Forecasting, Homework 2

Robby Jeffries

2/14/2022

Set working directory

```
rm(list = ls(all.names = TRUE))

setwd("~/MSEA2022/Spring 2022/ECON 5753, Forecasting")
```

Import packages and install them if necessary

```
list.of.packages <- c("tidyverse", "caTools", "pastecs", "ggplot2", "olsrr", "ggplotify", "lmtest")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
library(tidyverse)
library(caTools)
library(pastecs)
library(ggplot2)
library(olsrr)
library(ggplotify)
library(lmtest)
```

Import data

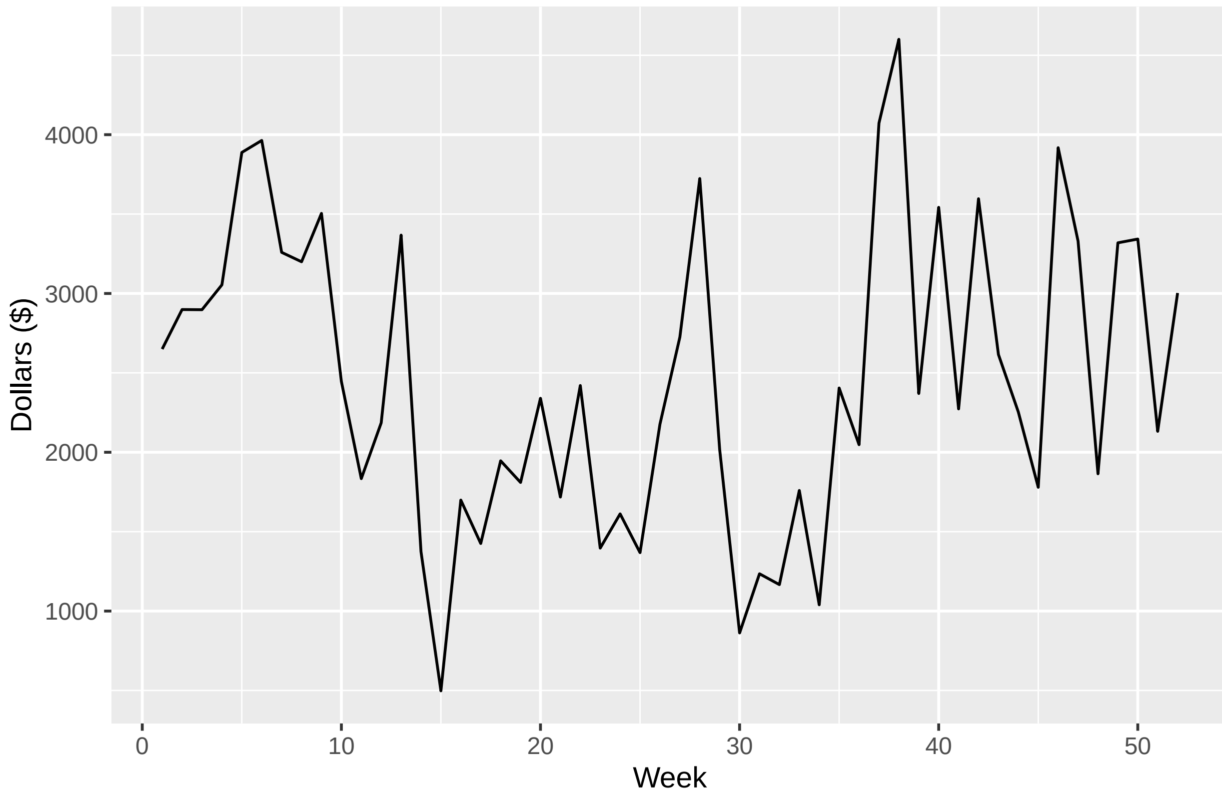
```
df = read.csv("Data/tablep21_hw2.csv")
```

1) Plot the sales data as a time series. Figure 1.

```
fig1 <- ggplot2::ggplot(df, aes(x=week, y=sales)) +
  geom_line() +
  xlab("Week") +
  ylab("Dollars ($)") +
  ggtitle("Figure 1: Sales Data as a Time Series")

fig1
```

Figure 1: Sales Data as a Time Series



2) Do you think this series is stationary or nonstationary? Explain.

I believe that the series is nonstationary for the range of data provided. The plot appears to indicate seasonality with a higher average dollar amount in the fourth quarter of the year. It is important to note that the plot does not appear to represent more than one complete cycle. It would be interesting to compare this plot to a two-year plot to confirm seasonality within the data. For example, year over year, if the seasonality is consistent, the data would be stationary.

With only one year of data, however, I observe a non-constant mean. Therefore, the data provided appears nonstationary.

3) Compute the autocorrelation coefficients of the sales series for the first 10 time lags. Is the behavior of the autocorrelation consistent with your conclusion of part 2? Explain.

```
# Ask R if the data is time series or not
is.ts(df$sales)
```

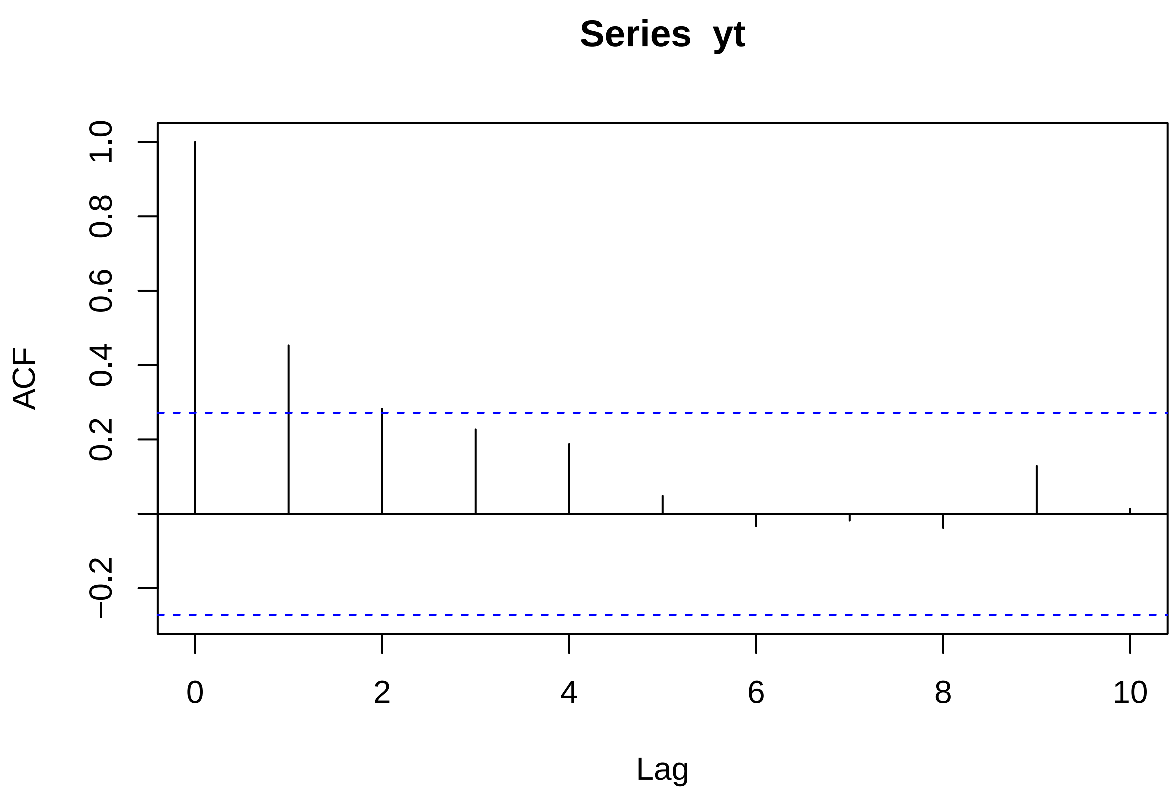
```
## [1] FALSE
```

```
# Change to time series data
yt <- ts(df$sales, start = 1, frequency = 1)
```

```r
# Ask R if the new yt variable is time series or not
is.ts(yt)
```

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

```r
# Autocorrelation function
rho <- acf(yt, lag.max = 10)
```




```r
rho
```

```
##
## Autocorrelations of series 'yt', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.453  0.283  0.227  0.188  0.049 -0.033 -0.018 -0.038  0.129  0.014
```

The autocorrelation plot actually seems to indicate a stationary time series. As the lag increases, the autocorrelation coefficient approaches zero. This demonstrates exponential decay, which is what indicates a stationary time series. According to the plot, lag 1 and lag 2 extend above the confidence interval line, but the correlation coefficients quickly lower to within the confidence interval.

Correlation coefficients above the confidence interval line indicate non-stationary data, but since only two lags are above the line, it's possible that these two are outliers. To confidently label the data as non-stationary, I would want to perform a similar test across a longer time-period.

Reference: https://coolstatsblog.com/2013/08/07/how-to-use-the-autocorreation-function-acf/

4) Calculate the mean of Sales

```r
avg_sales <- mean(df$sales)
avg_sales
```

```
## [1] 2460.056
```

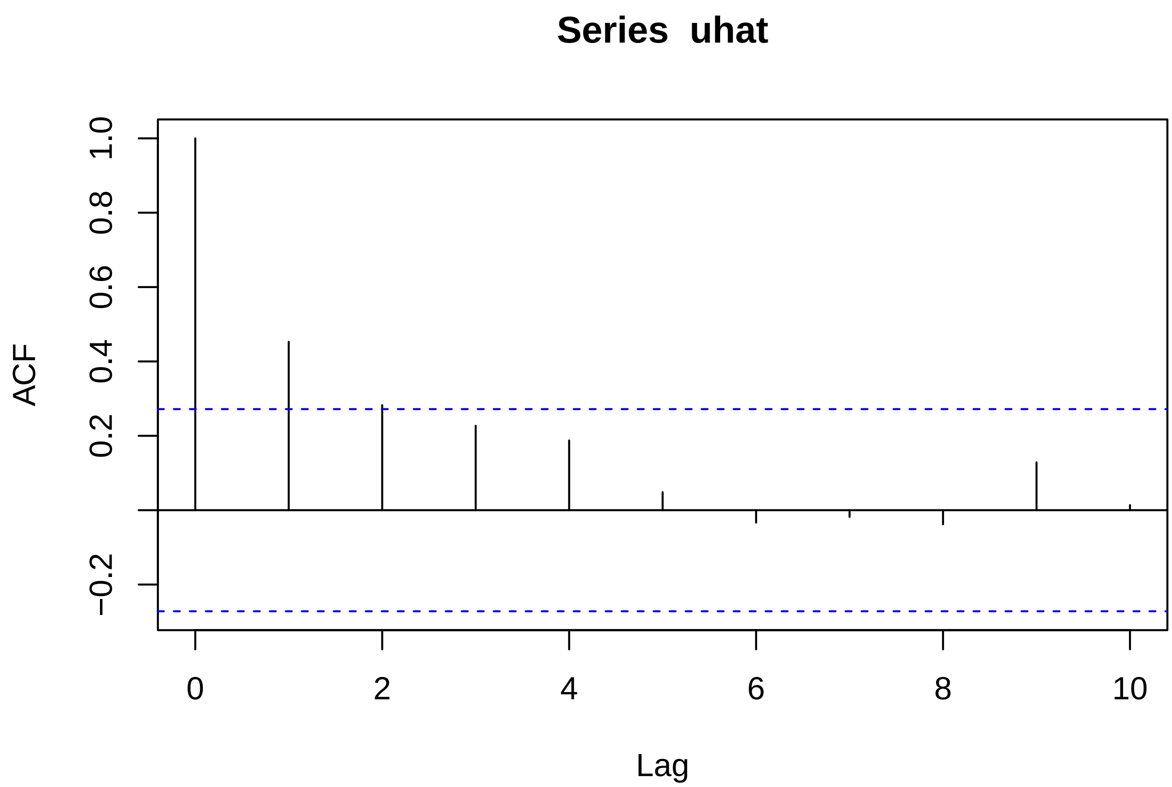
5) Fit a random model and compute residual

```r
m1 <- lm(sales ~ 1, data = df)  # Create a linear model
uhat <- resid(m1) # List of residuals
uhat
```

```
##           1           2           3           4           5           6 
##   189.84423   438.64423   437.74423   594.24423  1428.04423  1503.54423 
##           7           8           9          10          11          12 
##   798.84423   739.54423  1044.24423   -14.15577  -626.15577  -274.65577 
##          13          14          15          16          17          18 
##   907.34423 -1085.95577 -1962.55577  -761.05577 -1034.65577  -513.85577 
##          19          20          21          22          23          24 
##  -650.15577  -120.15577  -742.15577   -39.75577 -1063.55577  -847.95577 
##          25          26          27          28          29          30 
## -1092.15577  -283.25577   264.94423  1263.64423  -444.05577 -1597.85577 
##          31          32          33          34          35          36 
## -1225.15577 -1293.55577  -700.55577 -1420.65577   -55.25577  -412.25577 
##          37          38          39          40          41          42 
##  1612.54423  2140.44423   -89.95577  1082.24423  -187.05577  1136.54423 
##          43          44          45          46          47          48 
##   155.74423  -206.75577  -680.65577  1457.84423   869.24423  -595.65577 
##          49          50          51          52 
##   858.84423   882.54423  -328.15577   543.14423
```

6) Compute the autocorrelation function of epsilon_t for the first 10 time lags

```r
rho2 <- acf(uhat, lag.max = 10)
```




```
rho2
```

```
##
## Autocorrelations of series 'uhat', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.453  0.283  0.227  0.188  0.049 -0.033 -0.018 -0.038  0.129  0.014
```

7) Is the random model adequate for the sales data? (Review assumptions of OLS)

No. According to OLS assumptions, error terms should be independent. That is, the error terms should not be autocorrelated. The autocorrelation plot shows that the the correlation coefficient extends beyond the confidence interval for lags 1 and 2. This indicates that the OLS assumption fails and the random model is not adequate for the sales data.

Bonus Graphs

Throughout this assignment, I came across several packages and graphs that were helpful for my understanding. These are included below for reference, though I did not explicitly reference them in my solutions.

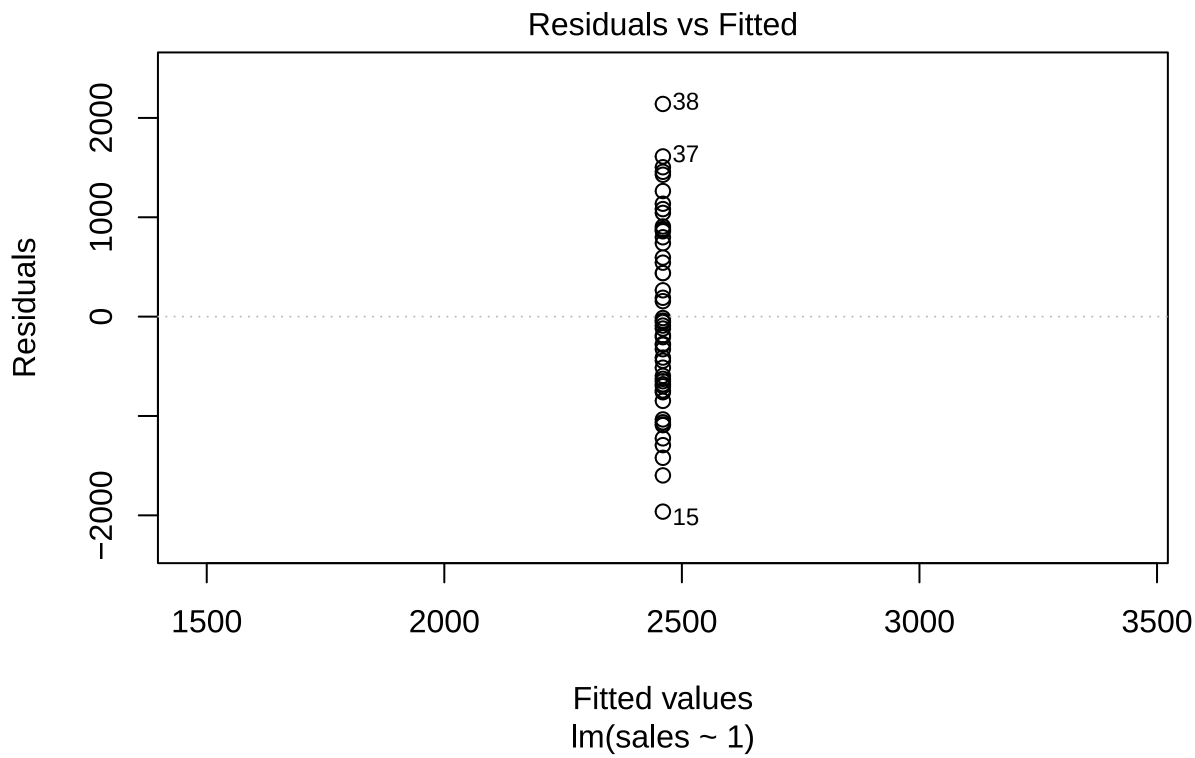
```
summary(m1)
```

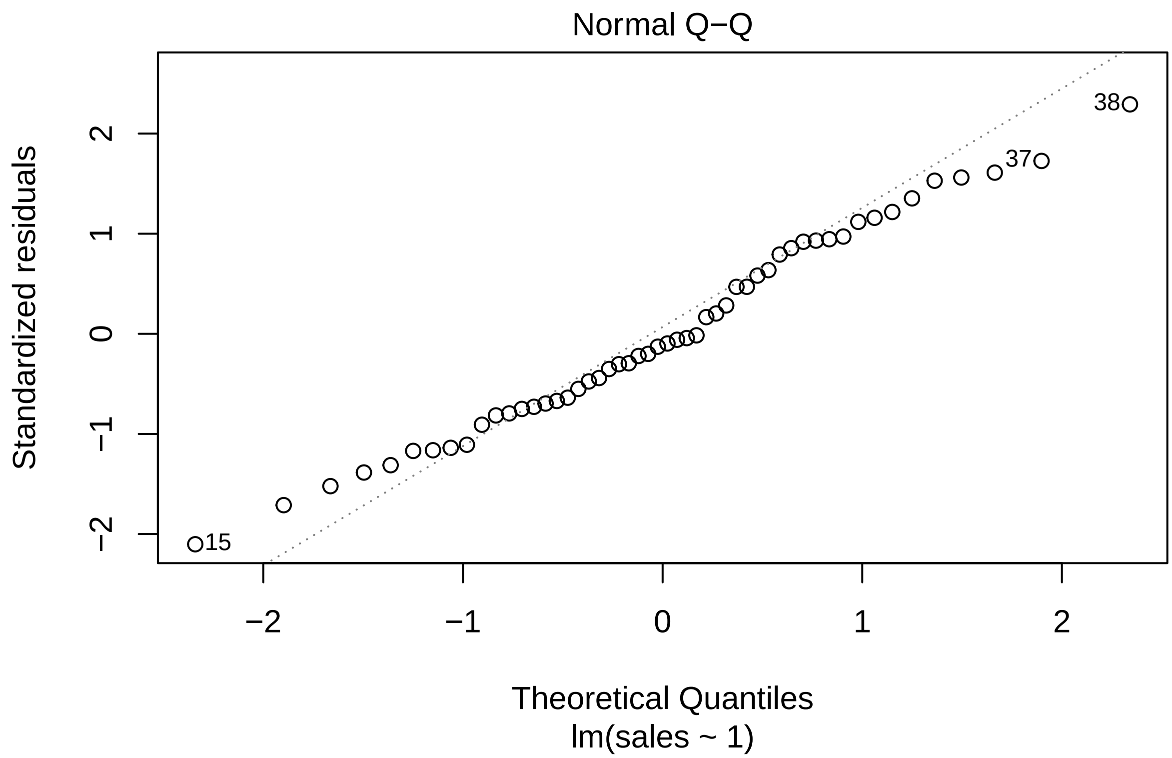
```
##
## Call:
## lm(formula = sales ~ 1, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1962.6  -685.6  -105.1   813.8  2140.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2460.1      130.8   18.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 942.9 on 51 degrees of freedom
```

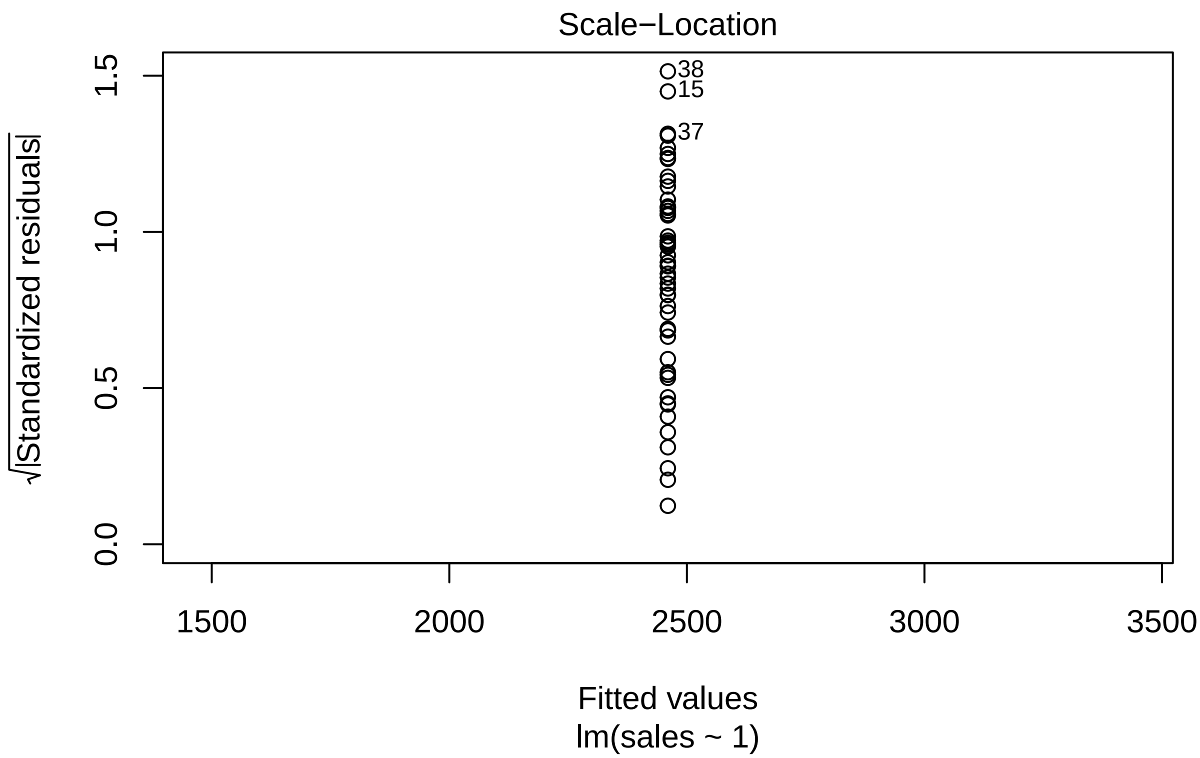
```
olsrr::ols_test_normality(m1)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9829         0.6559
## Kolmogorov-Smirnov        0.0829         0.8380
## Cramer-von Mises          4.641          0.0000
## Anderson-Darling          0.3484         0.4637
## -----------------------------------------------
```

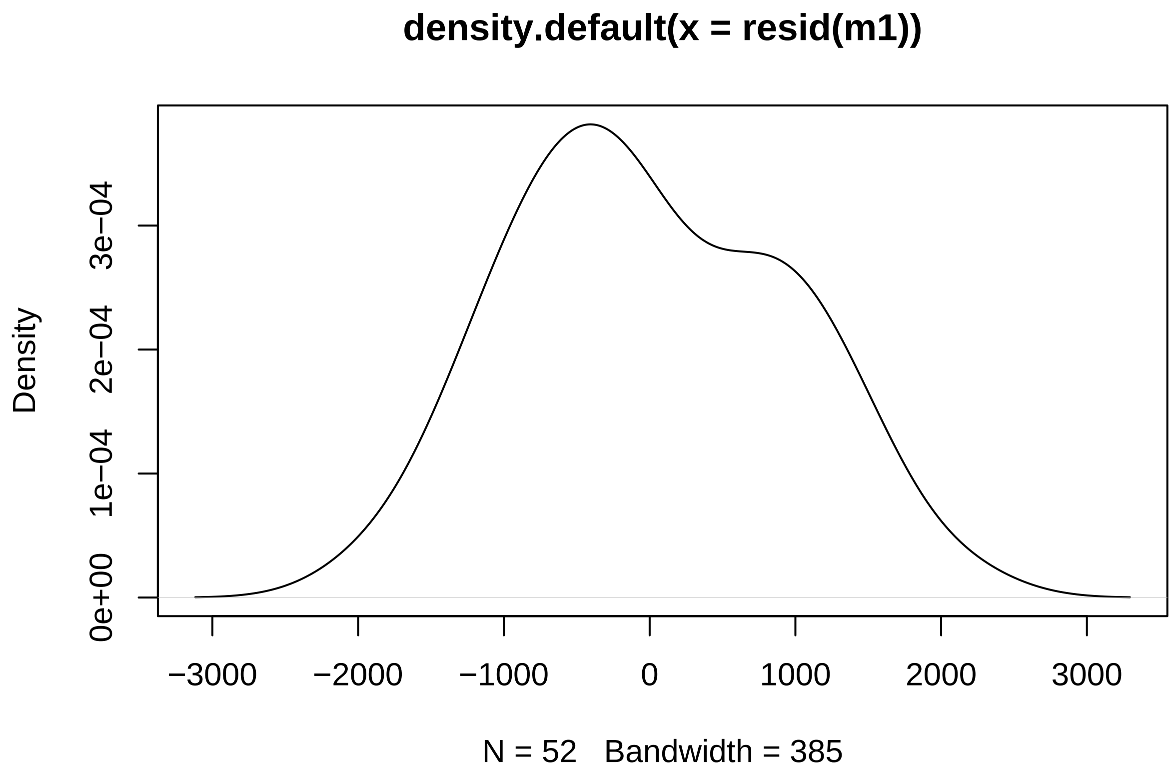
```
plot(m1)
```


Residuals vs Fitted
Residuals
2000
1000
0
−2000
38
37
15
1500
2000
2500
3000
3500
Fitted values
lm(sales ~ 1)

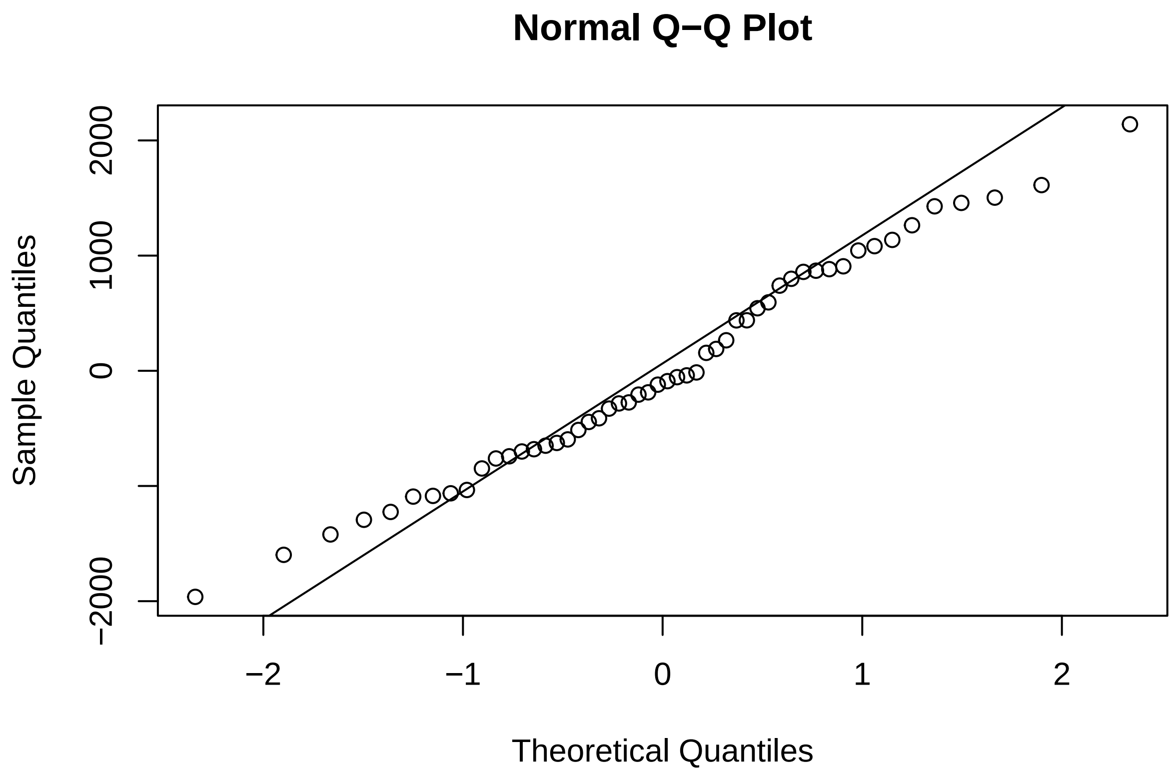

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(sales ~ 1)
15
37
38
−2
−1
0
1
2




```
plot(density(resid(m1))) # A density plot
```




```r
qqnorm(resid(m1)) # A quantile normal plot - good for checking normality
qqline(resid(m1))
```


Normal Q−Q Plot
Sample Quantiles
2000
1000
0
−2000
−2
−1
0
1
2
Theoretical Quantiles