Business Forecasting - Assignment 02

Ashita Shetty

2023-11-27

```
library(datasets)
library(fpp2)
```

```
## Warning: package 'fpp2' was built under R version 4.1.3
```

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

```
## -- Attaching packages --------------------------------------------- fpp2 2.5 --
```

```
## v ggplot2   3.4.3     v fma       2.5
## v forecast  8.21      v expsmooth 2.3
```

```
## Warning: package 'forecast' was built under R version 4.1.3
```

```
## Warning: package 'fma' was built under R version 4.1.3
```

```
## Warning: package 'expsmooth' was built under R version 4.1.3
```

```
##
```

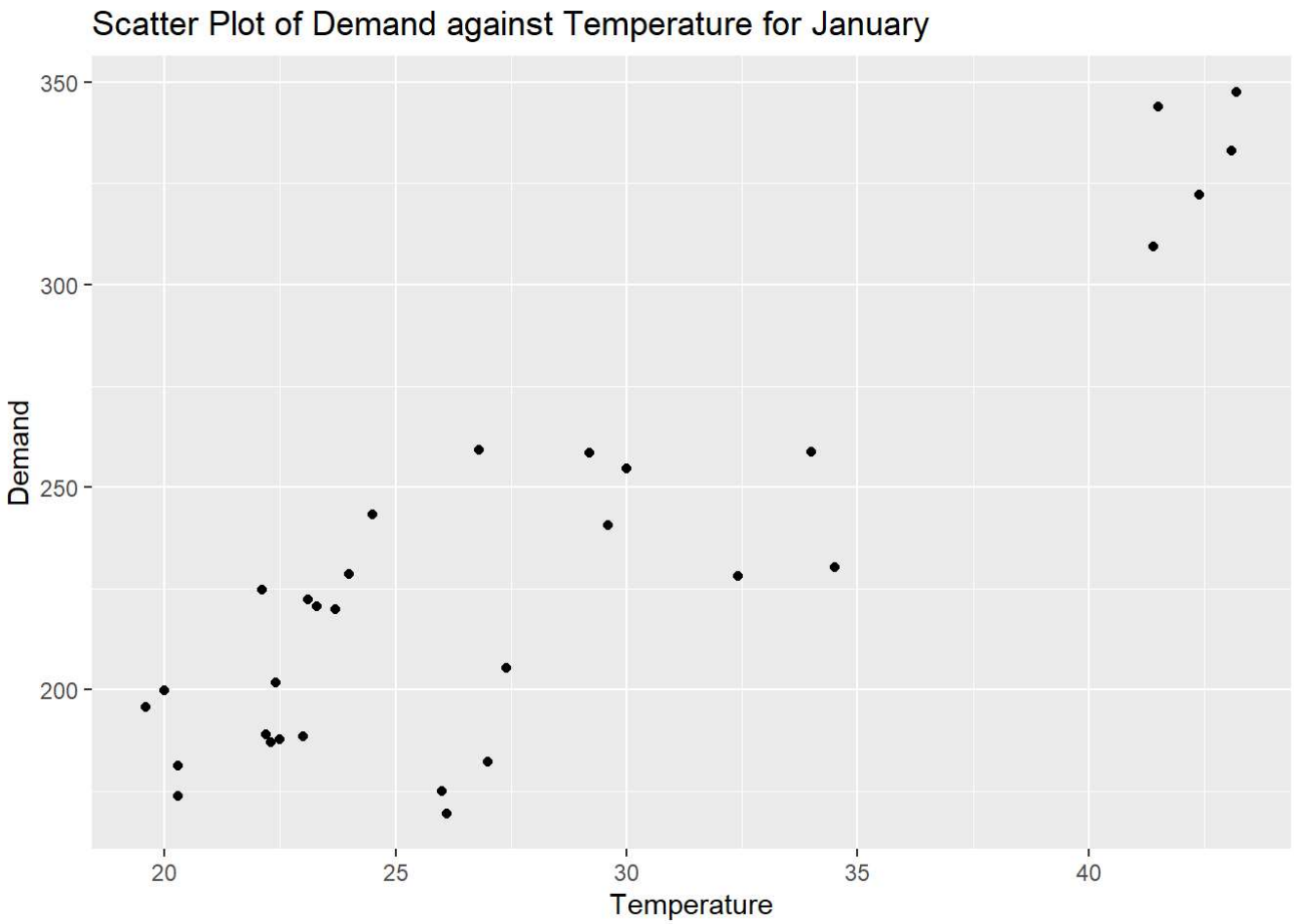
```
library(ggplot2)
```

Exercise 1

elecdaily

(i) Create a new dataset for only the month of January (i.e. the first 31 days) using the head function and create a scatter plot of Demand against Temperature.

```
elec_daily = data.frame(elecdaily)
elec_daily_jan = head(elec_daily, n=31)
ggplot(elec_daily_jan, aes(x = Temperature, y = Demand)) +
  geom_point() +
  labs(title = "Scatter Plot of Demand against Temperature for January",
       x = "Temperature",
       y = "Demand")
```



(ii) Give a possible explanation of the relationship you see in your scatterplot.

- In general, it can be observed that as the temperature increases, the demand also increases.
- However, it is not a linear relationship.

(iii) Create a simple linear regression model in R with Demand as the forecast variable and Temperature as the predictor variable. Write down the equation of the fitted model

```
lm_model <- lm(Demand ~ Temperature, data=elec_daily_jan)
summary(lm_model)
```

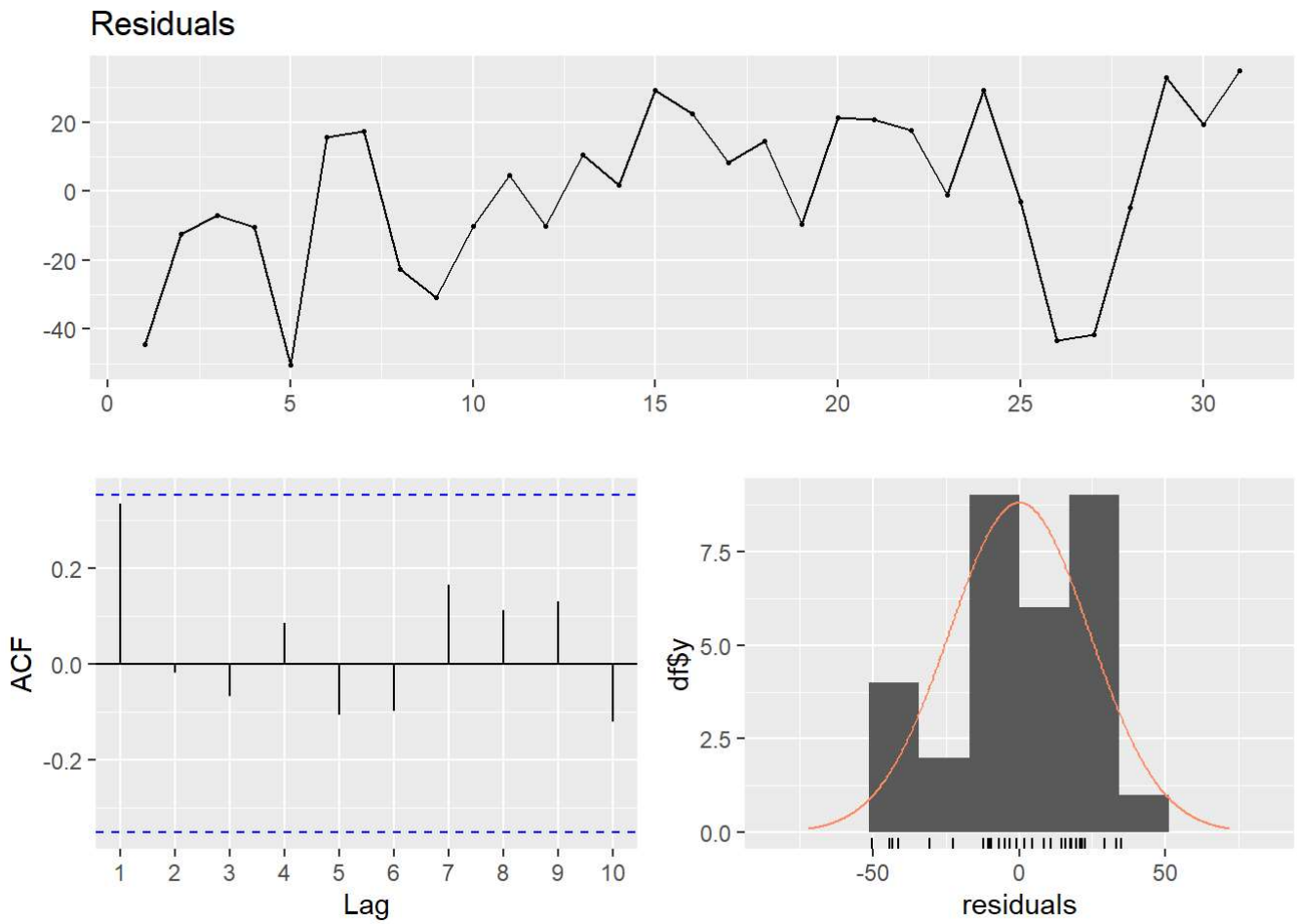
```
##
## Call:
## lm(formula = Demand ~ Temperature, data = elec_daily_jan)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.470 -10.333   1.915  18.520  35.012
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59.3294    17.2626   3.437   0.0018 **
## Temperature   6.1554     0.5963  10.322  3.2e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.43 on 29 degrees of freedom
## Multiple R-squared:  0.786,  Adjusted R-squared:  0.7787
## F-statistic: 106.5 on 1 and 29 DF,  p-value: 3.202e-11
```

```
#Fitted Model Equation
intercept <- coef(lm_model)[1]
slope <- coef(lm_model)[2]

cat("Fitted Model Equation: Demand =", round(intercept, 2), "+", round(slope, 2), "* Temperature
\n")
```

```
## Fitted Model Equation: Demand = 59.33 + 6.16 * Temperature
```

(iv) Is there any evidence of autocorrelation in the residuals? Do the residuals appear to show heteroscedasticity?

```
checkresiduals(lm_model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 6
##
## data:  Residuals
## LM test = 6.9378, df = 6, p-value = 0.3266
```

- Based on the graph above, it can be inferred that there are no autocorrelations between the residuals of the model because it is within the limits of the confidence intervals.
- The residuals do not appear to show heteroscedasticity since they do not have any pattern indicating there is no variance across data.

(v) Use your model to forecast the electricity demand if the temperature is 35 degrees or 15 degrees. Would you trust either or both of these forecasts? Explain your answer.

Hint: you can create a new data frame to use in the forecast function by using the code my.data <- data.frame(Temperature = c(15,35))

```
new_data <- data.frame(Temperature = c(15, 35))

new_forecasts <- predict(lm_model, newdata = new_data)

cat("Forecasted Demand for Temperature 15 degrees:", round(new_forecasts[1], 2), "\n")
```

```
## Forecasted Demand for Temperature 15 degrees: 151.66
```

```
cat("Forecasted Demand for Temperature 35 degrees:", round(new_forecasts[2], 2), "\n")
```

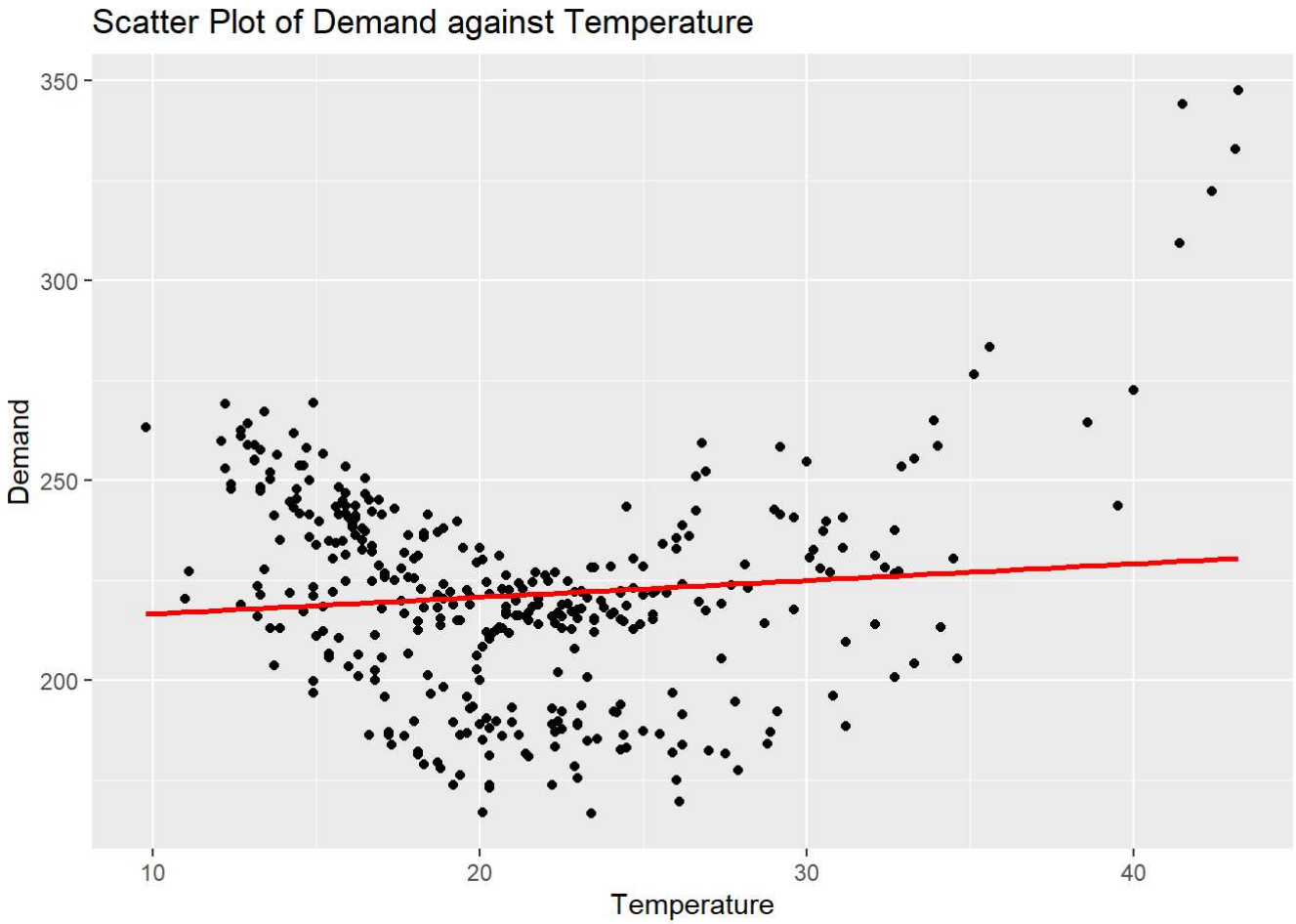
```
## Forecasted Demand for Temperature 35 degrees: 274.77
```

- Considering that our data has a positive slope, our demand increases with increase in temperature.
- Even for the temperatures mentioned, the forecasted values can be observed as increasing with temperature.

(vi) Plot Demand against Temperature for all available data in elecdaily. What does this say about your model?

```
ggplot(elec_daily, aes(x = Temperature, y = Demand)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatter Plot of Demand against Temperature",
       x = "Temperature",
       y = "Demand")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



- As observed, Linear Model Regression wouldn't be an ideal model to opt for since the data follows non-linearity.

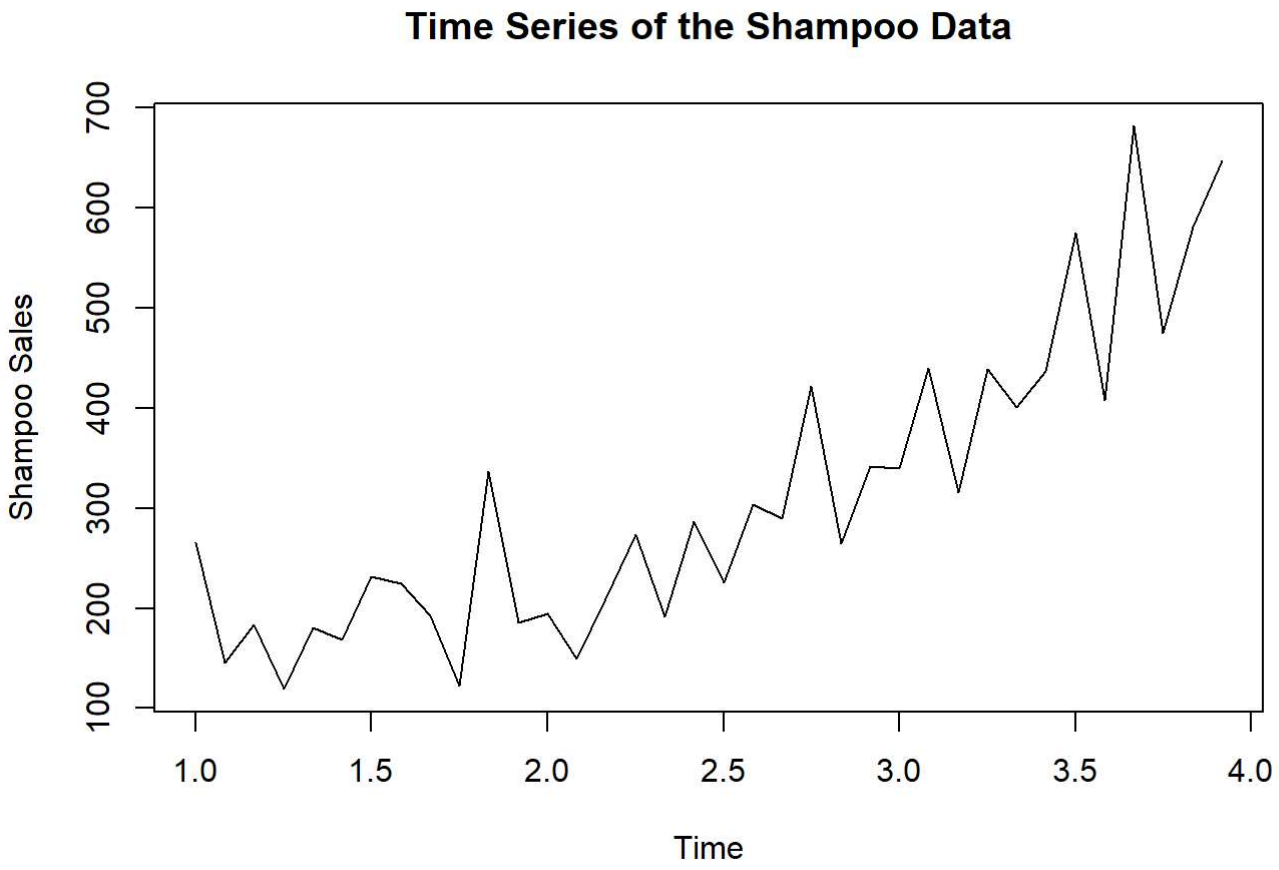
Exercise 2

shampoo

(i) Plot the data and comment on any patterns you see.

```
data(shampoo)
```

```
plot(shampoo, main='Time Series of the Shampoo Data', xlab='Time', ylab='Shampoo Sales')
```



- The data depicts an increasing trend in the Sales of Shampoo through the period of 3 years

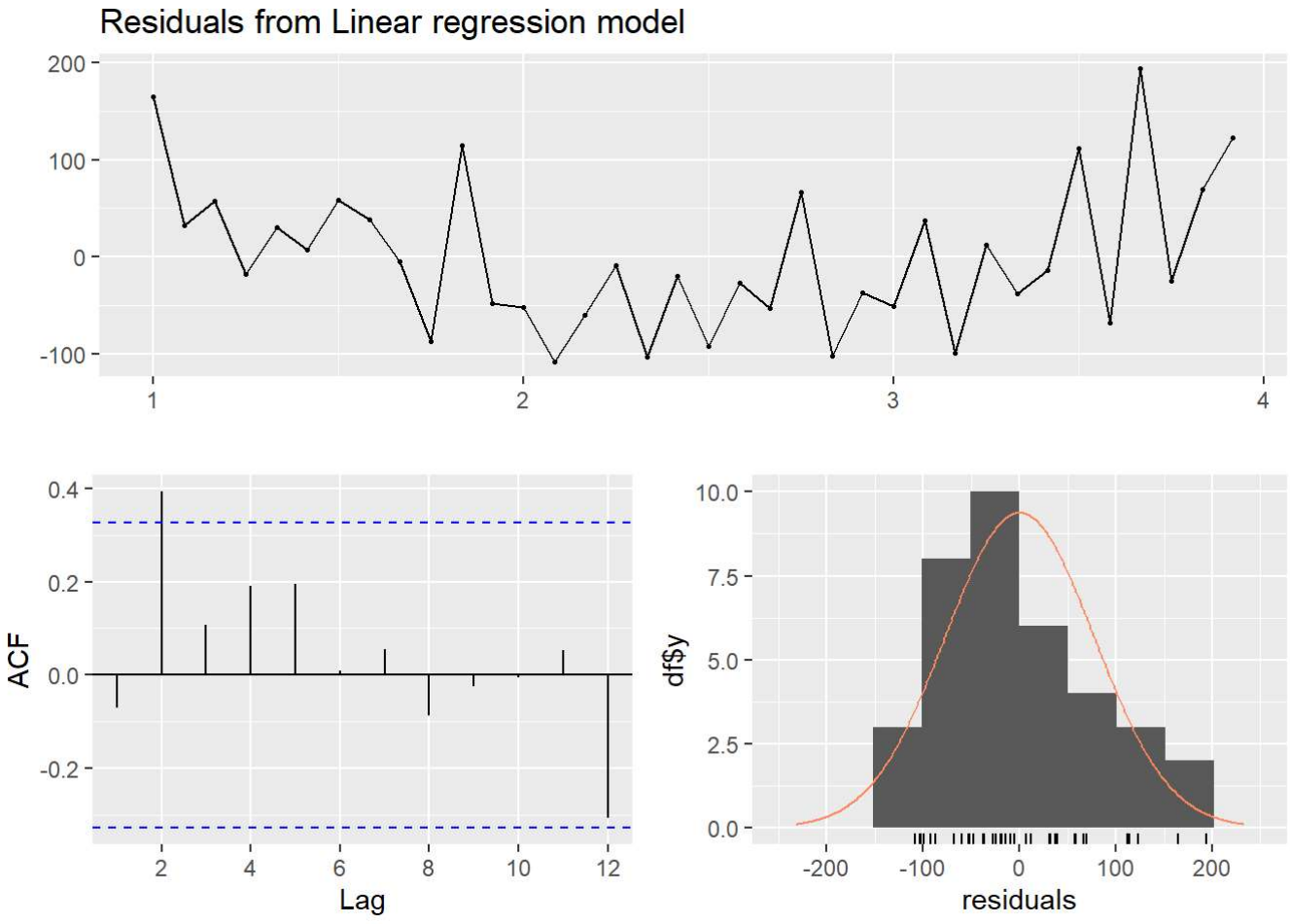
(ii) Fit a simple linear regression model for shampoo with a linear trend predictor variable

```
shampoo_lm_model <- tslm(shampoo~trend)
summary(shampoo_lm_model)
```

```
##
## Call:
## tslm(formula = shampoo ~ trend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -108.74  -52.12  -16.13   43.48  194.25
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    89.14      26.72   3.336  0.00207 **
## trend          12.08       1.26   9.590 3.37e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.5 on 34 degrees of freedom
## Multiple R-squared:  0.7301, Adjusted R-squared:  0.7222
## F-statistic: 91.97 on 1 and 34 DF,  p-value: 3.368e-11
```

(iii) Is there evidence of autocorrelation in the residuals?

```
checkresiduals(shampoo_lm_model)
```

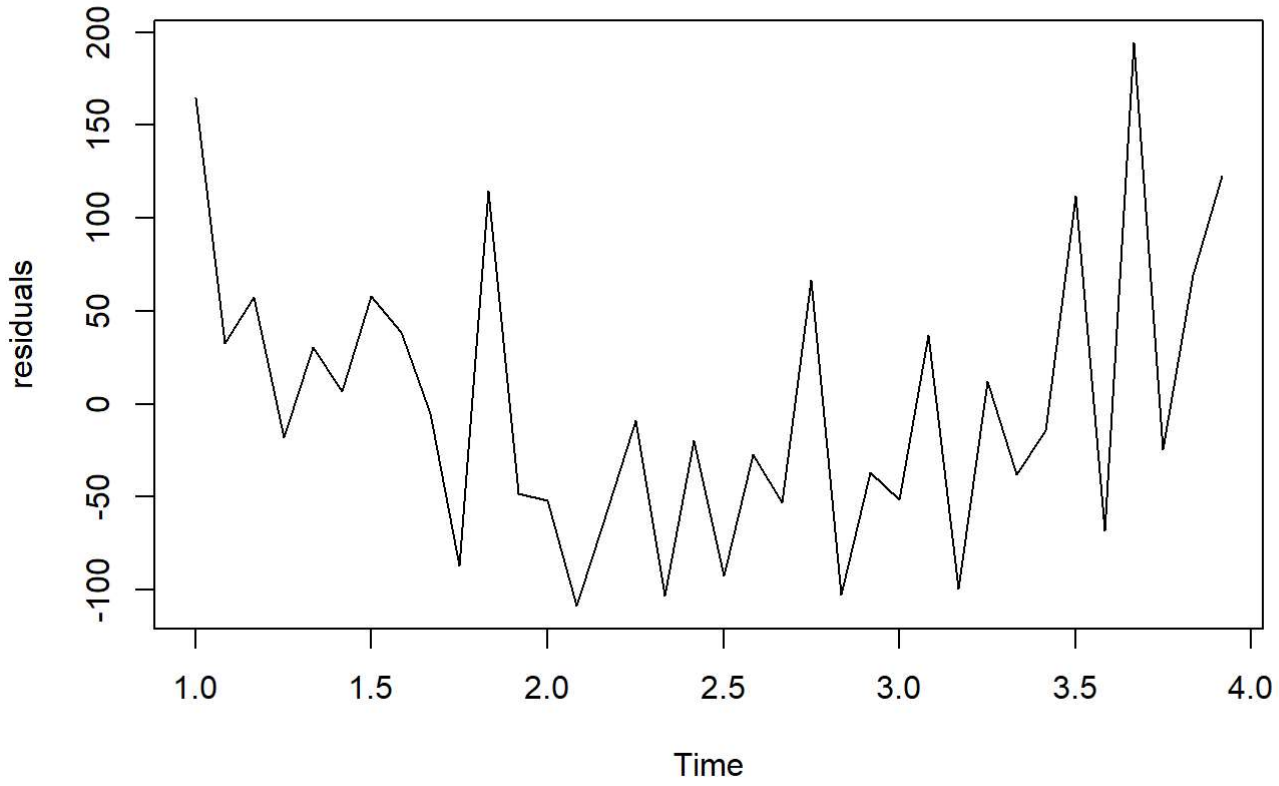


```
##
##  Breusch-Godfrey test for serial correlation of order up to 7
##
## data:  Residuals from Linear regression model
## LM test = 10.211, df = 7, p-value = 0.1769
```

- Based on the ACF plot above, it can be inferred that there is a correlation between the residuals.
- Since the ACF values have positive and negative values, it can be inferred that they have seasonality as well.

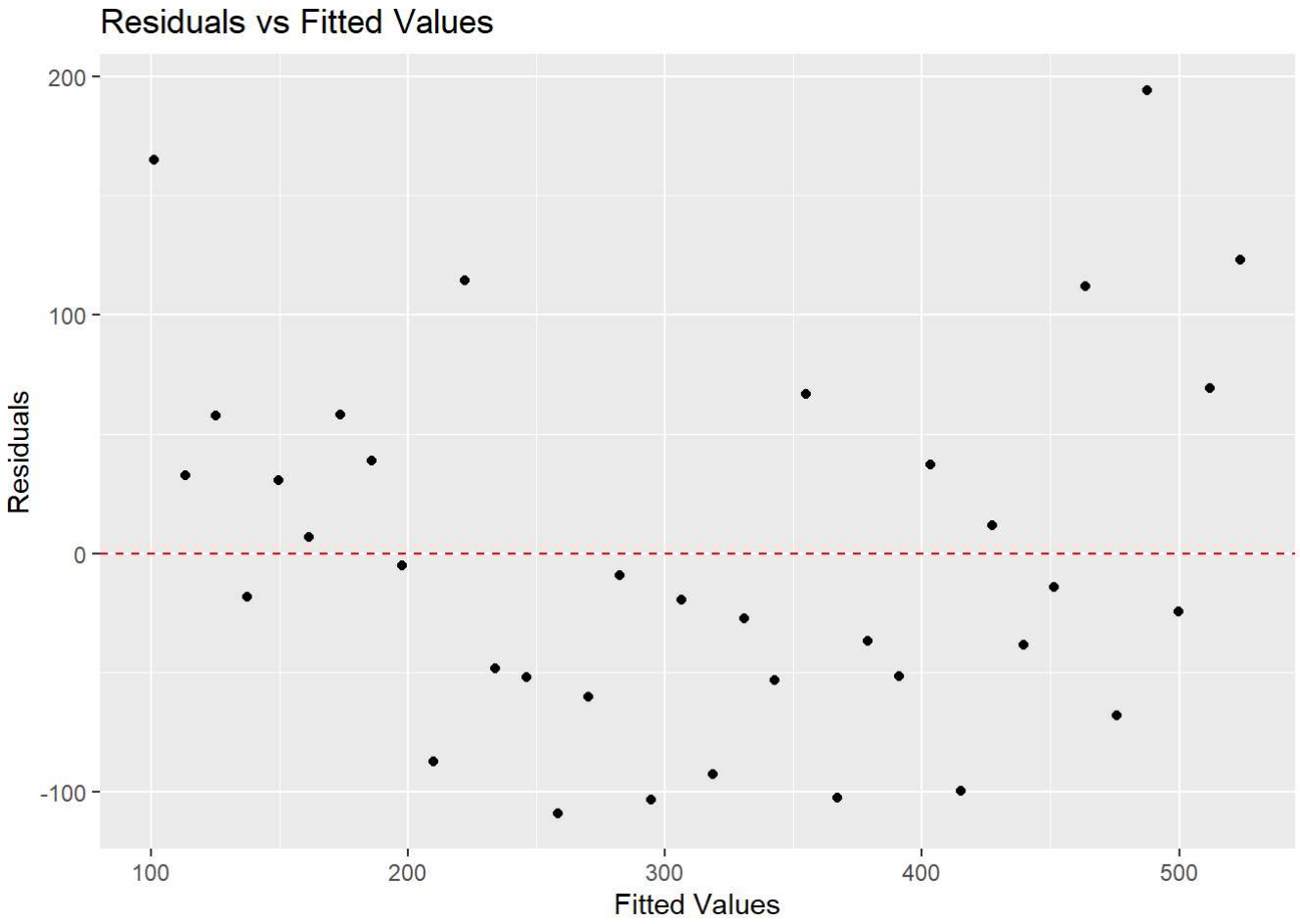
(iv) Plot the residuals against time and against the fitted values. Do the plots reveal any problems with the model?

```
residuals <- residuals(shampoo_lm_model)

plot(residuals)
```




```
plot_data <- data.frame(Residuals = residuals(shampoo_lm_model), FittedValues = fitted(shampoo_l
m_model))

ggplot(plot_data, aes(x = fitted(shampoo_lm_model), y = residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals")
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



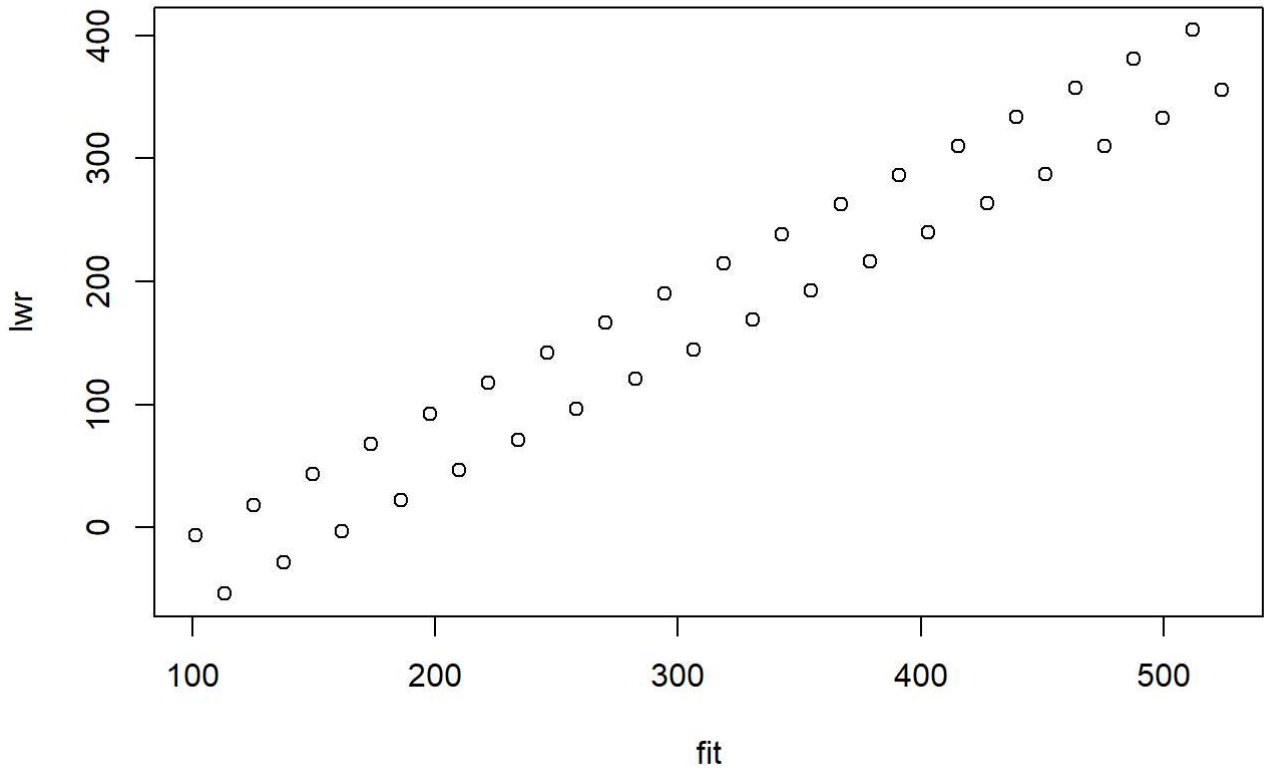
- There is no apparent change in the residuals spread across the fitted values, indicating that they are constant.
- Regression model assumptions are not violated by the residual's lack of autocorrelation or non-normality.

(v) Create a forecast for shampoo sales for the next year, along with 95% and 80% prediction intervals. Plot the forecast alongside the original data. Why should you be wary about trusting the prediction intervals?

```
future_period <- data.frame(Month=c(37:48))

forecast <- predict(shampoo_lm_model, newdata = future_period, interval = "prediction", level =
c(0.80, 0.95))
```

```
## Warning: 'newdata' had 12 rows but variables found have 36 rows
```

```
plot(forecast)
```



- The prediction intervals showcase the window on how much the forecast can vary.
- The higher the confidence interval, the more the forecast accuracy will be.

(vi) Create a multiple linear regression model for shampoo using both a linear trend predictor variable and seasonal dummy variables. Calculate the AICc for both models. Which model is better?

```
shampoo_sreg <- tslm(shampoo ~ trend + season)
summary(shampoo_sreg)
```

```
##
## Call:
## tslm(formula = shampoo ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -129.60  -62.32   -4.84   53.76  152.72
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  113.867     55.740   2.043   0.0527 .
## trend         11.754      1.534   7.664 8.88e-08 ***
## season2      -33.154     73.630  -0.450   0.6567
## season3      -53.808     73.678  -0.730   0.4726
## season4      -24.628     73.757  -0.334   0.7415
## season5      -56.015     73.869  -0.758   0.4560
## season6      -27.802     74.012  -0.376   0.7106
## season7        7.244     74.187   0.098   0.9231
## season8      -37.043     74.393  -0.498   0.6233
## season9       27.536     74.629   0.369   0.7155
## season10     -32.518     74.897  -0.434   0.6682
## season11       9.895     75.194   0.132   0.8964
## season12      -4.259     75.522  -0.056   0.9555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 90.16 on 23 degrees of freedom
## Multiple R-squared:  0.7592, Adjusted R-squared:  0.6336
## F-statistic: 6.043 on 12 and 23 DF,  p-value: 0.0001161
```

```
AIC(shampoo_sreg)
```

```
## [1] 438.1474
```

```
AIC(shampoo_lm_model)
```

```
## [1] 420.2535
```

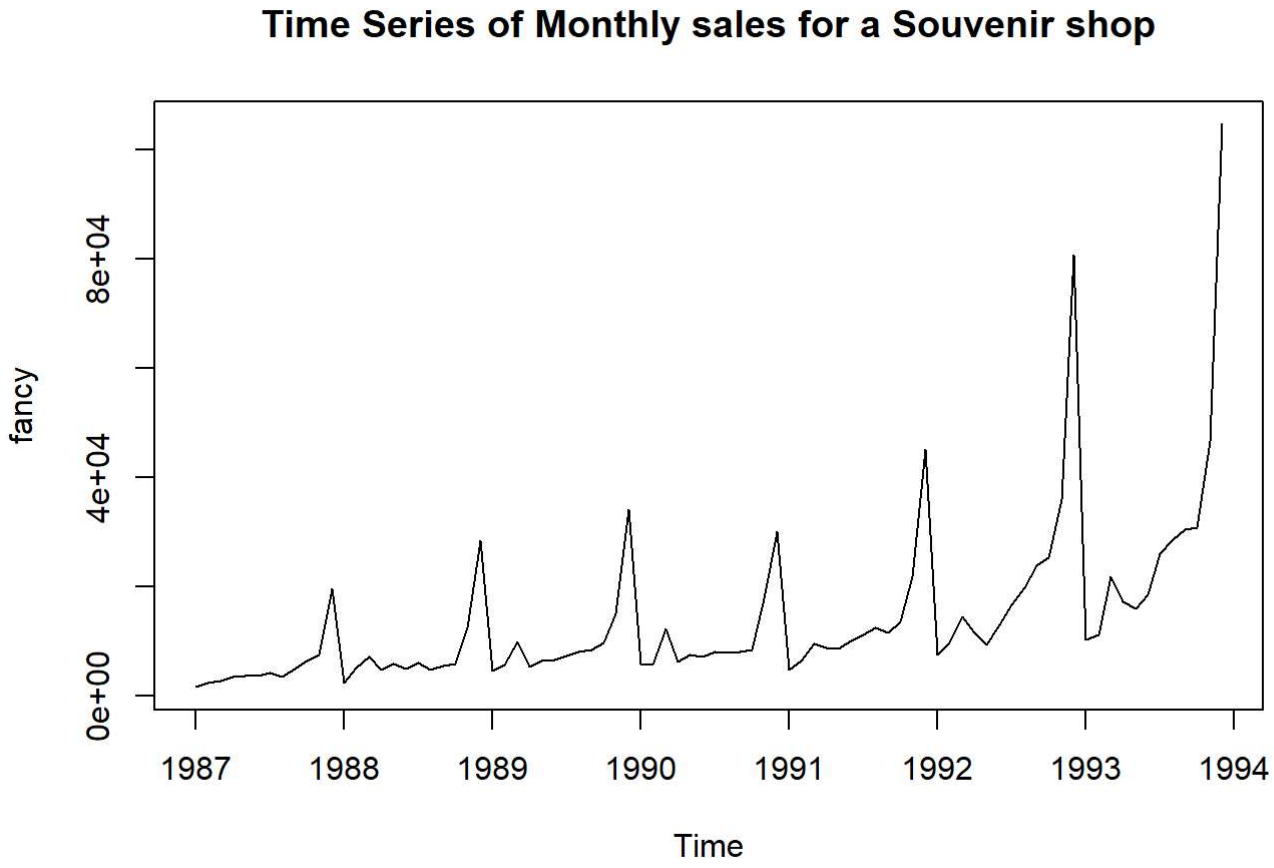
- The linear regression model with trend and seasonality had higher AIC, therefore it is the preferred model to go with.

Exercise 3

fancy

(i) Produce a time plot of the data and describe the patterns in the graph. Identify any unusual or unexpected fluctuations in the time series.

```
data("fancy")
plot(fancy, main='Time Series of Monthly sales for a Souvenir shop')
```



- The Time Series graph depicts *seasonality* and *trend* in its pattern.
- While most of the data, up until 1992 averaged around the same values, it can be observed that an increase was observed in the latter years.
- The *seasonality* in the trends can also be an indication of tourist seasons or any other festivity/event that attracts more sales.

(ii) Explain why it is appropriate to take logarithms of these data before fitting a model.

- Logarithmic Transformations are preferred when trying to stabilize the variance in the data across time.
- They help in normalizing the data when the distribution of data is skewed.

(iii) Use R to fit a regression model to the logarithms of these sales data with a linear trend, seasonal dummies and a “surfing festival” dummy variable.

Hint: the “surfing festival” dummy variable can be created using the following code.

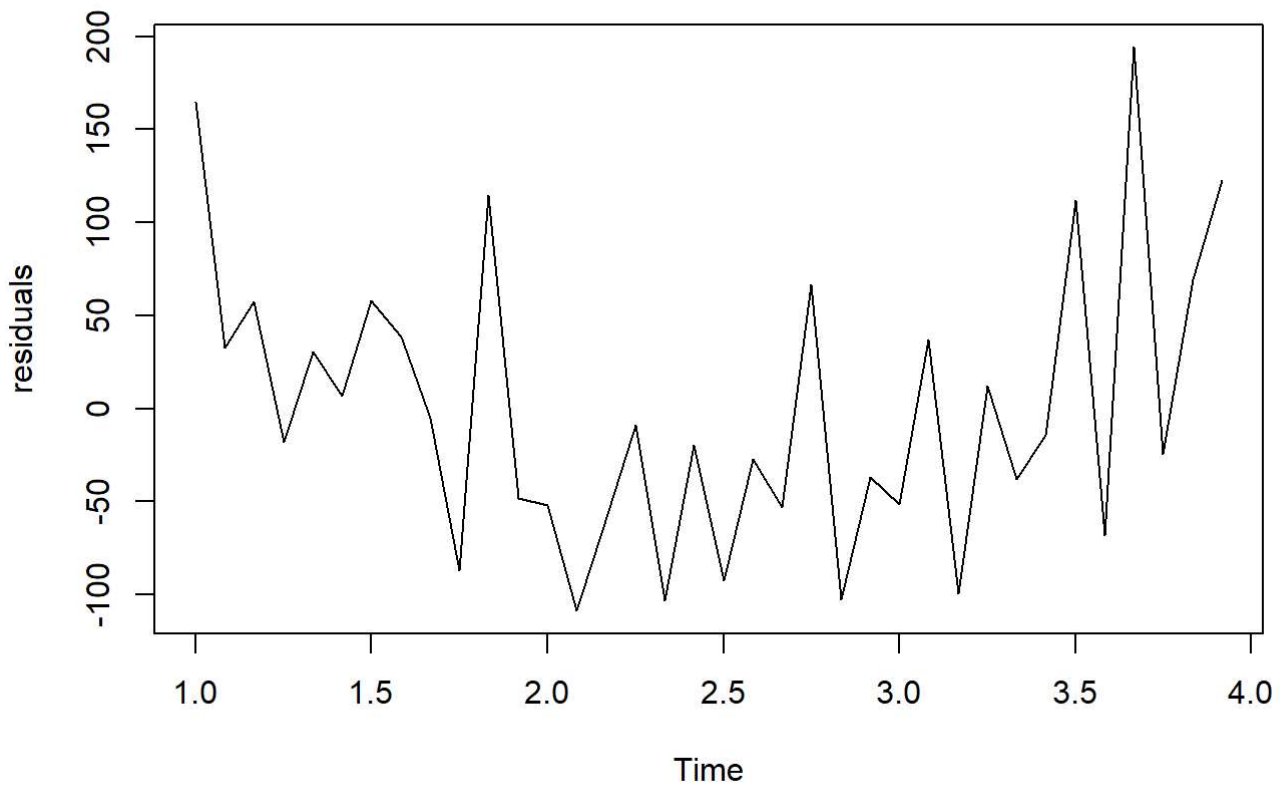
#####> festival <- cycle(fancy) == 3 #####> festival[3] == 0

```
log_fancy <- log(fancy)
time_index <- seq_along(log_fancy)
month <- cycle(log_fancy)
seasonal_dummies <- model.matrix(~ factor(month) - 1)
festival <- cycle(fancy) == 3
festival[1:12] <- 0
predictors <- data.frame(time_index, seasonal_dummies, festival)
fancy_lm <- lm(log_fancy ~ ., data = predictors)

summary(fancy_lm)
```

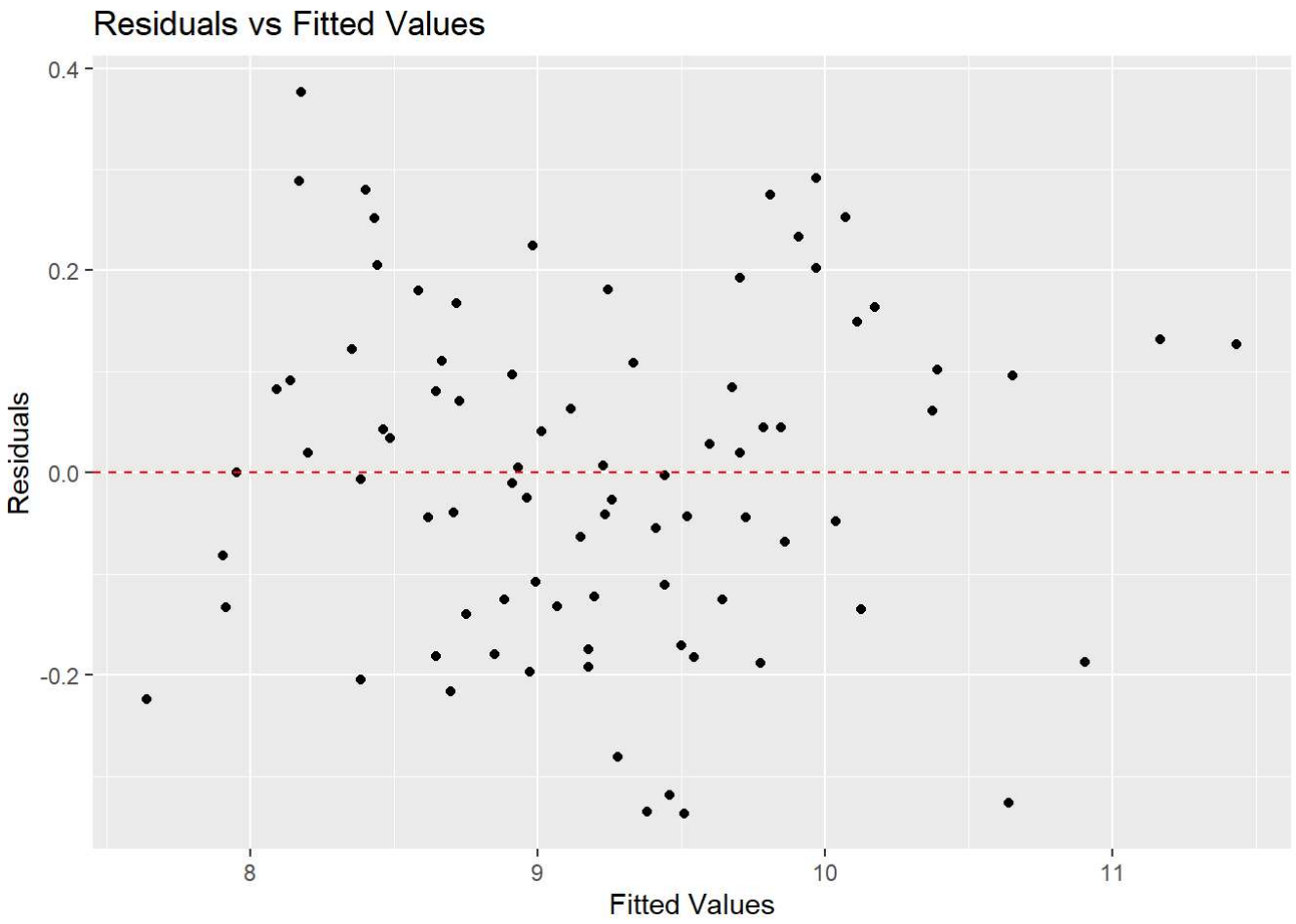
```
##
## Call:
## lm(formula = log_fancy ~ ., data = predictors)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33673 -0.12757  0.00257  0.10911  0.37671
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      9.5819082  0.0784330 122.167  < 2e-16 ***
## time_index       0.0220198  0.0008268  26.634  < 2e-16 ***
## factor.month.1  -1.9622412  0.0961066 -20.417  < 2e-16 ***
## factor.month.2  -1.7108244  0.0960319 -17.815  < 2e-16 ***
## factor.month.3  -1.6961584  0.1949340  -8.701 9.35e-13 ***
## factor.month.4  -1.5781877  0.0959037 -16.456  < 2e-16 ***
## factor.month.5  -1.5527543  0.0958503 -16.200  < 2e-16 ***
## factor.month.6  -1.5134129  0.0958039 -15.797  < 2e-16 ***
## factor.month.7  -1.3517867  0.0957647 -14.116  < 2e-16 ***
## factor.month.8  -1.3742768  0.0957325 -14.355  < 2e-16 ***
## factor.month.9  -1.2929114  0.0957075 -13.509  < 2e-16 ***
## factor.month.10 -1.2148493  0.0956897 -12.696  < 2e-16 ***
## factor.month.11 -0.7554933  0.0956790  -7.896 2.84e-11 ***
## factor.month.12         NA         NA      NA       NA
## festival         0.5015151  0.1964273   2.553   0.0129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.179 on 70 degrees of freedom
## Multiple R-squared:  0.9567, Adjusted R-squared:  0.9487
## F-statistic:   119 on 13 and 70 DF,  p-value: < 2.2e-16
```

(iv) Plot the residuals against time and against the fitted values. Do these plots reveal any problems with the model?

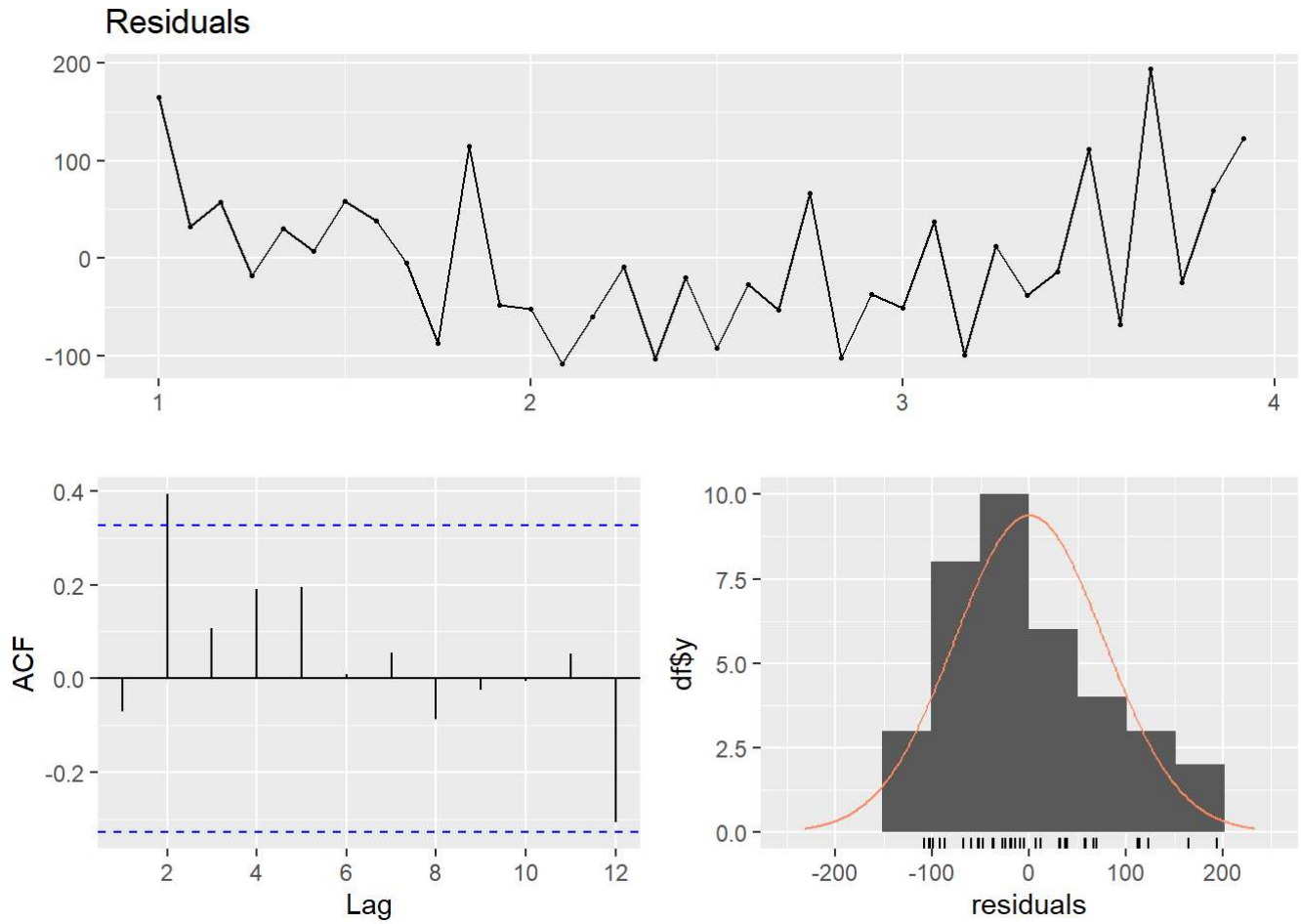
```
plot_data <- data.frame(FittedValues=fitted(fancy_lm), Residuals=residuals(fancy_lm))
plot(residuals)
```




```
ggplot(plot_data, aes(x = FittedValues, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals")
```




```
checkresiduals(residuals)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 10.224, df = 7, p-value = 0.1762
##
## Model df: 0.   Total lags used: 7
```

Residuals V/S Fitted Values

- No specific pattern can be observed in the graph indicating that there is no heterscedasticity depicting, there is no variance across the data.

Acf Plot

- Values of the Acf plot have crossed the confidence level indicating there is autocorrelation, and since the values also lead to the negative indicating there is seasonality in the data.

(v) Perform a Breusch-Godfrey test. What does it tell you?

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.1.3
```

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

```
## Warning: package 'zoo' was built under R version 4.1.3
```

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

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

```
bg_test <- bgtest(fancy_lm, order = 1)
print(bg_test)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  fancy_lm
## LM test = 25.031, df = 1, p-value = 5.642e-07
```

- The Breusch-Godfrey test gives us a very small p-value (<0.05) which suggests that there is an autocorrelation between the residuals.

(vi) Notwithstanding your answers to the questions above, create a forecast for monthly sales data in 1994. You will need to produce new data for the dummy variable to use in the forecast. One way to do this is to use the following code.

> future.festival <- rep(0, 12)

> future.festival[3] <- 1

```
length_fancy <- length(fancy)

future.festival <- rep(0, 12)
future.festival[3] <- 1

future_time_index <- (length_fancy + 1):(length_fancy + 12)

future_months <- rep(1:12, each=1)

future_data <- data.frame(
 time_index = future_time_index,
 festival = future.festival
)

for (i in 1:12) {
 future_data[paste0("factor.month.", i)] <- ifelse(future_months == i, 1, 0)
}

sales_forecast_1994 <- predict(fancy_lm, newdata = future_data)
```

```
## Warning in predict.lm(fancy_lm, newdata = future_data): prediction from a
## rank-deficient fit may be misleading
```

```
print(sales_forecast_1994)
```

```
##         1         2         3         4         5         6         7         8
##  9.491352  9.764789 10.302990  9.941465  9.988919 10.050280 10.233926 10.233456
##         9        10        11        12
## 10.336841 10.436923 10.918299 11.695812
```

(vii) Transform the forecast to obtain predictions for the original (untransformed) data.

Hint: for a forecast fc, you can extract the predictions using the code fc$mean

```
original_untransformed_data <- exp(sales_forecast_1994)
original_untransformed_data <- data.frame(original_untransformed_data)
print(original_untransformed_data)
```

```
##    original_untransformed_data
## 1                     13244.70
## 2                     17409.81
## 3                     29821.65
## 4                     20774.16
## 5                     21783.73
## 6                     23162.27
## 7                     27831.56
## 8                     27818.48
## 9                     30848.42
## 10                    34095.57
## 11                    55176.84
## 12                   120067.79
```

(viii) Recall that applying a log transformation to a time series is equivalent to using a Box-Cox transformation with ?? = 0. Check your answer to part (vii) by using the tslm function on the original (untransformed) data and specifying ?? = 0.

```
library(forecast)

fancy_tslm <- tslm(fancy ~ trend + season + I(festival), lambda = 0)

future_data <- data.frame(
 trend = seq(max(time(fancy)), by = 1/12, length.out = 12),
 festival = future.festival
)

boxcox_forecast_1994 <- forecast(fancy_tslm, newdata = future_data)
print(boxcox_forecast_1994)
```

```
##          Point Forecast        Lo 80        Hi 80        Lo 95        Hi 95
## Jan 1994   2.383328e+22 2.895980e+21 1.961427e+23 9.247855e+20 6.142239e+23
## Feb 1994   3.070221e+22 3.734258e+21 2.524266e+23 1.193105e+21 7.900610e+23
## Mar 1994   5.153962e+22 6.312981e+21 4.207731e+23 2.024722e+21 1.311949e+24
## Apr 1994   3.518576e+22 4.287926e+21 2.887264e+23 1.371449e+21 9.027222e+23
## May 1994   3.615842e+22 4.410753e+21 2.964191e+23 1.411478e+21 9.262853e+23
## Jun 1994   3.767837e+22 4.600640e+21 3.085787e+23 1.473020e+21 9.637747e+23
## Jul 1994   4.436928e+22 5.422898e+21 3.630224e+23 1.737204e+21 1.133219e+24
## Aug 1994   4.346223e+22 5.317211e+21 3.552550e+23 1.704246e+21 1.108388e+24
## Sep 1994   4.723300e+22 5.784158e+21 3.857010e+23 1.854888e+21 1.202744e+24
## Oct 1994   5.116163e+22 6.271362e+21 4.173754e+23 2.012187e+21 1.300829e+24
## Nov 1994   8.114038e+22 9.955829e+21 6.612972e+23 3.196045e+21 2.059971e+24
## Dec 1994   1.730376e+23 2.125219e+22 1.408891e+24 6.826032e+21 4.386446e+24
```

Exercise 4

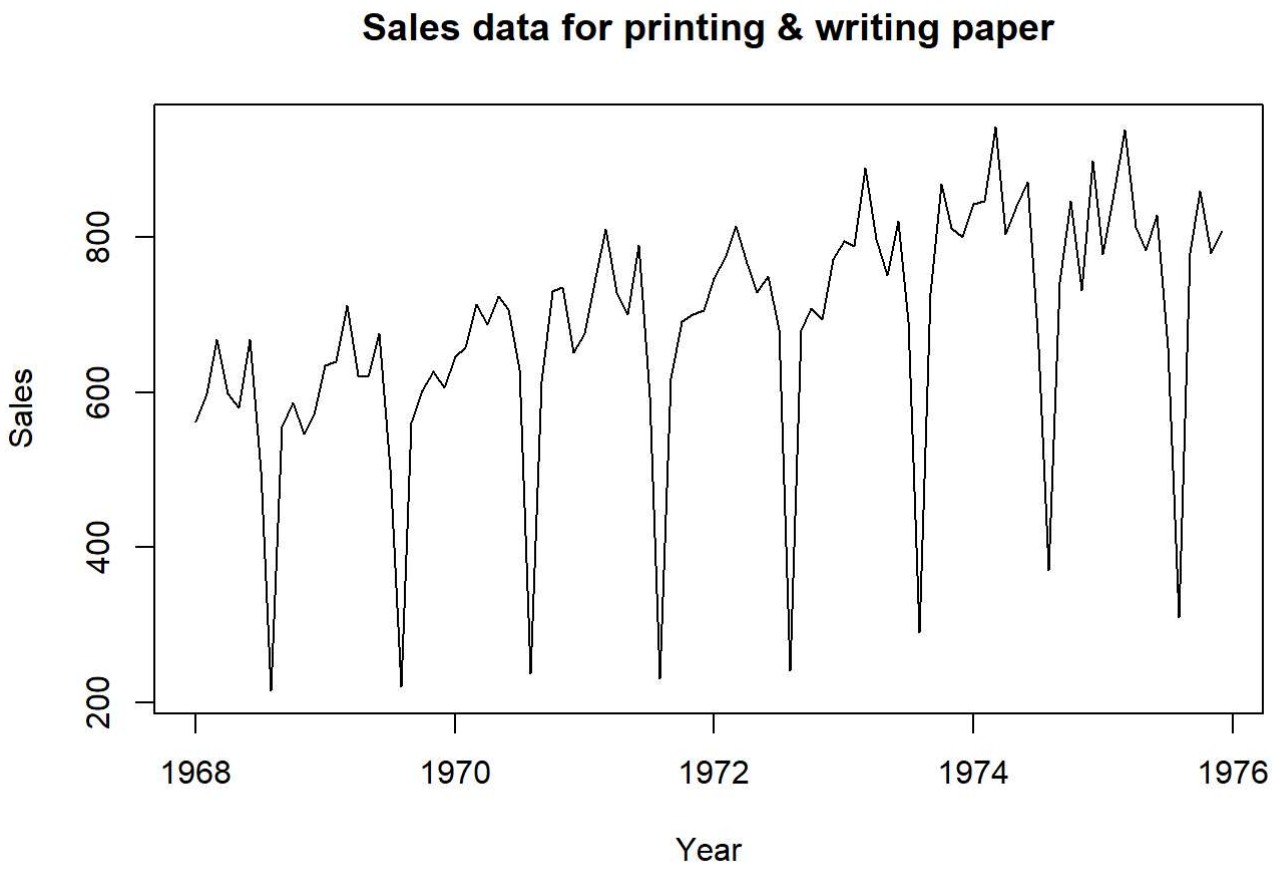
writing

(i) Split the data into a training set from the beginning of 1968 to the end of 1975 and a test set from the beginning of 1976 to the end of 1977

```
writing_training <- window(writing, start=c(1968,1), end=c(1975,12))
writing_testing <- window(writing, start=c(1976,1), end=c(1977,12))
```

(ii)Plot the sales data for the training set. Propose an appropriate regression model based on the patterns you see in the plot.

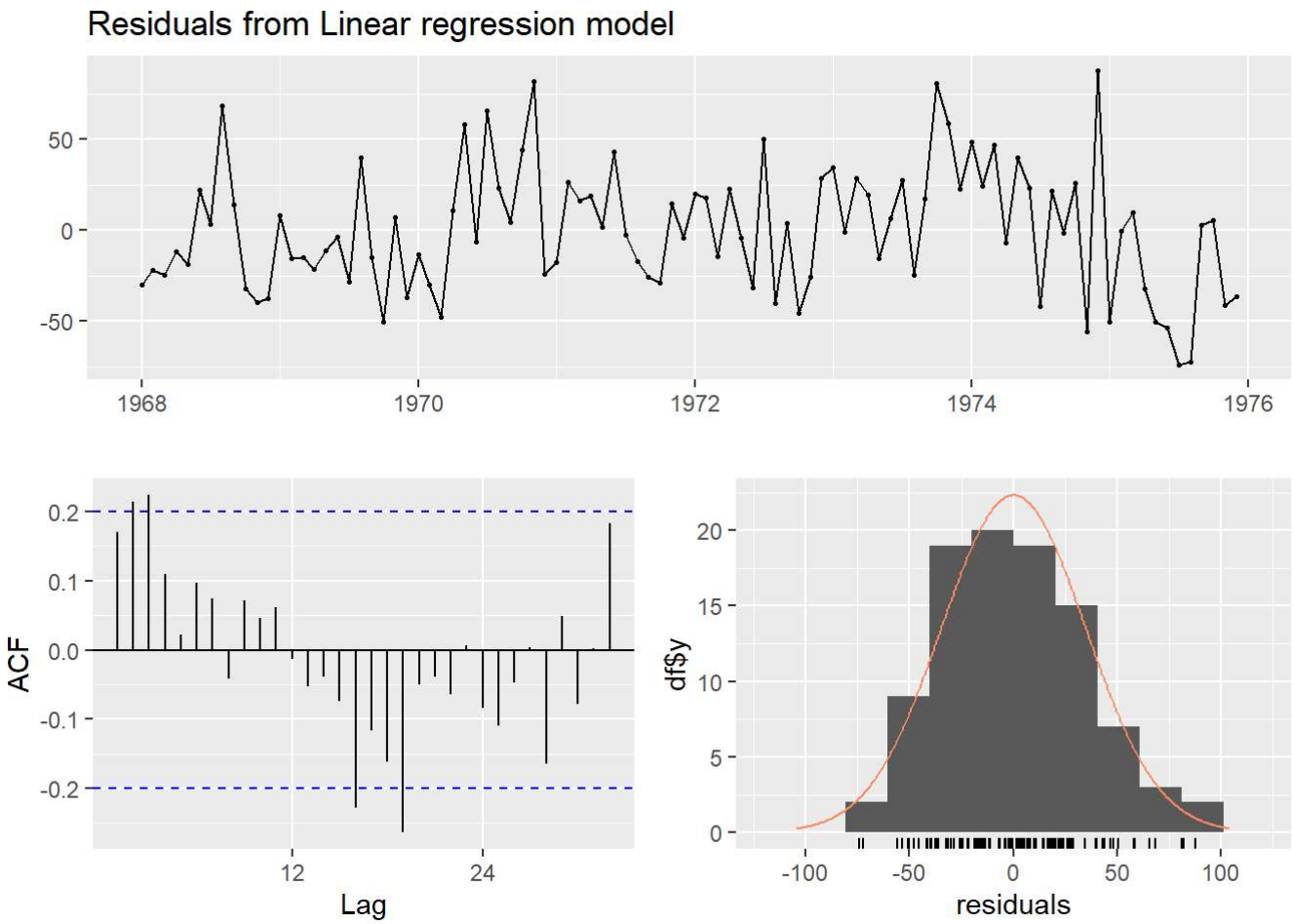
```
plot(writing_training, main ='Sales data for printing & writing paper', xlab='Year', ylab='Sale
s')
```



- The plot clearly indicates *seasonality* and *trend* with frequent increases and decreases during specific periods through a year.
- Proposed model to better forecast would be *Time Series Linear Regression* - it can incorporate both seasonality and trend.

(iii) For your chosen model, check for autocorrelation and seasonality in the residuals. Do the residuals appear to be normally distributed?

```
writing_model <- tslm(writing_training ~ trend+ season)

forecast_training <- forecast(writing_model)

checkresiduals(forecast_training)
```

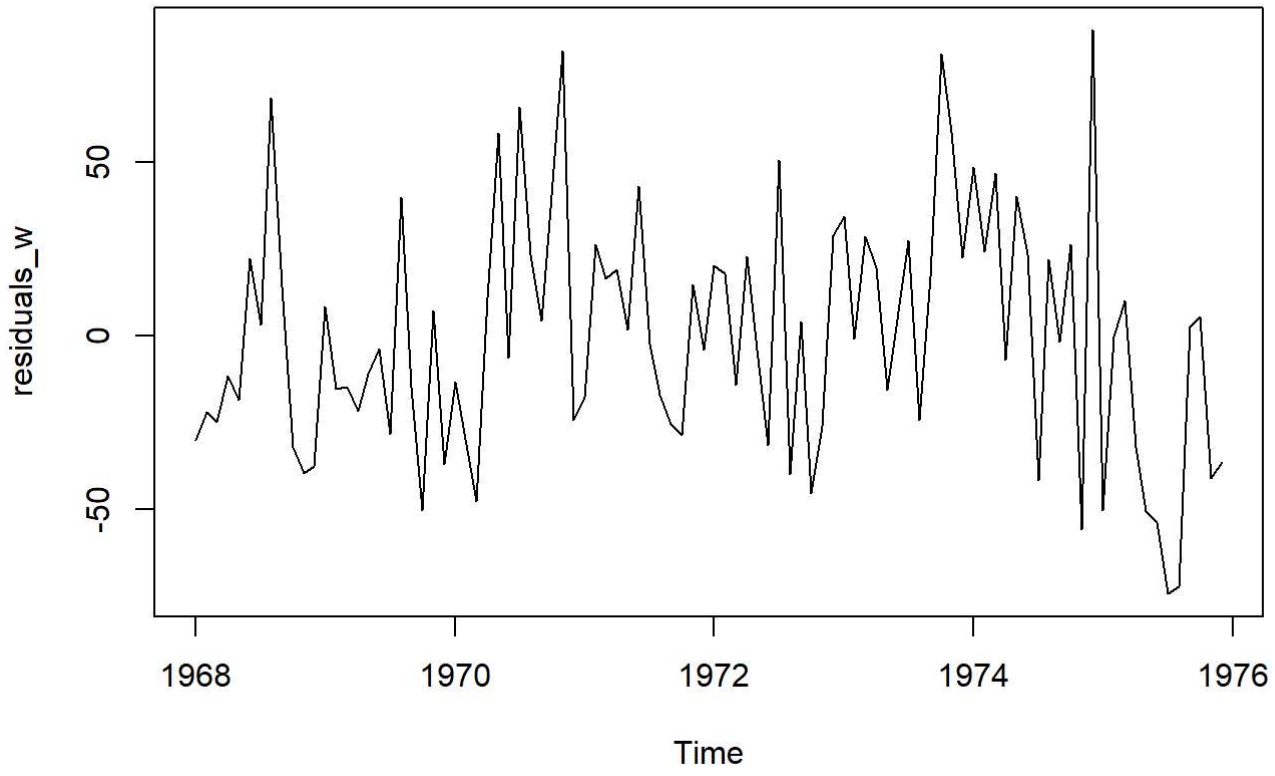



```
##
##  Ljung-Box test
##
## data:  Residuals from Linear regression model
## Q* = 37.286, df = 19, p-value = 0.007307
##
## Model df: 0.   Total lags used: 19
```

- Based on the ACF plot, it can be inferred that since the values of ACF exceed the confidence levels - there is autocorrelation between the residuals.
- Secondly, the ACF values have positive and negative values indicating seasonality.
- The residuals are not normally distributed.

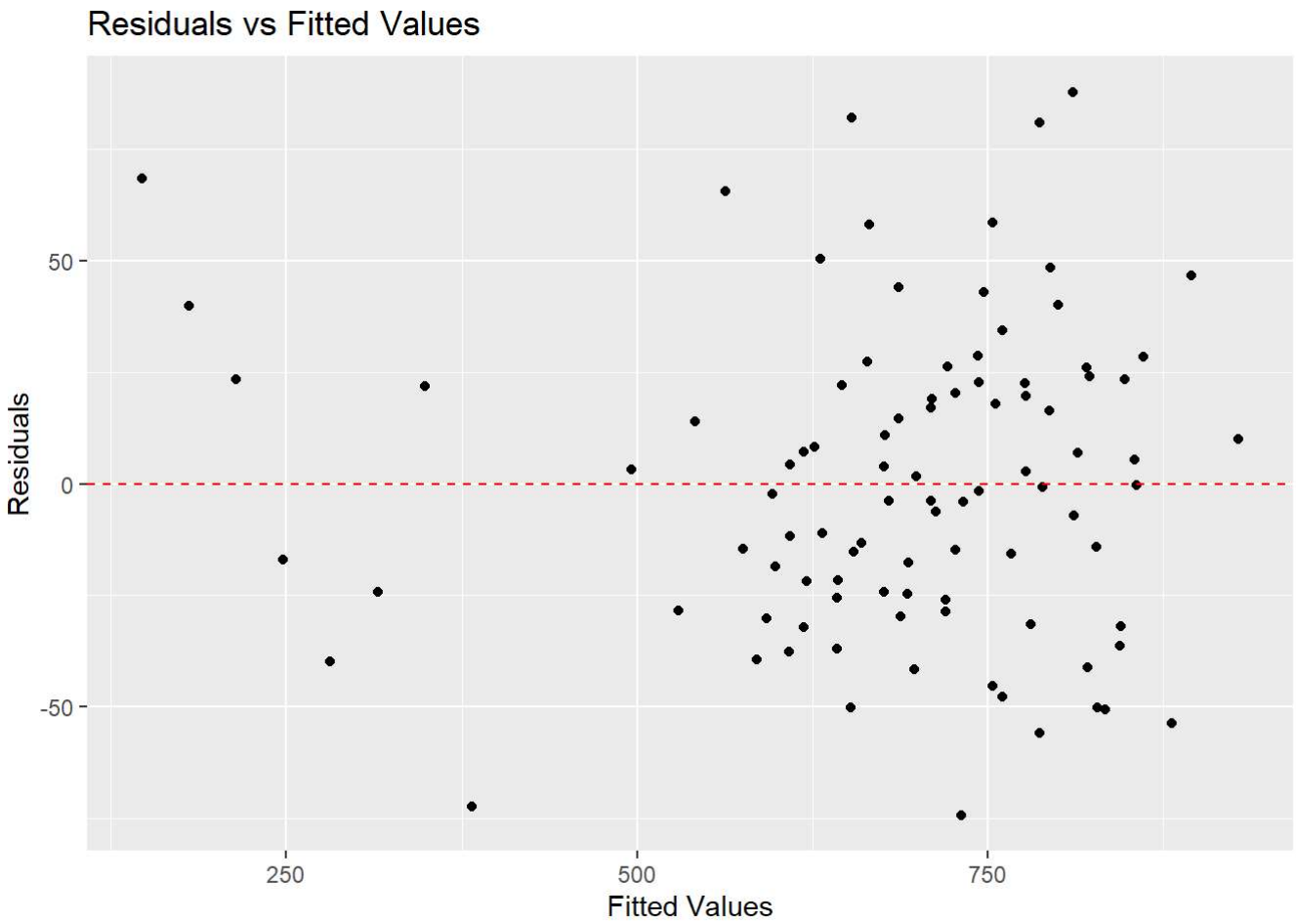
(iv) Plot the residuals against time and against the fitted values. Do these plots reveal any problems with the model

```
residuals_w <- residuals(writing_model)

plot_data <- data.frame(Residuals = residuals_w, FittedValues = fitted(writing_model))

plot(residuals_w)
```




```
ggplot(plot_data, aes(x = fitted(writing_model), y = residuals_w)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals")
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



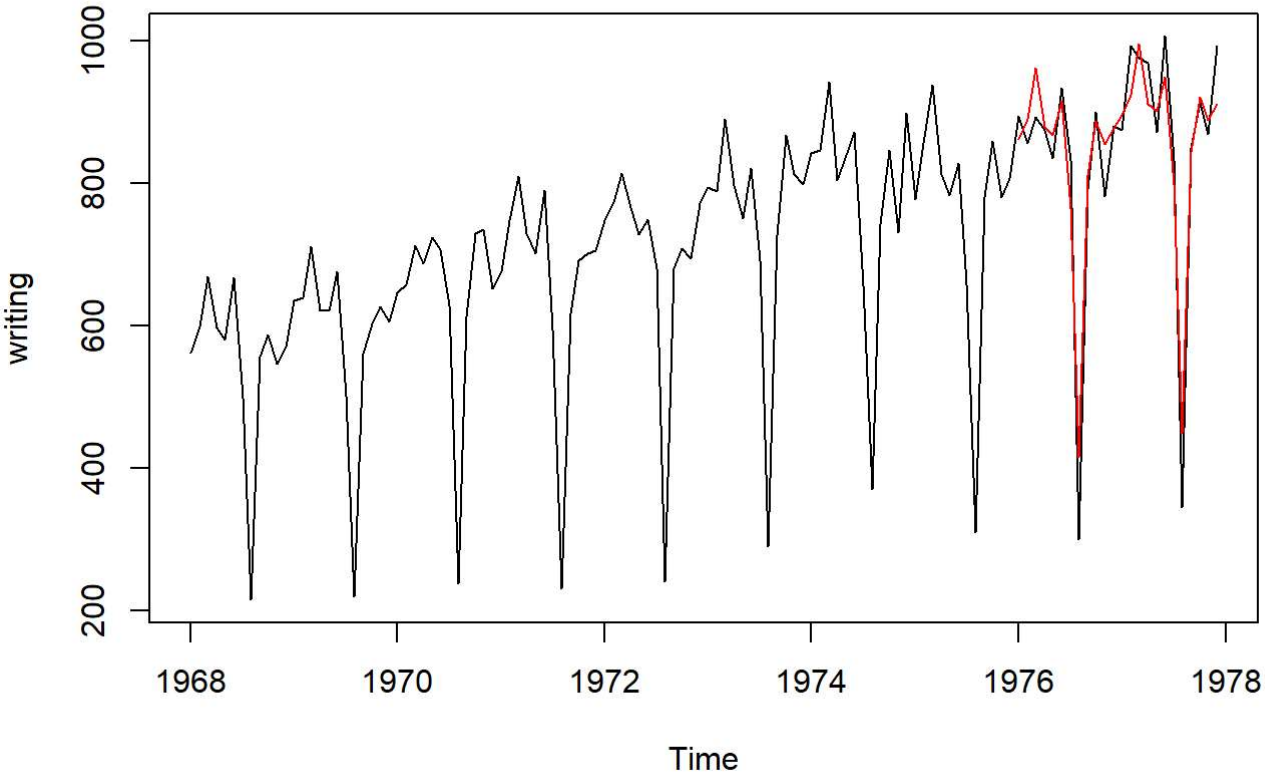
- The data shows a funnel pattern, and therefore it has heterscedasticity.

(v) Create a forecast for 1976 and 1977 using your model, and plot it alongside the full data set from 1968 to 1977.

```
train_forecast <- forecast(writing_model, h = 24)
train_forecast
```

```
##          Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## Jan 1976       861.9835 810.1056  913.8614 782.1103  941.8567
## Feb 1976       890.1487 838.2709  942.0266 810.2755  970.0220
## Mar 1976       962.5275 910.6496 1014.4054 882.6543 1042.4007
## Apr 1976       878.6580 826.7801  930.5359 798.7848  958.5312
## May 1976       867.6132 815.7354  919.4911 787.7400  947.4865
## Jun 1976       915.3710 863.4931  967.2489 835.4978  995.2442
## Jul 1976       765.2211 713.3433  817.0990 685.3479  845.0943
## Aug 1976       415.9665 364.0886  467.8444 336.0933  495.8397
## Sep 1976       810.9750 759.0971  862.8529 731.1018  890.8482
## Oct 1976       888.3156 836.4378  940.1935 808.4424  968.1888
## Nov 1976       854.7755 802.8976  906.6534 774.9023  934.6487
## Dec 1976       877.9686 826.0908  929.8465 798.0954  957.8418
## Jan 1977       895.6333 843.3156  947.9510 815.0829  976.1837
## Feb 1977       923.7986 871.4809  976.1163 843.2482 1004.3490
## Mar 1977       996.1773 943.8596 1048.4950 915.6269 1076.7277
## Apr 1977       912.3078 859.9901  964.6255 831.7574  992.8582
## May 1977       901.2631 848.9454  953.5808 820.7127  981.8135
## Jun 1977       949.0208 896.7031 1001.3385 868.4704 1029.5712
## Jul 1977       798.8710 746.5533  851.1887 718.3206  879.4214
## Aug 1977       449.6163 397.2986  501.9340 369.0659  530.1667
## Sep 1977       844.6248 792.3071  896.9425 764.0744  925.1752
## Oct 1977       921.9655 869.6478  974.2832 841.4151 1002.5159
## Nov 1977       888.4253 836.1076  940.7430 807.8749  968.9757
## Dec 1977       911.6185 859.3008  963.9362 831.0681  992.1689
```

```
plot(writing)
  lines(train_forecast$mean, col = "red")
```



(vi) Check the accuracy of the forecast by comparing the errors to the standard deviation of the forecast variable in the test set. How well does your model perform?

```
residuals <- residuals(writing_model)

stddev_test <- sd(writing_testing)

accuracy <- (mean(abs(residuals) / stddev_test))*100

cat("Accuracy of the model:", round((accuracy),2),'%')
```

```
## Accuracy of the model: 16.32 %
```

- Based on the accuracy observed, it can inferred that the model performs poorly.