SKS104_MIDTERM2

Skasko_Stephen

11/15/2023

```r
data <- read.csv("amazon.csv")
data
```

```
##    Quarter Year    Sales
## 1        1 2000  573.889
## 2        2 2000  577.876
## 3        3 2000  637.858
## 4        4 2000  972.360
## 5        1 2001  700.356
## 6        2 2001  667.625
## 7        3 2001  639.281
## 8        4 2001 1115.171
## 9        1 2002  847.422
## 10       2 2002  805.605
## 11       3 2002  851.299
## 12       4 2002 1428.610
## 13       1 2003 1083.559
## 14       2 2003 1099.912
## 15       3 2003 1134.456
## 16       4 2003 1945.772
## 17       1 2004 1530.349
## 18       2 2004 1387.341
## 19       3 2004 1462.475
## 20       4 2004 2540.959
## 21       1 2005 1901.600
## 22       2 2005 1753.000
## 23       3 2005 1858.000
## 24       4 2005 2977.000
## 25       1 2006 2279.000
## 26       2 2006 2139.000
## 27       3 2006 2307.000
## 28       4 2006 3986.000
## 29       1 2007 3015.000
## 30       2 2007 2886.000
## 31       3 2007 3262.000
## 32       4 2007 5673.000
## 33       1 2008 4135.000
## 34       2 2008 4063.000
## 35       3 2008 4264.000
## 36       4 2008 6704.000
## 37       1 2009 4889.000
```

```
## 38       2 2009  4651.000
## 39       3 2009  5449.000
## 40       4 2009  9519.000
## 41       1 2010  7131.000
## 42       2 2010  6566.000
## 43       3 2010  7560.000
## 44       4 2010 12948.000
## 45       1 2011  9857.000
## 46       2 2011  9913.000
## 47       3 2011 10876.000
## 48       4 2011 17431.000
## 49       1 2012 13185.000
## 50       2 2012 12834.000
## 51       3 2012 13806.000
## 52       4 2012 21268.000
## 53       1 2013 16070.000
## 54       2 2013 15704.000
## 55       3 2013 17092.000
## 56       4 2013 25587.000
## 57       1 2014 19741.000
## 58       2 2014 19340.000
```

1.

```r
# a) Roots are greater than 1, which does make it stationary.
coefficients_a <- c(1, -1.2, 0.85)
roots_a <- polyroot(coefficients_a)
roots_a
```

```
## [1] 0.7058824+0.8235294i 0.7058824-0.8235294i
```

```r
# b) Roots are looking like they are greater than 1, which also makes it stationary but is a bit inconc
coefficients_b <- c(1, -11/6, 1, -1/6)
roots_b <- polyroot(coefficients_b)
roots_b
```

```
## [1] 1-0i 2+0i 3-0i
```

```r
# c) Like b, there is uncertainty but one root is above 1, which implies potential for invertibility is
coefficients_c <- c(1, -15/4, 0, 1/4)
roots_c <- polyroot(coefficients_c)
roots_c
```

```
## [1]  0.2679492+0i -4.0000000-0i  3.7320508-0i
```

```r
# d) The roots show greater than 1, which implies stationary and potential for invertibility.
coefficients_d <- c(1, -0.8, 0.15)
roots_d <- polyroot(coefficients_d)

coefficients_i_d <- c(1, 0.3)
roots_i_d <- polyroot(coefficients_i_d)

roots_d
```

```
## [1] 2.000000-0i 3.333333+0i
```

```r
roots_i_d
```

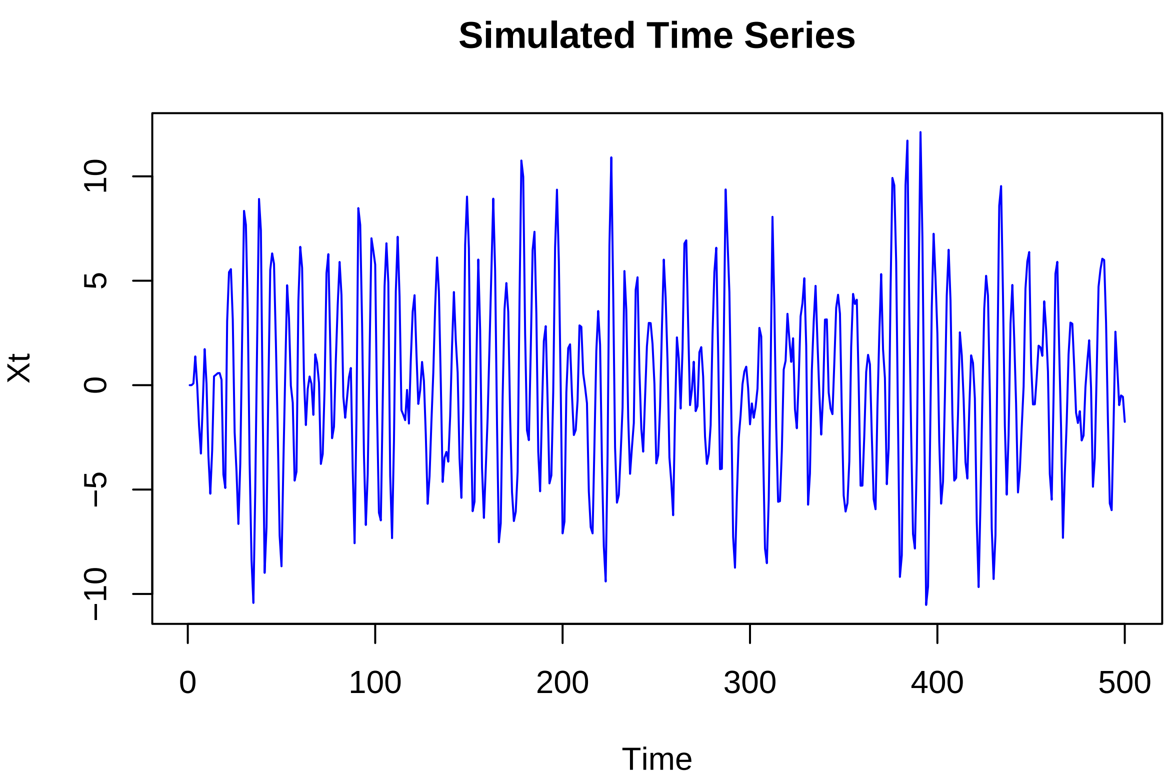
```
## [1] -3.333333+0i
```

2.

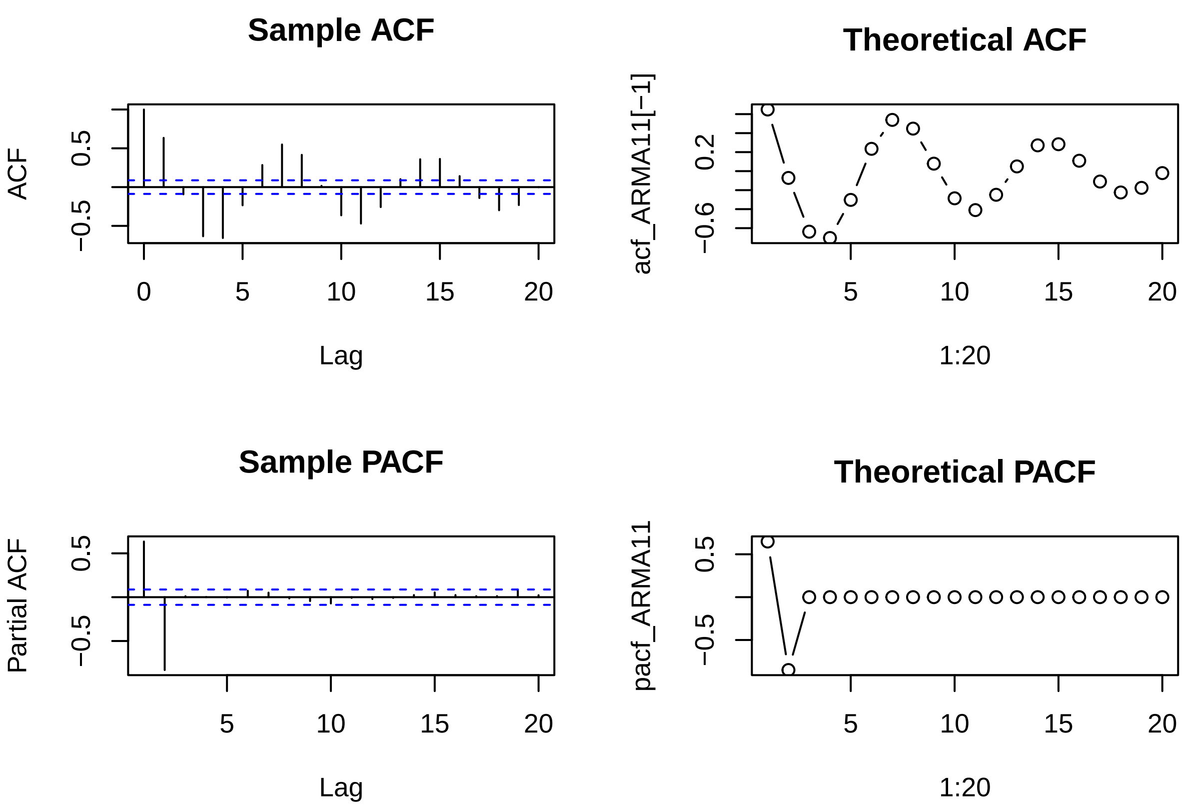
```r
library(forecast)
```

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

```r
# a)
set.seed(1000)

n <- 500
Wt <- rnorm(n, mean = 0, sd = sqrt(4))
Xt <- numeric(n)

for (t in 3:n) {
  Xt[t] <- 1.2 * Xt[t-1] - 0.85 * Xt[t-2] + Wt[t]
}

plot(Xt, type = "l", col = "blue", xlab = "Time", ylab = "Xt", main = "Simulated Time Series")
```



```r
# b) The sample ACF plot shows a quick up-and-down pattern, suggesting a dynamic time series with decre
acf_ARMA11 <- ARMAacf(ar = c(1.2, -0.85), lag.max = 20)
pacf_ARMA11 <- ARMAacf(ar = c(1.2, -0.85), lag.max = 20, pacf = TRUE)

sample_acf_ARMA11 <- acf(Xt, lag.max = 20, plot = FALSE)
sample_pacf_ARMA11 <- pacf(Xt, lag.max = 20, plot = FALSE)

par(mfrow = c(2, 2))

# Create sample ACF plot
plot(sample_acf_ARMA11, main = "Sample ACF")

plot(1:20, acf_ARMA11[-1], type = "b",  main = "Theoretical ACF")

# Create sample PACF plot
plot(sample_pacf_ARMA11, main = "Sample PACF")
plot(1:20, pacf_ARMA11, type = "b", main = "Theoretical PACF")
```

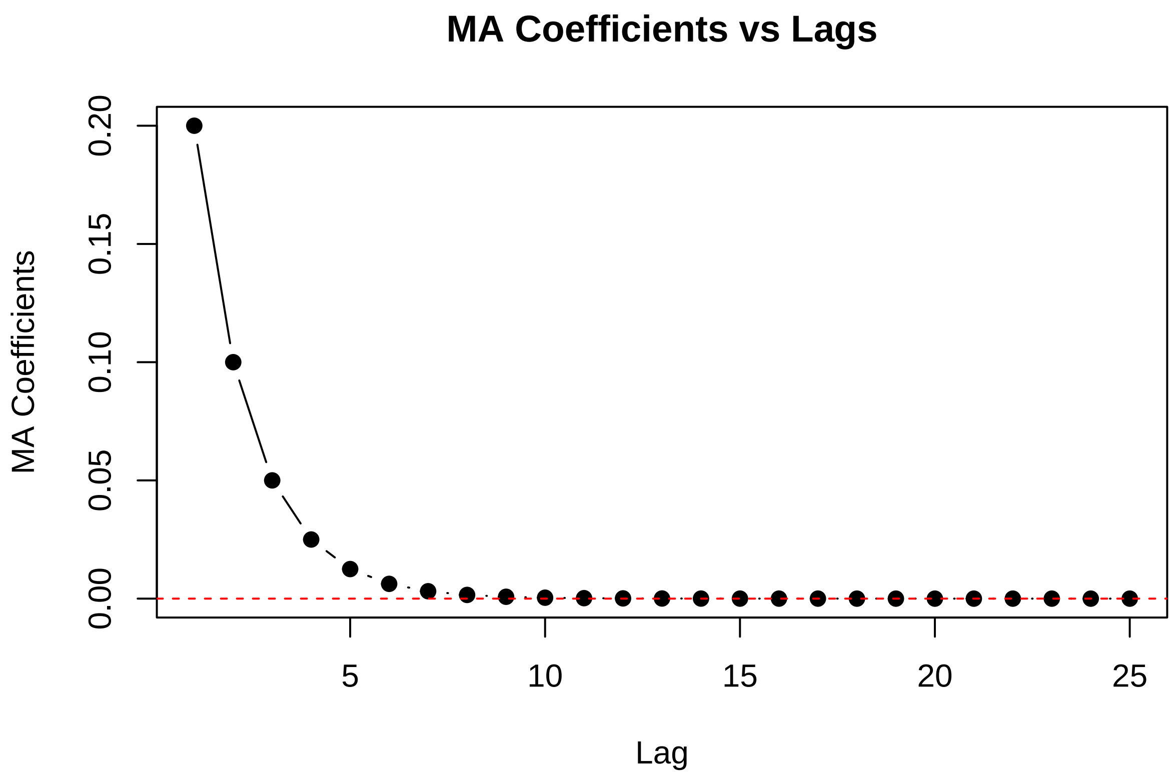


3.

```r
# a)
phi <- 0.5
theta <- -0.3

ma_coefs <- ARMAtoMA(ar = phi, ma = theta, lag.max = 25)
ma_coefs
```

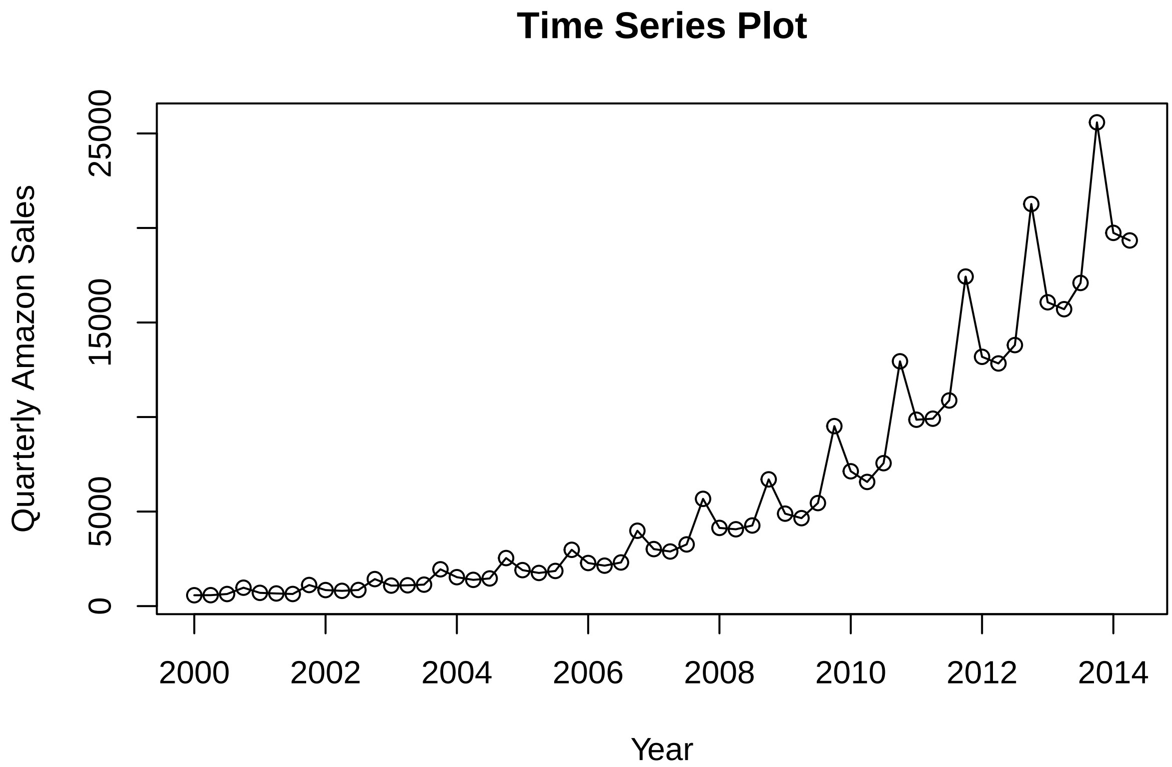
```
##  [1] 2.000000e-01 1.000000e-01 5.000000e-02 2.500000e-02 1.250000e-02
##  [6] 6.250000e-03 3.125000e-03 1.562500e-03 7.812500e-04 3.906250e-04
## [11] 1.953125e-04 9.765625e-05 4.882813e-05 2.441406e-05 1.220703e-05
## [16] 6.103516e-06 3.051758e-06 1.525879e-06 7.629395e-07 3.814697e-07
## [21] 1.907349e-07 9.536743e-08 4.768372e-08 2.384186e-08 1.192093e-08
```

```
# b) The shape of the plot shows a decay in coefficients rapidly and consistence with a consistent poin

plot(1:25, ma_coefs, pch = 19,type = "b",  xlab = "Lag", ylab = "MA Coefficients", main = "MA Coefficie
abline(h = 0, col = "red", lty = 2)
```



4.

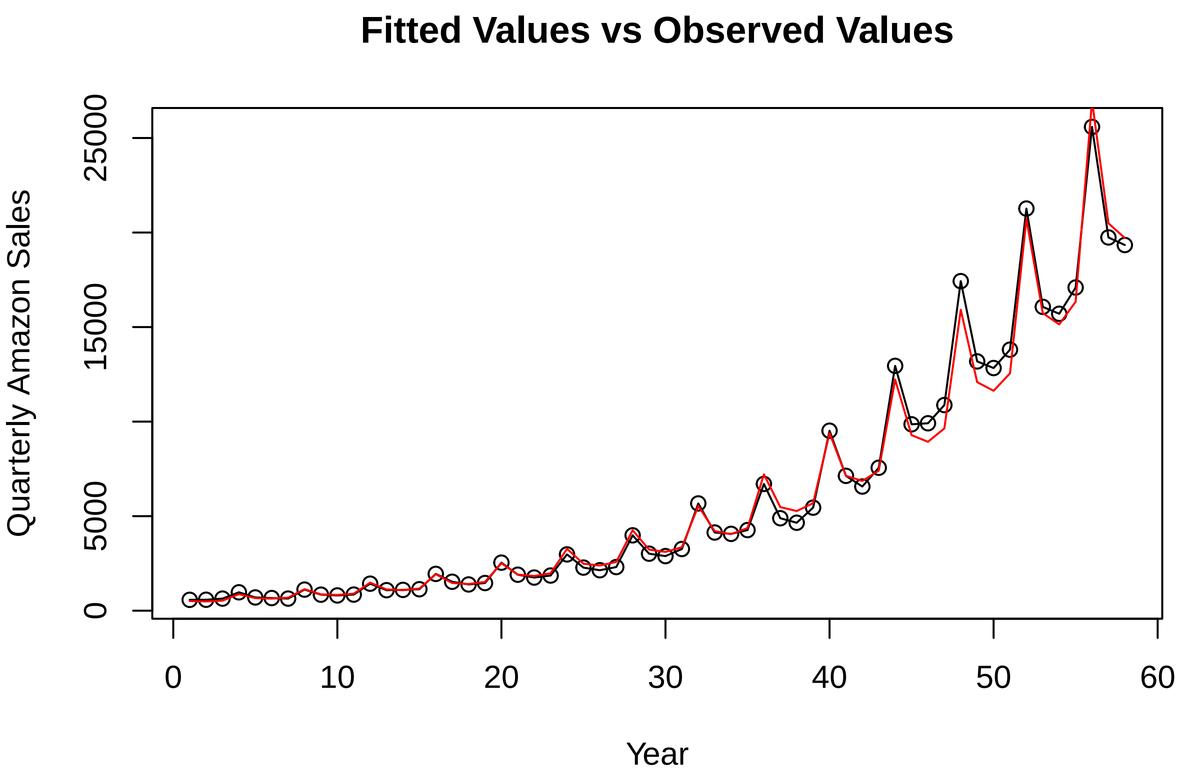
```
# (a)
data <- read.csv("amazon.csv")
sales_ts <- ts(data$Sales, start = c(data$Year[1], data$Quarter[1]), frequency = 4)

# (b)
plot(sales_ts, type = "o", xlab = "Year", ylab = "Quarterly Amazon Sales", main = "Time Series Plot")
```




```r
# (c) The model appears to provide a close fit qdto the original data, with fitted values closely resem

time <- 1:length(sales_ts)
model <- lm(log(sales_ts) ~ time + as.factor(data$Quarter))
summary(model)
```

```
##
## Call:
## lm(formula = log(sales_ts) ~ time + as.factor(data$Quarter))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12512 -0.04523 -0.01658  0.04419  0.18626
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.168183   0.025352 243.303  < 2e-16 ***
## time                      0.065954   0.000578 114.103  < 2e-16 ***
## as.factor(data$Quarter)2 -0.104492   0.026903  -3.884 0.000287 ***
## as.factor(data$Quarter)3 -0.094190   0.027373  -3.441 0.001138 **
## as.factor(data$Quarter)4  0.340925   0.027379  12.452  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07366 on 53 degrees of freedom
## Multiple R-squared:  0.9961, Adjusted R-squared:  0.9958
## F-statistic:  3357 on 4 and 53 DF,  p-value: < 2.2e-16
```

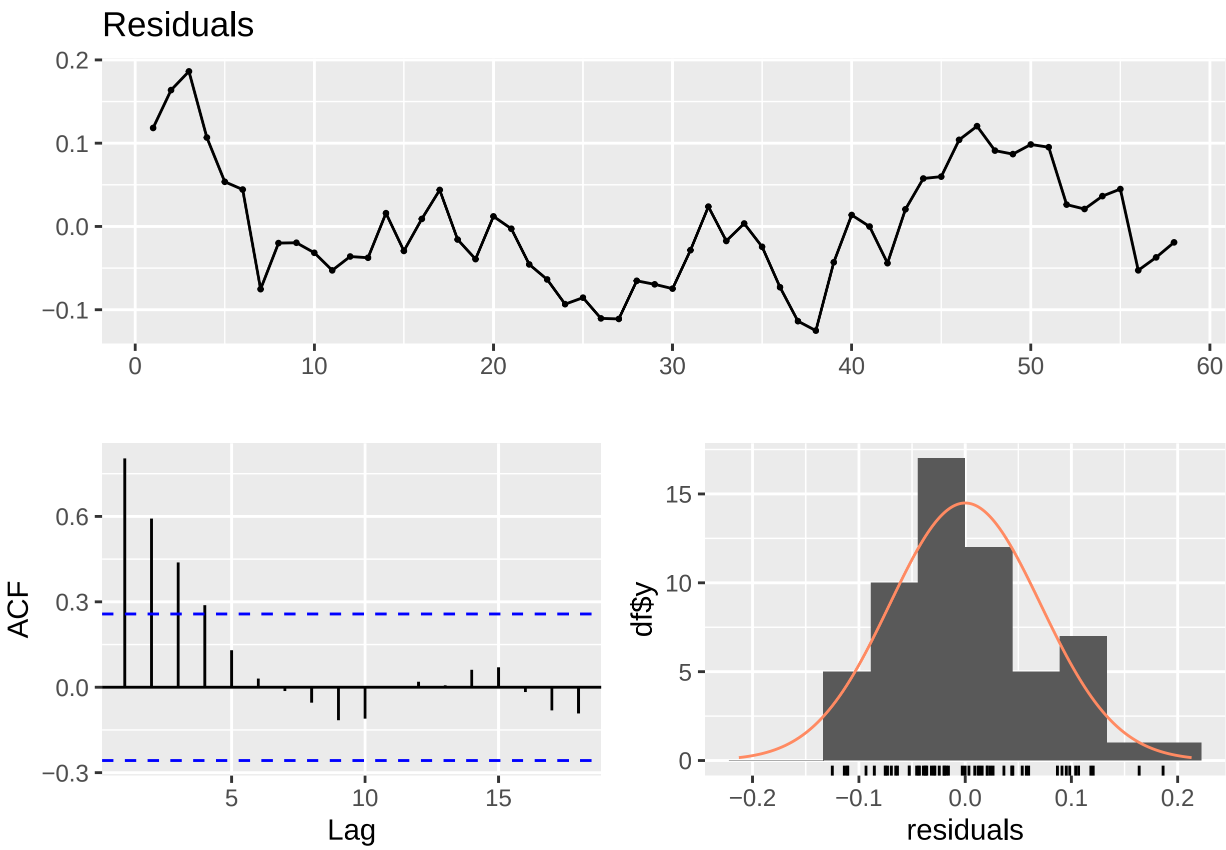
```
fitted_values <- exp(predict(model))
plot(time, sales_ts, type = "o", xlab = "Year", ylab = "Quarterly Amazon Sales", main = "Fitted Values v
lines(time, fitted_values, col = "red")
```




```
# (d)
rsquared <- summary(model)$r.squared
aic <- AIC(model)
bic <- BIC(model)
cat("R-squared:", rsquared, "\nAIC:", aic, "\nBIC:", bic, "\n")
```

```
## R-squared: 0.9960685
## AIC: -131.1925
## BIC: -118.8299
```

```
# (e) The Ljung-Box test result (p-value = 2.049e-13) suggests that there is strong evidence of pattern
residuals_check <- checkresiduals(model, test = "LB", LjungBox = TRUE)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 81.983, df = 10, p-value = 2.049e-13
##
## Model df: 0.   Total lags used: 10
```

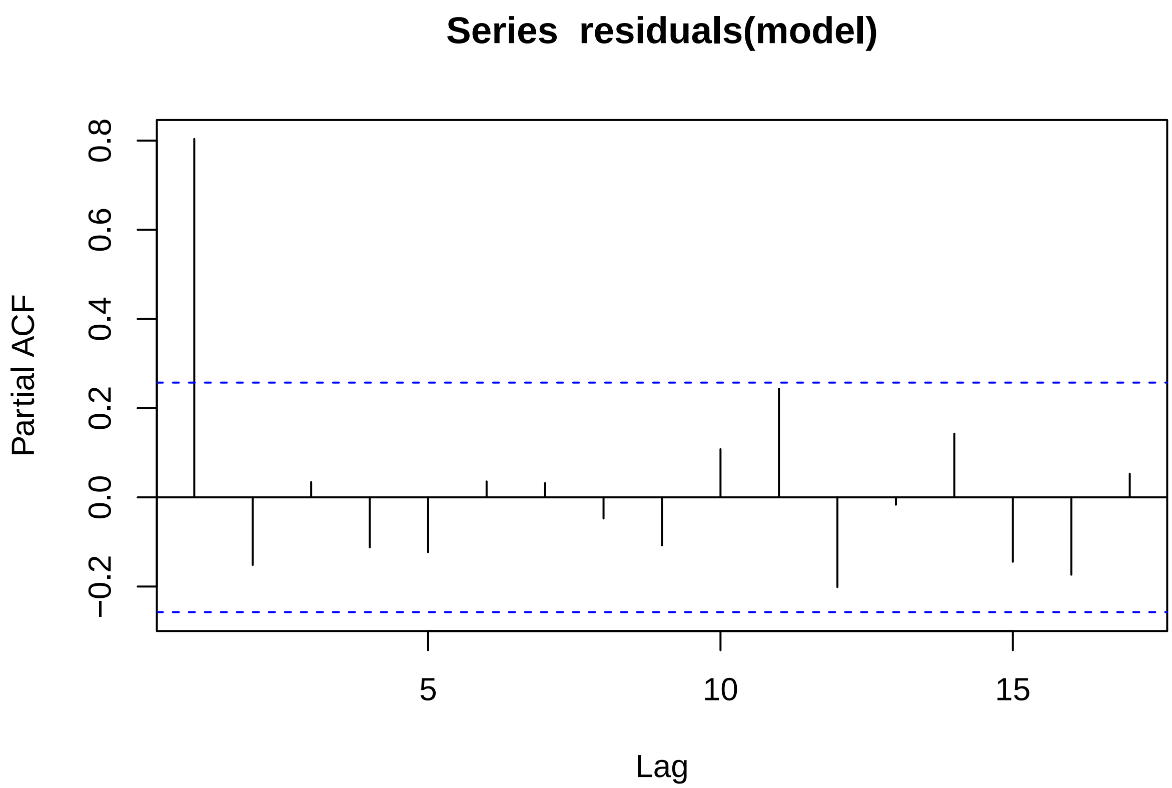
```
residuals_check
```

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 81.983, df = 10, p-value = 2.049e-13
```

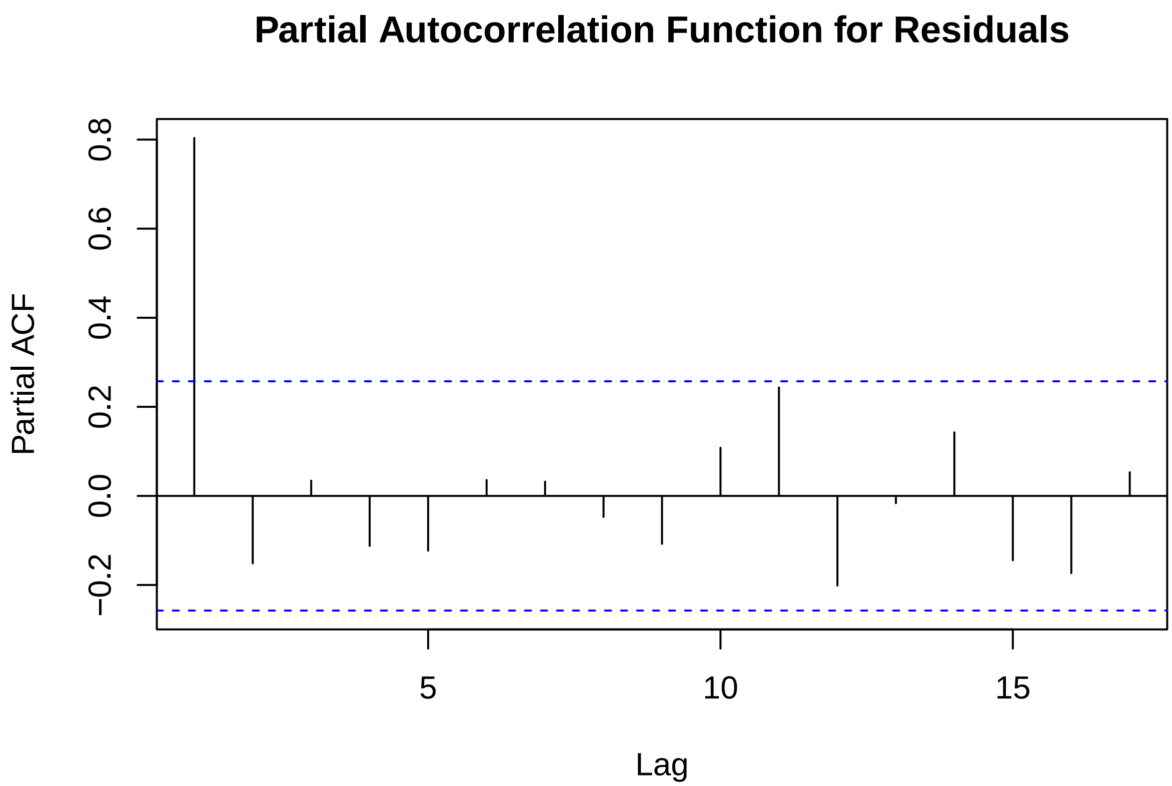
```
# (f)
library(tseries)

adf_test <- adf.test(residuals(model))
cat("ADF Test p-value:", adf_test$p.value, "\n")
```

```
## ADF Test p-value: 0.09728275
```

```r
# (g) The plot shows a large lag at 1 then a large drop with consistency overtime around 0, which can i
pacf_res <- pacf(residuals(model))
```



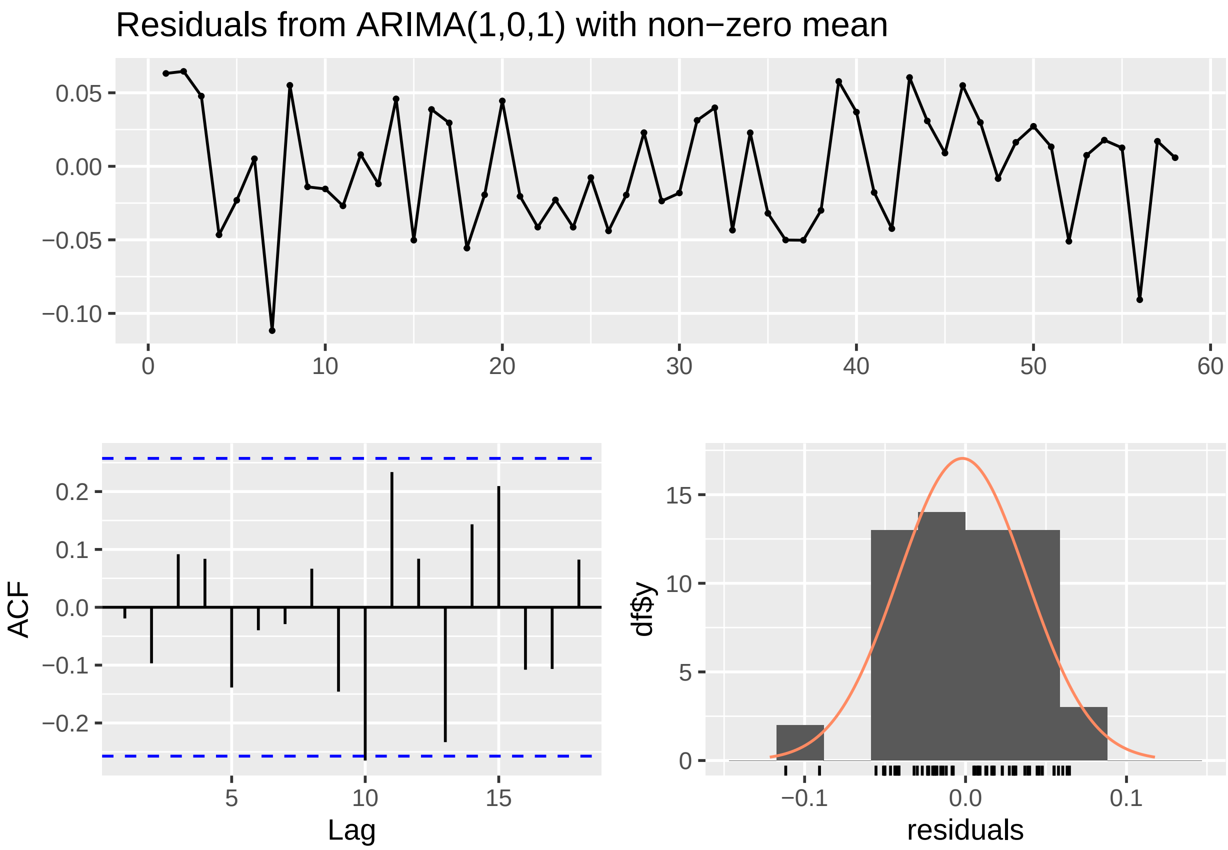

```r
plot(pacf_res, main = "Partial Autocorrelation Function for Residuals")
```




```
# h) Sure, same model.
library(forecast)

residuals_model <- residuals(model)
auto.arima(residuals_model)
```

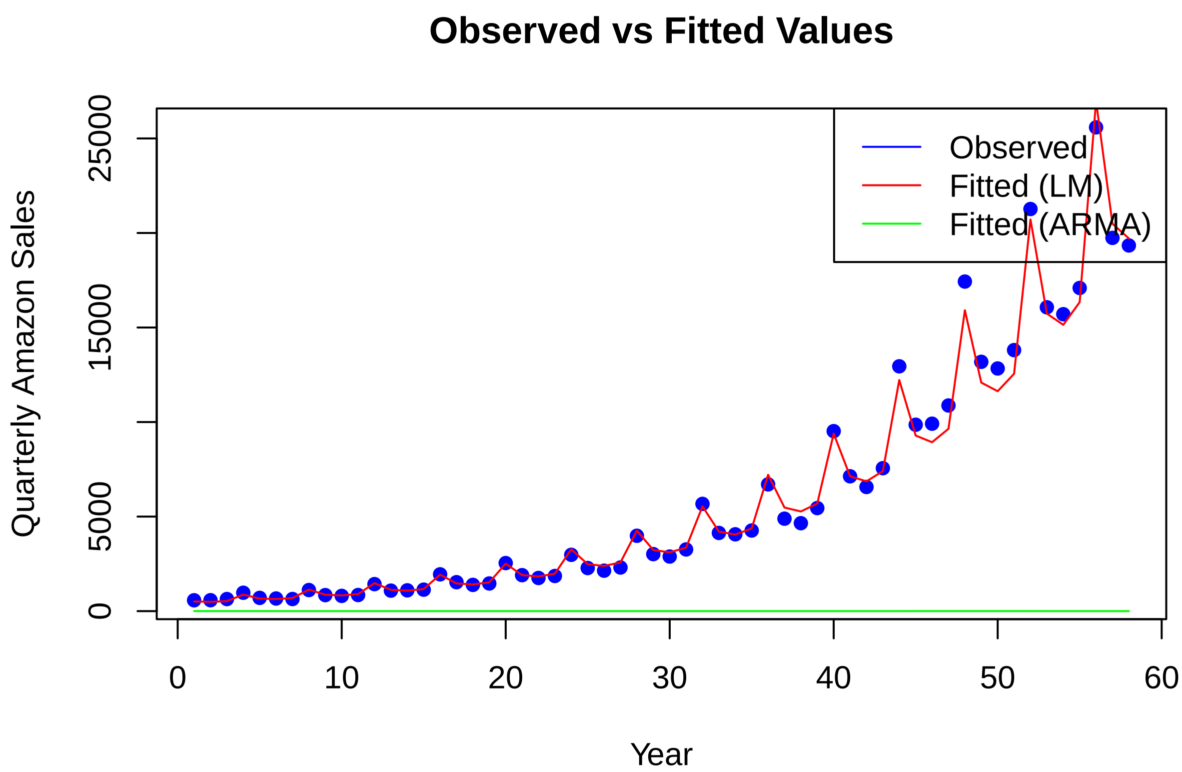
```
## Series: residuals_model
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##       0.8303
## s.e.  0.0726
##
## sigma^2 = 0.001616:  log likelihood = 104.03
## AIC=-204.06   AICc=-203.85   BIC=-199.94
```

```
#wont work after 7 horus sorry
#armasubsets(residuals_model, nar = 2, nma = 2)

# (i)
arma_model <- arima(residuals(model), order = c(1, 0, 1))
checkresiduals(arma_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 9.9234, df = 8, p-value = 0.2704
##
## Model df: 2.   Total lags used: 10
```

```
# (j) The model fits quite well with the model with some anomalies.
fitted_lm <- exp(predict(model))
fitted_arma <- fitted(arma_model)
plot(time, as.vector(sales_ts), type = "p", col = "blue", pch = 16, xlab = "Year", ylab = "Quarterly Am
lines(time, fitted_lm, col = "red", type = "l")
lines(time, fitted_arma, col = "green", type = "l")
legend("topright", legend = c("Observed", "Fitted (LM)", "Fitted (ARMA)"), col = c("blue", "red", "gree
```




```r
# (k) Yes, very similar.
future_time <- max(time) + 1:2
lm_future_data <- data.frame(time = future_time, Quarter = rep(1:4, times = 2))
#lm_forecast <- exp(predict(model, newdata = lm_future_data))
future_data <- data.frame(Quarter = rep(1:4, times = 2), Year = rep(max(data$Year) + 1, each = 4))
future_predictors <- data.frame(time = rep(future_time, each = 4), Quarter = rep(1:4, times = 2))
arma_forecast <- forecast(arma_model, h = 2)
arma_forecast
```

```
##    Point Forecast       Lo 80      Hi 80       Lo 95      Hi 95
## 59   -0.012743141 -0.06347563 0.03798935 -0.09033179 0.06484551
## 60   -0.008721917 -0.07818796 0.06074413 -0.11496107 0.09751724
```