DSCI5340_HW2_Group13

Sai Vamshi Palakurthi, Kiran Kumar, Pavan Naramreddy, Sai Hari Vignesh

2024-02-17

Load all the required libraries.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(forecast)
```

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

Load the data and view the dataset contents.

```
library(fpp3)
```

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

## v tibble      3.2.1     v feasts      0.3.1
## v lubridate   1.9.3     v fable       0.3.3
## v tsibble     1.1.4     v fabletools  0.4.0
## v tsibbledata 0.4.1

## -- 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()
```

```
data(insurance)
data <- insurance
```

The head() gives us the dataset information

```
head(insurance)
```

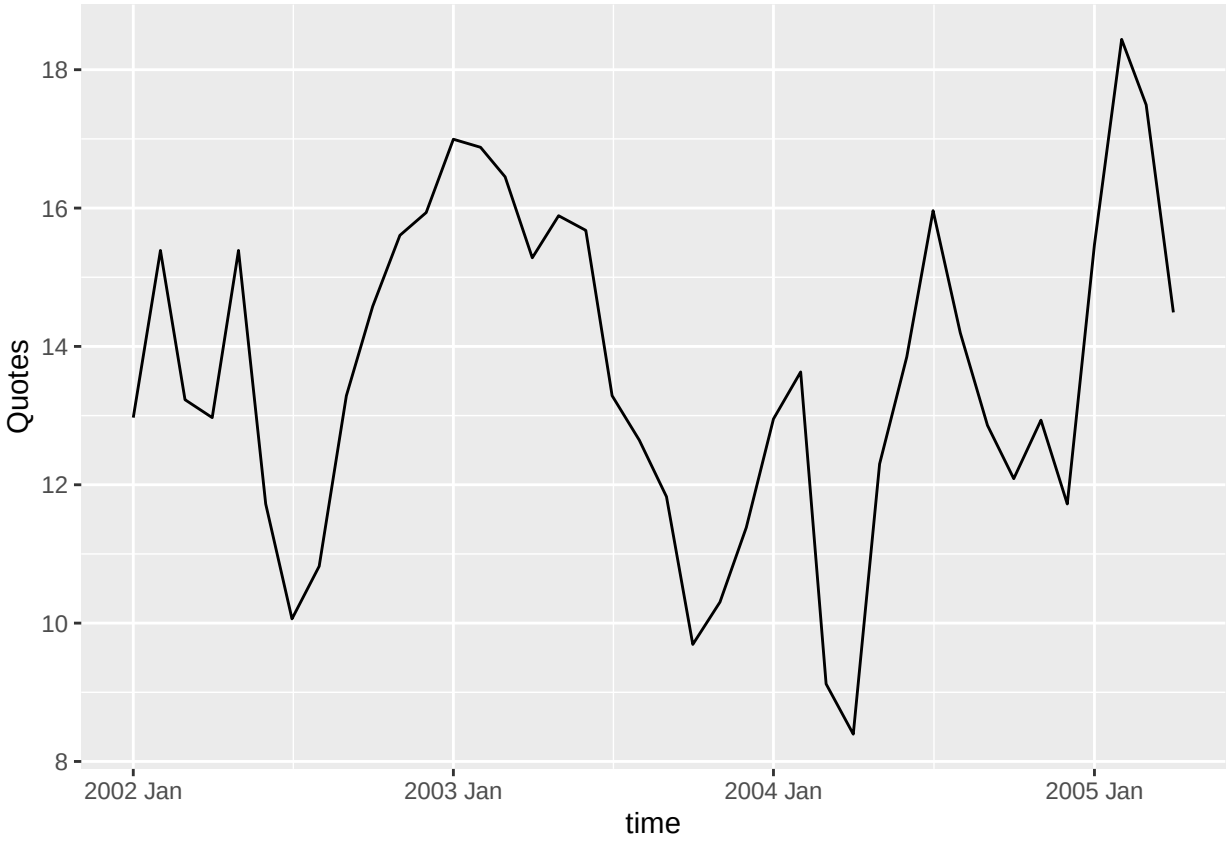
```
## # A tsibble: 6 x 3 [1M]
##      Month Quotes TVadverts
##      <mth>  <dbl>     <dbl>
## 1 2002 Jan   13.0      7.21
## 2 2002 Feb   15.4      9.44
## 3 2002 Mar   13.2      7.53
## 4 2002 Apr   13.0      7.21
## 5 2002 May   15.4      9.44
## 6 2002 Jun   11.7      6.42
```

Question 1 - Produce a time plot of the data and describe the patterns. Identify any unusual or unexpected fluctuations in the time series. Plot the data

We have used autoplot to plot.

```
insurance %>%
  autoplot() +
  xlab("time") +
  ylab("Quotes")
```

```
## Plot variable not specified, automatically selected `.vars = Quotes`
```



Here, after the year 2002 I can see a downtrend, after the year Jan 2004 I can see deep downtrend, which is greater than the years between 2002 and 2003, and after the year 2005 Jan we can see the highest uptend.

Question 2 - Fit a regression model with Quotes as the dependent variable and a linear trend and seasonal dummies as explanatory variables

Here, TSLM means Time Series Linear Regression, season() for seasonal dummies

```
library(fpp3)
fit_regression <- insurance %>%
  model(TSLM(Quotes ~ trend() + season()))
report(fit_regression)
```

```
## Series: Quotes
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.01858 -1.60766  0.07939  1.61455  3.22002
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    14.38763    1.43309  10.040  1.3e-10 ***
## trend()         0.01102    0.03521   0.313    0.757
## season()year2   1.47572    1.79272   0.823    0.418
## season()year3  -0.54569    1.79376  -0.304    0.763
## season()year4  -1.84559    1.79548  -1.028    0.313
```

```
## season()year5  -0.04938    1.93726  -0.025    0.980
## season()year6  -0.83649    1.93630  -0.432    0.669
## season()year7  -1.49306    1.93598  -0.771    0.447
## season()year8  -2.05308    1.93630  -1.060    0.298
## season()year9  -1.96111    1.93726  -1.012    0.320
## season()year10 -2.51062    1.93886  -1.295    0.206
## season()year11 -1.69338    1.94110  -0.872    0.391
## season()year12 -1.63884    1.94397  -0.843    0.407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.535 on 27 degrees of freedom
## Multiple R-squared: 0.2273,  Adjusted R-squared: -0.1161
## F-statistic: 0.6619 on 12 and 27 DF, p-value: 0.77112
```

Question 3 - Create a plot showing two lines – a fitted line from the above regression and a line with actual quotes. What do you observe in this plot?

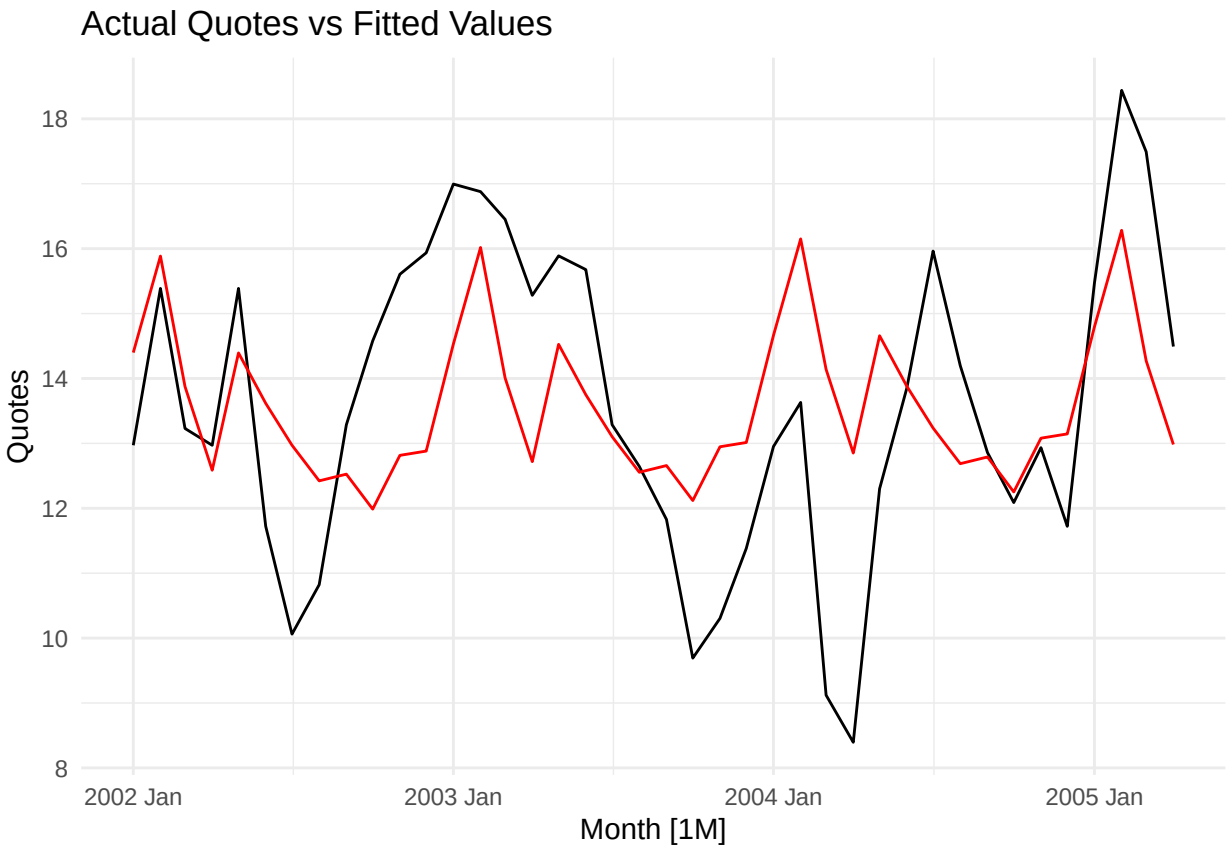
```
fitted_regression <- augment(fit_regression)
```

The below plot gives us the information shwoing two lines a fitted line from ques 2 and and quotes from insurance dataset

```
autoplot(insurance, series = "Quotes") +
  geom_line(data = fitted_regression, aes(y = .fitted), color = "red") +
  labs(title = "Actual Quotes vs Fitted Values",
       y = "Quotes") +
  theme_minimal()
```

```
## Plot variable not specified, automatically selected `.vars = Quotes`
```

```
## Warning in geom_line(...): Ignoring unknown parameters: `series`
```



What do we observe for this plot?

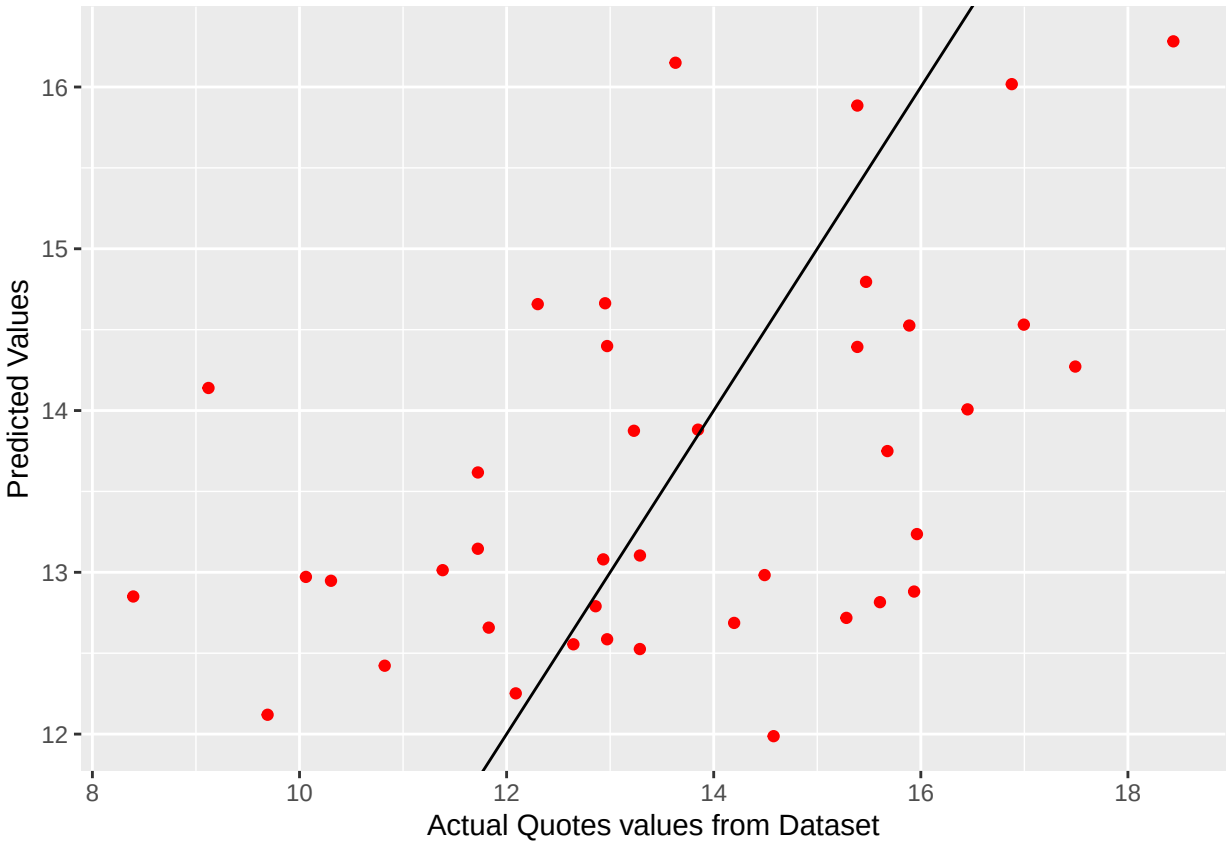
The black line in this plot displays the actual quotes, and the red line reflects the fitted values using the regression model. Plotting shows that, although there are minor differences between the two lines, the fitted line essentially matches the overall pattern of the real statements, fitted values seem to be less stable than the actual quotes raises the possibility that the regression model did not account for all of the sources of variation in the quotes. The fact that the real quotes occasionally differ from the fitted values suggests that the model may not be a perfect fit for the data.

Question 4 - Create a scatter plot showing fitted v actual. Do you observe any patterns?

```
library(ggplot2)
```

The below scatter plot gives the information about fitted vs actual

```
augment(fit_regression) %>%
  ggplot(aes(x = Quotes, y = .fitted)) +
  geom_point(color = " red") +
  labs( x = "Actual Quotes values from Dataset",
        y = "Predicted Values") +
  geom_abline(intercept = 0, slope = 1, color = "black")
```

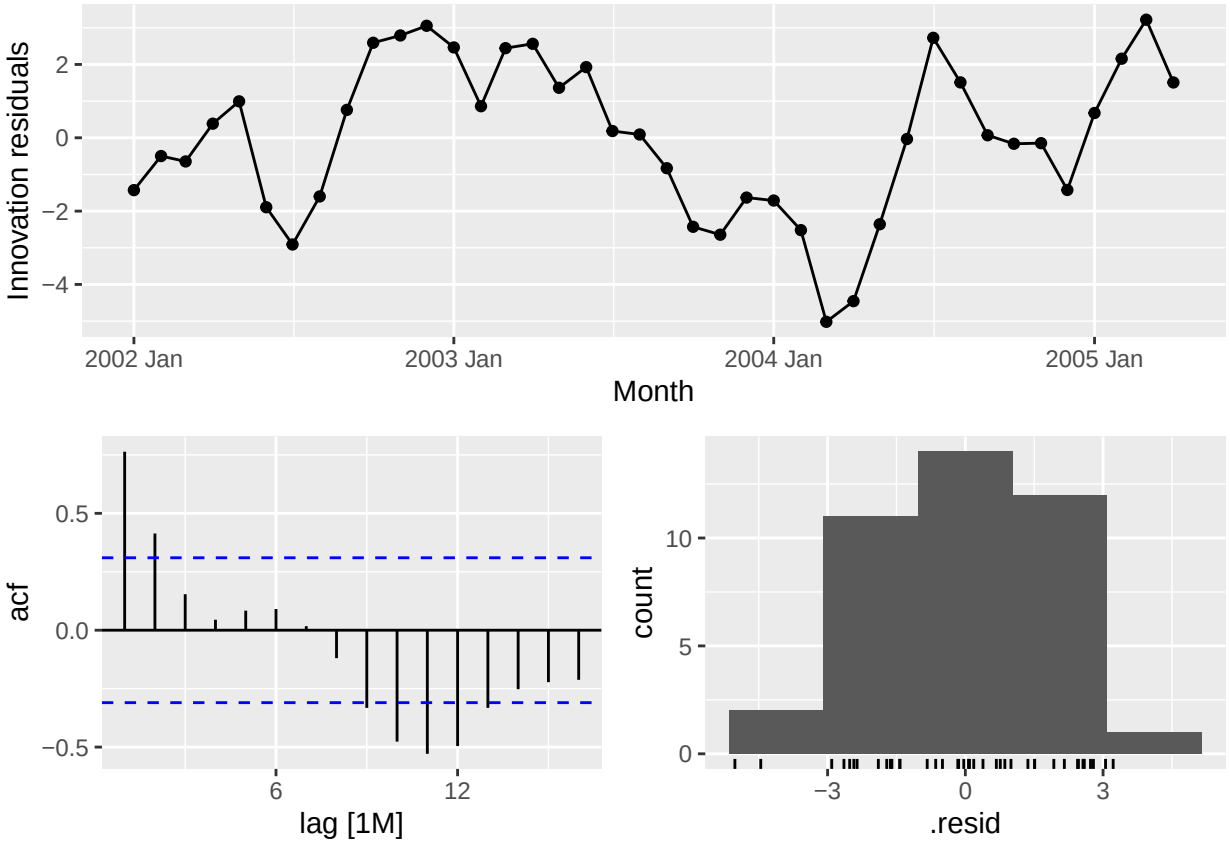


Yes, we can see a pattern in the scatter plot where the fitted values and real quotes are typically rather near to each other. Given that the points are dispersed throughout the 45-degree line, it appears likely that the fitted values represent an accurate approximation of the original statements. It's not quite a perfect fit, though, as some points stray off the line. Although there is still some unexplained fluctuation, the pattern indicates that the regression model offers a decent overall match to the data. The plot also reveals that, although the fitted values vary from roughly 12 to 16, the actual quotes range from about 8 to 18.

Question 5 - Plot the residuals against time. Do these plots reveal any autocorrelation in the model?

Here the gg_tsresiduals() function is used from fpp3 package, which helps us to plot residuals against time.

```
fit_regression %>% gg_tsresiduals()
```



The ACF plot indicates that the innovation residuals at lag 1 exhibit autocorrelation. There is a positive autocorrelation between the residuals at lag 1 and the ACF value of 0.5, which is above the 95% significance level (dashed lines). This implies the possibility of some unmodeled autocorrelation and the possibility that the model did not fully capture the underlying pattern in the data. As such, it might be required to take into account different models or modify the existing model in order to take the autocorrelation into account.

Question 6 - Generate box plots of the residuals for each month. Do these plots reveal any patterns in the above model?

```
head(insurance)
```

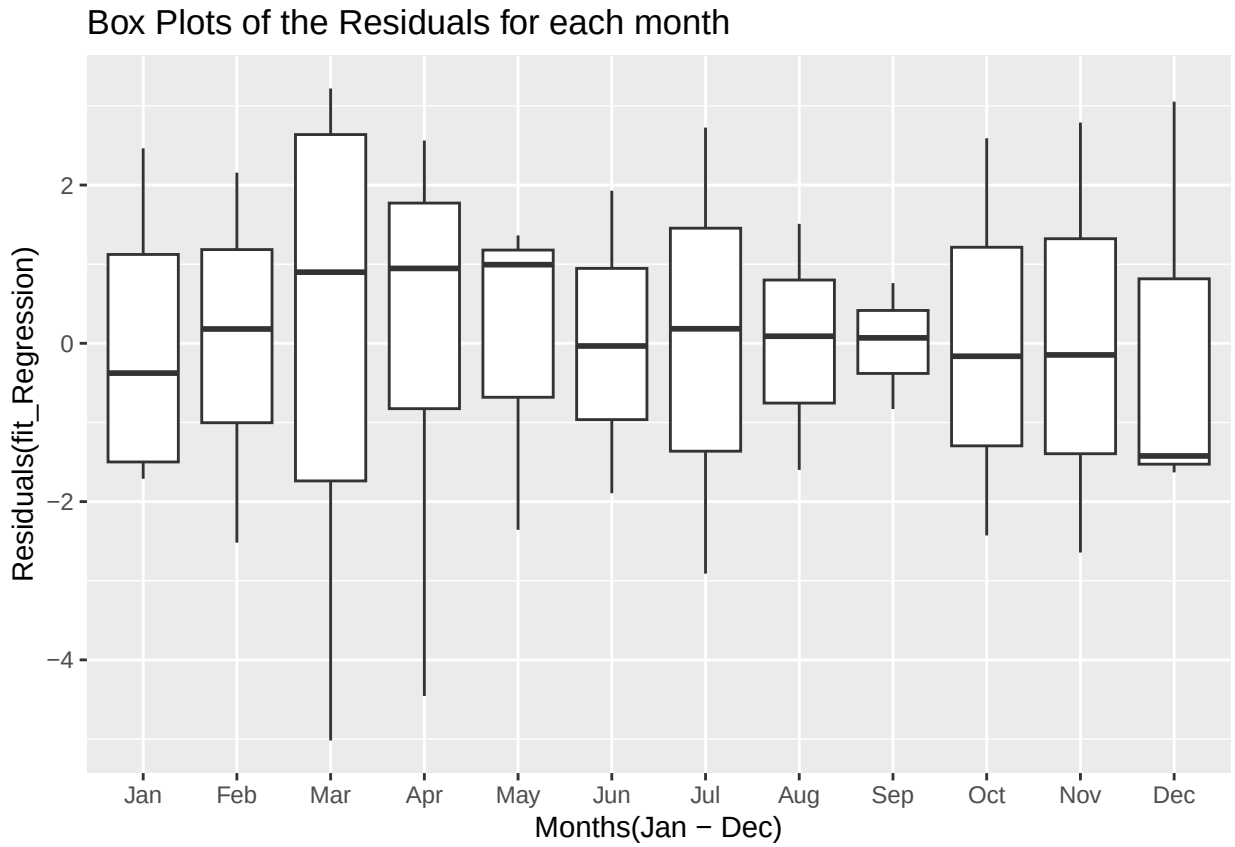
```
## # A tsibble: 6 x 3 [1M]
##      Month Quotes TVadverts
##      <mth>  <dbl>     <dbl>
## 1 2002 Jan   13.0      7.21
## 2 2002 Feb   15.4      9.44
## 3 2002 Mar   13.2      7.53
## 4 2002 Apr   13.0      7.21
## 5 2002 May   15.4      9.44
## 6 2002 Jun   11.7      6.42
```

Here we have created a vector format where 2000 is the year 1:12 is the months from Jan - Dec and time (0 hrs, 0mins,0sec)

```
Month_from_insurance_dataset <- format(ISOdatetime(2000, 1:12, 1, 0, 0, 0), "%b")
```

```
#let us left join residuals with the month column of insurance dataset with fit_regression
insurance <- left_join(insurance, residuals(fit_regression), by = "Month")

# let us create the box plot
insurance %>%
  ggplot(aes(y = .resid, x = factor(format(Month, "%b"), level = Month_from_insurance_dataset), group =
  geom_boxplot() +
  xlab('Months(Jan - Dec)') +
  ylab("Residuals(fit_Regression)") +
  ggtitle("Box Plots of the Residuals for each month")
```



Question 7 - Run a Ljung-Box test and interpret the results

The below line helps us to run the Ljung-Box test

```
augment(fit_regression) %>%
  features(.innov, ljung_box, lag = 10, dof = 5)
```

```
## # A tibble: 1 x 3
##   .model                           lb_stat lb_pvalue
##   <chr>                              <dbl>     <dbl>
## 1 TSLM(Quotes ~ trend() + season())    54.1  2.03e-10
```

Question 8 - Interpret the coefficients – the one associated with the trend variable and at least one associated with a seasonal variable.

```
report(fit_regression)
```

```
## Series: Quotes
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.01858 -1.60766  0.07939  1.61455  3.22002
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    14.38763    1.43309  10.040  1.3e-10 ***
## trend()         0.01102    0.03521   0.313    0.757
## season()year2   1.47572    1.79272   0.823    0.418
## season()year3  -0.54569    1.79376  -0.304    0.763
## season()year4  -1.84559    1.79548  -1.028    0.313
## season()year5  -0.04938    1.93726  -0.025    0.980
## season()year6  -0.83649    1.93630  -0.432    0.669
## season()year7  -1.49306    1.93598  -0.771    0.447
## season()year8  -2.05308    1.93630  -1.060    0.298
## season()year9  -1.96111    1.93726  -1.012    0.320
## season()year10 -2.51062    1.93886  -1.295    0.206
## season()year11 -1.69338    1.94110  -0.872    0.391
## season()year12 -1.63884    1.94397  -0.843    0.407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.535 on 27 degrees of freedom
## Multiple R-squared: 0.2273,  Adjusted R-squared: -0.1161
## F-statistic: 0.6619 on 12 and 27 DF, p-value: 0.77112
```

```
coefficients(fit_regression)
```

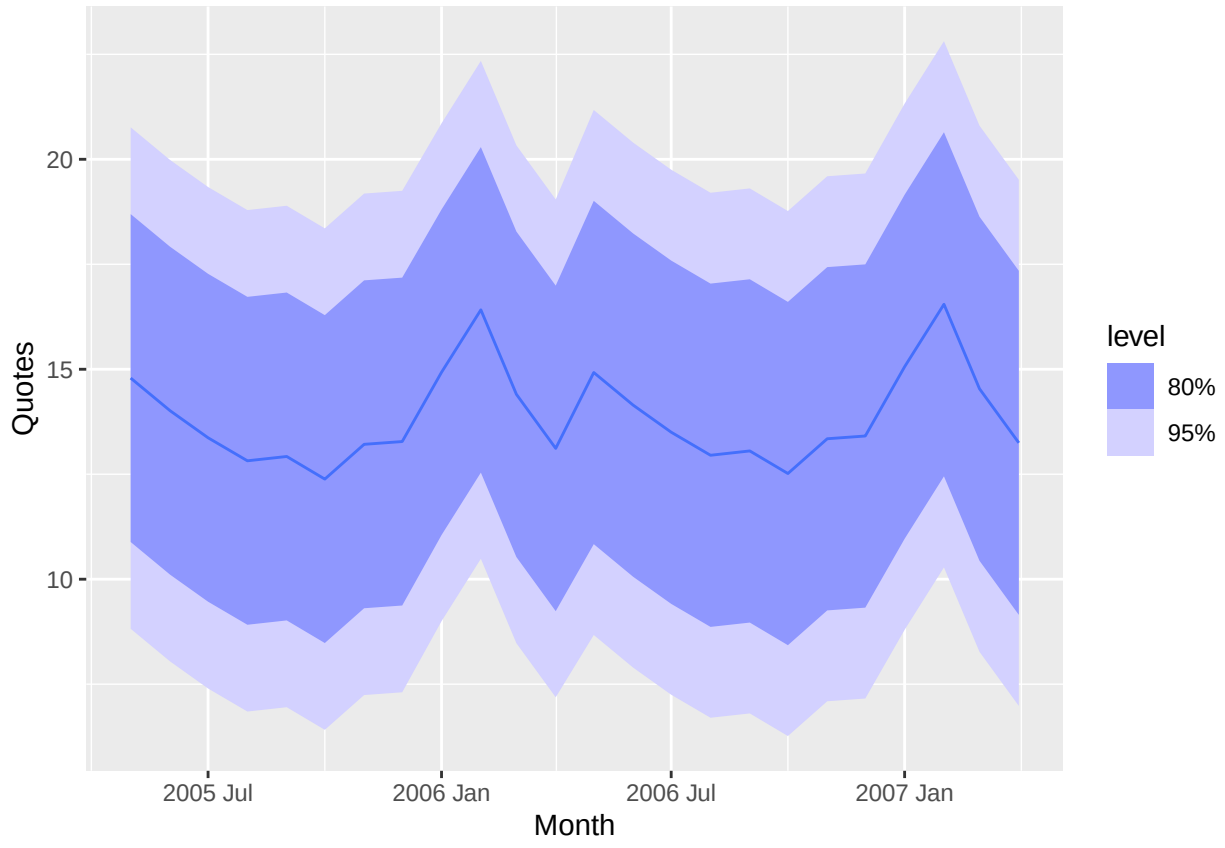
```
## # A tibble: 13 x 6
##    .model                           term  estimate std.error statistic  p.value
##    <chr>                            <chr>    <dbl>     <dbl>     <dbl>    <dbl>
##  1 TSLM(Quotes ~ trend() + season()) (Int~  14.4       1.43     10.0    1.30e-10
##  2 TSLM(Quotes ~ trend() + season()) tren~   0.0110    0.0352    0.313  7.57e- 1
##  3 TSLM(Quotes ~ trend() + season()) seas~   1.48      1.79      0.823  4.18e- 1
##  4 TSLM(Quotes ~ trend() + season()) seas~  -0.546     1.79     -0.304  7.63e- 1
##  5 TSLM(Quotes ~ trend() + season()) seas~  -1.85      1.80     -1.03   3.13e- 1
##  6 TSLM(Quotes ~ trend() + season()) seas~  -0.0494    1.94     -0.0255 9.80e- 1
##  7 TSLM(Quotes ~ trend() + season()) seas~  -0.836     1.94     -0.432  6.69e- 1
##  8 TSLM(Quotes ~ trend() + season()) seas~  -1.49      1.94     -0.771  4.47e- 1
##  9 TSLM(Quotes ~ trend() + season()) seas~  -2.05      1.94     -1.06   2.98e- 1
## 10 TSLM(Quotes ~ trend() + season()) seas~  -1.96      1.94     -1.01   3.20e- 1
## 11 TSLM(Quotes ~ trend() + season()) seas~  -2.51      1.94     -1.29   2.06e- 1
## 12 TSLM(Quotes ~ trend() + season()) seas~  -1.69      1.94     -0.872  3.91e- 1
## 13 TSLM(Quotes ~ trend() + season()) seas~  -1.64      1.94     -0.843  4.07e- 1
```

Question 9 - Use your regression model to forecast the monthly Quotes for 24 months ahead. Produce prediction intervals for those forecasts.

```
library(forecast)
forecast_monthly_quotes <- forecast(fit_regression, h = 24)
print(forecast_monthly_quotes)
```

```
## # A fable: 24 x 4 [1M]
## # Key:     .model [1]
##    .model                                Month     Quotes .mean
##    <chr>                                 <mth>     <dist> <dbl>
##  1 TSLM(Quotes ~ trend() + season()) 2005 May N(15, 9.3)  14.8
##  2 TSLM(Quotes ~ trend() + season()) 2005 Jun N(14, 9.3)  14.0
##  3 TSLM(Quotes ~ trend() + season()) 2005 Jul N(13, 9.3)  13.4
##  4 TSLM(Quotes ~ trend() + season()) 2005 Aug N(13, 9.3)  12.8
##  5 TSLM(Quotes ~ trend() + season()) 2005 Sep N(13, 9.3)  12.9
##  6 TSLM(Quotes ~ trend() + season()) 2005 Oct N(12, 9.3)  12.4
##  7 TSLM(Quotes ~ trend() + season()) 2005 Nov N(13, 9.3)  13.2
##  8 TSLM(Quotes ~ trend() + season()) 2005 Dec N(13, 9.3)  13.3
##  9 TSLM(Quotes ~ trend() + season()) 2006 Jan N(15, 9.1)  14.9
## 10 TSLM(Quotes ~ trend() + season()) 2006 Feb N(16, 9.1)  16.4
## # i 14 more rows
```

```
autoplot(forecast_monthly_quotes)
```



Question 10 - Do you have any recommendations for improving the model?

There could be some recommendations for improving the model. Firstly, we can use TV advertising as a predictor. We can other transformations to improve the model. There are chances of using other decomposition

techniques like ARIMA, STL etc.