growth,
## 1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.99196 -0.38636  0.02771  0.43996  2.19578
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.54264   0.09858   5.505 1.17e-07 ***
## L(cons_growth, 1:3)1  0.02923   0.08142   0.359  0.7200
## L(cons_growth, 1:3)2  0.37233   0.08123   4.584 8.19e-06 ***
## L(cons_growth, 1:3)3  0.10805   0.08143   1.327  0.1861
## L(gdp_growth, 1:3)1   0.15173   0.07255   2.091  0.0378 *
## L(gdp_growth, 1:3)2  -0.17542   0.07313  -2.399  0.0174 *
## L(gdp_growth, 1:3)3  -0.10377   0.06894  -1.505  0.1339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7344 on 193 degrees of freedom
## Multiple R-squared: 0.1736, Adjusted R-squared: 0.1479
## F-statistic: 6.756 on 6 and 193 DF, p-value: 1.632e-06
```

```
summary(cons_gdp_5)
```

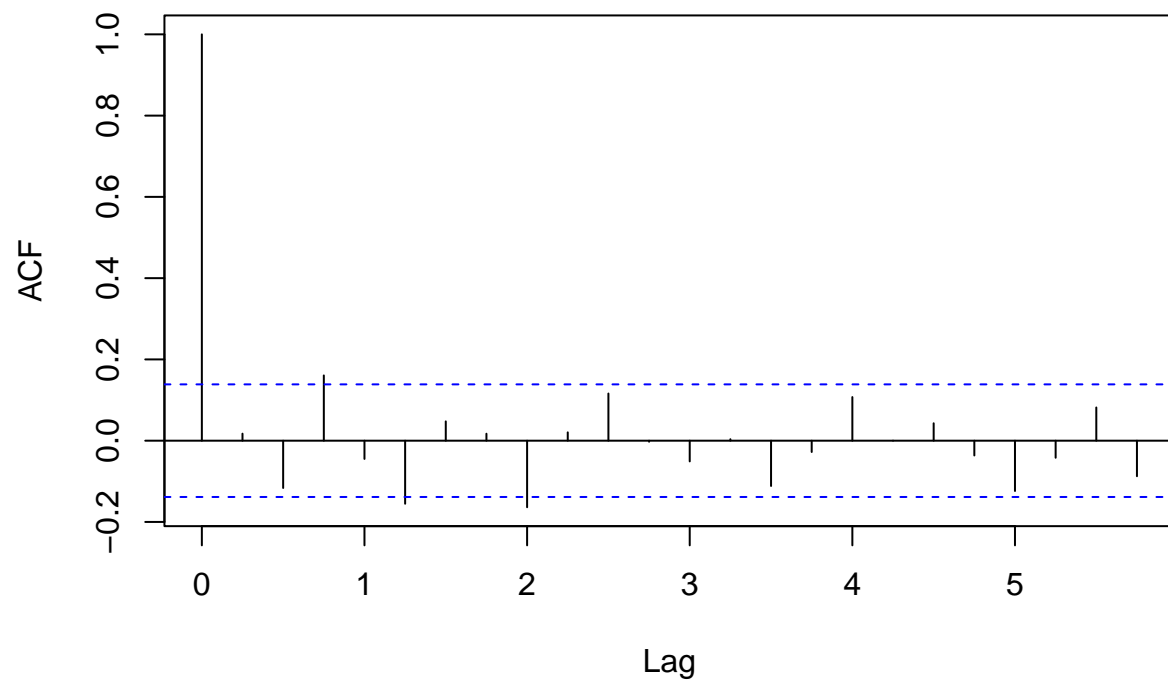
```
##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:5) + L(gdp_growth,
## 1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.11154 -0.34003 -0.00136  0.41019  2.34272
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.64891    0.11153   5.818 2.54e-08 ***
## L(cons_growth, 1:5)1  0.04642    0.08913   0.521  0.6031
## L(cons_growth, 1:5)2  0.21633    0.09049   2.391  0.0178 *
## L(cons_growth, 1:5)3  0.16213    0.08564   1.893  0.0599 .
## L(cons_growth, 1:5)4  0.00246    0.08297   0.030  0.9764
## L(cons_growth, 1:5)5 -0.06543    0.08055  -0.812  0.4176
## L(gdp_growth, 1:5)1   0.15732    0.07314   2.151  0.0328 *
## L(gdp_growth, 1:5)2  -0.11407    0.07271  -1.569  0.1184
## L(gdp_growth, 1:5)3  -0.03924    0.07188  -0.546  0.5858
## L(gdp_growth, 1:5)4  -0.10588    0.07202  -1.470  0.1432
## L(gdp_growth, 1:5)5   0.01627    0.06774   0.240  0.8104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7061 on 187 degrees of freedom
## Multiple R-squared: 0.1539, Adjusted R-squared: 0.1087
## F-statistic: 3.401 on 10 and 187 DF, p-value: 0.0004151
```

When looking at the residual plots, there are no overly significant spikes except for maybe Lag 2 in the PACF plot for order 5. So far, so good. We now come to an interesting discrepancy. For the MSE, order 5 wins out in both, but when we look at AIC and BIC, order 3 wins out in both. Since order 3 has 2/3 in its favor, for now, we will say it's better. When looking at the order 3 summary, we see that lag 2 for cons\_growth is significant and gdp growth lags 1 and 2 are significant. In the order 5 model, we see that cons\_growth lag 2 and gdp\_growth lag 1 are significant. Once again, similar to last time, it may be a good idea to run an ARDL model of order 1, or perhaps drop cons\_growth and only base it on gdp\_growth. This one has some room for flexibility.

```
#Cons-Invest
cons_invest_3 <- dynlm(cons_growth~L(cons_growth, 1:3) + L(inv_growth, 1:3), data = my_dataframe)
acf(cons_invest_3$residuals)
```

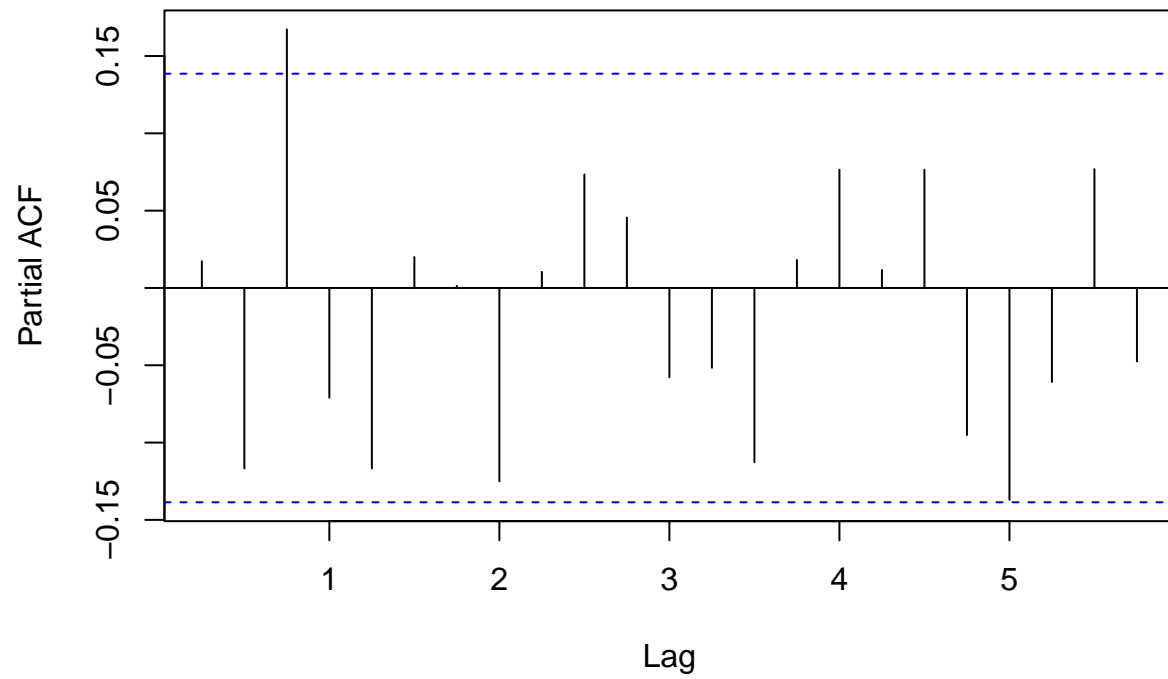


### Series cons\_invest\_3\$residuals



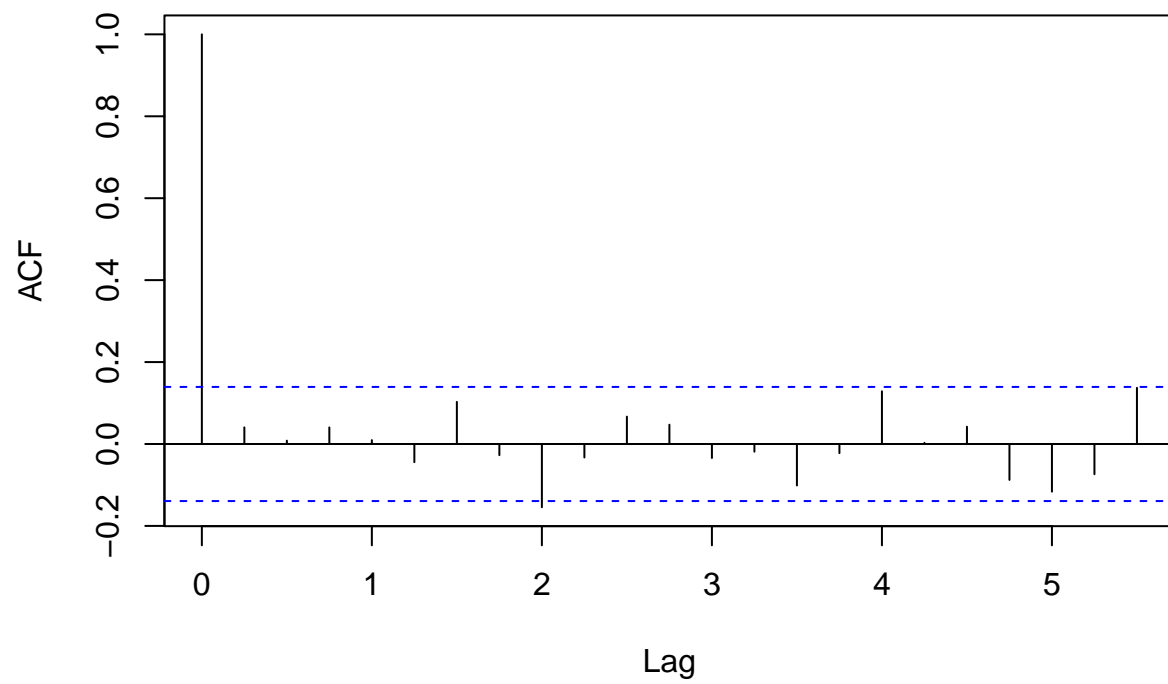
```
pacf(cons_invest_3$residuals)
```

### Series cons\_invest\_3\$residuals



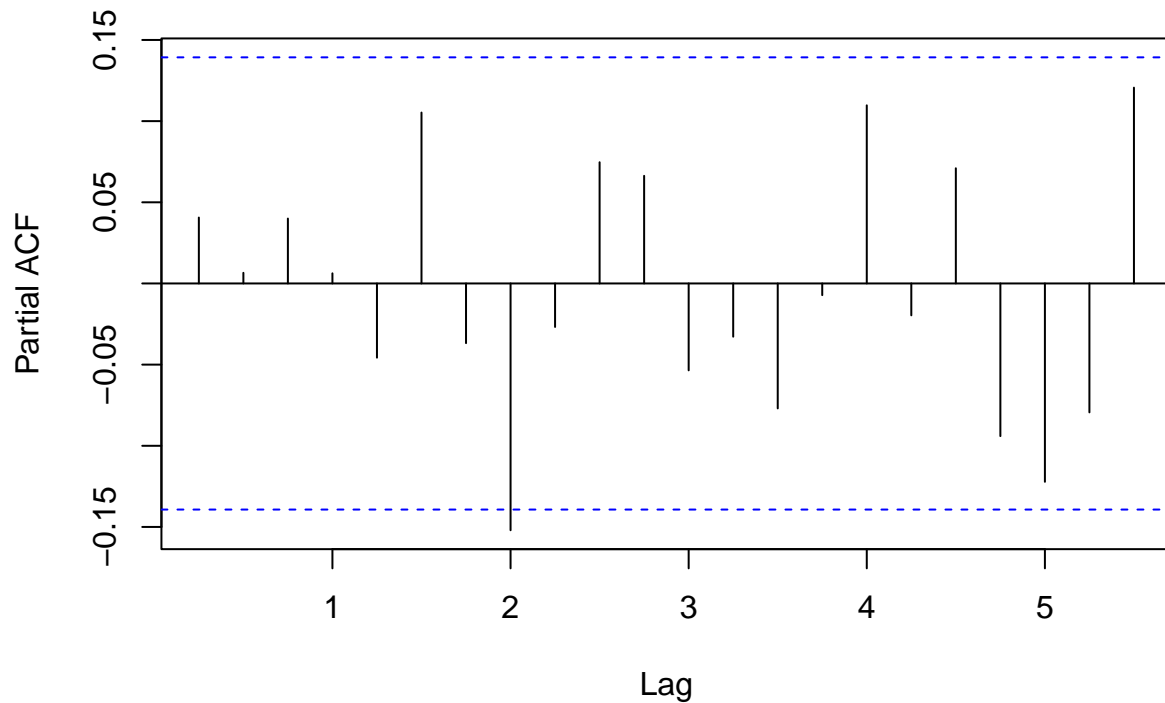
```
cons_invest_5 <- dynlm(cons_growth~L(cons_growth, 1:5) + L(inv_growth, 1:5), data=my_dataframe)
acf(cons_invest_5$residuals)
```

### Series cons\_invest\_5\$residuals



```
pacf(cons_invest_5$residuals)
```

### Series cons\_invest\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
cons_invest_3_train <- dynlm(cons_growth~L(cons_growth, 1:3) + L(inv_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(cons_invest_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.140272
```

```
predictions <- predict(cons_invest_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2689332
```

```
cons_invest_5_train <- dynlm(cons_growth~L(cons_growth, 1:5) + L(inv_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(cons_invest_5_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.111136
```

```
predictions <- predict(cons_invest_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2555573
```

```
AIC(cons_invest_3_train)
```

```
## [1] 454.8521
```

```
AIC(cons_invest_5_train)
```

```
## [1] 438.2524
```

```
BIC(cons_invest_3_train)
```

```
## [1] 481.2386
```

```
BIC(cons_invest_5_train)
```

```
## [1] 477.7116
```

```
#Step 3: forecast 10 steps ahead
```

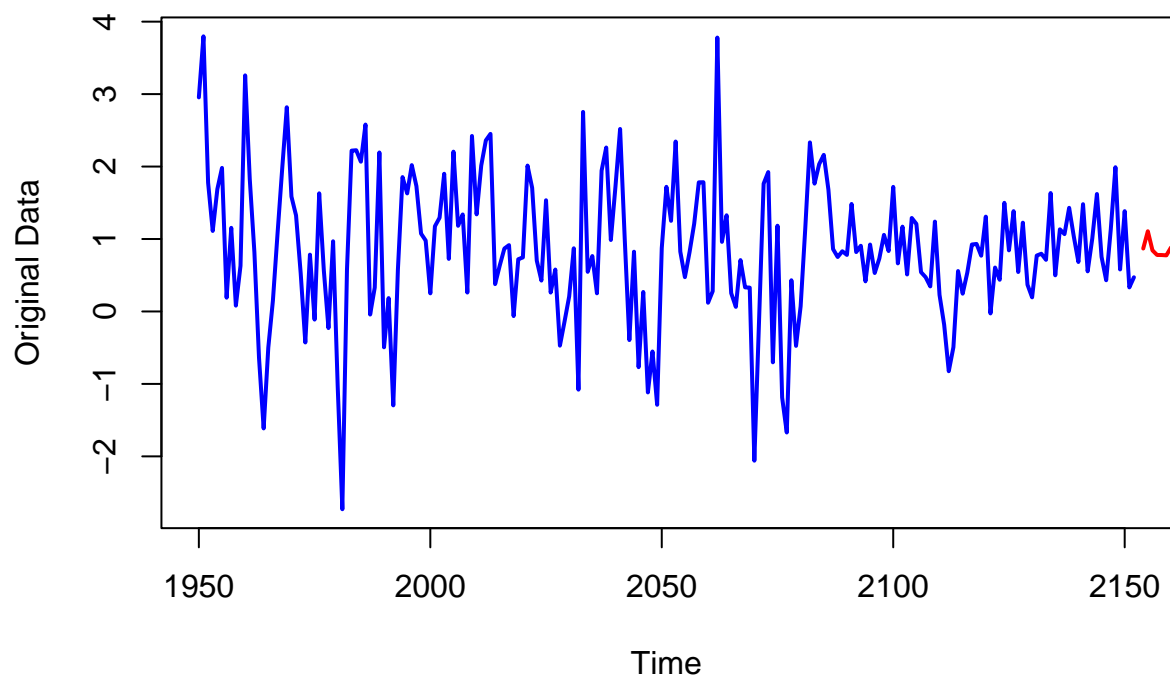
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + inv_growth, data = data.frame(my_dataframe), p = 3, d = 1, q = 0, h = 10)
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data", main = "Original Data and 10-step ahead forecast")
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data)), col = "red", lty = 1, lwd = 2)
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

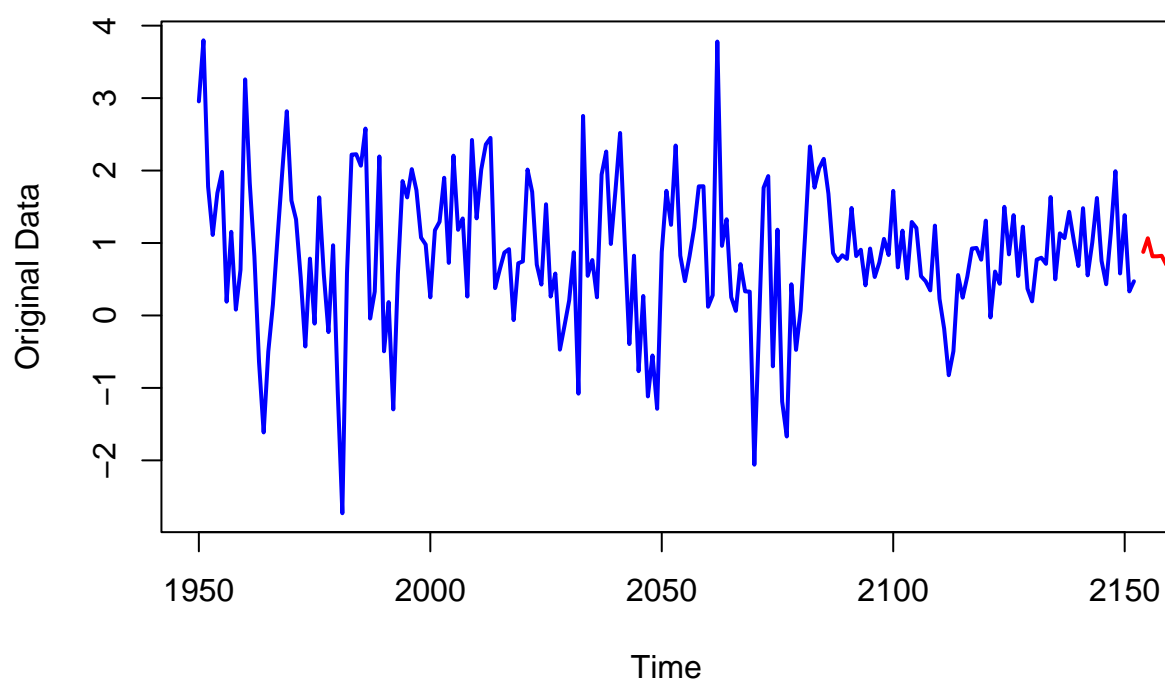
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + inv_growth, data = data.frame(my_dataframe), p = 1, d = 1, q = 1)
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data", main = "Original Data")
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data)), col = "red", lty = 1, lwd = 1)
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(cons_invest_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:3) + L(inv_growth,
##      1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.99254 -0.31889 -0.00485  0.45088  2.12195
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.54977    0.11311   4.860 2.42e-06 ***
## L(cons_growth, 1:3)1  0.09545    0.06777   1.408  0.16063
## L(cons_growth, 1:3)2  0.24123    0.07514   3.210  0.00155 **
## L(cons_growth, 1:3)3  0.03836    0.07218   0.531  0.59576
## L(inv_growth, 1:3)1   0.03191    0.01278   2.498  0.01333 *
## L(inv_growth, 1:3)2  -0.01634    0.01291  -1.266  0.20719
## L(inv_growth, 1:3)3  -0.01404    0.01143  -1.228  0.22102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7379 on 193 degrees of freedom
## Multiple R-squared: 0.1658, Adjusted R-squared: 0.1398
## F-statistic: 6.392 on 6 and 193 DF, p-value: 3.706e-06
```

```
summary(cons_invest_5)
```

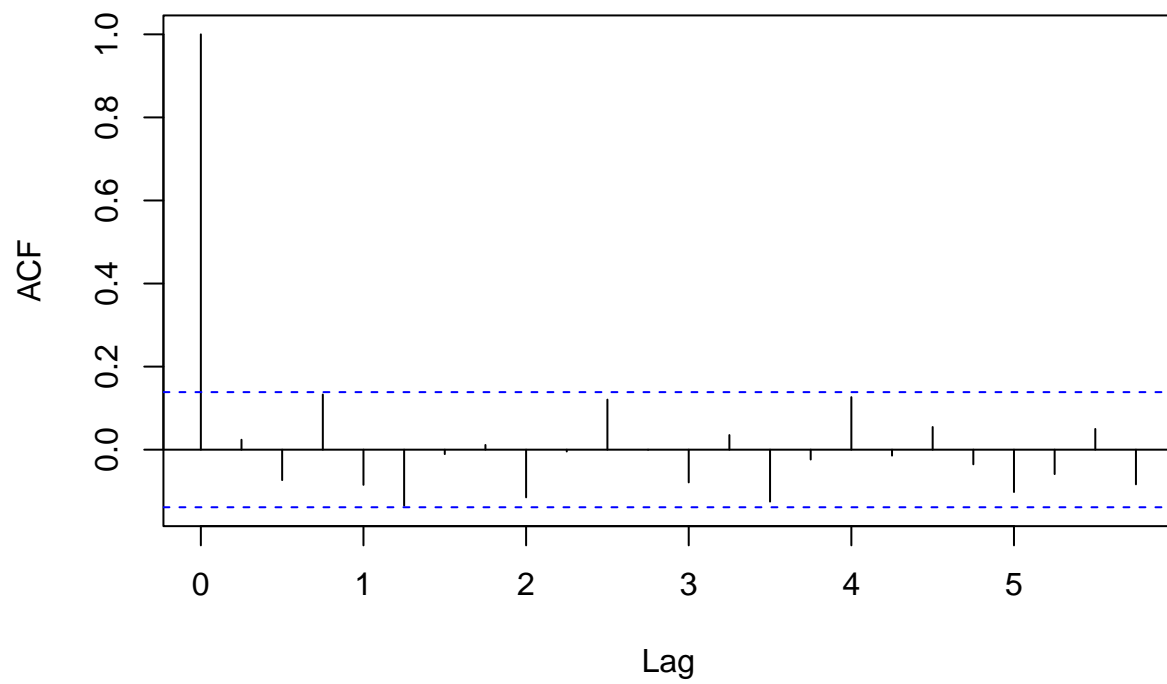
```
##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:5) + L(inv_growth,
##      1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.07234 -0.36948 -0.00648  0.42587  2.13174
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.623901   0.131246   4.754 3.98e-06 ***
## L(cons_growth, 1:5)1  0.132064   0.072286   1.827  0.0693 .
## L(cons_growth, 1:5)2  0.133434   0.080264   1.662  0.0981 .
## L(cons_growth, 1:5)3  0.138356   0.077641   1.782  0.0764 .
## L(cons_growth, 1:5)4 -0.040625   0.074703  -0.544  0.5872
## L(cons_growth, 1:5)5 -0.047300   0.071401  -0.662  0.5085
## L(inv_growth, 1:5)1   0.026834   0.012741   2.106  0.0365 *
## L(inv_growth, 1:5)2  -0.015831   0.012670  -1.250  0.2130
## L(inv_growth, 1:5)3  -0.001992   0.012601  -0.158  0.8745
## L(inv_growth, 1:5)4  -0.015916   0.012500  -1.273  0.2045
## L(inv_growth, 1:5)5  -0.001996   0.011263  -0.177  0.8596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7087 on 187 degrees of freedom
## Multiple R-squared: 0.1476, Adjusted R-squared: 0.1021
## F-statistic: 3.239 on 10 and 187 DF, p-value: 0.0007097
```

When looking at the ACF and PACF residual plots, except for the spike at lag = 0.75, it looks pretty good for both orders. We do have a discrepancy for training and testing MSE, but order 3 beats out order 5 for both the AIC and BIC, so we can say that order 3 is better in that regard. In the order 3 summary, we see that cons\_growth Lag 2 and inv\_growth Lag 1 are both significant, implying that an ARDL could work, albeit at a lower order. In the order 5 model, only inv\_growth of Lag 1 is significant. If we were to redo this model, it would be best to use ARDL of order 1, or we could even drop cons\_growth entirely and not do ARDL.

```
#Cons-Gov
cons_gov_3 <- dynlm(cons_growth~L(cons_growth, 1:3) + L(gov_growth, 1:3), data = my_dataframe)
acf(cons_gov_3$residuals)
```



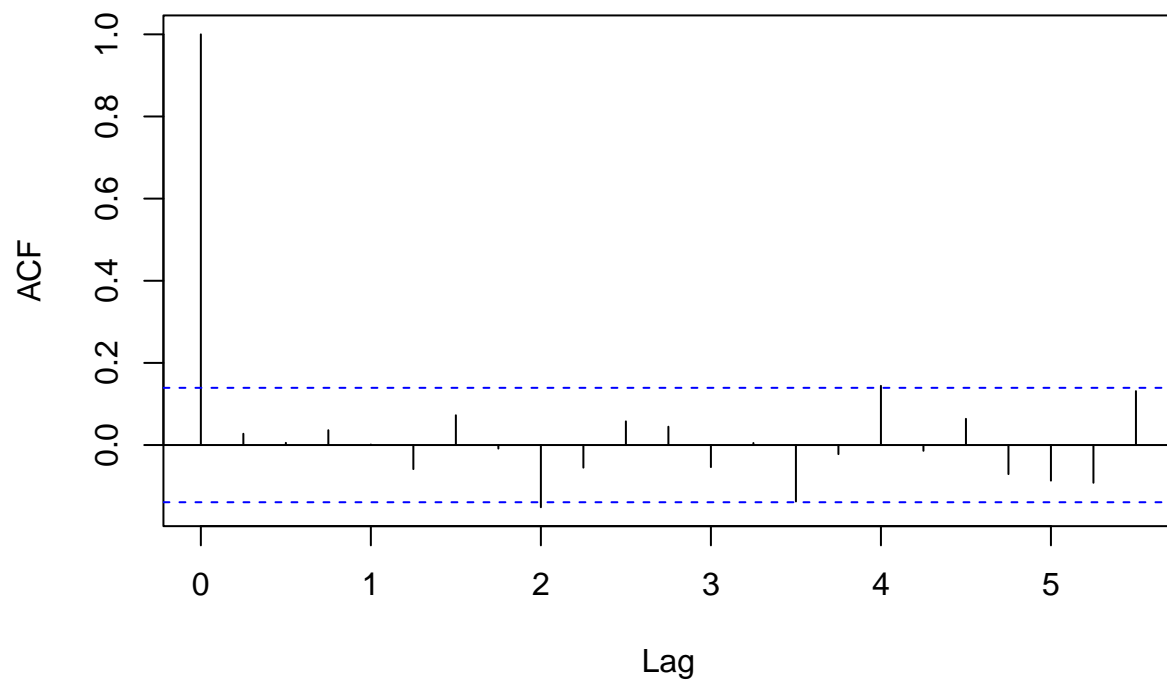
### Series cons\_gov\_3\$residuals



```
pacf(cons_gov_3$residuals)
```

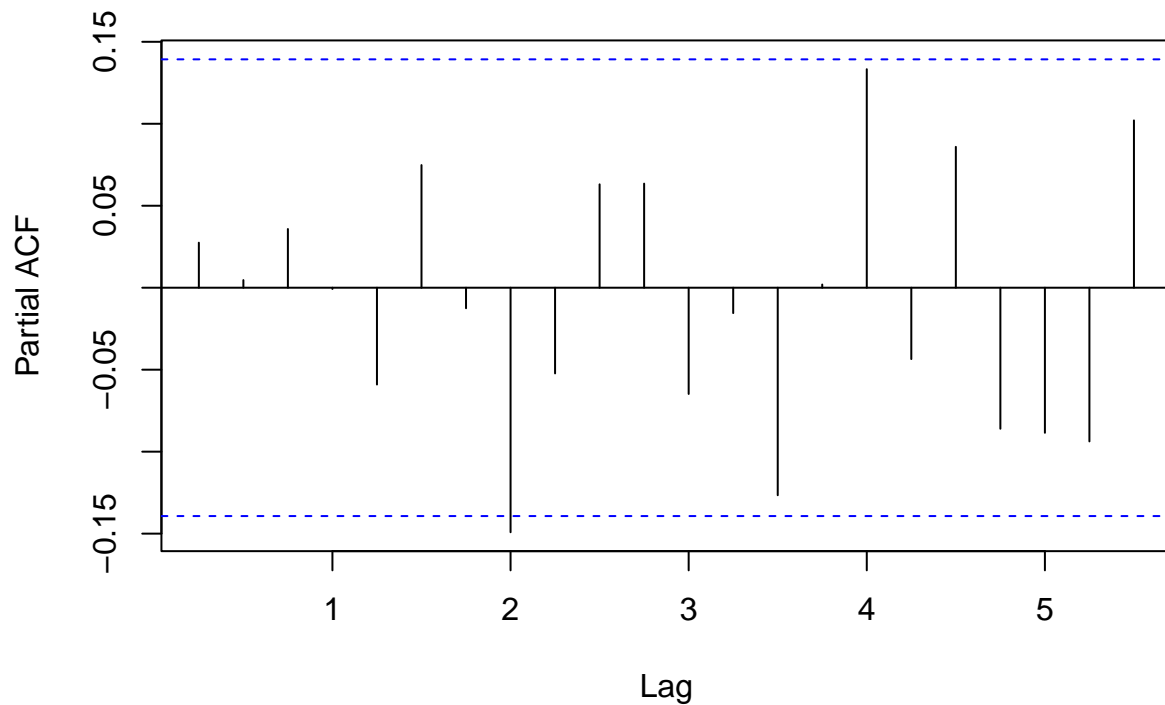
```
cons_gov_5 <- dynlm(cons_growth~L(cons_growth, 1:5) + L(gov_growth, 1:5), data=my_dataframe)
acf(cons_gov_5$residuals)
```

### Series cons\_gov\_5\$residuals



```
pacf(cons_gov_5$residuals)
```

## Series cons\_gov\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
cons_gov_3_train <- dynlm(cons_growth~L(cons_growth, 1:3) + L(gov_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(cons_gov_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.178543
```

```
predictions <- predict(cons_gov_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2122369
```

```
cons_gov_5_train <- dynlm(cons_growth~L(cons_growth, 1:5) + L(gov_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(cons_gov_3_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.321638
```

```
predictions <- predict(cons_gov_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2339647
```

```
AIC(cons_gov_3_train)
```

```
## [1] 459.59
```

```
AIC(cons_gov_5_train)
```

```
## [1] 444.3865
```

```
BIC(cons_gov_3_train)
```

```
## [1] 485.9766
```

```
BIC(cons_gov_5_train)
```

```
## [1] 483.8457
```

#Step 3: forecast 10 steps ahead

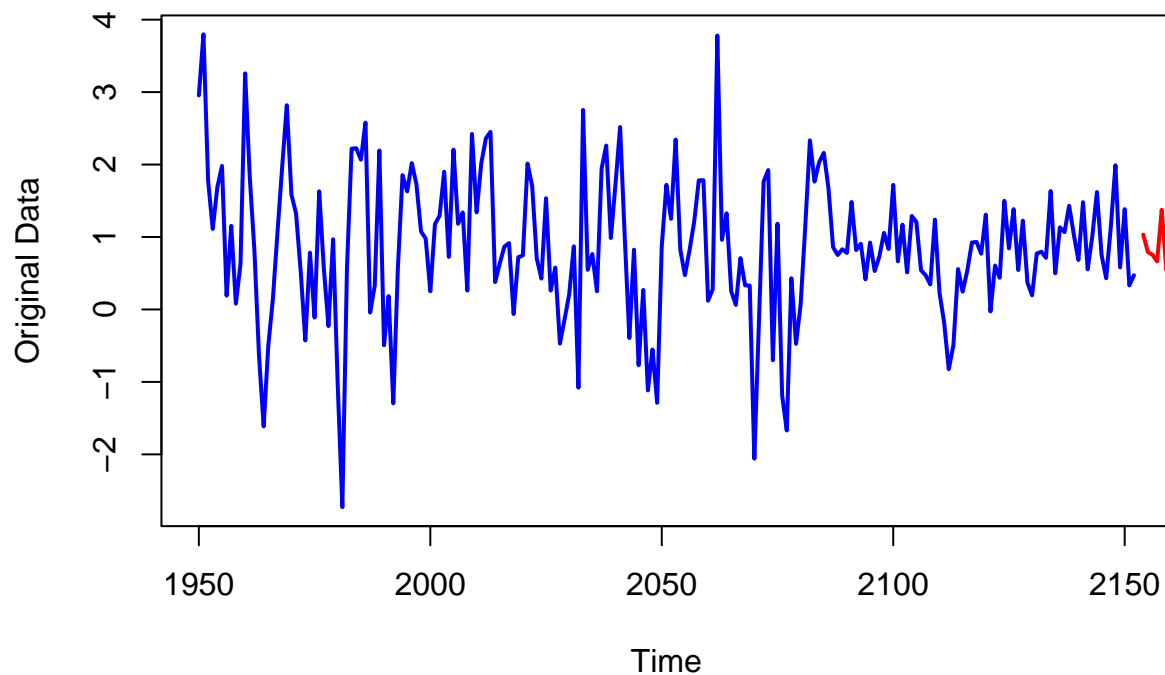
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + gov_growth, data = data.frame(my_dataframe), p = 3)
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data", r
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

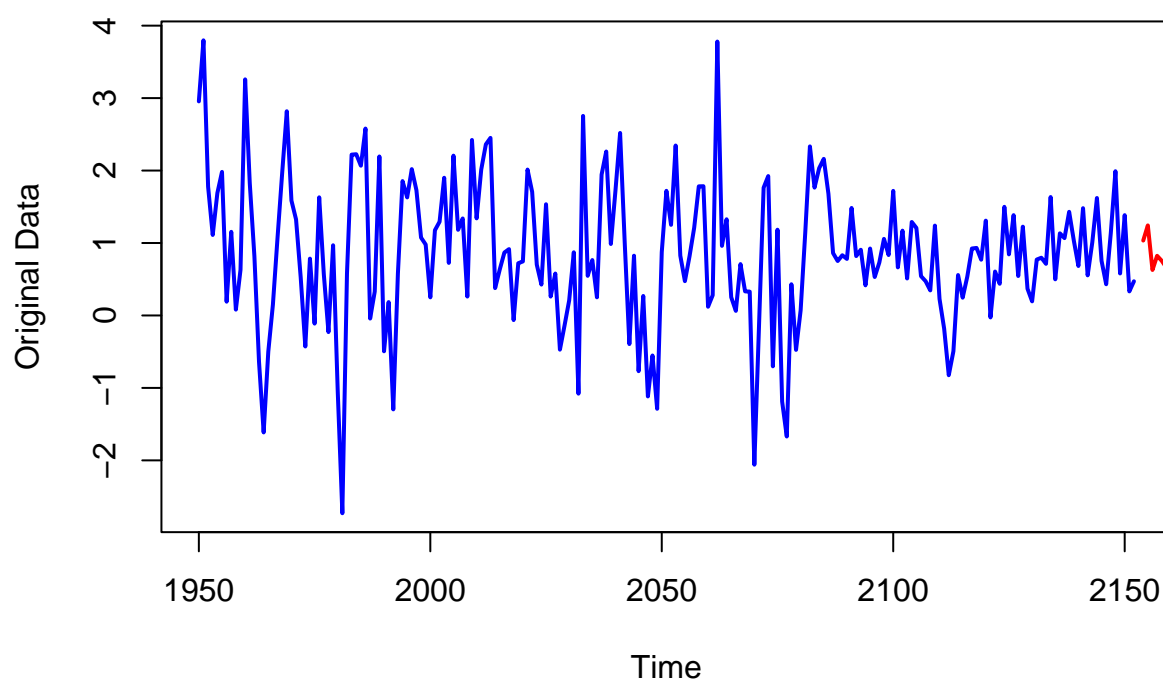
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + gov_growth, data = data.frame(my_dataframe), p = 1, d = 1, q = 0,
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data", main = "Original Data")
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data)))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(cons_gov_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:3) + L(gov_growth,
## 1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.84991 -0.33242 -0.01873  0.47036  2.30805
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.533318   0.105559   5.052 1.01e-06 ***
## L(cons_growth, 1:3)1  0.132819   0.066371   2.001  0.0468 *
## L(cons_growth, 1:3)2  0.270996   0.060096   4.509 1.13e-05 ***
## L(cons_growth, 1:3)3 -0.004345   0.062584  -0.069  0.9447
## L(gov_growth, 1:3)1 -0.011359   0.037122  -0.306  0.7599
## L(gov_growth, 1:3)2 -0.073494   0.042090  -1.746  0.0824 .
## L(gov_growth, 1:3)3  0.078738   0.036902   2.134  0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7467 on 193 degrees of freedom
## Multiple R-squared: 0.1458, Adjusted R-squared: 0.1192
## F-statistic: 5.489 on 6 and 193 DF, p-value: 2.867e-05
```

```
summary(cons_gov_5)
```

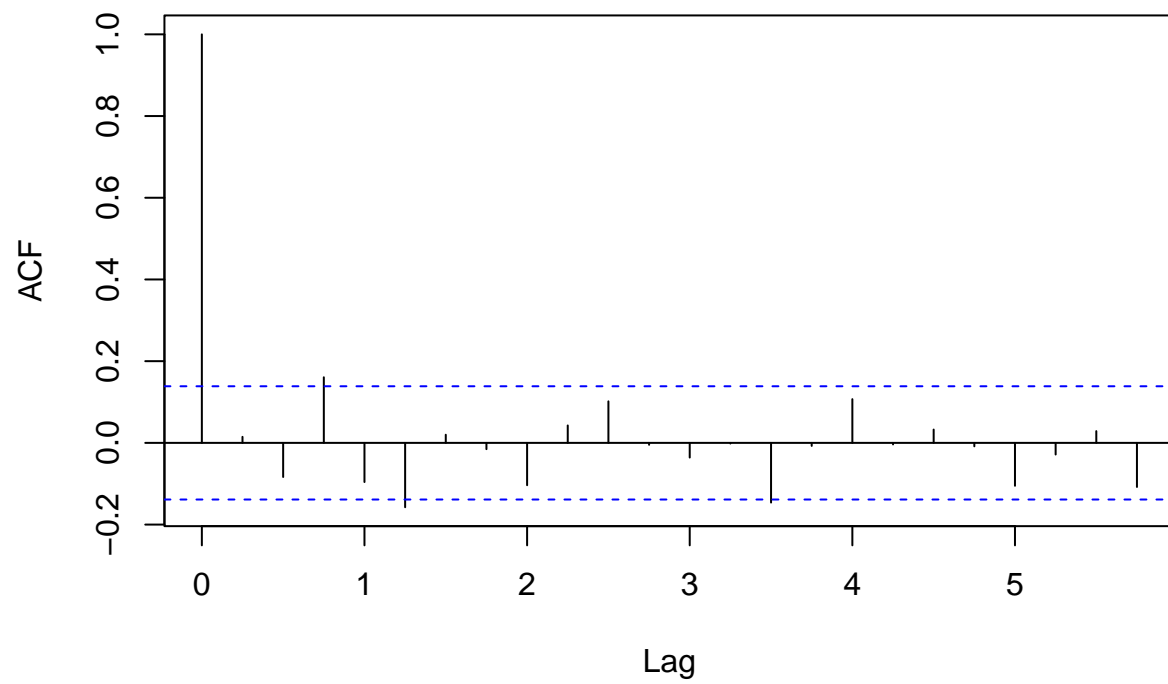
```
##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:5) + L(gov_growth,
##      1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.99181 -0.34492 -0.00002  0.40197  2.24409
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.61742    0.11684   5.284 3.49e-07 ***
## L(cons_growth, 1:5)1  0.16429    0.07032   2.336 0.02054 *
## L(cons_growth, 1:5)2  0.18799    0.07034   2.673 0.00819 **
## L(cons_growth, 1:5)3  0.09241    0.06833   1.352 0.17789
## L(cons_growth, 1:5)4 -0.09877    0.06107  -1.617 0.10747
## L(cons_growth, 1:5)5 -0.06064    0.06155  -0.985 0.32578
## L(gov_growth, 1:5)1   0.03629    0.04045   0.897 0.37080
## L(gov_growth, 1:5)2  -0.03950    0.04266  -0.926 0.35565
## L(gov_growth, 1:5)3   0.02726    0.04289   0.636 0.52581
## L(gov_growth, 1:5)4  -0.02626    0.04163  -0.631 0.52897
## L(gov_growth, 1:5)5   0.03702    0.03635   1.018 0.30983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7198 on 187 degrees of freedom
## Multiple R-squared: 0.1208, Adjusted R-squared: 0.0738
## F-statistic: 2.57 on 10 and 187 DF, p-value: 0.00619
```

The ACF and PACF residuals plot look good. Nothing to be concerned about. The order 3 model does a better job in both train/test MSE and the AIC/BIC tests. Order 3 so far is clearly the superior model. In the order 3 summary, we see that cons\_growth Lag 1 and 2 are significant, and gov\_growth lag 3 is significant. In the order 5 model, only lags 1 and 2 for gdp\_growth is significant. It is very unusual for gov\_growth Lag 3 to be significant, considering that lag 1 and 2 are insignificant, so it could be a fluke. If we were to redo this model, it would be better to use a regular AR(2) model since the gov\_growth isn't really that important and it may not be worth taking that extra step to add a gov\_growth Lag 3 component.

```
#Cons-RDPI
cons_dpi_3 <- dynlm(cons_growth~L(cons_growth, 1:3) + L(rdpi_growth, 1:3), data = my_dataframe)
acf(cons_dpi_3$residuals)
```

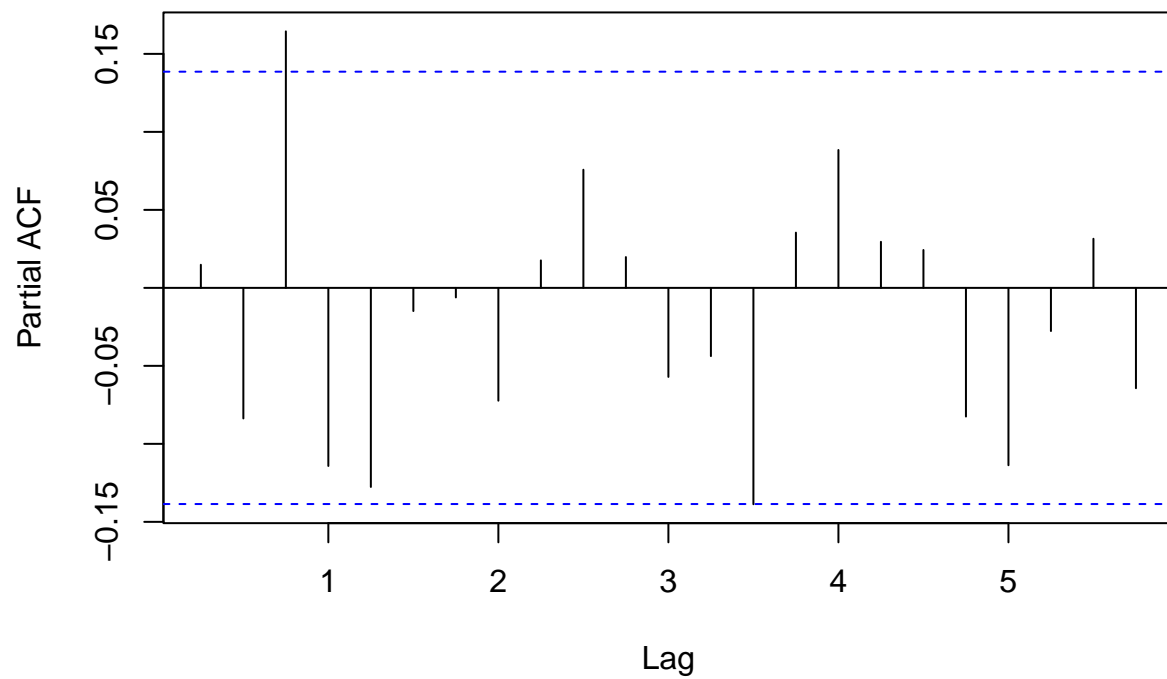


### Series cons\_dpi\_3\$residuals



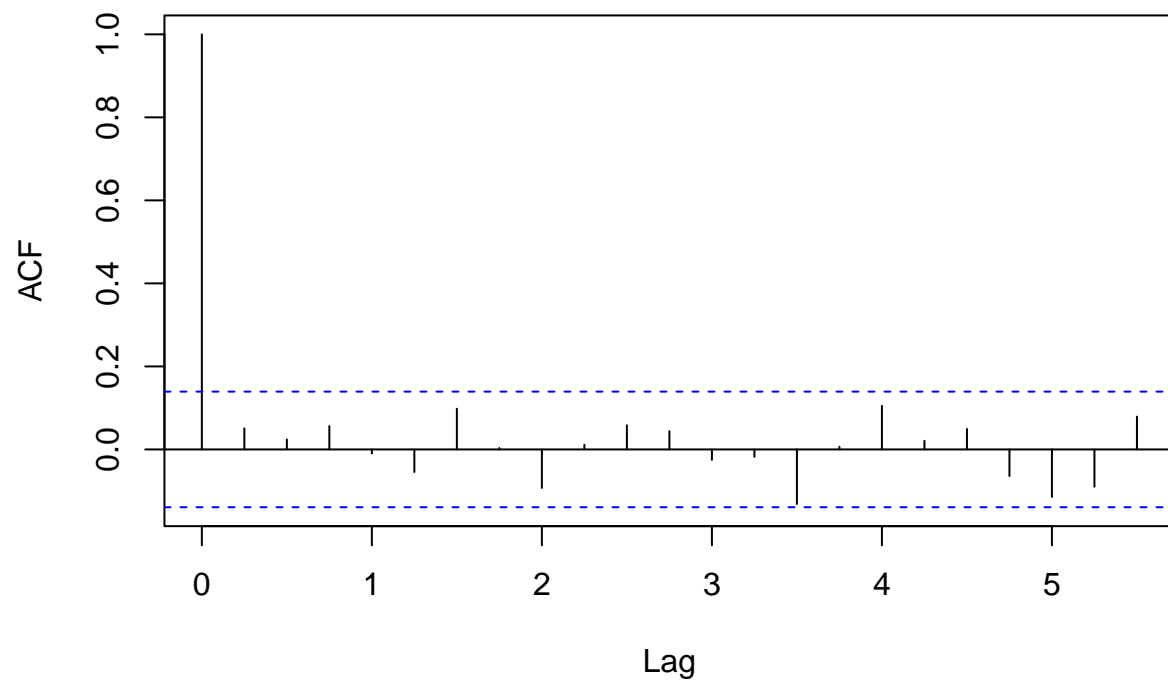
```
pacf(cons_dpi_3$residuals)
```

### Series cons\_dpi\_3\$residuals



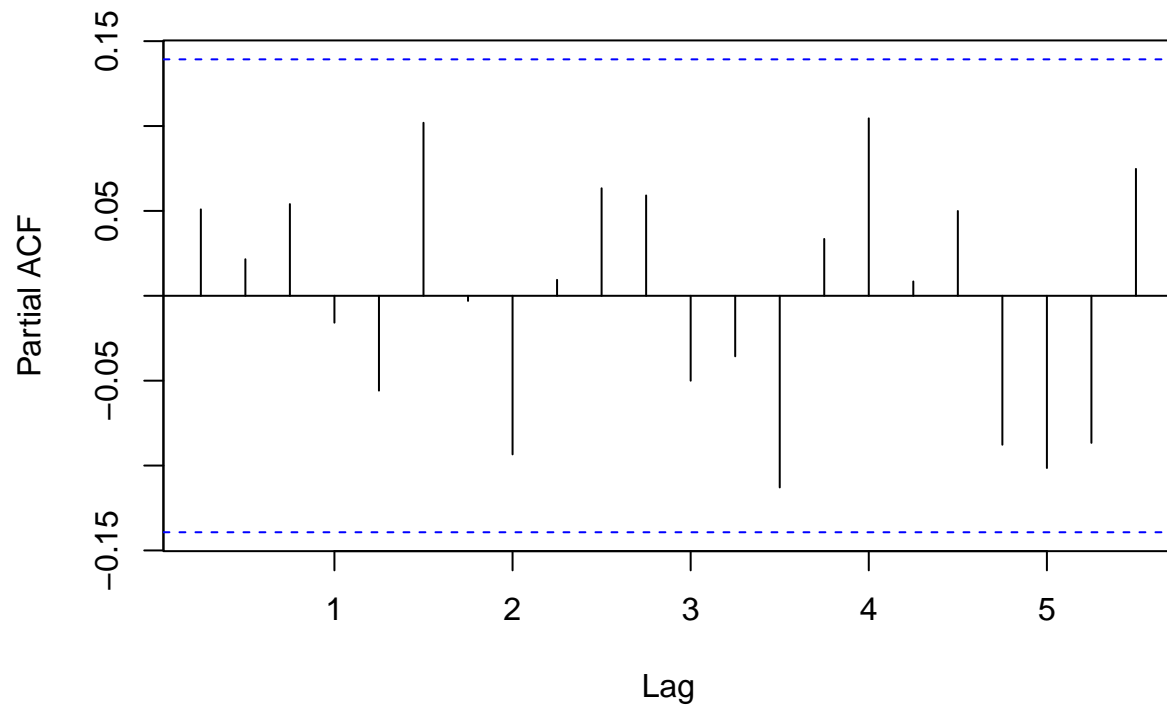
```
cons_dpi_5 <- dynlm(cons_growth~L(cons_growth, 1:5) + L(rdpi_growth, 1:5), data=my_dataframe)
acf(cons_dpi_5$residuals)
```

### Series cons\_dpi\_5\$residuals



```
pacf(cons_dpi_5$residuals)
```

## Series cons\_dpi\_5\$residuals



```
#Step 2. Split into train and test and get mses, aic, bic
cons_dpi_3_train <- dynlm(cons_growth~L(cons_growth, 1:3) + L(rdpi_growth, 1:3), data = train_data)

train_mse <- mean((as.numeric(predict(cons_dpi_3_train)[1:133]) - train_data_df$gdp_growth[4:136])^2)
print(train_mse)
```

```
## [1] 1.111105
```

```
predictions <- predict(cons_dpi_3_train, n.ahead=67)
test_mse <- mean(na.omit((as.numeric(predictions[137:203]) - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2517516
```

```
cons_dpi_5_train <- dynlm(cons_growth~L(cons_growth, 1:5) + L(rdpi_growth, 1:5), data = train_data)

train_mse <- mean((as.numeric(predict(cons_dpi_5_train)[1:131]) - train_data_df$gdp_growth[6:136])^2)
print(train_mse)
```

```
## [1] 1.077032
```

```
predictions <- predict(cons_dpi_5_train, n.ahead=67)
test_mse <- mean(na.omit((predictions[137:203] - test_data_df$gdp_growth))^2)
print(test_mse)
```

```
## [1] 0.2879298
```

```
AIC(cons_dpi_3_train)
```

```
## [1] 456.6023
```

```
AIC(cons_dpi_5_train)
```

```
## [1] 434.7935
```

```
BIC(cons_dpi_3_train)
```

```
## [1] 482.9888
```

```
BIC(cons_dpi_5_train)
```

```
## [1] 474.2527
```

```
#Step 3: forecast 10 steps ahead
```

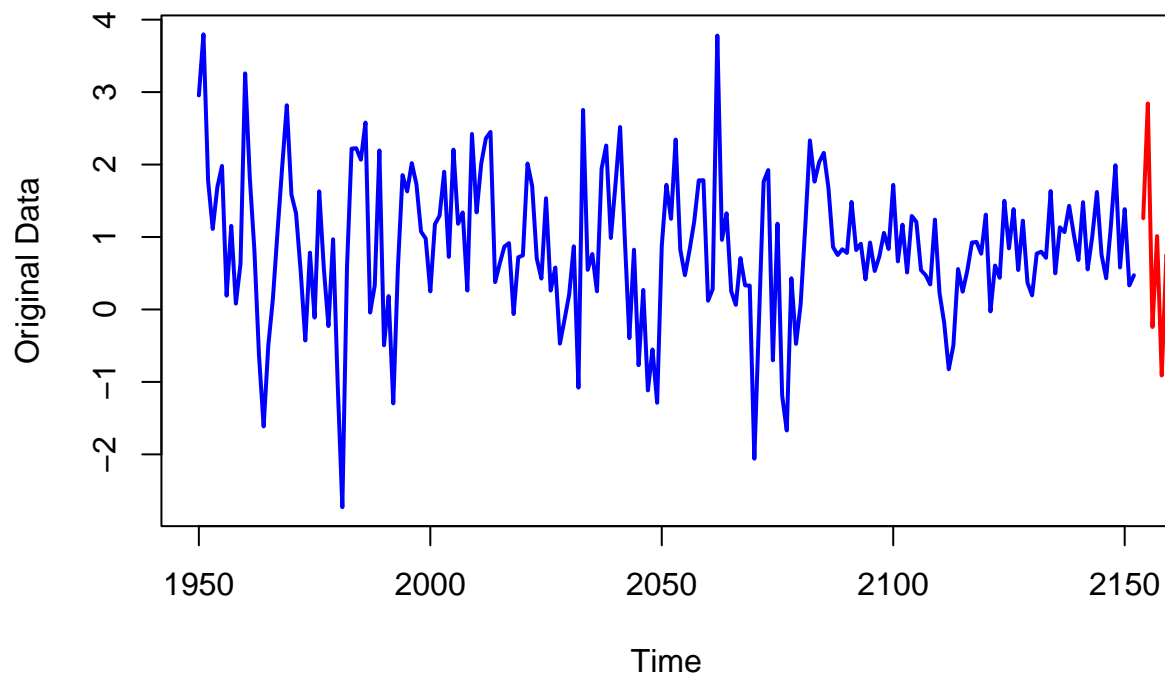
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + rdpi_growth, data = data.frame(my_dataframe), p =  
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
#5 order model
```

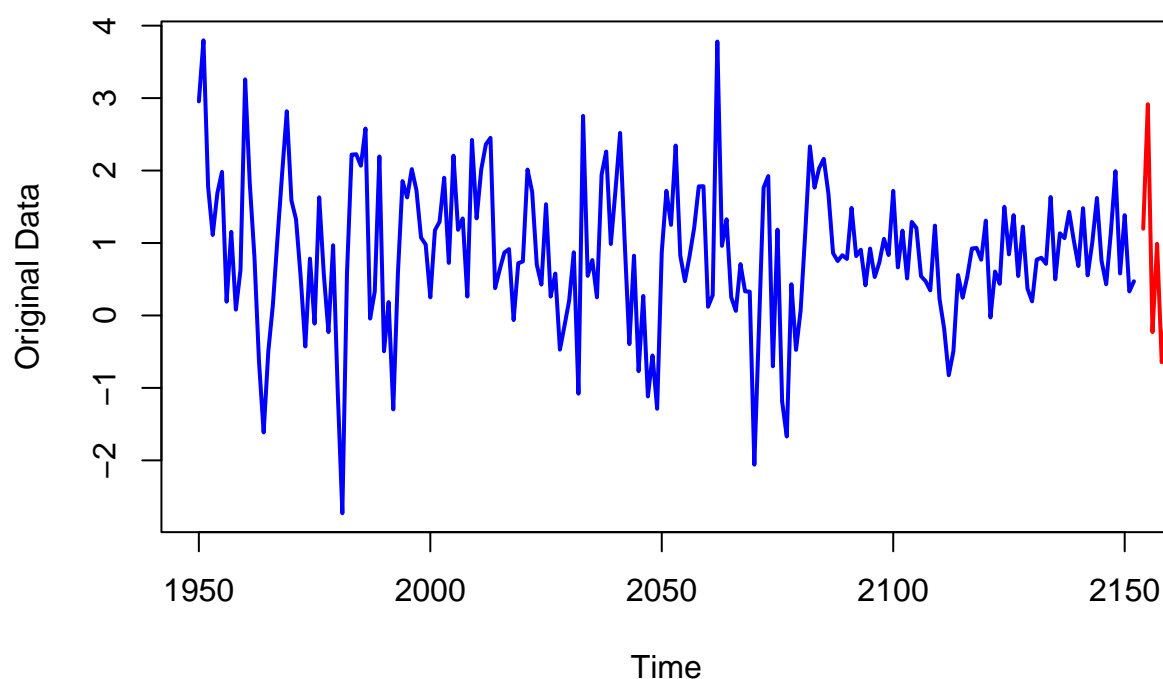
```
ardl <- ardlDlm(formula = cons_growth ~ cons_growth + rdpi_growth, data = data.frame(my_dataframe), p =  
forecast_data <- forecast(model = ardl, x = cons_growth, h = 10)
```

```
original_data <- ts(my_dataframe[, 1], start = 1950)
```

```
plot(original_data, type = "l", col = "blue", lty = 1, lwd = 2, xlab = "Time", ylab = "Original Data",
```

```
lines(ts(forecast_data$forecasts, start = end(original_data) + 1, frequency = frequency(original_data))
```

## 10-Step-Ahead Forecast with Prediction Interval



```
summary(cons_dpi_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(1), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:3) + L(rdpi_growth,
## 1:3), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.93733 -0.37064  0.00213  0.44067  2.11785
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.54001   0.10342   5.222 4.57e-07 ***
## L(cons_growth, 1:3)1  0.07260   0.07713   0.941  0.3477
## L(cons_growth, 1:3)2  0.31273   0.07422   4.213 3.86e-05 ***
## L(cons_growth, 1:3)3  0.06459   0.07148   0.904  0.3673
## L(rdpi_growth, 1:3)1  0.14239   0.07311   1.948  0.0529 .
## L(rdpi_growth, 1:3)2 -0.11373   0.07465  -1.524  0.1293
## L(rdpi_growth, 1:3)3 -0.09545   0.06868  -1.390  0.1662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7411 on 193 degrees of freedom
## Multiple R-squared: 0.1584, Adjusted R-squared: 0.1323
## F-statistic: 6.056 on 6 and 193 DF, p-value: 7.923e-06
```

```
summary(cons_dpi_5)
```

```
##
## Time series regression with "ts" data:
## Start = 1951(3), End = 2000(4)
##
## Call:
## dynlm(formula = cons_growth ~ L(cons_growth, 1:5) + L(rdpi_growth,
##      1:5), data = my_dataframe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.12431 -0.35074 -0.02005  0.40980  2.21700
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.71475    0.11434   6.251 2.7e-09 ***
## L(cons_growth, 1:5)1  0.07161    0.08278   0.865  0.3881
## L(cons_growth, 1:5)2  0.19882    0.08228   2.416  0.0166 *
## L(cons_growth, 1:5)3  0.16767    0.08036   2.086  0.0383 *
## L(cons_growth, 1:5)4 -0.04911    0.07432  -0.661  0.5096
## L(cons_growth, 1:5)5 -0.02227    0.07138  -0.312  0.7554
## L(rdpi_growth, 1:5)1  0.15796    0.07262   2.175  0.0309 *
## L(rdpi_growth, 1:5)2 -0.06827    0.07264  -0.940  0.3485
## L(rdpi_growth, 1:5)3 -0.07216    0.07164  -1.007  0.3151
## L(rdpi_growth, 1:5)4 -0.07421    0.07134  -1.040  0.2996
## L(rdpi_growth, 1:5)5 -0.11382    0.06608  -1.723  0.0866 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7026 on 187 degrees of freedom
## Multiple R-squared: 0.1624, Adjusted R-squared: 0.1176
## F-statistic: 3.626 on 10 and 187 DF, p-value: 0.0001971
```

When looking at the ACF and PACF residual plot, we see a spike in the PACF of order 3 for Lag = 0.75. This could imply that there is some pattern in the residuals that we need to fix if we want to improve our model. We have a discrepancy in the MSE, but the AIC and BIC both favor the order 3 model. In the summary, we see that cons\_growth Lag 2 is significant in the order 3 model. In the order 5 model, we see that Lag 2 and 3 for cons\_growth are significant and for rdpi\_growth, Lag 1 is significant. We could make an ARDL model work, but it is probably better to go a different path for this.

##Question 5

```
#CCF
library(vars)
```

```
## Loading required package: MASS
```



```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##      select

## Loading required package: strucchange

## Loading required package: urca

##
## Attaching package: 'vars'

## The following object is masked from 'package:fable':
##
##      VAR
```

```
var_model <- VAR(my_dataframe, p = 3)
summary(var_model)
```

```
##
## VAR Estimation Results:
## =====
## Endogenous variables: gdp_growth, cons_growth, inv_growth, gov_growth, rdpi_growth
## Deterministic variables: const
## Sample size: 200
## Log Likelihood: -1290.717
## Roots of the characteristic polynomial:
## 0.749 0.6002 0.6002 0.5969 0.5969 0.5921 0.5921 0.5056 0.5056 0.4575 0.3221 0.3221 0.3215 0.3215 0.0
## Call:
## VAR(y = my_dataframe, p = 3)
##
## Estimation results for equation gdp_growth:
## =====
## gdp_growth = gdp_growth.l1 + cons_growth.l1 + inv_growth.l1 + gov_growth.l1 + rdpi_growth.l1 + gdp_g
##
##              Estimate Std. Error t value Pr(>|t|)
## gdp_growth.l1 -0.327142   0.249547  -1.311 0.191510
## cons_growth.l1  0.538308   0.160488   3.354 0.000967 ***
## inv_growth.l1   0.057243   0.039787   1.439 0.151923
## gov_growth.l1   0.130382   0.070647   1.846 0.066566 .
## rdpi_growth.l1  0.136161   0.099127   1.374 0.171237
## gdp_growth.l2 -0.148630   0.253734  -0.586 0.558746
## cons_growth.l2  0.360259   0.172537   2.088 0.038174 *
## inv_growth.l2   0.015300   0.040257   0.380 0.704345
## gov_growth.l2   0.056720   0.077819   0.729 0.467006
## rdpi_growth.l2 -0.156328   0.096031  -1.628 0.105257
## gdp_growth.l3 -0.105197   0.253998  -0.414 0.679236
## cons_growth.l3  0.030797   0.165555   0.186 0.852630
## inv_growth.l3 -0.001185   0.039394  -0.030 0.976036
```

```

## gov_growth.l3 -0.024442 0.070833 -0.345 0.730437
## rdpi_growth.l3 0.037180 0.092788 0.401 0.689106
## const 0.317244 0.151007 2.101 0.037016 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.8872 on 184 degrees of freedom
## Multiple R-Squared: 0.2248, Adjusted R-squared: 0.1616
## F-statistic: 3.557 on 15 and 184 DF, p-value: 2.144e-05
##
##
## Estimation results for equation cons_growth:
## =====
## cons_growth = gdp_growth.l1 + cons_growth.l1 + inv_growth.l1 + gov_growth.l1 + rdpi_growth.l1 + gdp_
##
## Estimate Std. Error t value Pr(>|t|)
## gdp_growth.l1 -0.006204 0.203548 -0.030 0.97572
## cons_growth.l1 0.023891 0.130905 0.183 0.85539
## inv_growth.l1 0.022341 0.032453 0.688 0.49207
## gov_growth.l1 -0.011171 0.057624 -0.194 0.84650
## rdpi_growth.l1 0.142894 0.080855 1.767 0.07884 .
## gdp_growth.l2 -0.302764 0.206963 -1.463 0.14520
## cons_growth.l2 0.399695 0.140733 2.840 0.00502 **
## inv_growth.l2 0.030381 0.032837 0.925 0.35607
## gov_growth.l2 0.005301 0.063474 0.084 0.93354
## rdpi_growth.l2 -0.037296 0.078329 -0.476 0.63454
## gdp_growth.l3 -0.482083 0.207178 -2.327 0.02106 *
## cons_growth.l3 0.329933 0.135039 2.443 0.01550 *
## inv_growth.l3 0.051852 0.032133 1.614 0.10831
## gov_growth.l3 0.153977 0.057776 2.665 0.00838 **
## rdpi_growth.l3 0.003349 0.075684 0.044 0.96476
## const 0.585994 0.123172 4.758 3.95e-06 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.7237 on 184 degrees of freedom
## Multiple R-Squared: 0.2349, Adjusted R-squared: 0.1726
## F-statistic: 3.767 on 15 and 184 DF, p-value: 8.365e-06
##
##
## Estimation results for equation inv_growth:
## =====
## inv_growth = gdp_growth.l1 + cons_growth.l1 + inv_growth.l1 + gov_growth.l1 + rdpi_growth.l1 + gdp_g
##
## Estimate Std. Error t value Pr(>|t|)
## gdp_growth.l1 -1.69762 1.15506 -1.470 0.14335
## cons_growth.l1 4.10897 0.74284 5.531 1.08e-07 ***
## inv_growth.l1 0.19365 0.18416 1.052 0.29440
## gov_growth.l1 0.09670 0.32700 0.296 0.76777
## rdpi_growth.l1 0.07855 0.45882 0.171 0.86426
## gdp_growth.l2 0.30684 1.17444 0.261 0.79418
## cons_growth.l2 1.07585 0.79861 1.347 0.17959

```

```

## inv_growth.l2 -0.03993    0.18634   -0.214   0.83057
## gov_growth.l2  0.07770    0.36019    0.216   0.82945
## rdpi_growth.l2 -1.27077    0.44449   -2.859   0.00474 **
## gdp_growth.l3 -0.07160    1.17567   -0.061   0.95150
## cons_growth.l3 -0.67763    0.76630   -0.884   0.37769
## inv_growth.l3 -0.04190    0.18234   -0.230   0.81850
## gov_growth.l3 -0.31315    0.32786   -0.955   0.34076
## rdpi_growth.l3  0.30034    0.42948    0.699   0.48525
## const         -0.94715    0.69896   -1.355   0.17705
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.107 on 184 degrees of freedom
## Multiple R-Squared:  0.3446, Adjusted R-squared:  0.2912
## F-statistic: 6.451 on 15 and 184 DF, p-value: 5.94e-11
##
##
## Estimation results for equation gov_growth:
## =====
## gov_growth = gdp_growth.l1 + cons_growth.l1 + inv_growth.l1 + gov_growth.l1 + rdpi_growth.l1 + gdp_g
##
##
##           Estimate Std. Error t value Pr(>|t|)
## gdp_growth.l1 -0.786910   0.381087  -2.065  0.04033 *
## cons_growth.l1  0.106145   0.245083   0.433  0.66545
## inv_growth.l1  0.161247   0.060759   2.654  0.00865 **
## gov_growth.l1  0.733794   0.107886   6.802  1.4e-10 ***
## rdpi_growth.l1  0.189456   0.151378   1.252  0.21233
## gdp_growth.l2  0.248248   0.387480   0.641  0.52253
## cons_growth.l2 -0.234014   0.263484  -0.888  0.37562
## inv_growth.l2 -0.023890   0.061478  -0.389  0.69803
## gov_growth.l2  0.034805   0.118838   0.293  0.76994
## rdpi_growth.l2 -0.035318   0.146650  -0.241  0.80995
## gdp_growth.l3  0.016366   0.387883   0.042  0.96639
## cons_growth.l3  0.130811   0.252822   0.517  0.60550
## inv_growth.l3 -0.003319   0.060160  -0.055  0.95607
## gov_growth.l3  0.022092   0.108170   0.204  0.83839
## rdpi_growth.l3 -0.043512   0.141698  -0.307  0.75913
## const         0.333643   0.230605   1.447  0.14965
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.355 on 184 degrees of freedom
## Multiple R-Squared:  0.4714, Adjusted R-squared:  0.4283
## F-statistic: 10.94 on 15 and 184 DF, p-value: < 2.2e-16
##
##
## Estimation results for equation rdpi_growth:
## =====
## rdpi_growth = gdp_growth.l1 + cons_growth.l1 + inv_growth.l1 + gov_growth.l1 + rdpi_growth.l1 + gdp_
##
##
##           Estimate Std. Error t value Pr(>|t|)
## gdp_growth.l1  0.30000    0.23549   1.274  0.2043

```

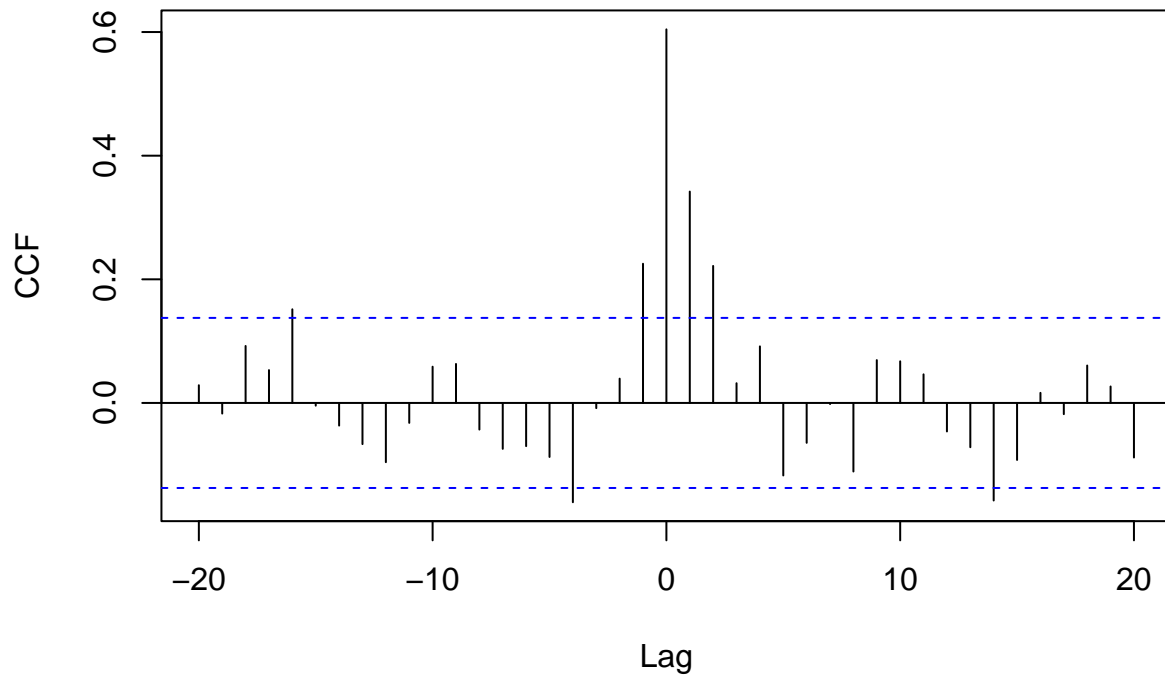
```

## cons_growth.l1  0.13852    0.15145    0.915    0.3616
## inv_growth.l1   -0.04020    0.03755   -1.071    0.2857
## gov_growth.l1   -0.09610    0.06667   -1.441    0.1512
## rdpi_growth.l1  -0.09789    0.09354   -1.046    0.2967
## gdp_growth.l2   -0.26267    0.23944   -1.097    0.2741
## cons_growth.l2   0.19998    0.16282    1.228    0.2209
## inv_growth.l2    0.07373    0.03799    1.941    0.0538 .
## gov_growth.l2    0.08814    0.07343    1.200    0.2316
## rdpi_growth.l2  -0.08477    0.09062   -0.935    0.3508
## gdp_growth.l3   -0.17473    0.23969   -0.729    0.4669
## cons_growth.l3   0.10545    0.15623    0.675    0.5005
## inv_growth.l3    0.03175    0.03717    0.854    0.3942
## gov_growth.l3    0.03543    0.06684    0.530    0.5967
## rdpi_growth.l3  -0.01228    0.08756   -0.140    0.8886
## const           0.66342    0.14250    4.656 6.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.8372 on 184 degrees of freedom
## Multiple R-Squared:  0.152,    Adjusted R-squared:  0.08285
## F-statistic: 2.199 on 15 and 184 DF,  p-value: 0.007804
##
##
## Covariance matrix of residuals:
##
##      gdp_growth cons_growth inv_growth gov_growth rdpi_growth
## gdp_growth      0.7872    0.40394    2.8512    0.35648    0.4287
## cons_growth      0.4039    0.52373    0.7384   -0.02042    0.3312
## inv_growth       2.8512    0.73840   16.8648   -0.13669    1.3042
## gov_growth        0.3565   -0.02042   -0.1367    1.83576    0.1234
## rdpi_growth       0.4287    0.33125    1.3042    0.12344    0.7010
##
## Correlation matrix of residuals:
##
##      gdp_growth cons_growth inv_growth gov_growth rdpi_growth
## gdp_growth      1.0000    0.62911    0.78253    0.29654    0.5771
## cons_growth      0.6291    1.00000    0.24846   -0.02083    0.5467
## inv_growth       0.7825    0.24846    1.00000   -0.02457    0.3793
## gov_growth       0.2965   -0.02083   -0.02457    1.00000    0.1088
## rdpi_growth      0.5771    0.54670    0.37931    0.10882    1.0000

ccf_plot <- ccf(data.frame(my_dataframe)$gdp_growth, data.frame(my_dataframe)$cons_growth, main = "Cross-correlation plot of GDP growth and consumption growth")

```

## Cross-Correlation Function (CCF)



We see that there is very visible cross-correlation near the middle and when looking at the graph, it does exhibit a specific pattern.

### *#Granger Test*

```
granger_test <- causality(var_model)
print(granger_test)
```

```
## $Granger
```

```
##
```

```
## Granger causality H0: gdp_growth do not Granger-cause cons_growth
```

```
## inv_growth gov_growth rdpi_growth
```

```
##
```

```
## data: VAR object var_model
```

```
## F-Test = 1.8528, df1 = 12, df2 = 920, p-value = 0.03655
```

```
##
```

```
##
```

```
## $Instant
```

```
##
```

```
## H0: No instantaneous causality between: gdp_growth and cons_growth
```

```
## inv_growth gov_growth rdpi_growth
```

```
##
```

```
## data: VAR object var_model
```

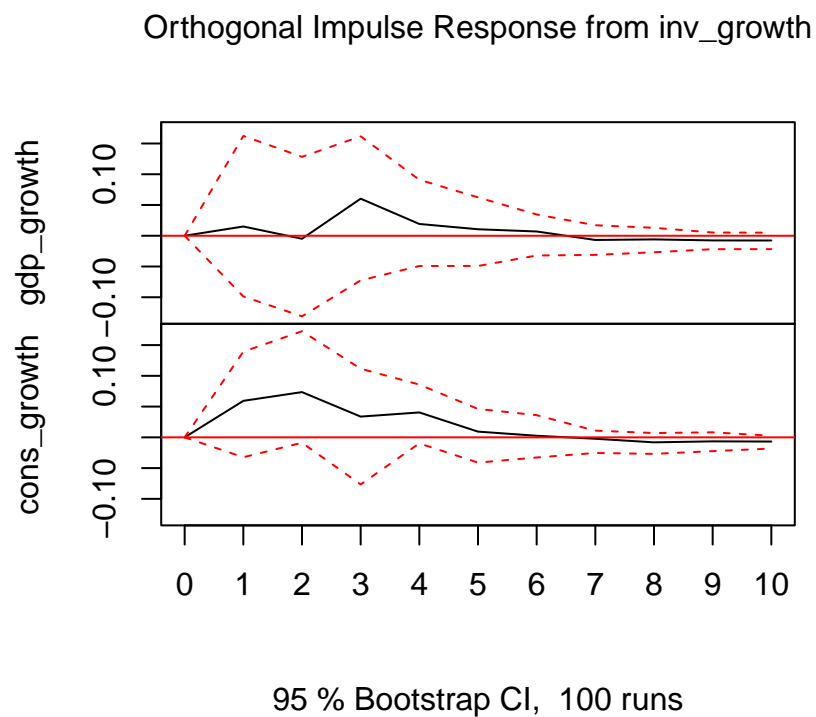
```
## Chi-squared = 95.798, df = 4, p-value < 2.2e-16
```

```
#reject null hypothesis
```

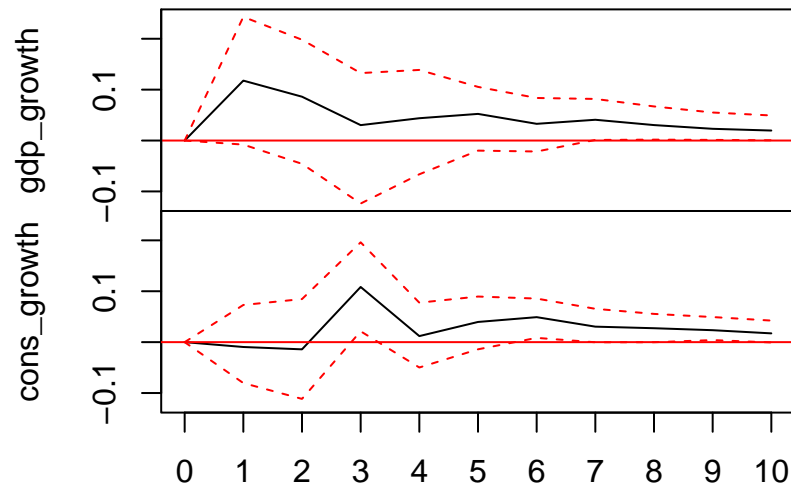
We can reject the null hypothesis and say that there is causality in some form and one of the predictor variables does Granger cause gdp\_growth. We can also reject the null hypothesis of instantaneous causality between gdp\_growth, cons\_growth, and the predictor variables.

```
#IRF Test
```

```
irf_result <- irf(var_model, impulse = c("inv_growth", "gov_growth", "rdpi_growth"), response = c("gdp_growth", "cons_growth"), nboot = 1000, nrep = 1000)
plot(irf_result)
```

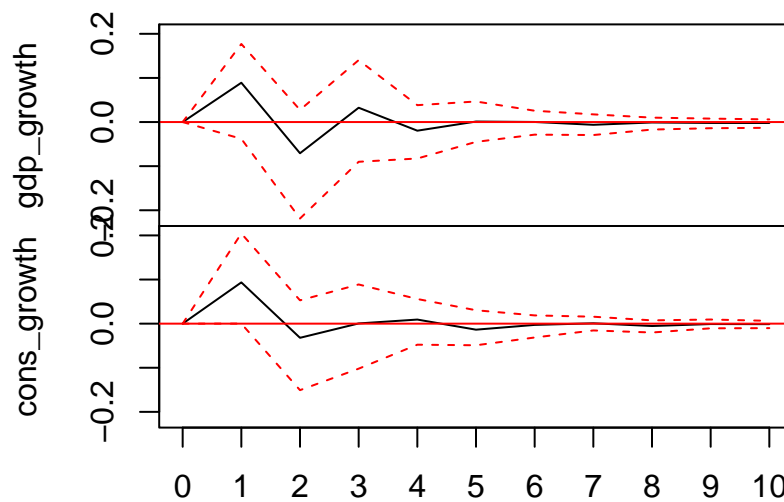


Orthogonal Impulse Response from gov\_growth



95 % Bootstrap CI, 100 runs

### Orthogonal Impulse Response from rdpi\_growth



95 % Bootstrap CI, 100 runs

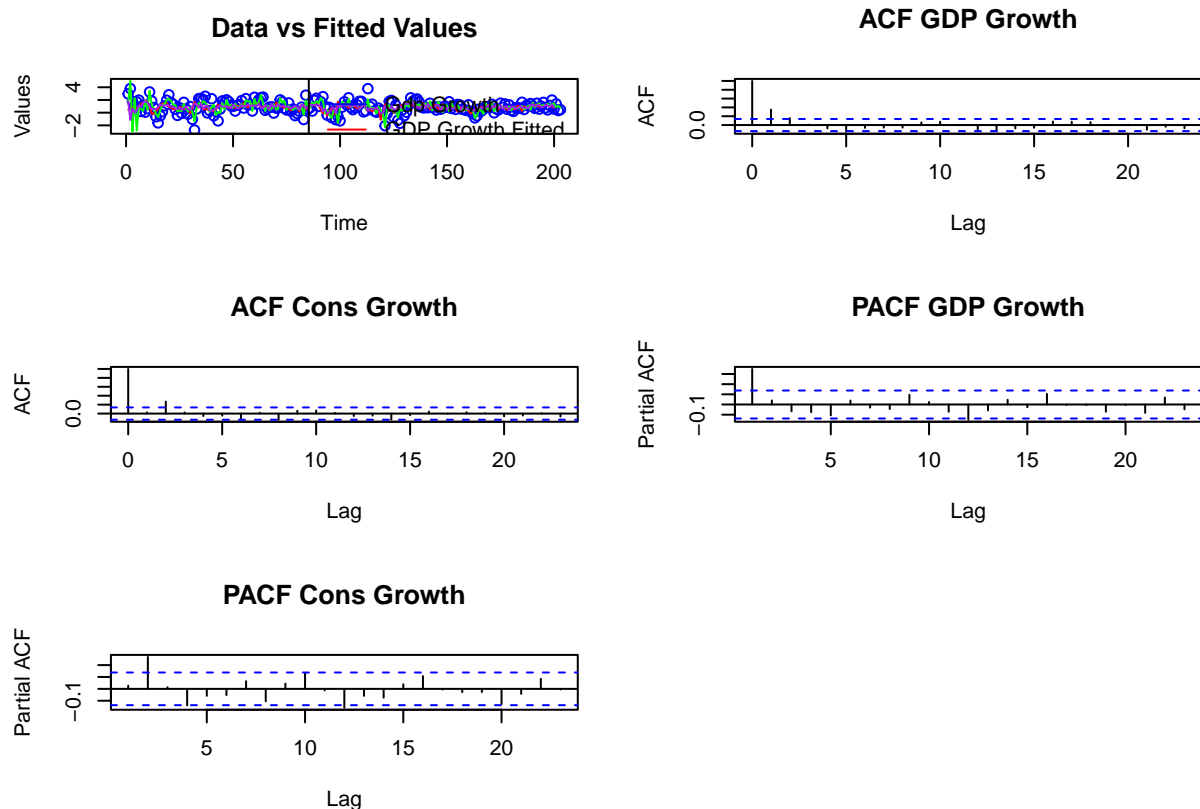
The first comparison is inv\_growth. Early on, inv\_growth has a pretty sizable impact on cons\_growth, but later on, it falls off. As for gdp\_growth, inv\_growth doesn't really seem to have too much of an effect on that. Gov\_growth on the other hand seems to have a pretty sizable impact on gdp\_growth, but the same can't be said for cons\_growth. At the start, rdpi\_growth has a noticeable effect on cons and gdp\_growth, but it becomes irrelevant later on.

```
#Plot of fitted value, data, P/ACF
fit <- data.frame(fitted(var_model))

par(mfrow = c(3, 2))
plot(data.frame(my_dataframe)$gdp_growth, col = "blue", ylim = range(c(data.frame(my_dataframe)$gdp_growth, fit$gdp_growth)))
lines(fit$gdp_growth, col = "red")
lines(data.frame(my_dataframe)$cons_growth, col = "green")
lines(fit$cons_growth, col = "purple")
legend("topright", legend = c("Gdp Growth", "GDP Growth Fitted", "Cons Growth", "Cons Growth Fitted"), col = c("blue", "red", "green", "purple"))

acf(data.frame(my_dataframe)$gdp_growth, main = "ACF GDP Growth")
acf(data.frame(my_dataframe)$cons_growth, main = "ACF Cons Growth")
pacf(data.frame(my_dataframe)$gdp_growth, main = "PACF GDP Growth")
pacf(data.frame(my_dataframe)$cons_growth, main = "PACF Cons Growth")
```





```
#Training, testing, MSE, AIC, BIC
library(forecast)
train_size <- round(2/3 * length(my_dataframe[, 1]))

# Split the data
train_data <- my_dataframe[, 1][1:train_size]
test_data <- my_dataframe[, 1][(train_size + 1):length(my_dataframe[, 1])]

# Forecasting

forecast_1 <- predict(var_model, n.ahead=length(test_data))
predictions <- forecast_1$endog
mse_1 <- mean((predictions - test_data)^2)
cat("MSE for Model 1:", mse_1, "\n")
```

```
## MSE for Model 1: 6.20567
```

```
forecast_2 <- predict(var_model, n.ahead=length(train_data))
predictions <- forecast_2$endog
mse_1 <- mean((predictions - train_data)^2)
cat("MSE for Model 2:", mse_1, "\n")
```

```
## MSE for Model 2: 6.465269
```

```
aic <- AIC(var_model)
bic <- BIC(var_model)
cat("AIC:", aic, "\n")
```

```
## AIC: 2741.433
```

```
cat("BIC:", bic, "\n")
```

```
## BIC: 3005.299
```

```
forecast_10 <- predict(var_model, n.ahead=10)
```

```
print("10-step ahead forecast for VAR Model:")
```

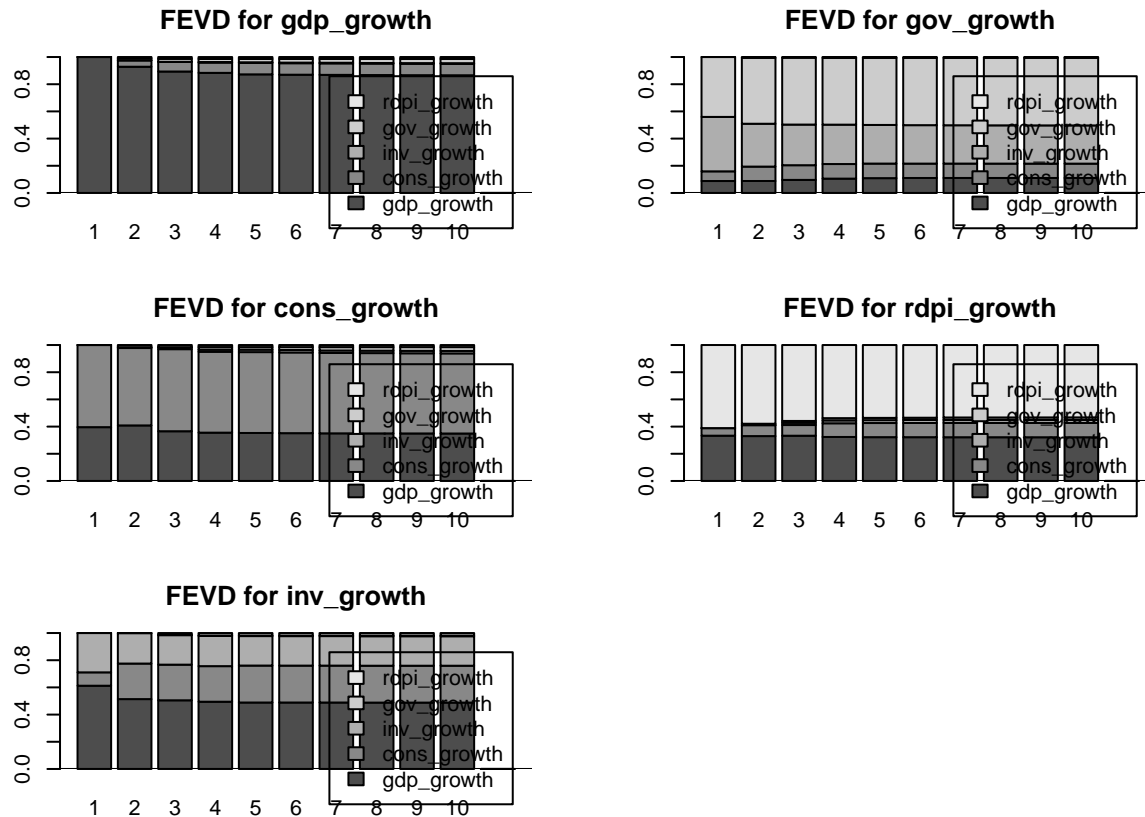
```
## [1] "10-step ahead forecast for VAR Model:"
```

```
print(forecast_10)
```

```
## $gdp_growth
##          fcst      lower    upper      CI
## [1,] 0.8868831 -0.8520580 2.625824 1.738941
## [2,] 0.9432692 -0.9355022 2.822041 1.878771
## [3,] 0.9514864 -0.9827521 2.885725 1.934238
## [4,] 0.9098127 -1.0354214 2.855047 1.945234
## [5,] 0.8998722 -1.0581807 2.857925 1.958053
## [6,] 0.8637192 -1.0974901 2.824929 1.961209
## [7,] 0.8590185 -1.1034566 2.821494 1.962475
## [8,] 0.8519285 -1.1122600 2.816117 1.964188
## [9,] 0.8471096 -1.1181404 2.812360 1.965250
## [10,] 0.8469978 -1.1188821 2.812878 1.965880
##
## $cons_growth
##          fcst      lower    upper      CI
## [1,] 1.0230392 -0.3953643 2.441443 1.418404
## [2,] 0.9249635 -0.5370890 2.387016 1.462052
## [3,] 1.0283757 -0.5208894 2.577641 1.549265
## [4,] 0.9324602 -0.6400532 2.504974 1.572513
## [5,] 0.9177334 -0.6601538 2.495621 1.577887
## [6,] 0.9155762 -0.6665741 2.497727 1.582150
## [7,] 0.8975215 -0.6875778 2.482621 1.585099
## [8,] 0.8904551 -0.6957761 2.476686 1.586231
## [9,] 0.8898632 -0.6974235 2.477150 1.587287
## [10,] 0.8872080 -0.7009046 2.475321 1.588113
##
## $inv_growth
##          fcst      lower    upper      CI
## [1,] 1.088528 -6.960418  9.137475 8.048947
## [2,] 1.464704 -7.790297 10.719705 9.255001
## [3,] 1.615370 -7.796710 11.027449 9.412079
## [4,] 1.486736 -8.085332 11.058805 9.572068
## [5,] 1.295868 -8.390964 10.982701 9.686832
```

```
## [6,] 1.068443 -8.629503 10.766388 9.697946
## [7,] 1.133810 -8.570008 10.837629 9.703819
## [8,] 1.077444 -8.630832 10.785720 9.708276
## [9,] 1.061879 -8.647446 10.771205 9.709326
## [10,] 1.084003 -8.625726 10.793733 9.709730
##
## $gov_growth
##      fcst      lower      upper      CI
## [1,] 0.6411644 -2.014398 3.296727 2.655562
## [2,] 0.5794947 -2.561200 3.720189 3.140695
## [3,] 0.5506427 -2.781802 3.883087 3.332444
## [4,] 0.6309223 -2.813299 4.075144 3.444222
## [5,] 0.6415121 -2.858158 4.141182 3.499670
## [6,] 0.6505006 -2.873837 4.174838 3.524338
## [7,] 0.6456541 -2.891103 4.182411 3.536757
## [8,] 0.6472711 -2.895238 4.189780 3.542509
## [9,] 0.6428318 -2.903448 4.189112 3.546280
## [10,] 0.6414250 -2.906823 4.189673 3.548248
##
## $rdpi_growth
##      fcst      lower      upper      CI
## [1,] 0.7445567 -0.8964149 2.385528 1.640972
## [2,] 0.8703935 -0.8232853 2.564072 1.693679
## [3,] 0.9167378 -0.8089776 2.642453 1.725715
## [4,] 0.9072558 -0.8514182 2.665930 1.758674
## [5,] 0.8910277 -0.8718701 2.653926 1.762898
## [6,] 0.8970720 -0.8680313 2.662175 1.765103
## [7,] 0.8775619 -0.8904380 2.645562 1.768000
## [8,] 0.8606746 -0.9077229 2.629072 1.768397
## [9,] 0.8641167 -0.9043865 2.632620 1.768503
## [10,] 0.8622324 -0.9066621 2.631127 1.768894
```

```
#FEVD
fevd_result <- fevd(var_model, n.ahead = 10)
par(mar = c(3, 3, 3, 3))
plot(fevd_result)
```

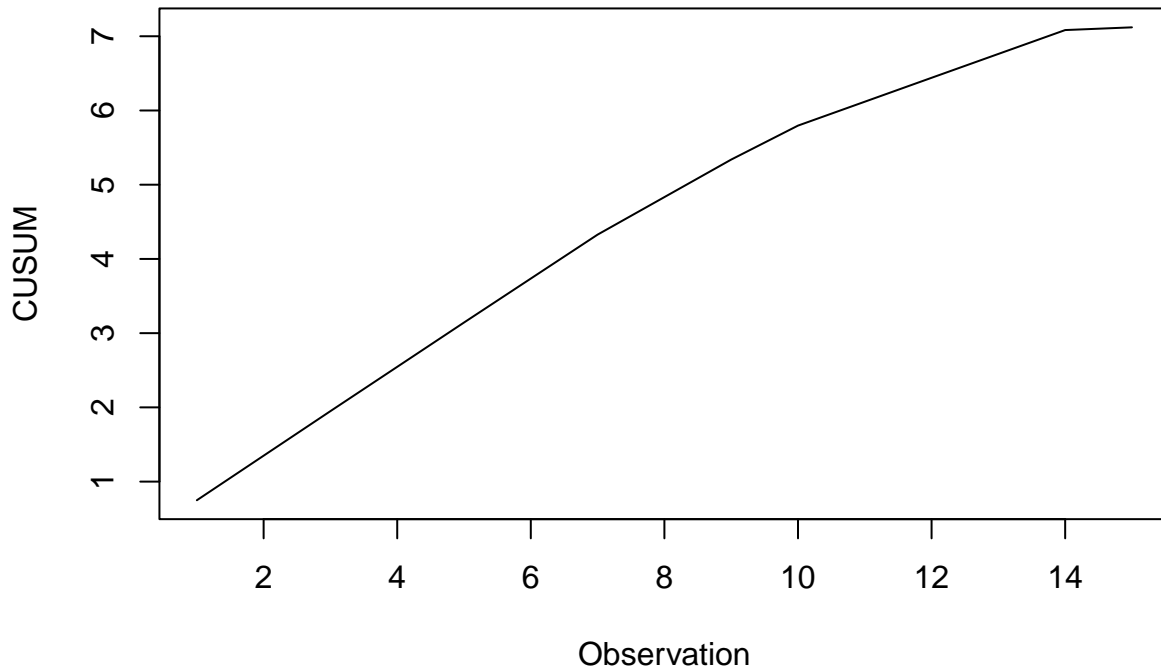


*#due to margins, couldn't print out full plot, so the #legend does override some of the graph*

When looking at `gdp_growth`, it is almost entirely affected by `gdp_growth` and nothing else. This does align with #4 where we saw that for most models, an `AR()` might be better. For the `cons_growth` graph, we see a high reliance on `cons_growth` and `gdp_growth`. When modeling `cons_growth`, we could add a `gdp_growth` component to it if we like. As for the other graphs, they're not really that important since we're seeing what affects `cons_growth` and `gdp_growth`, but if we were using those as our predictors, it may be worth checking out.

```
#CUSUM
par(mfrow=c(1,1))
residuals <- roots(var_model)
cusum <- cumsum(residuals)
plot(cusum, type = "l", xlab = "Observation", ylab = "CUSUM", main = "CUSUM of Recursive Residuals")
abline(h = 0, col = "red")
```

## CUSUM of Recursive Residuals



CUSUM starts off with a rapid increase and then it slows down as time goes on. It kind of looks like a square root function, but it is hard to tell unless the graph is bigger. This means the process is starting to stabilize and is adapting well to new observations.

### #Part 6

When looking at the initial plots of the data, we decided upon 3 and 5 because it gave a lot of room to work with and there was a possible fit to work with. When looking at Question 3, we see for both the `gdp_growth` and `consumption_growth`, it can be better presented by an  $AR(3)$  model as opposed to an  $AR(5)$ . When looking at Question 4, we see that `gdp_growth` doesn't really do well with other variables, so regardless of whatever we chose, the  $AR(3)$  from Question 3 was better. However, for predicting `cons_growth`, we see that  $ARDL(3, 3)$  with `cons_growth` and `gdp_growth` were pretty effective and the FEVD model in Question 5 also backs this up. When comparing  $ARDL(3, 3)$  vs  $AR(3)$  for the `cons_growth` model, we decided upon the  $ARDL(3, 3)$  just because we felt it was important to note the importance of `gdp_growth`. In hindsight, it may have been better to use  $ARDL(1, 1)$ . While the `var_models` were nice to work with, we felt like this dataset wasn't really meant for the `var_model` since the variables weren't really related to each other enough to justify using one.