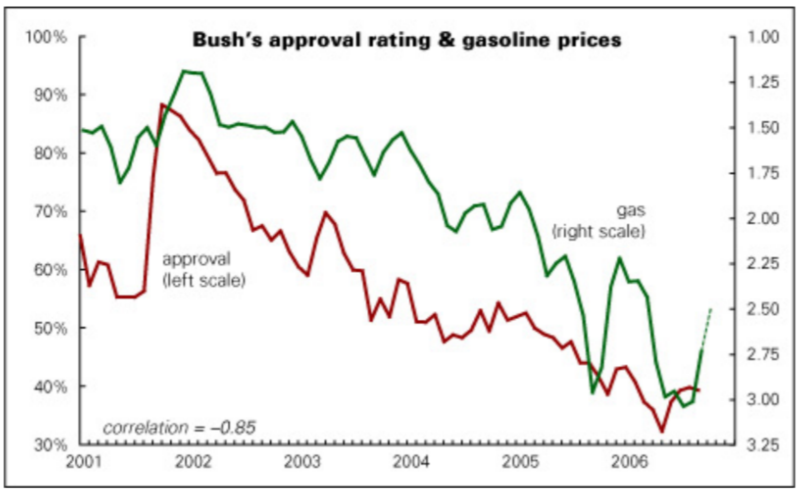
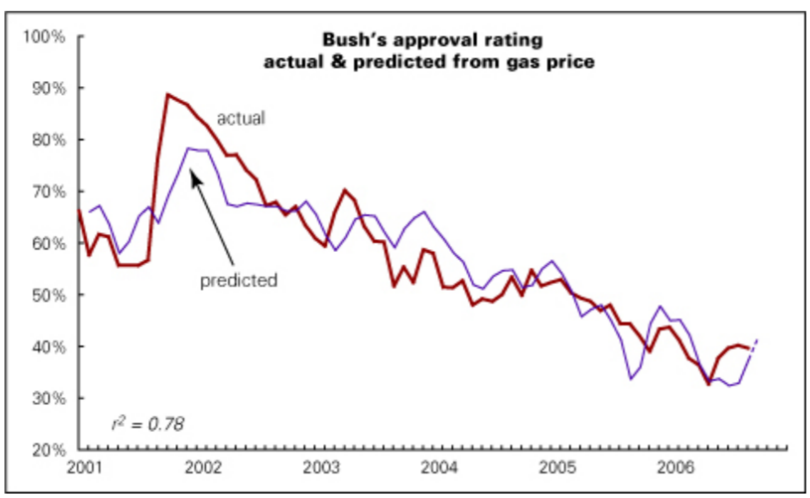
6.1.1 - R: Time Series

Stat 5100: Dr. Bean

Example 1: Bush and the Price of Gas

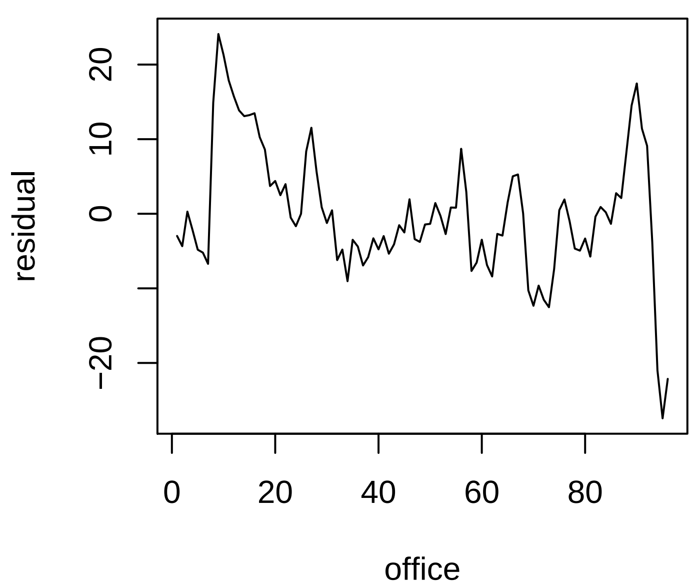
1. http://www.leftbusinessobserver.com/BushNGas.html
2. "... no occupant of the White House has ever seen his popularity so closely tied to the price of gas."
3. "There's no precedent for this tight relationship."





But - can we justify a conclusion that gas price significantly affects approval rating? (HW 7 will address this more completely).

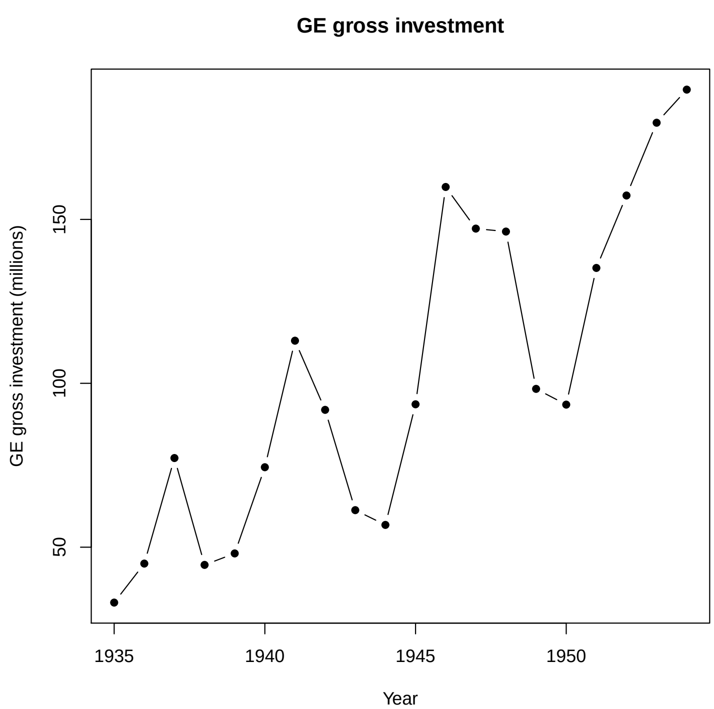
```r
library(stat5100)

data(bush_gas)

tlm <- lm(rating ~ price, data = bush_gas)
summary(tlm)

##
## Call:
## lm(formula = rating ~ price, data = bush_gas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.423  -4.779  -1.294   3.779  24.099
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 88.80015    2.82573   31.43   <2e-16 ***
## price       -0.18281    0.01242  -14.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.77 on 94 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.6975,Adjusted R-squared:  0.6943
## F-statistic: 216.7 on 1 and 94 DF,  p-value: < 2.2e-16

plot(bush_gas$office[!is.na(bush_gas$rating) & !is.na(bush_gas$price)],
     tlm$residuals, type = "l", xlab = "office", ylab = "residual")
```



Example 2: General Electric's gross investment (in millions of dollars) for years 1935 - 1954.

Originally presented in Grunfeld, Y. (1958), "The Determinants of Corporate Investment," Ph.D. dissertation, University of Chicago; discussed in Boot, J.C.G. (1960), "Investment Demand: An Empirical Contribution to the Aggregation Problem," International Economic Review, 1, 3-30. See also Damodar N. Gujarati, Basic Econometrics, Third Edition, 1995, McGraw-Hill, [1995, pp. 522-525].

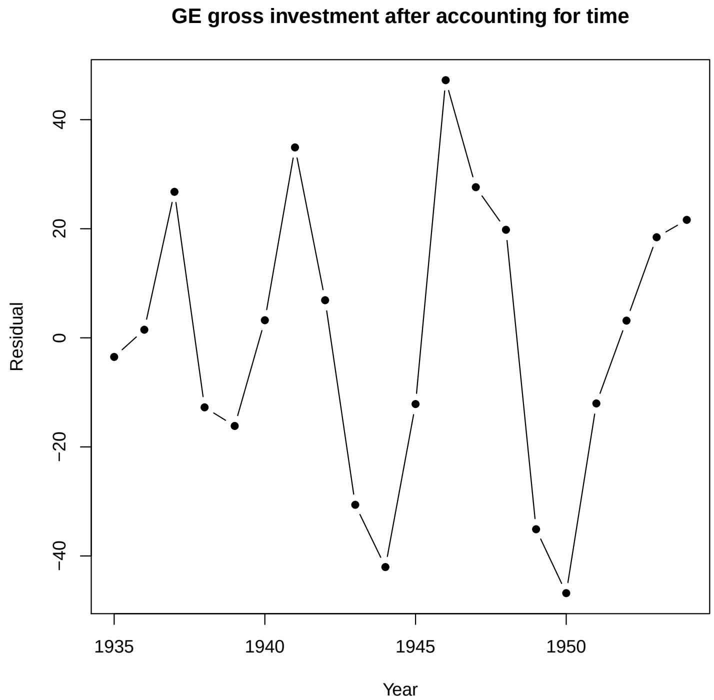
```r
# Load the GE data
data(ge)
head(ge)

##   year GEinv
## 1 1935  33.1
## 2 1936  45.0
## 3 1937  77.2
## 4 1938  44.6
## 5 1939  48.1
## 6 1940  74.4

# Create a line plot of GE's gross investment each year.
plot(ge$year, ge$GEinv, main = "GE gross investment", xlab = "Year",
     ylab = "GE gross investment (millions)", type = "b", pch = 16)
```



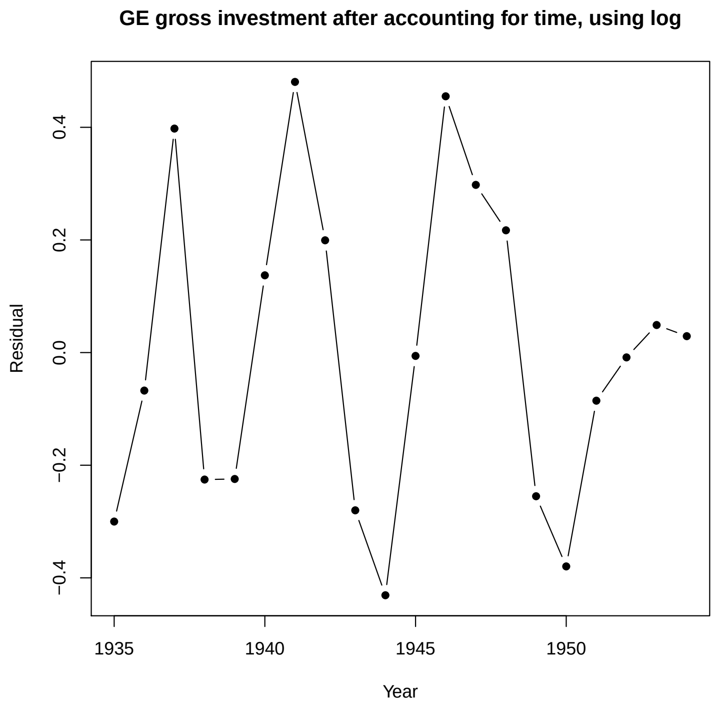
(1) Make the data stationary

```r
# Create a regression model predicting GE investment from year. Next we will
# examine the residuals (residuals represent the structure after accounting
# for the time dependence)
ge_time_lm <- lm(GEinv ~ year, data = ge)

plot(ge$year, ge_time_lm$residuals, xlab = "Year", ylab = "Residual",
```

```r
     main = "GE gross investment after accounting for time", type = "b", pch = 16)
```

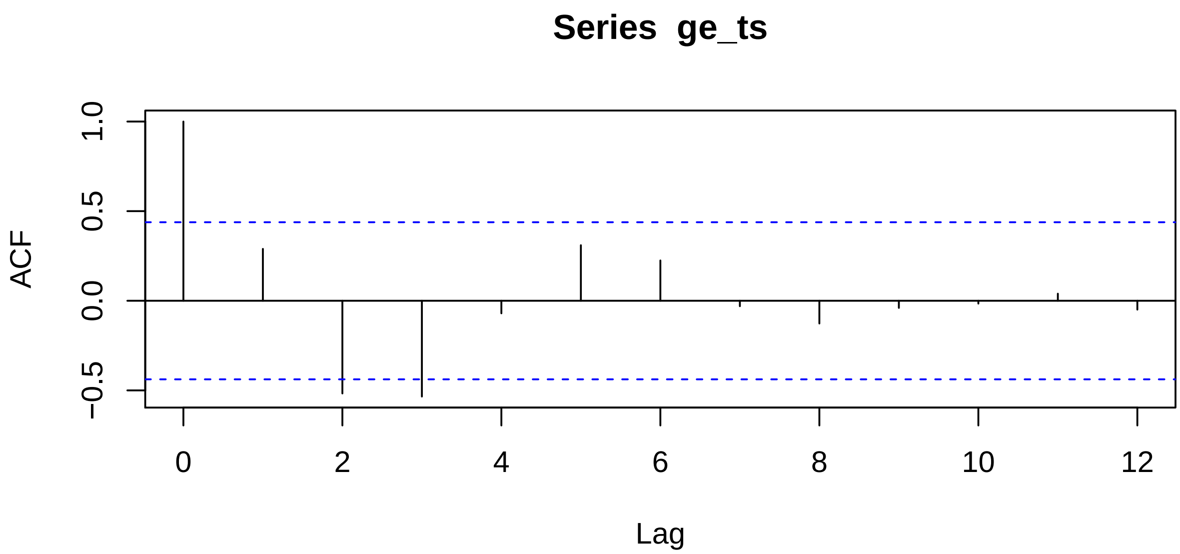


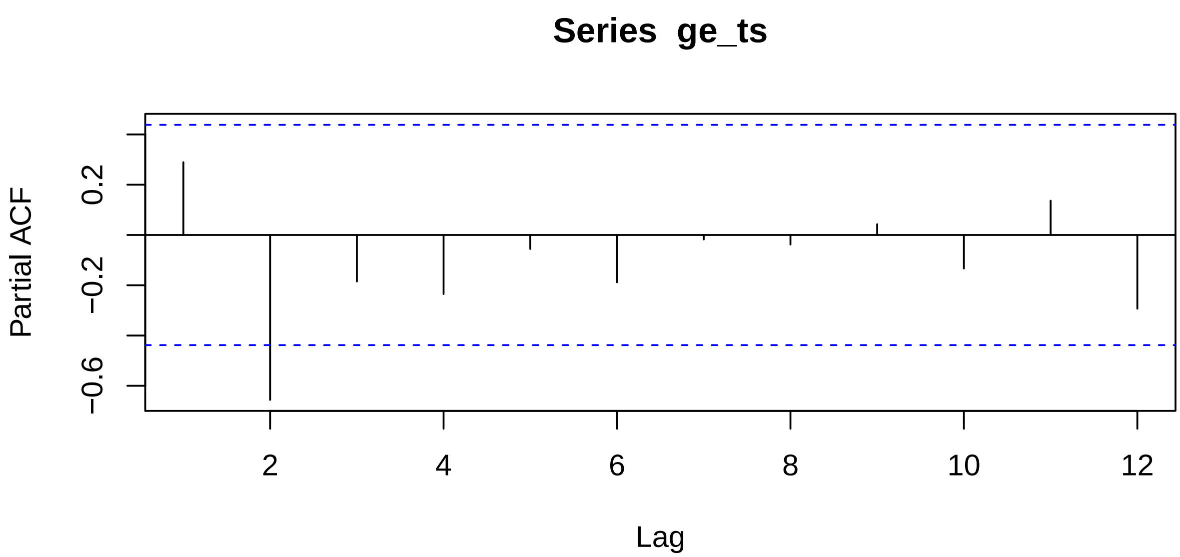

```r
# Alternatively, we can make the data stationary after transforming the
# response variable with a log transformation.
ge <- cbind(ge, logGEinv = log(ge$GEinv))
ge_time_log_lm <- lm(logGEinv ~ year, data = ge)

plot(ge$year, ge_time_log_lm$residuals, xlab = "Year", ylab = "Residual",
     main = "GE gross investment after accounting for time, using log",
     type = "b", pch = 16)
```



(2) Test for independence and (3) investigate potential dependence structures

```
# Create a time-series object for our data
ge_ts <- ts(ge_time_log_lm$residuals)

# Sample Autocorrelation Plot (ACF) / Sample Partial Autocorrelation Plots (PACF)
par(mfrow = c(2, 1))
acf(ge_ts, lag.max = 12)
pacf(ge_ts, lag.max = 12)
```



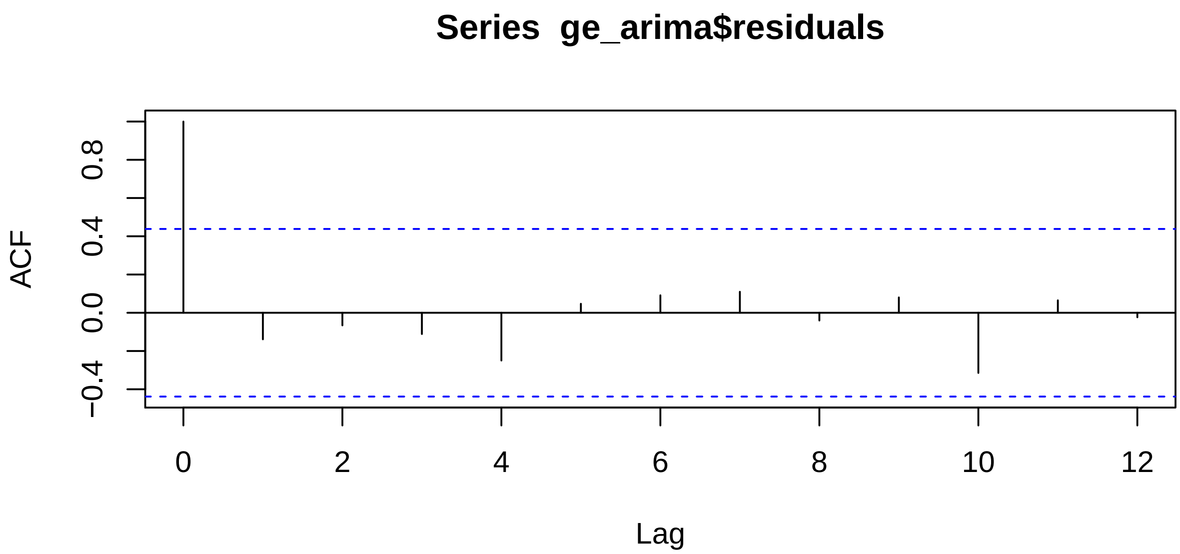


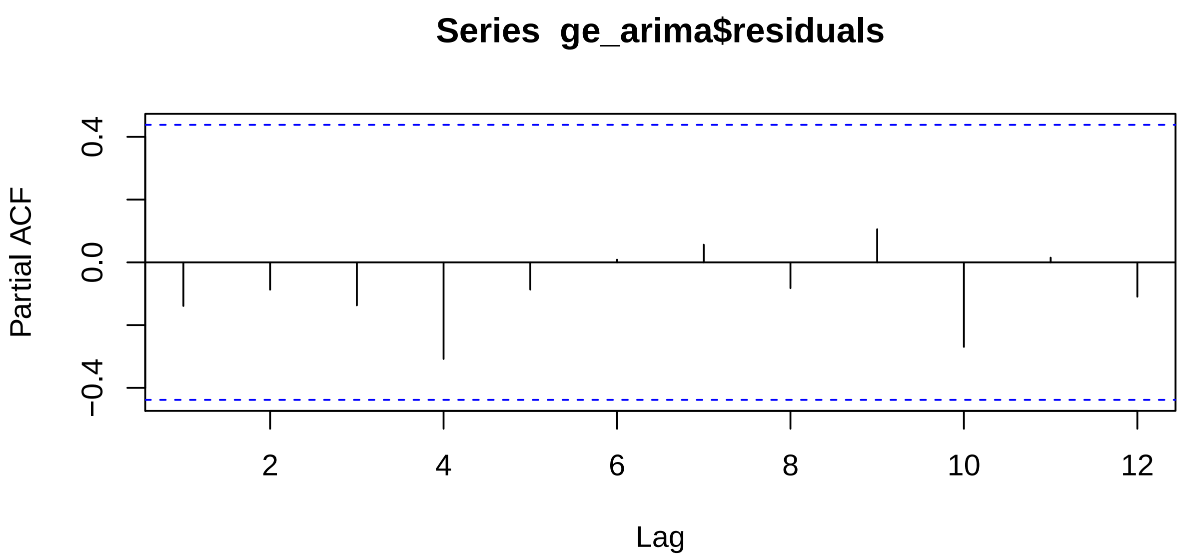

```
par(mfrow = c(1, 1))

# Autocorrelation check for white noise
Box.test(ge_ts, lag = 6, type = "Ljung")

##
##  Box-Ljung test
##
## data:  ge_ts
## X-squared = 20.425, df = 6, p-value = 0.002326

Box.test(ge_ts, lag = 12, type = "Ljung")

##
##  Box-Ljung test
##
```

```
## data:  ge_ts
## X-squared = 21.331, df = 12, p-value = 0.04574
```

(4) Fit a dependence structure and assess model adequacy

```
# Fit the original (log transformed) data (not the residuals) and
# include a drift term which will fit a linear time dependent trend.
ge_ts <- ts(ge$logGEinv)
ge_arima <- forecast::Arima(ge_ts, order = c(2, 0, 0), include.drift = TRUE)

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

summary(ge_arima)

## Series: ge_ts
## ARIMA(2,0,0) with drift
##
## Coefficients:
##          ar1      ar2  intercept   drift
##       0.4871  -0.6507     3.7527  0.0722
## s.e.  0.1668   0.1539     0.0836  0.0071
##
## sigma^2 estimated as 0.04461:  log likelihood=4.35
## AIC=1.29   AICc=5.58   BIC=6.27
##
## Training set error measures:
##                        ME      RMSE      MAE       MPE     MAPE      MASE
## Training set -0.004326995 0.1889213 0.143847 -0.321654 3.312231 0.5164973
##                    ACF1
## Training set -0.1389596

# Note that forecast::Arima does not automatically provide p-values.
# The lmtest package will provide these for you.
lmtest::coeftest(ge_arima)

##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar1        0.4870567  0.1667691  2.9205  0.003494 **
## ar2       -0.6507009  0.1538594 -4.2292 2.345e-05 ***
## intercept  3.7526822  0.0836412 44.8664 < 2.2e-16 ***
## drift      0.0721521  0.0070884 10.1788 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Create a panel of plots to diagnose the results of the ARIMA predictions
par(mfrow = c(2, 1))
acf(ge_arima$residuals, lag.max = 12)
pacf(ge_arima$residuals, lag.max = 12)
```






```
par(mfrow = c(1, 1))

# Autocorrelation check of residuals
Box.test(ge_arima$residuals, lag = 6, type = "Ljung")

##
##  Box-Ljung test
##
## data:  ge_arima$residuals
## X-squared = 2.9163, df = 6, p-value = 0.8193

Box.test(ge_arima$residuals, lag = 12, type = "Ljung")

##
##  Box-Ljung test
##
```

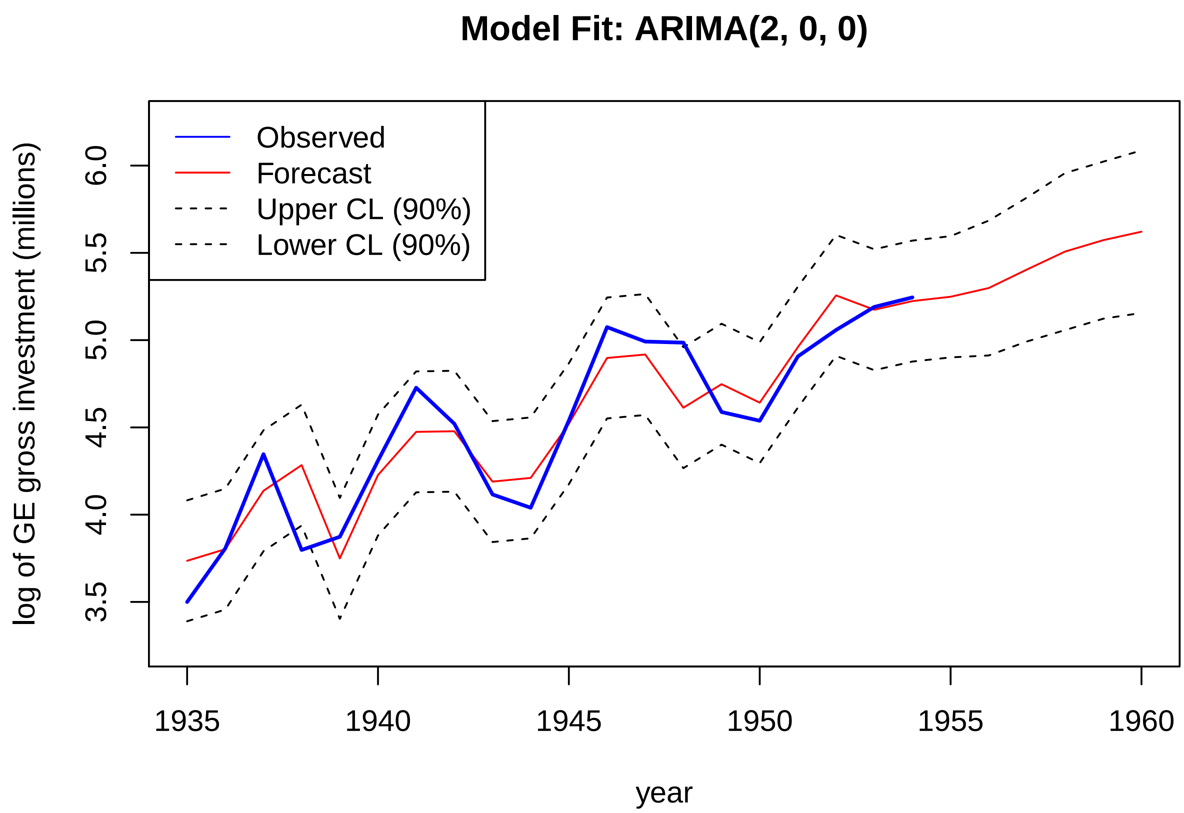
```
## data:  ge_arima$residuals
## X-squared = 8.2374, df = 12, p-value = 0.7663
```

```
Box.test(ge_arima$residuals, lag = 18, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  ge_arima$residuals
## X-squared = 12.576, df = 18, p-value = 0.8161
```

(5) Forecast

```
# 1.645 is the z-score associated with a 90 percent confidence interval
current <- data.frame(fit = as.numeric(ge_arima$fitted),
                      lower = as.numeric(ge_arima$fitted -
                                         1.64*sqrt(ge_arima$sigma2)),
                      upper = as.numeric(ge_arima$fitted +
                                         1.64*sqrt(ge_arima$sigma2)),
                      year = ge$year)

ahead <- forecast::forecast(ge_arima, h = 6, level = 90)
ahead <- data.frame(fit = as.numeric(ahead$mean),
                    lower = as.numeric(ahead$lower[, 1]),
                    upper = as.numeric(ahead$upper[ ,1]),
                    year = (max(ge$year) + 1):(max(ge$year) + 6))

final <- rbind(current, ahead)


plot(final$year, final$fit, col = "red", type = "l",
     xlab = "year", ylab = "log of GE gross investment (millions)",
     ylim = c(3.25, 6.25),
     main = "Model Fit: ARIMA(2, 0, 0)")
lines(final$year, final$lower, lty = 2)
lines(final$year, final$upper, lty = 2)
lines(ge$year, ge$logGEinv, lwd = 2, col = "blue")
legend("topleft", legend = c("Observed", "Forecast",
                             "Upper CL (90%)",
                             "Lower CL (90%)"),
       lty = c(1, 1, 2, 2),
       col = c("blue", "red", "black", "black"))
```



ARMA(1, 1) model (for comparison purposes only)

```
ge_arima_2 <- forecast::Arima(ge_ts, order = c(1, 0, 1), include.drift = TRUE)
summary(ge_arima_2)

## Series: ge_ts
## ARIMA(1,0,1) with drift
##
## Coefficients:
##           ar1     ma1  intercept   drift
##       -0.2775  1.0000     3.7411  0.0733
## s.e.   0.2234  0.2261     0.1503  0.0125
##
## sigma^2 estimated as 0.05709:  log likelihood=1.23
## AIC=7.54   AICc=11.83   BIC=12.52
##
## Training set error measures:
##                        ME      RMSE      MAE        MPE     MAPE      MASE
## Training set -0.002126457 0.2137022 0.167283 -0.3245719 3.847204 0.6006467
##                     ACF1
## Training set -0.02210677

lmtest::coeftest(ge_arima)

##
## z test of coefficients:
##
```

```
##              Estimate Std. Error z value  Pr(>|z|)    
## ar1        0.4870567  0.1667691  2.9205  0.003494 ** 
## ar2       -0.6507009  0.1538594 -4.2292 2.345e-05 ***
## intercept  3.7526822  0.0836412 44.8664 < 2.2e-16 ***
## drift      0.0721521  0.0070884 10.1788 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```