DSCI_5340_HW2_Group10

```
pacman::p_load(dplyr, fpp3, GGally, gridExtra, fma, forecast, expsmooth, zoo, tseries, l
mtest)
theme_set(theme_classic())
```

#Reading the data set from fpp3 package

```
insurance
```

Month <mth>	Quotes <dbl>	TVadverts <dbl>
2002 Jan	12.97065	7.212725
2002 Feb	15.38714	9.443570
2002 Mar	13.22957	7.534250
2002 Apr	12.97065	7.212725
2002 May	15.38714	9.443570
2002 Jun	11.72288	6.415215
2002 Jul	10.06177	5.806990
2002 Aug	10.82279	6.203600
2002 Sep	13.28707	7.586430
2002 Oct	14.57832	8.004935

1-10 of 40 rows Previous **1** 2 3 4 Next

STRUCTURE OF INSURANCE

```
str(insurance)
```

```
## tbl_ts [40 × 3] (S3: tbl_ts/tbl_df/tbl/data.frame)
##  $ Month    : mth [1:40] 2002 Jan, 2002 Feb, 2002 Mar, 2002 Apr, 2002 May, 2002 Ju
n,...
##  $ Quotes   : num [1:40] 13 15.4 13.2 13 15.4 ...
##  $ TVadverts: num [1:40] 7.21 9.44 7.53 7.21 9.44 ...
##  - attr(*, "key")= tibble [1 × 1] (S3: tbl_df/tbl/data.frame)
##   ..$ .rows: list<int> [1:1]
##   .. ..$ : int [1:40] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..@ ptype: int(0)
##  - attr(*, "index")= chr "Month"
##   ..- attr(*, "ordered")= logi TRUE
##  - attr(*, "index2")= chr "Month"
##  - attr(*, "interval")= interval [1:1] 1M
##   ..@ .regular: logi TRUE
```

```
head(insurance)
```

Month <mth>	Quotes <dbl>	TVadverts <dbl>
2002 Jan	12.97065	7.212725
2002 Feb	15.38714	9.443570
2002 Mar	13.22957	7.534250
2002 Apr	12.97065	7.212725
2002 May	15.38714	9.443570
2002 Jun	11.72288	6.415215

6 rows

```
tail(insurance)
```

Month <mth>	Quotes <dbl>	TVadverts <dbl>
2004 Nov	12.93375	8.244881
2004 Dec	11.72235	6.675540
2005 Jan	15.47126	9.219604
2005 Feb	18.43898	10.963800
2005 Mar	17.49186	10.456290
2005 Apr	14.49168	8.728600

6 rows

```
attributes(insurance)
```

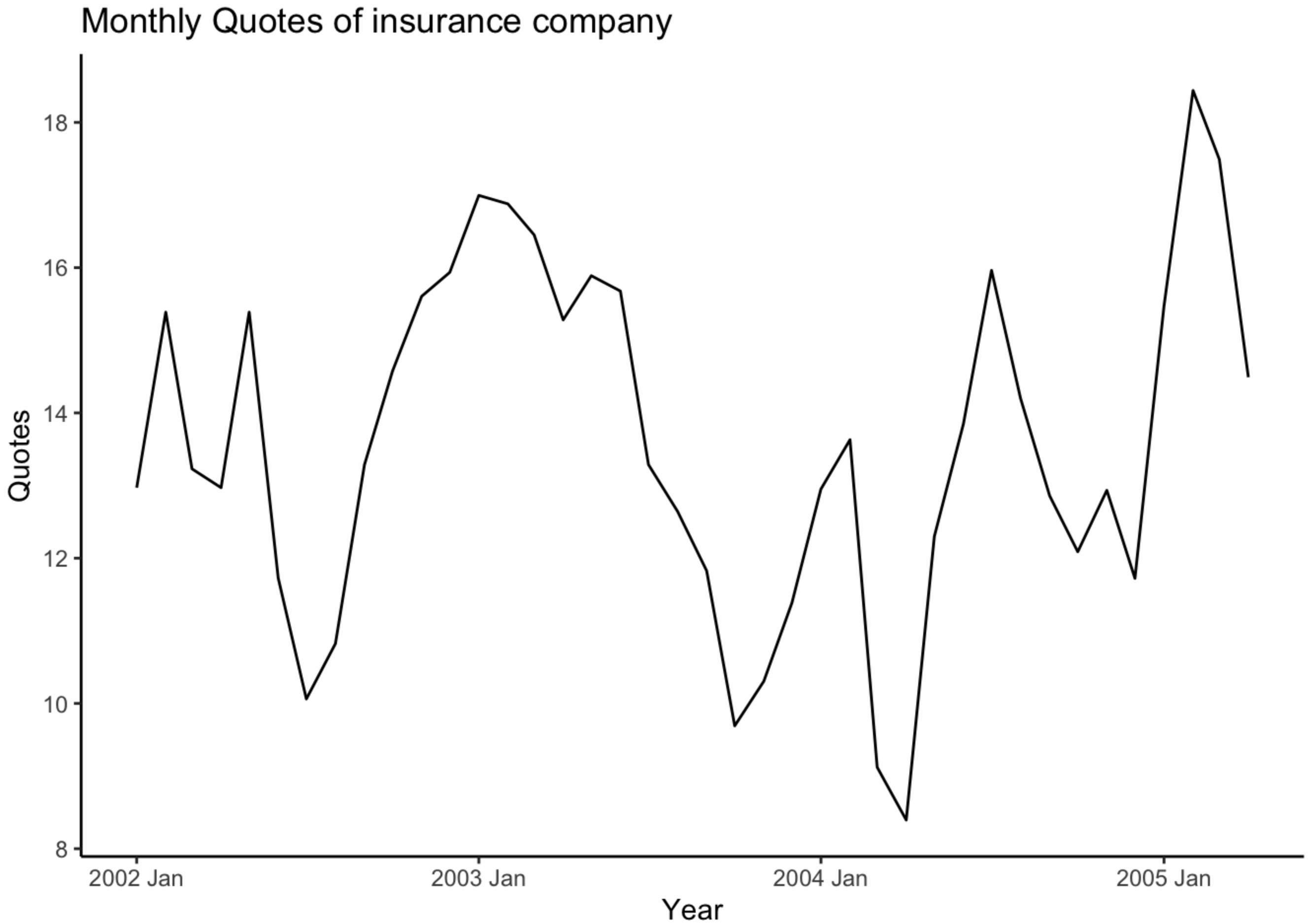
```
## $names
## [1] "Month"     "Quotes"    "TVadverts"
##
## $row.names
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##
## $key
## # A tibble: 1 × 1
##         .rows
##   <list<int>>
## 1        [40]
##
## $index
## [1] "Month"
## attr(,"ordered")
## [1] TRUE
##
## $index2
## [1] "Month"
##
## $interval
## <interval[1]>
## [1] 1M
##
## $class
## [1] "tbl_ts"     "tbl_df"     "tbl"        "data.frame"
```

```
dim(insurance)
```

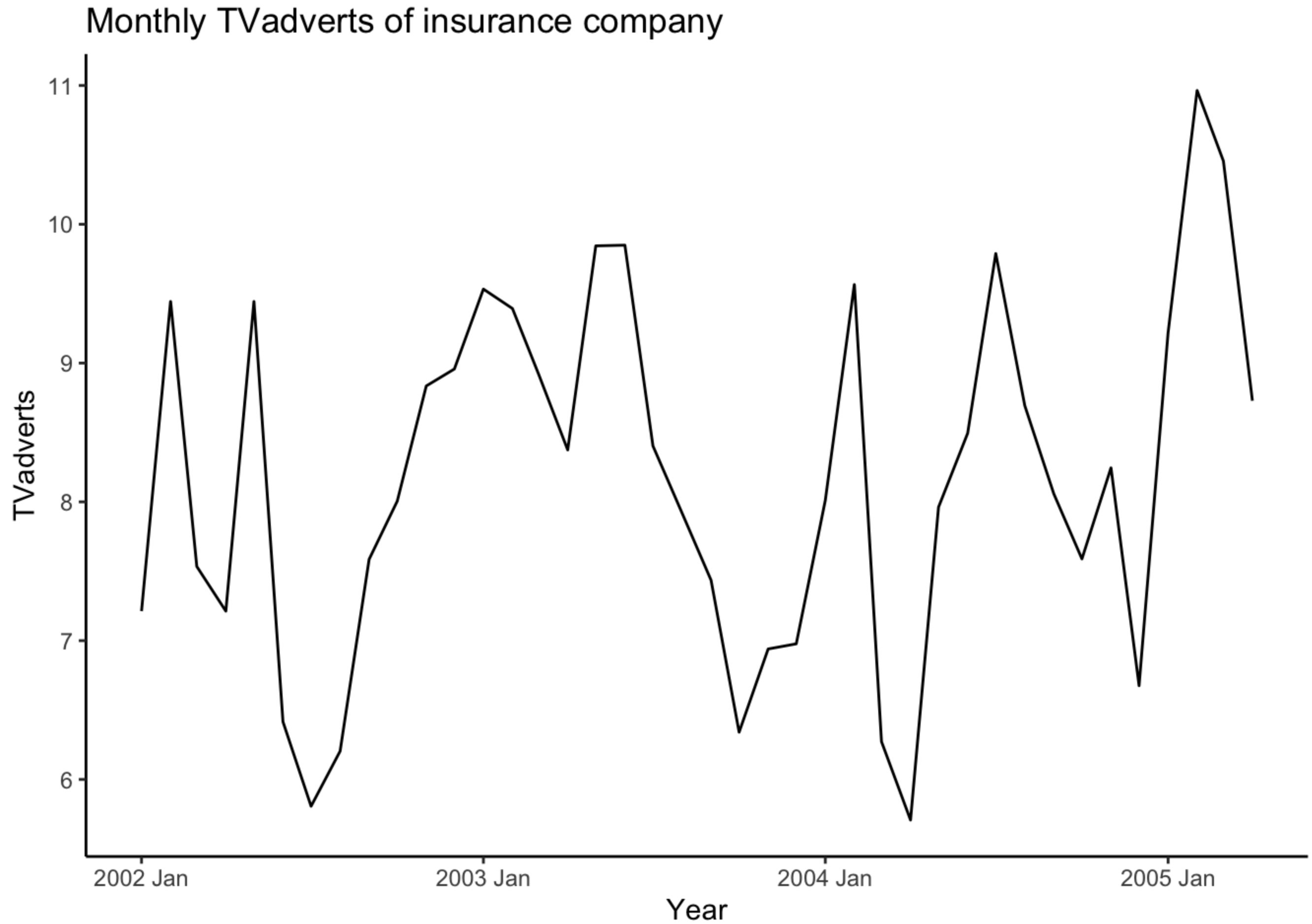
```
## [1] 40  3
```

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

```
time_Quotes <- insurance %>%
  autoplot(Quotes)+ylab("Quotes")+xlab("Year")+ggtitle("Monthly Quotes of insurance comp
any")
time_Quotes
```




```
time_TVadverts <- insurance %>%
  autoplot(TVadverts)+ylab("TVadverts")+xlab("Year")+ggtitle("Monthly TVadverts of insur
ance company")
time_TVadverts
```



#The

above pattern is irregular from 2002 january to 2005 january .we didnt find any breakouts or diversion from a regular or in repeating pattern.

#Question2 -Fit a regression model with Quotes as the dependent variable and a linear trend andseasonal dummies as explanatory variables

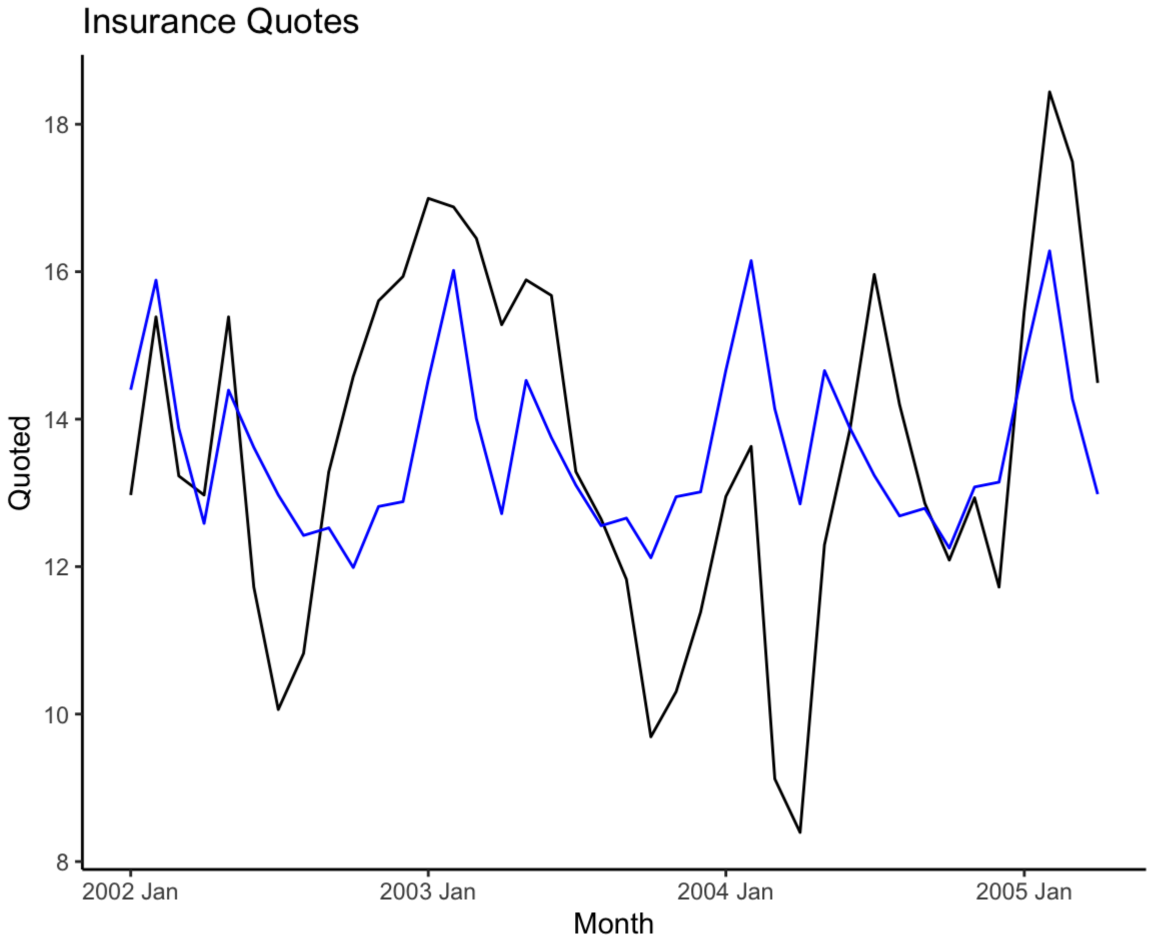
```
fit_quotes <- insurance %>% model(TSLM (Quotes~trend ()+season ()))
report (fit_quotes)
```

```
## 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
```

since Adjusted R-squared value is -0.1161 .we can observe that model is overfitted

#Question3-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?

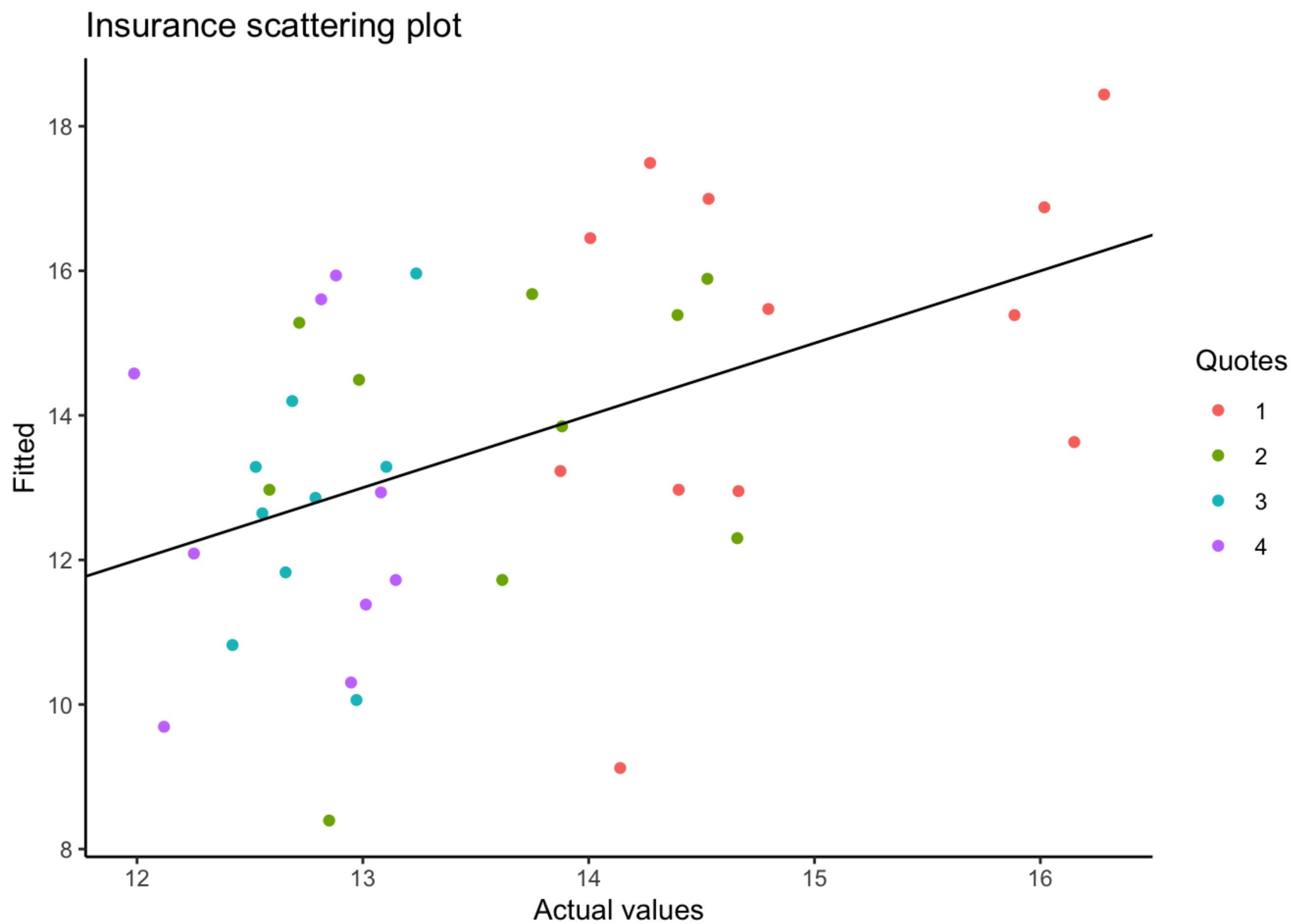
```
augment(fit_quotes)%>%
  ggplot(aes(x= Month)) +
  geom_line(aes(y=Quotes,colour = "Data"))+
  geom_line(aes(y=.fitted,colour = "Fitted")) +
  scale_colour_manual(
    values = c(Data = "black",Fitted ="blue")
  ) +
  labs(y="Quoted",title = "Insurance Quotes") +
  guides(colour =guide_legend(title="Series"))
```



From the above plot, we see the pattern between the fitted line and data is a little too similar which might be because the model is overfitting which would also explain the negetive Adjusted R-Squared value from the above TSLR.

#Question4. Create a scatter plot showing fitted v actual. Do you observe any patterns?

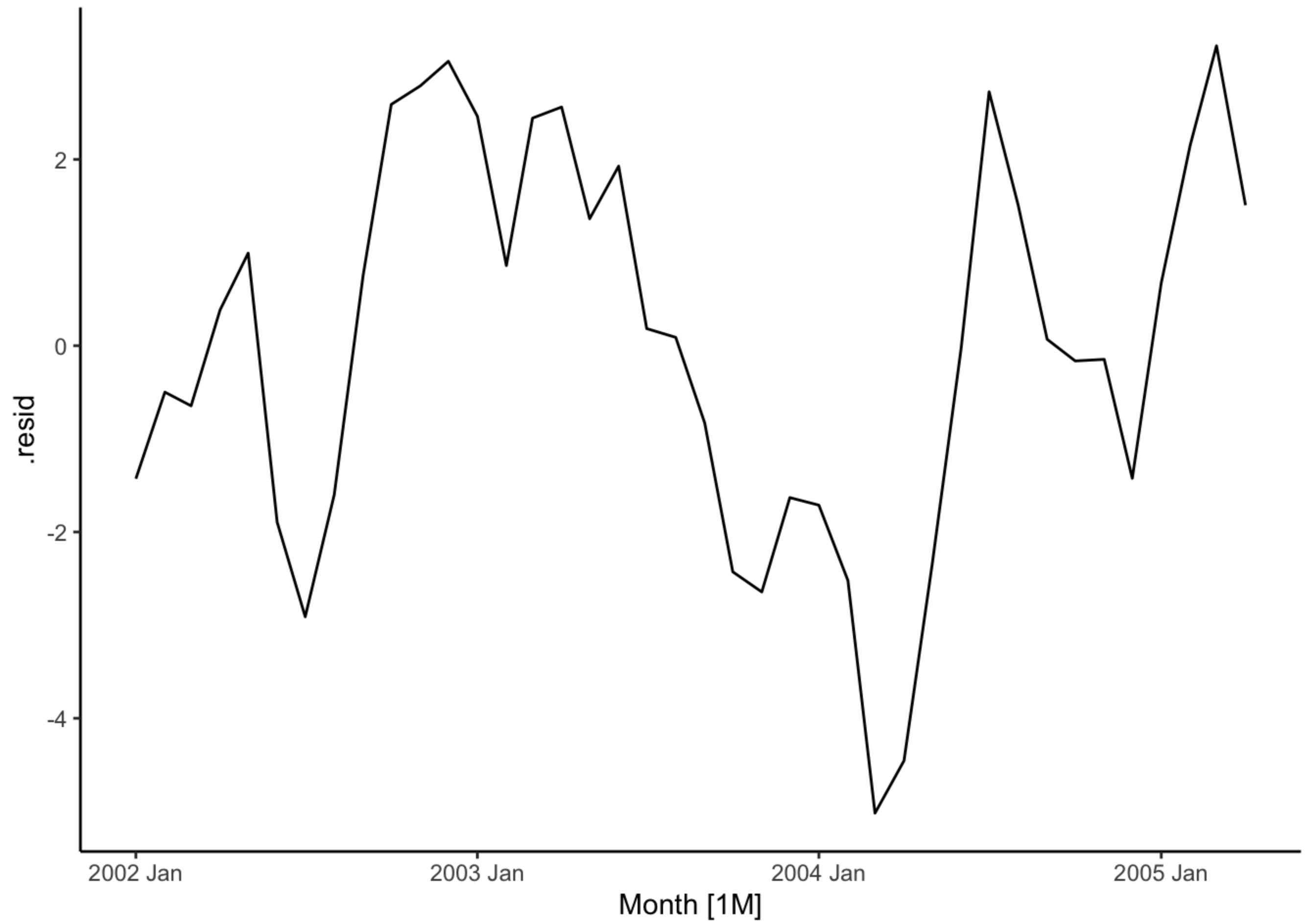
```
augment(fit_quotes) %>%
  ggplot(aes (y = Quotes, x = .fitted,
              colour = factor (quarter (Month)))) +
  geom_point () +
  labs(y ="Fitted", x="Actual values" ,
       title = "Insurance scattering plot") +
  geom_abline (intercept = 0, slope = 1) +
  guides (colour = guide_legend(title = "Quotes"))
```



#There is no discernible pattern in the scatter plot which might be indicative of the model overfitting or underfitting or having a non-linear relationship.

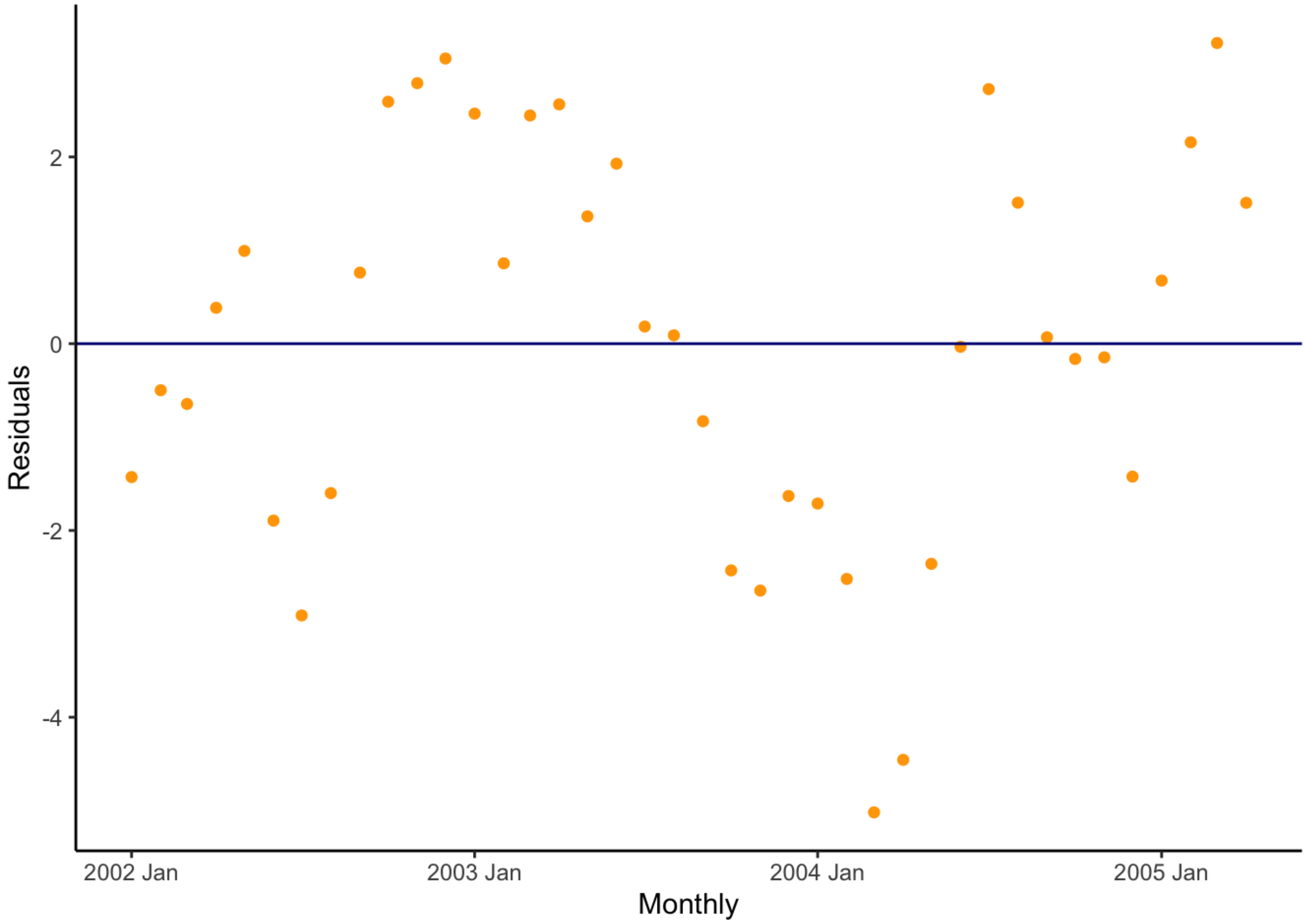
#Question5. Plot the residuals against time. Do these plots reveal any autocorrelation in the model?

```
augment(fit_quotes) %>%autoplot(.resid)
```

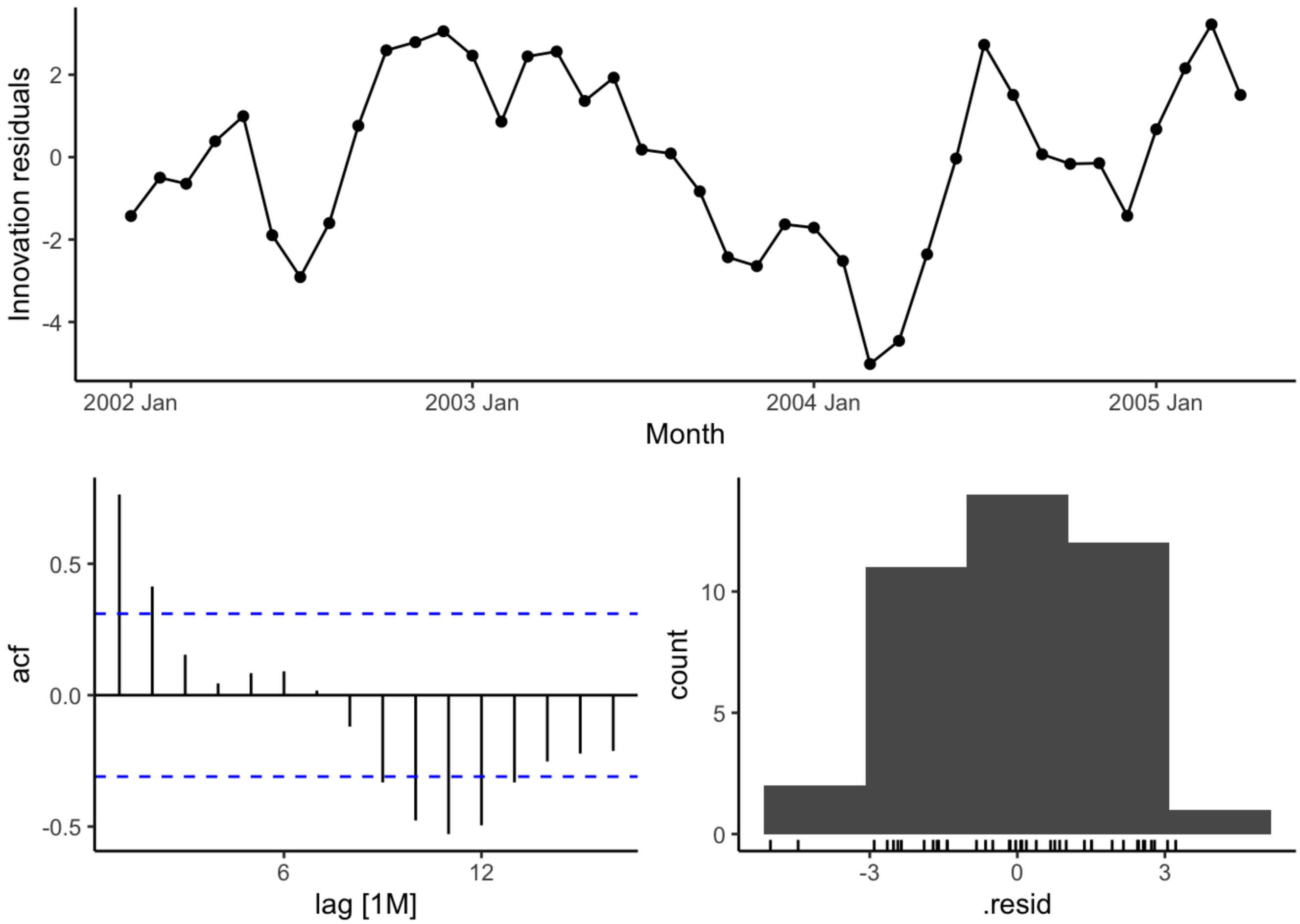



```
augment(fit_quotes)%>%
ggplot(aes (x= Month, y=.resid))+
geom_point(color = "ORANGE") +
geom_abline(intercept = 0, slope = 0,color = "navy", Ity = 2) +
labs(x = "Monthly", y="Residuals","Residual v time")
```

```
## Warning in geom_abline(intercept = 0, slope = 0, color = "navy", Ity = 2):
## Ignoring unknown parameters: `Ity`
```



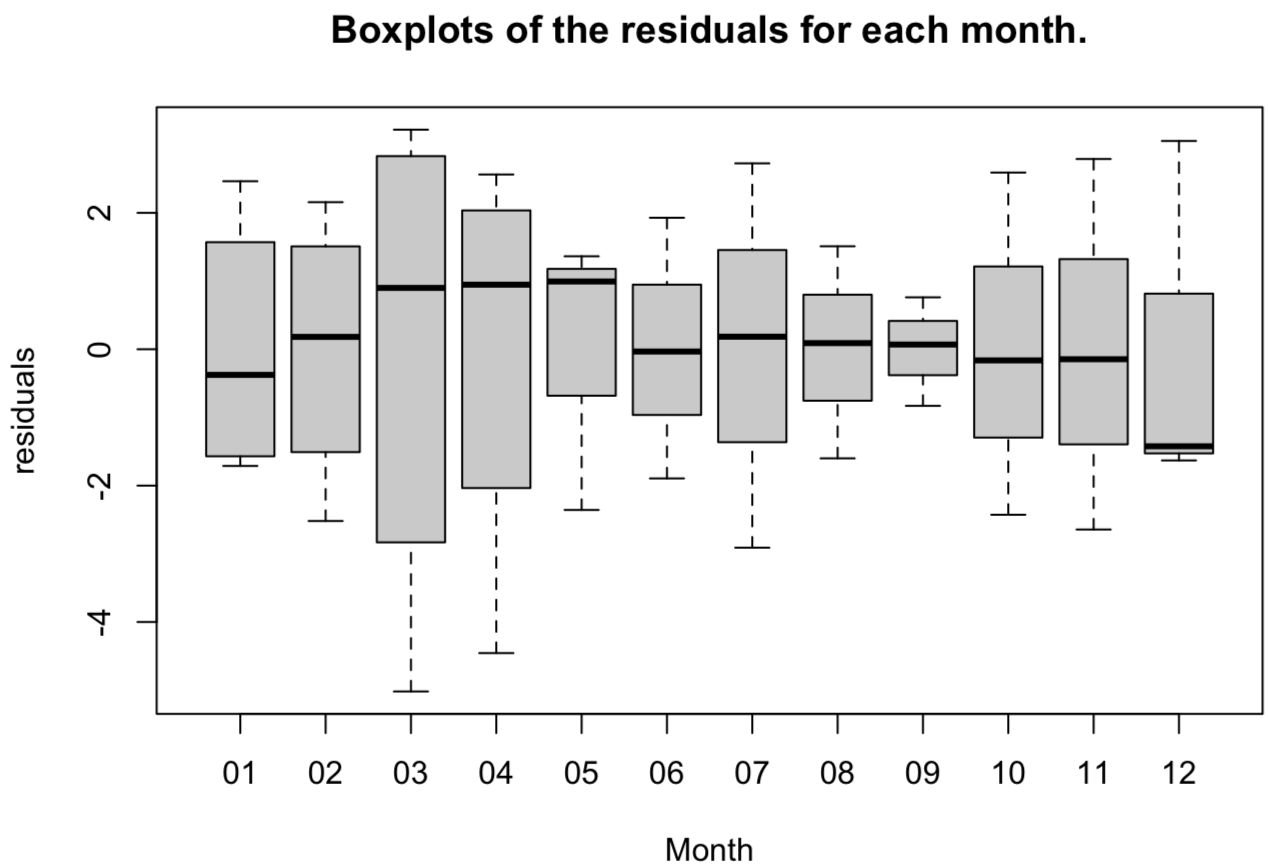

```
fit_quotes%>%gg_tsresiduals()
```



##The ACF plot does reveal a significant auto correlation in the data with 7 out of the 24 lag values cross the 95% threshold. which means that the data is not white noise.

#Question6. Generate box plots of the residuals for each month. Do these plots reveal any patterns in the above model?

```
boxplot(augment(fit_quotes)$.resid ~
format (augment (fit_quotes)$Month, '%m'),xlab ='Month', ylab= 'residuals',
main='Boxplots of the residuals for each month.')
```



#From this plot we can say that some months like march and april have higher range of residuals than compared to the rest.

#7.Run a Ljung-Box test and interpret the results.

```
augment(fit_quotes) %>%
  features(.innov, ljung_box, lag = 10, dof = 27)
```

```
## Warning in pchisq(STATISTIC, lag - fitdf): NaNs produced
```

.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
TSLM(Quotes ~ trend() + season())	54.06362	NaN

1 row

#p value lessthan 0.05(level of significance) we can reject the null hypothesis and conclude that there is auto correleation in this model.

#question8. Interpret the coefficients – the one associated with the trend variable and at least one associated with a seasonal variable.

```
report(fit_quotes)
```

```
## 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
```

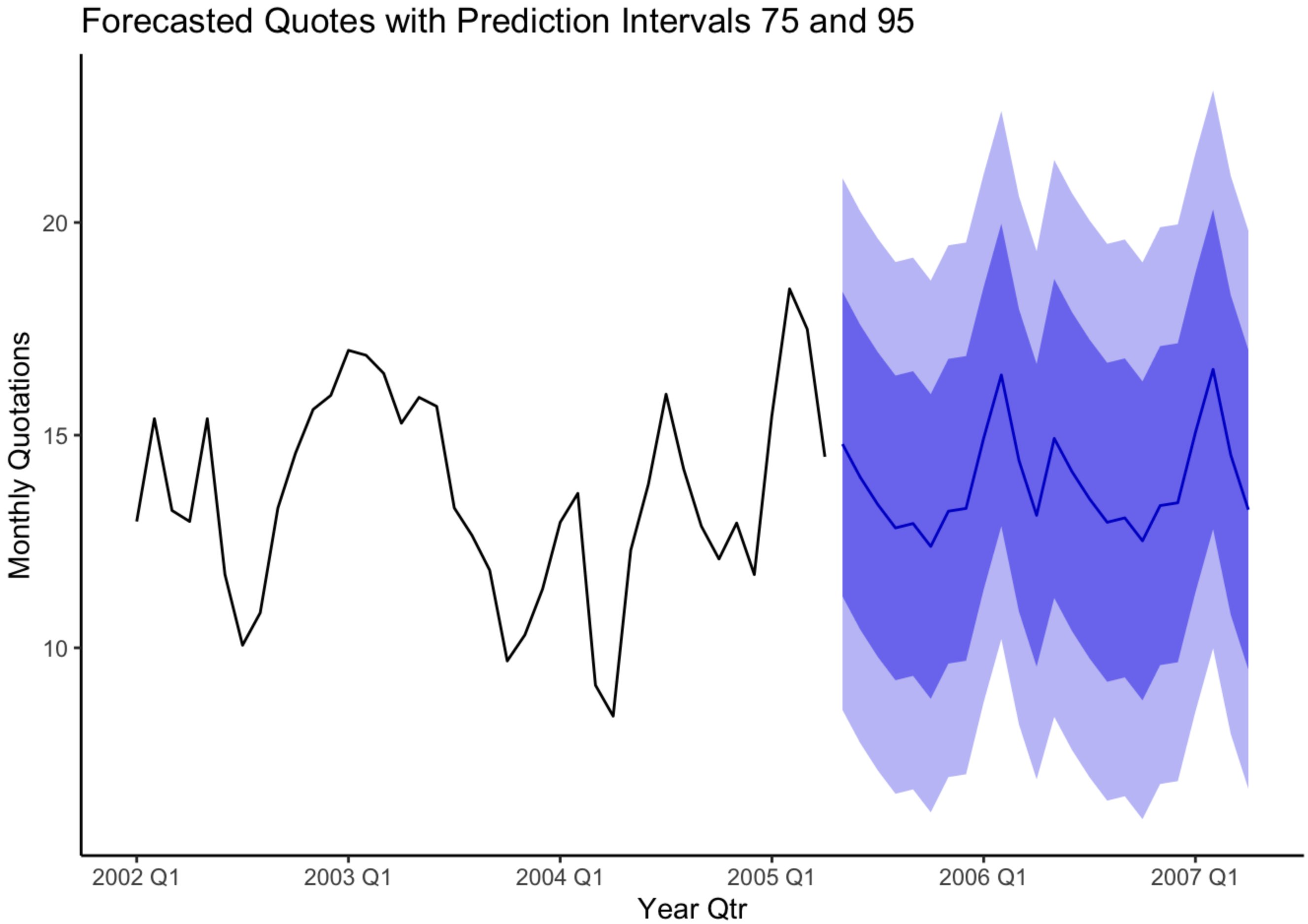
#with a p value of 0.77112 which is significantly greater that 0.05, we fail to reject null hypothesis that there is trend and with a unit increase in the value of trend, the value of quotes will increase by 0.01102.

#question9. Use your regression model to forecast the monthly Quotes for 24 months ahead. Produce prediction intervals for those forecasts.

#The graph below displays forecasted Insurance Quotes from May 2005 to April 2007, using past data from January 2002 to April 2004, along with prediction intervals (confidence levels) of 75% and 95%

```
 library(forecast)
library(ggplot2)
Quotes <- ts(insurance$Quotes, start=c(2002,1), end=c(2005,4), frequency=12)
reg_fit <- tslm(Quotes ~ trend + season)
  ft_quotes <- forecast(reg_fit, h = 24, level=c(75,95))
  autoplot(ft_quotes, prediction.interval = TRUE) + xlab("Year Qtr") +
     ylab("Monthly Quotations") +
  ggtitle("Forecasted Quotes with Prediction Intervals 75 and 95") +
    scale_x_yearqtr(format = "%Y Q%q")
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```




```
summary(ft_quotes)
```

```
##
## Forecast method: Linear regression model
##
## Model Information:
##
## Call:
## tslm(formula = Quotes ~ trend + season)
##
## Coefficients:
## (Intercept)        trend      season2      season3      season4      season5
##    14.38763      0.01102      1.47572     -0.54569     -1.84559     -0.04938
##     season6      season7      season8      season9     season10     season11
##    -0.83649     -1.49306     -2.05308     -1.96111     -2.51062     -1.69338
##    season12
##    -1.63884
##
##
## Error measures:
##                         ME     RMSE      MAE      MPE    MAPE      MASE
## Training set -3.053113e-16 2.082549 1.716869 -2.68927 13.5543 0.5002478
##                   ACF1
## Training set 0.7636135
##
## Forecasts:
##          Point Forecast     Lo 75    Hi 75     Lo 95    Hi 95
## May 2005       14.79019 11.208651 18.37173  8.539395 21.04099
## Jun 2005       14.01411 10.432568 17.59565  7.763312 20.26490
## Jul 2005       13.36856  9.787025 16.95010  7.117769 19.61936
## Aug 2005       12.81956  9.238018 16.40110  6.568762 19.07035
## Sep 2005       12.92255  9.341015 16.50409  6.671759 19.17335
## Oct 2005       12.38406  8.802525 15.96560  6.133269 18.63486
## Nov 2005       13.21233  9.630788 16.79387  6.961532 19.46312
## Dec 2005       13.27790  9.696358 16.85944  7.027102 19.52869
## Jan 2006       14.92776 11.372141 18.48338  8.722204 21.13331
## Feb 2006       16.41451 12.858888 19.97012 10.208951 22.62006
## Mar 2006       14.40412 10.848498 17.95973  8.198561 20.60967
## Apr 2006       13.11524  9.559623 16.67086  6.909686 19.32080
## May 2006       14.92247 11.172691 18.67225  8.378050 21.46689
## Jun 2006       14.14639 10.396607 17.89616  7.601967 20.69080
## Jul 2006       13.50084  9.751064 17.25062  6.956423 20.04526
## Aug 2006       12.95184  9.202057 16.70161  6.407417 19.49625
## Sep 2006       13.05483  9.305054 16.80461  6.510413 19.59925
## Oct 2006       12.51634  8.766564 16.26612  5.971923 19.06076
## Nov 2006       13.34461  9.594827 17.09438  6.800187 19.88902
## Dec 2006       13.41018  9.660397 17.15995  6.865757 19.95459
## Jan 2007       15.06004 11.302044 18.81803  8.501281 21.61879
## Feb 2007       16.54678 12.788791 20.30478  9.988028 23.10554
## Mar 2007       14.53639 10.778401 18.29439  7.977638 21.09515
## Apr 2007       13.24752  9.489526 17.00551  6.688763 19.80627
```

#The above graph displays forecasted Insurance Quotes from May 2005 to April 2007, using past data from January 2002 to April 2004, along with prediction intervals (confidence levels) of 75% and 95%

#Question10. Do you have any recommendations for improving the model?

1)We can remove the missing values from data which helps to improve predictions. 2)we have to eliminate outliers 3)We can use STL decomposition to get a better idea about the trend and seasonal properties of the data