Final_Exam_Energy_Analytics

Chanukya

12/11/2019

```
library(fpp2)
```

```
## Loading required package: ggplot2
```

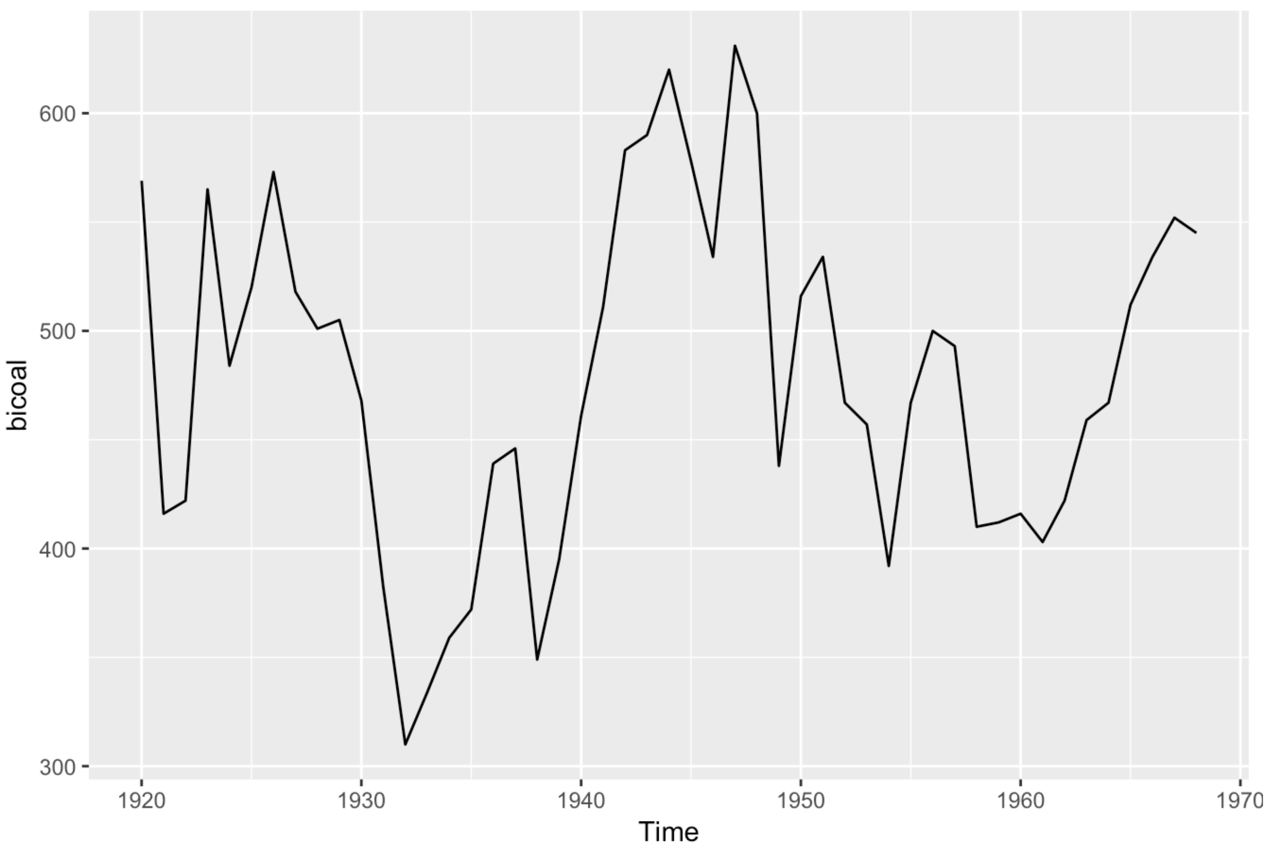
```
## Loading required package: forecast
```

```
## Warning: package 'forecast' was built under R version 3.5.2
```

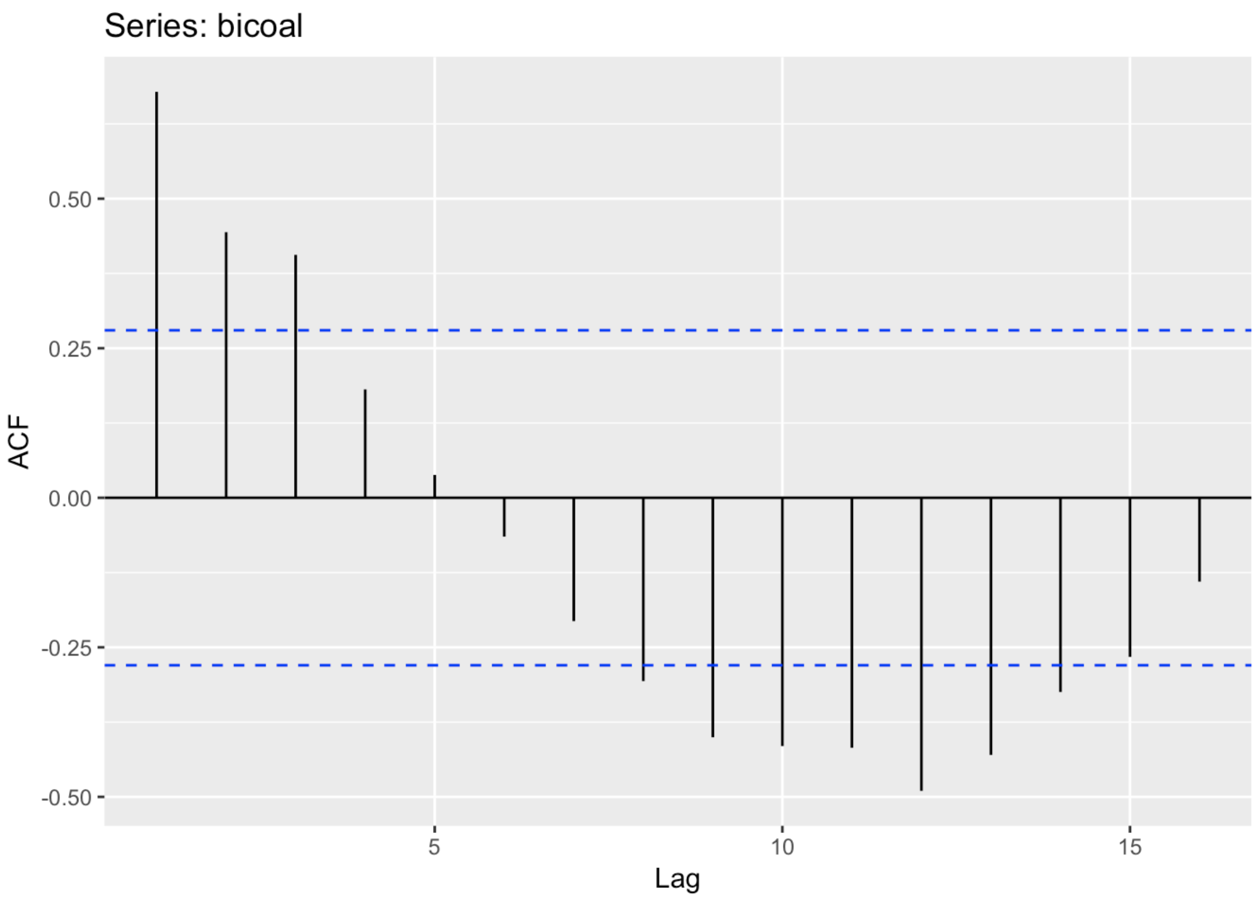
```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

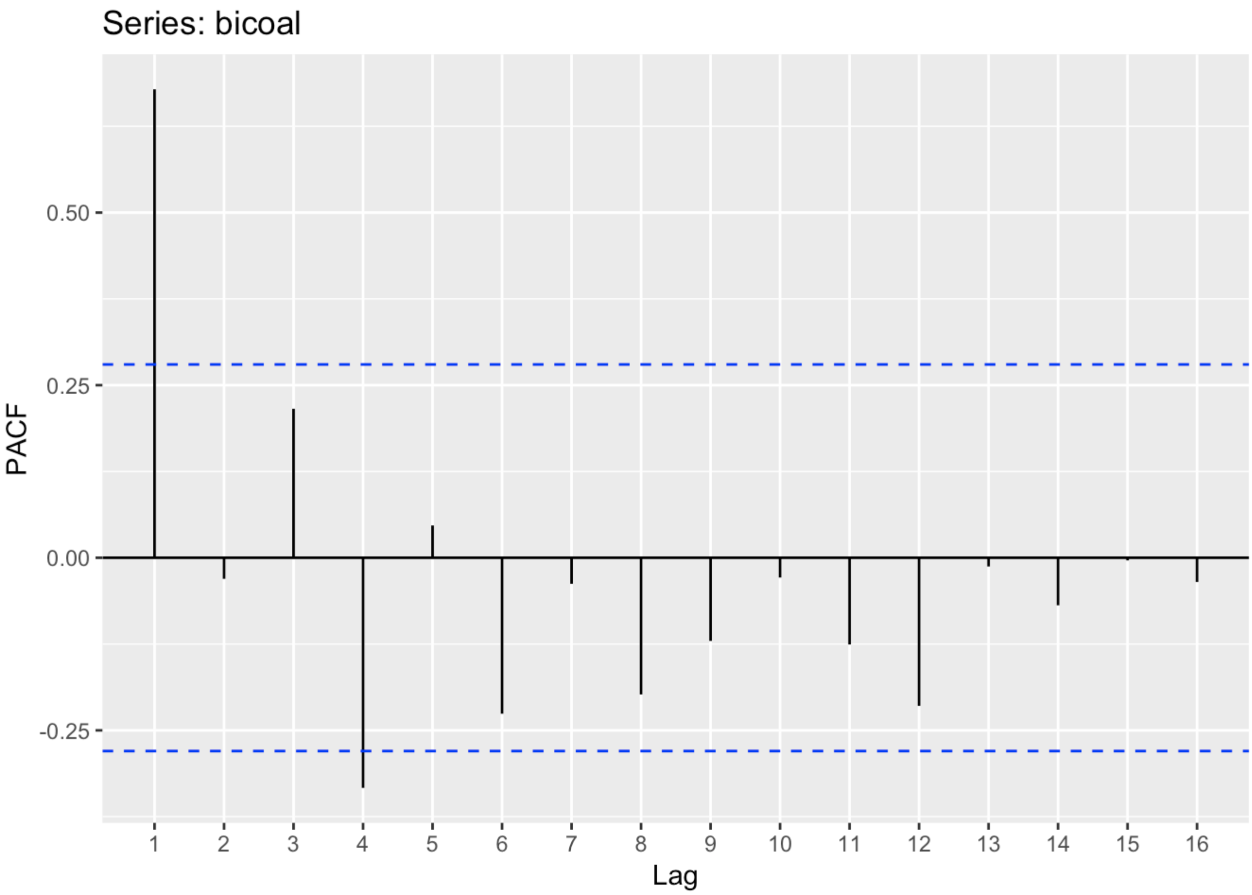
```
# 1)

#a)
autoplot(bicoal)
```



```
# From the plot we can see that there is not particular trend or seasonlity.

#b)
 # The give equation : yt = c + φ1 yt−1 + φ2 yt−2 + φ3 yt−3 +φ4 yt−4 + et
 #  p : the number of autoregressive terms, from the above equation we can say
p value is equal to 4
 #  d : the number of nonseasonal differences, from the the above equation we
see that there is no difference taken into consideration.
 #  q : the number of moving average terms whic can also be said as number of
lagged forecast errors, from the given equation we see that none of lagged for
ecast errors are taken into consideration
 # so the final (p,d,q) values for the above equation is (4,0,0).

#c)

##
ggAcf(bicoal)
```




```
# There decreasing correlation till lag 12. The plot depicts more of a sinusoi
dal wave

ggPacf(bicoal)
```




```
# There is lage first and fourth spikes

#d)

c = 162.00
pi1 = 0.83
pi2 = -.34
pi3 = 0.55
pi4 = -.38

y_1968 = 545
y_1967 = 552
y_1966 = 534
y_1965 = 512



 y_1969 = c + (pi1*y_1968) + (pi2*y_1967) + (pi3*y_1966) + (pi4*y_1965)
 y_1970 = c + (pi1*y_1969) + (pi2*y_1968) + (pi3*y_1967) + (pi4*y_1966)
 y_1971 = c + (pi1*y_1970) + (pi2*y_1969) + (pi3*y_1968) + (pi4*y_1967)

print(paste0("y_1969 : ", y_1969))
```

```
## [1] "y_1969 : 525.81"
```

```
print(paste0("y_1970 : ", y_1970))
```

```
## [1] "y_1970 : 513.8023"
```

```
print(paste0("y_1971 : ", y_1971))
```

```
## [1] "y_1971 : 499.670509"
```

```
# e)

 fitARIMA <- arima(bicoal, order=c(4,0,0))
 predict(fitARIMA,n.ahead = 3)
```

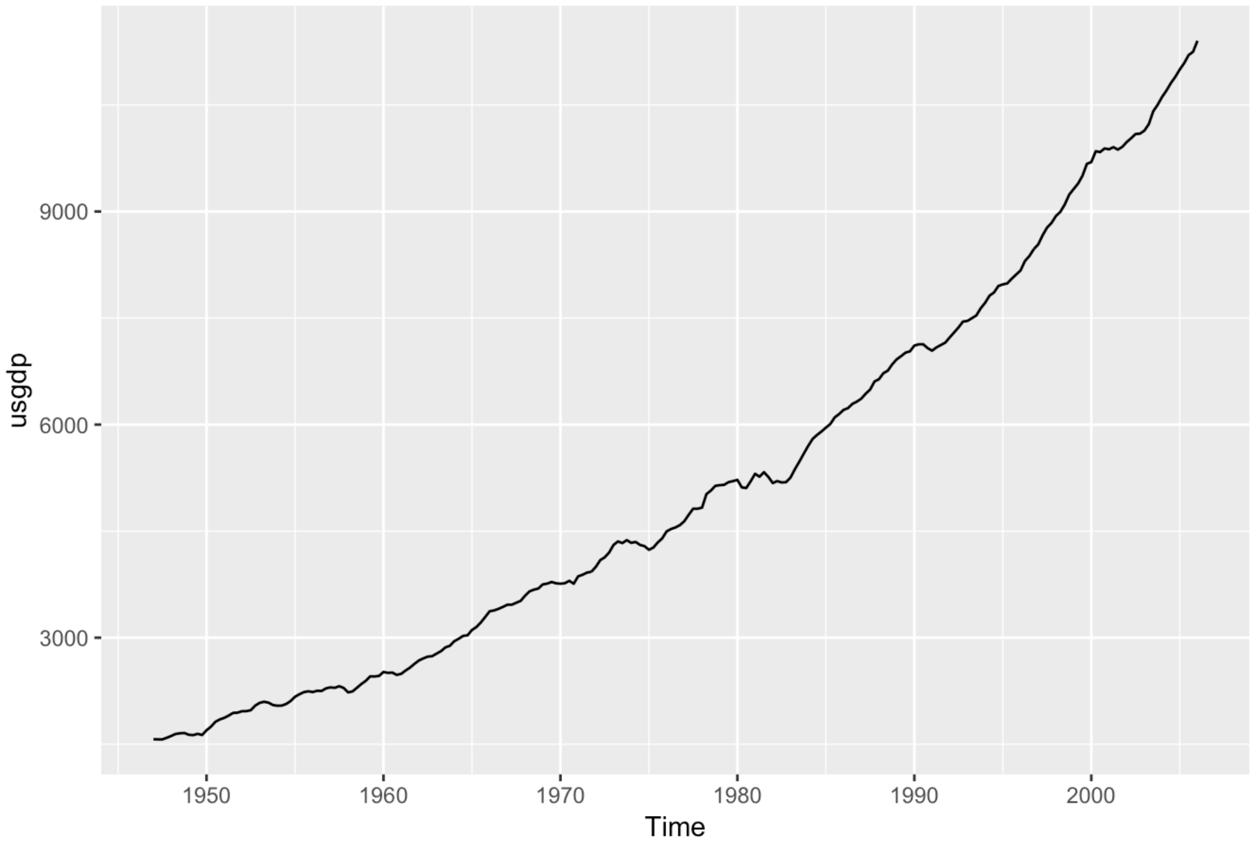
```
## $pred
## Time Series:
## Start = 1969
## End = 1971
## Frequency = 1
## [1] 527.6291 517.1923 503.8051
##
## $se
## Time Series:
## Start = 1969
## End = 1971
## Frequency = 1
## [1] 50.09487 65.21025 67.52875
```

```
 fitARIMA$coef
```

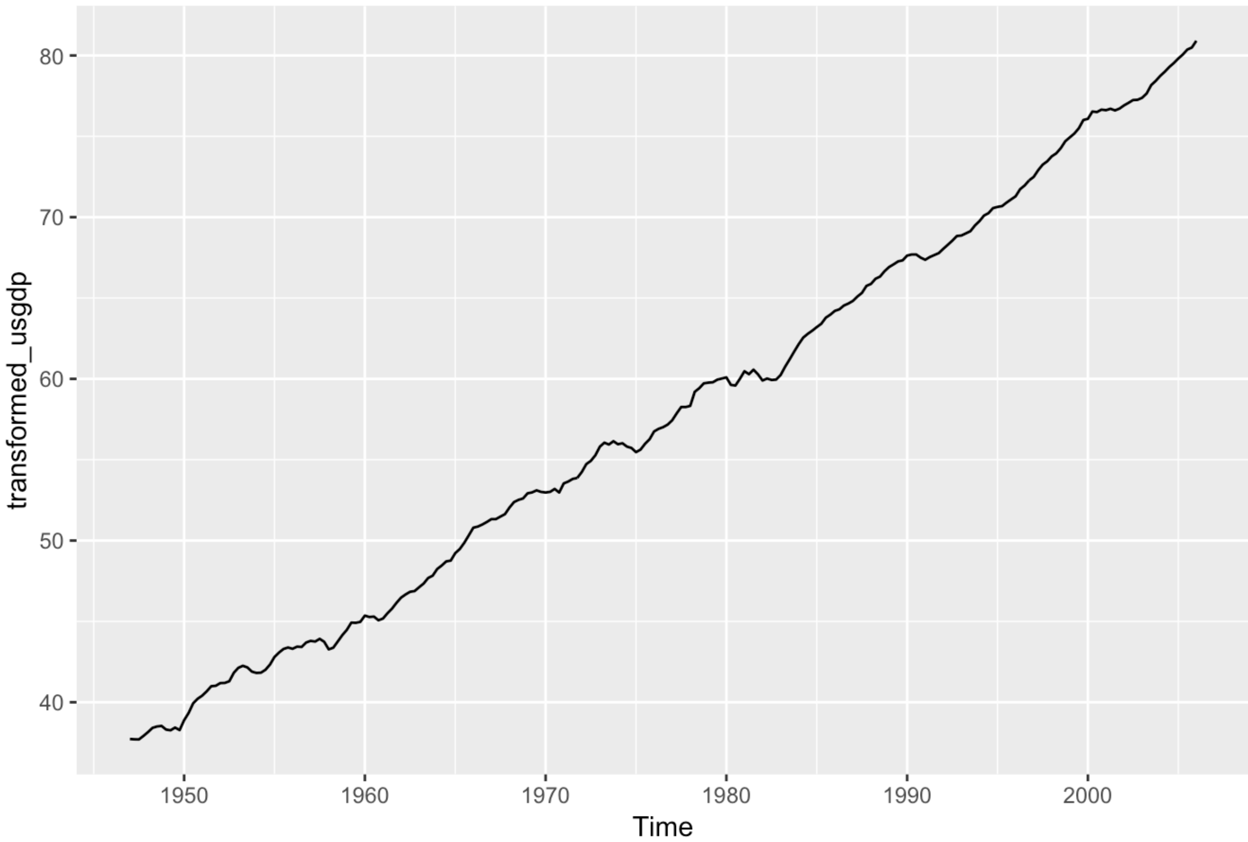
```
##          ar1          ar2          ar3          ar4   intercept
##    0.8333751   -0.3443200    0.5524712   -0.3779598 481.5220637
```

```
 # The values obtained are [527.6291,517.1923,503.8051]

 # They slightly greater than previously predicted ones because of larger inte
rcept.
```

```
##2)

 # a)
 autoplot(usgdp)
```




```
# from the plot we can see upward trend and seasonality.

 lambdaa = BoxCox.lambda(usgdp)
 transformed_usgdp = BoxCox(usgdp,lambdaa)
 autoplot(transformed_usgdp)
```



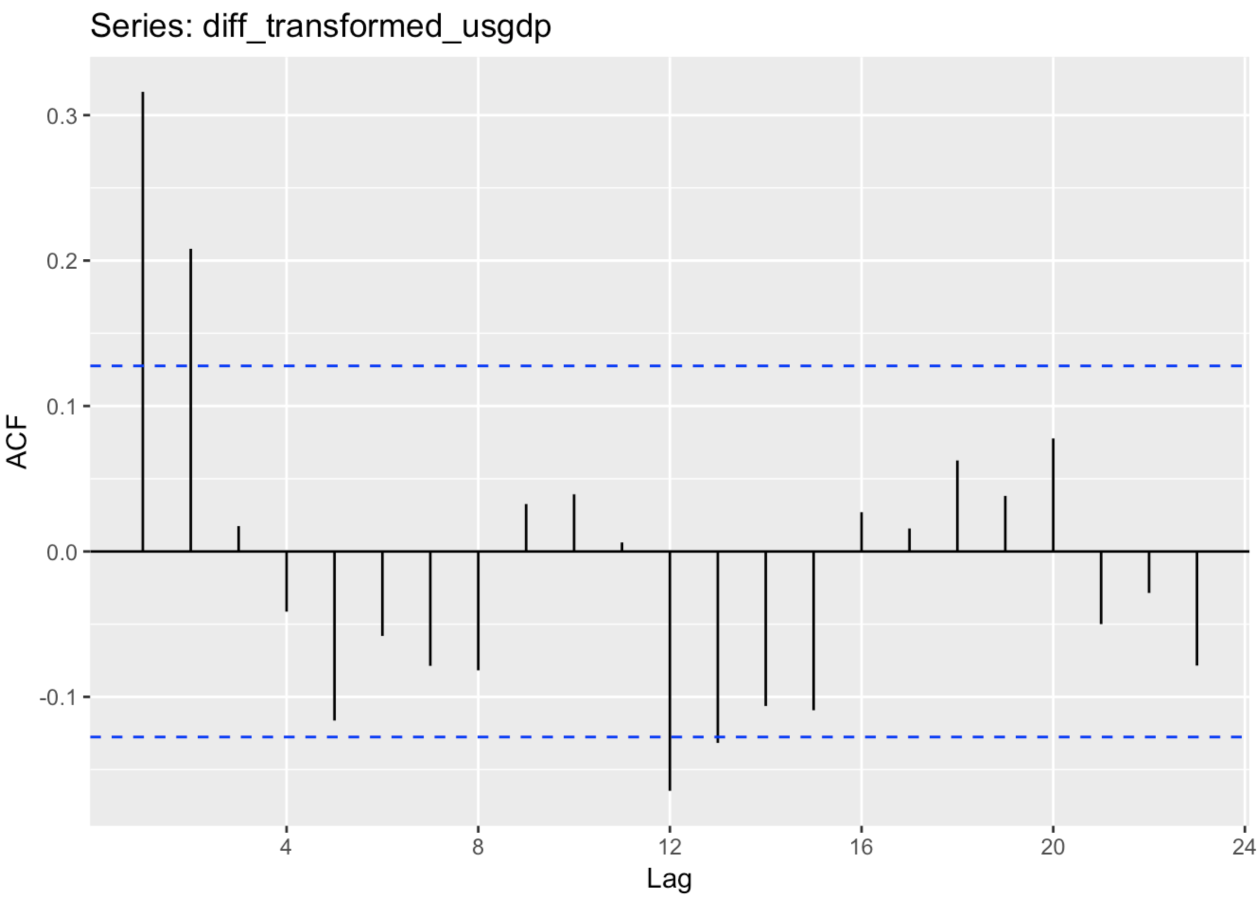

```
# After trasformation the data seems to be more linear than data without trans
formation.


#b)

autoar_usgdp <- auto.arima(usgdp, lambda = lambdaa)
autoar_usgdp
```

```
## Series: usgdp
## ARIMA(2,1,0) with drift
## Box Cox transformation: lambda= 0.366352
##
## Coefficients:
##          ar1     ar2   drift
##       0.2795  0.1208  0.1829
## s.e.  0.0647  0.0648  0.0202
##
## sigma^2 estimated as 0.03518:  log likelihood=61.56
## AIC=-115.11   AICc=-114.94   BIC=-101.26
```

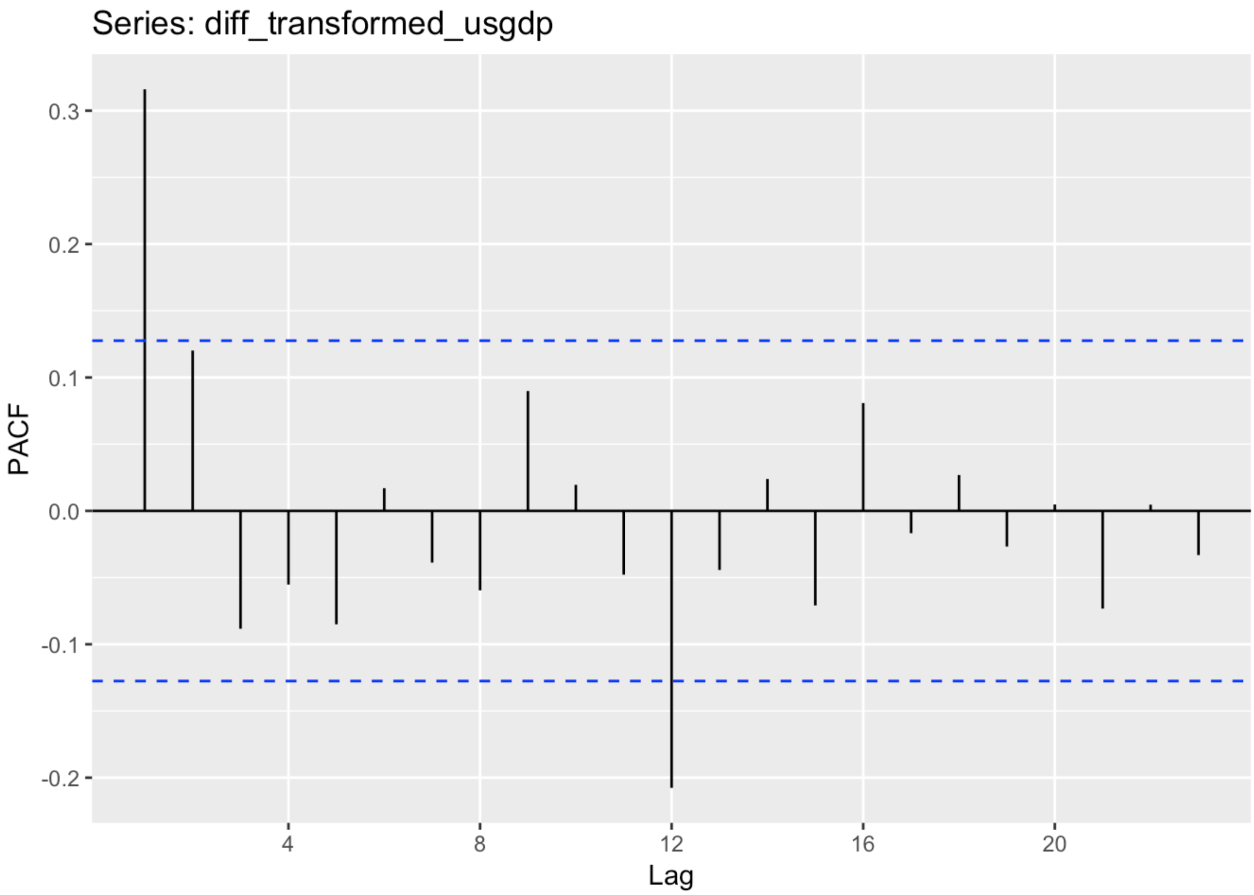
```
# auto_arima  gave (2,1,0) is the best model.



#c)

# first order differencing
ndiffs(BoxCox(usgdp,lambdaa))
```

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

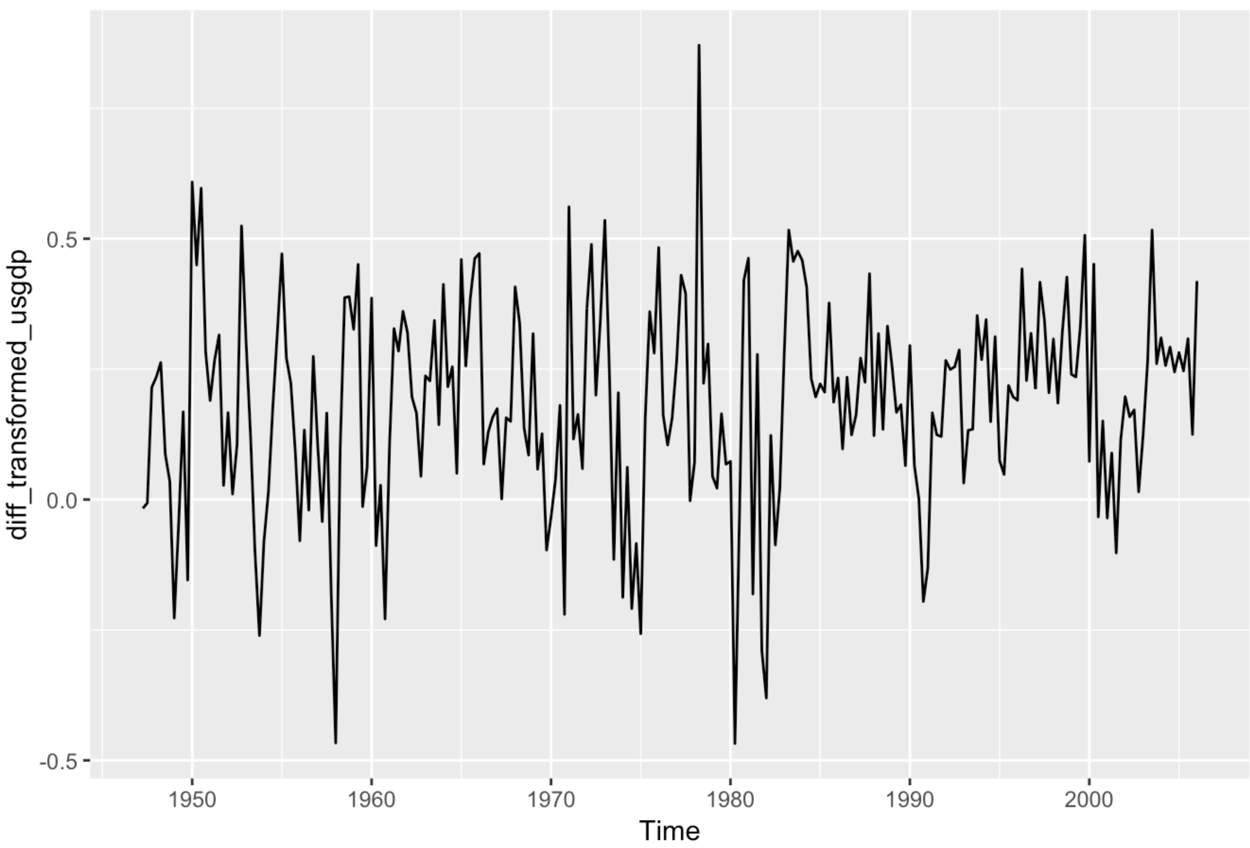
```
# requires 1 differencing

diff_transformed_usgdp <- diff(transformed_usgdp)

ggAcf(diff_transformed_usgdp)
```



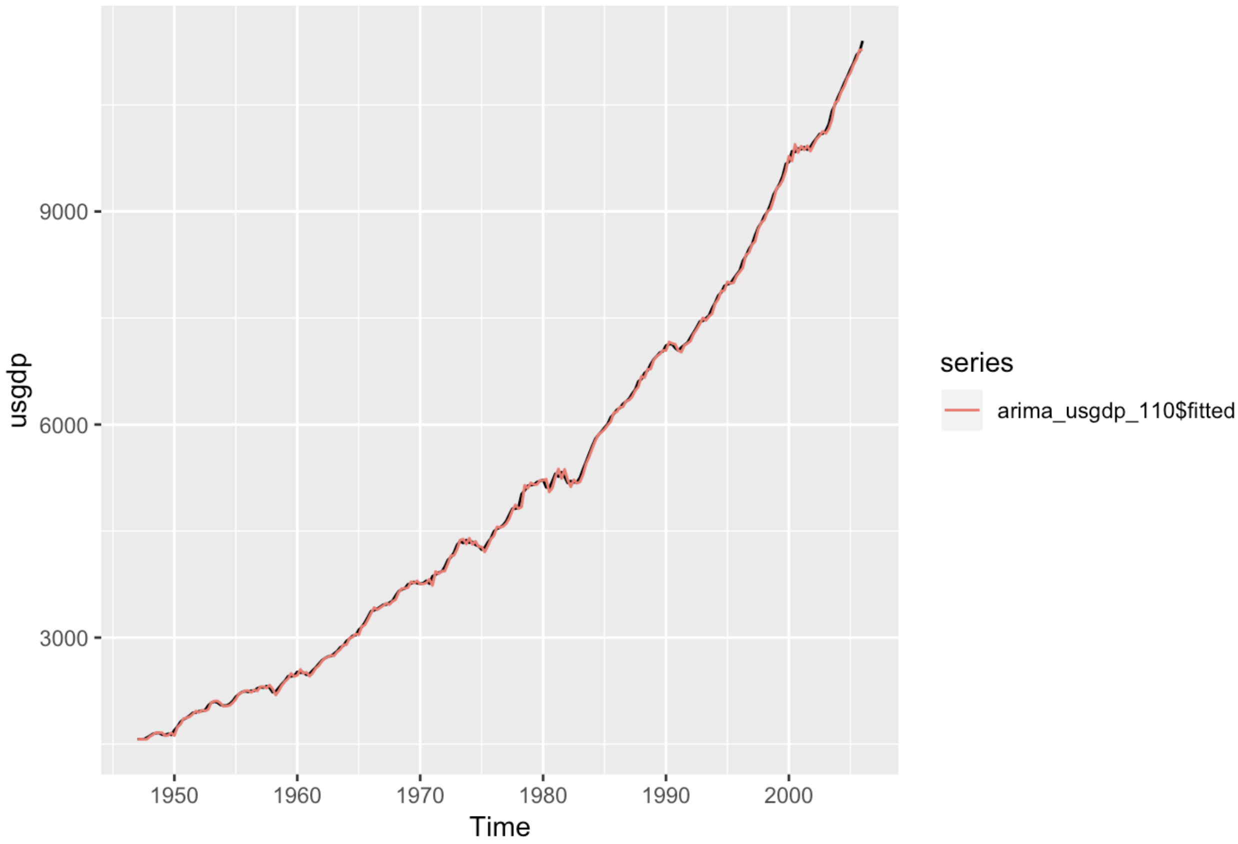

```
# From the above plot, we can see the spikes at lag 1, lag2 and lag 12

ggPacf(diff_transformed_usgdp)
```



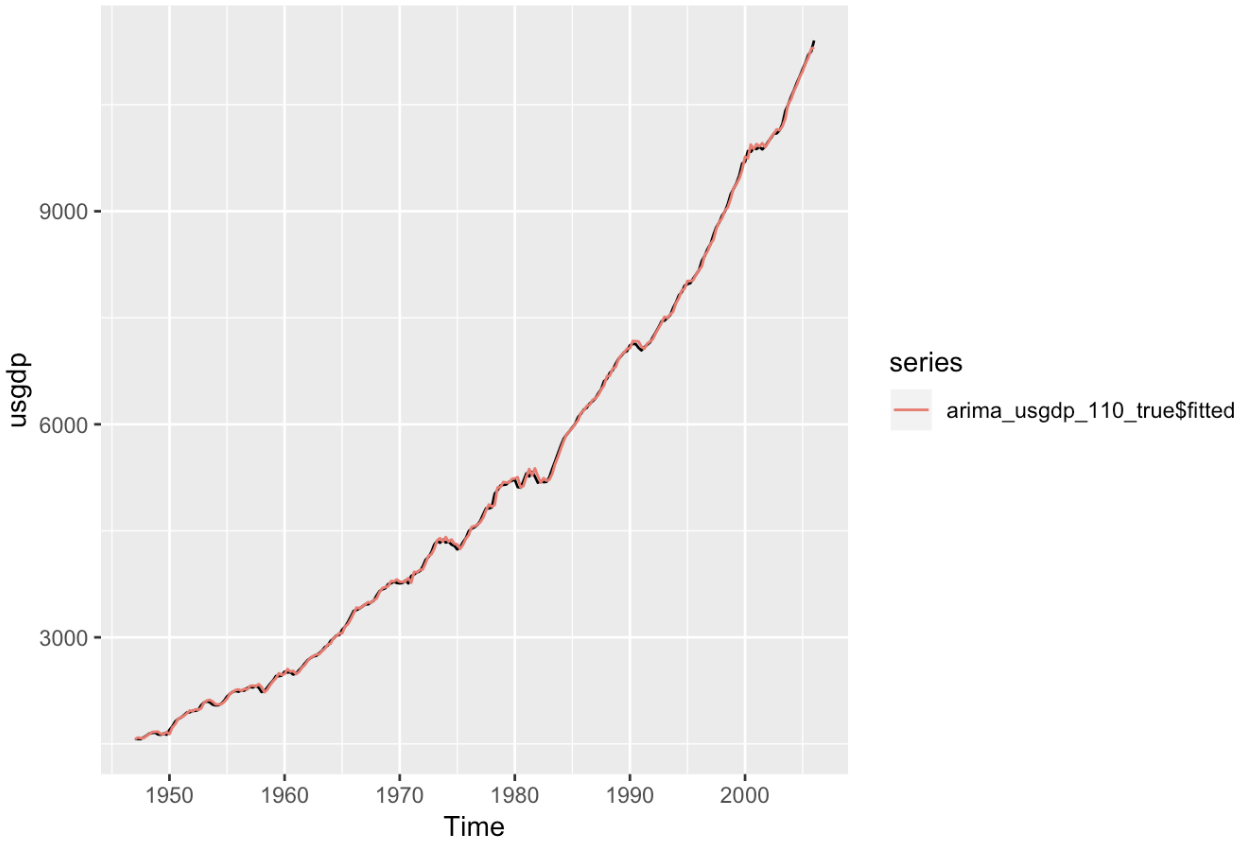

```
# From the above plot, we can see the spikes at lag 1 and lag 12

autoplot(diff_transformed_usgdp)
```



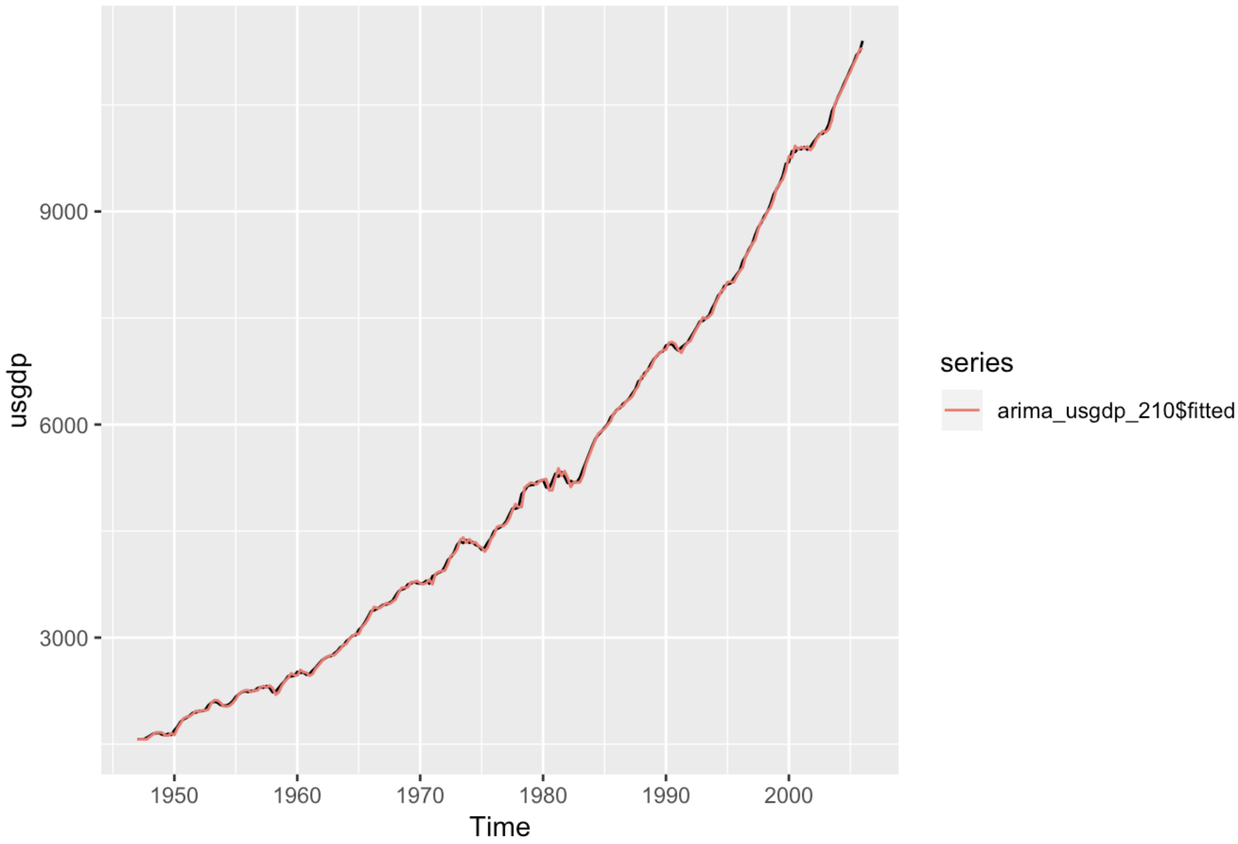

```
# The data seems to be stationary

#(1,1,0)
arima_usgdp_110 <- Arima( usgdp, lambda = lambdaa, order = c(1, 1, 0),include.
drift = FALSE)
autoplot(usgdp)+autolayer(arima_usgdp_110$fitted)
```



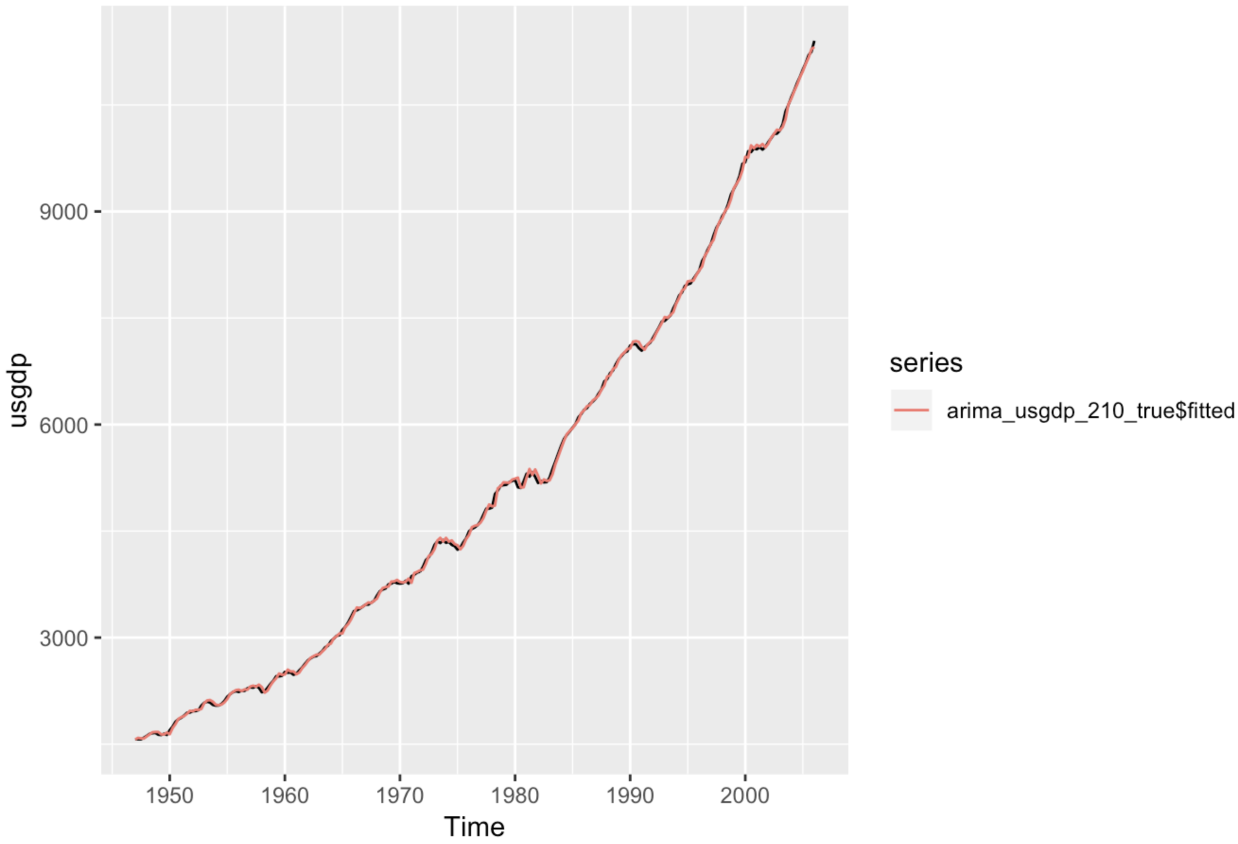

```
#include Drift
arima_usgdp_110_true <- Arima( usgdp, lambda = lambdaa, order = c(1, 1, 0), in
clude.drift = TRUE)
autoplot(usgdp)+autolayer(arima_usgdp_110_true$fitted)
```




```
# from the graphs both are fitting properly

#(2,1,0)

arima_usgdp_210 <- Arima( usgdp, lambda = lambdaa, order = c(2, 1, 0),include.
drift = FALSE)
autoplot(usgdp)+autolayer(arima_usgdp_210$fitted)
```




```
#include Drift
arima_usgdp_210_true <- Arima( usgdp, lambda = lambdaa, order = c(2, 1, 0), in
clude.drift = TRUE)
autoplot(usgdp)+autolayer(arima_usgdp_210_true$fitted)
```




```
#d)
accuracy(arima_usgdp_110)
```

```
##                    ME     RMSE      MAE       MPE      MAPE     MASE
## Training set 15.45449 45.49569 35.08393 0.3101283 0.7815664 0.198285
##                    ACF1
## Training set -0.3381619
```

```
accuracy(arima_usgdp_110_true)
```

```
##                    ME     RMSE     MAE         MPE      MAPE      MASE
## Training set 1.315796 39.90012 29.5802 -0.01678591 0.6834509 0.1671794
##                     ACF1
## Training set -0.08544569
```

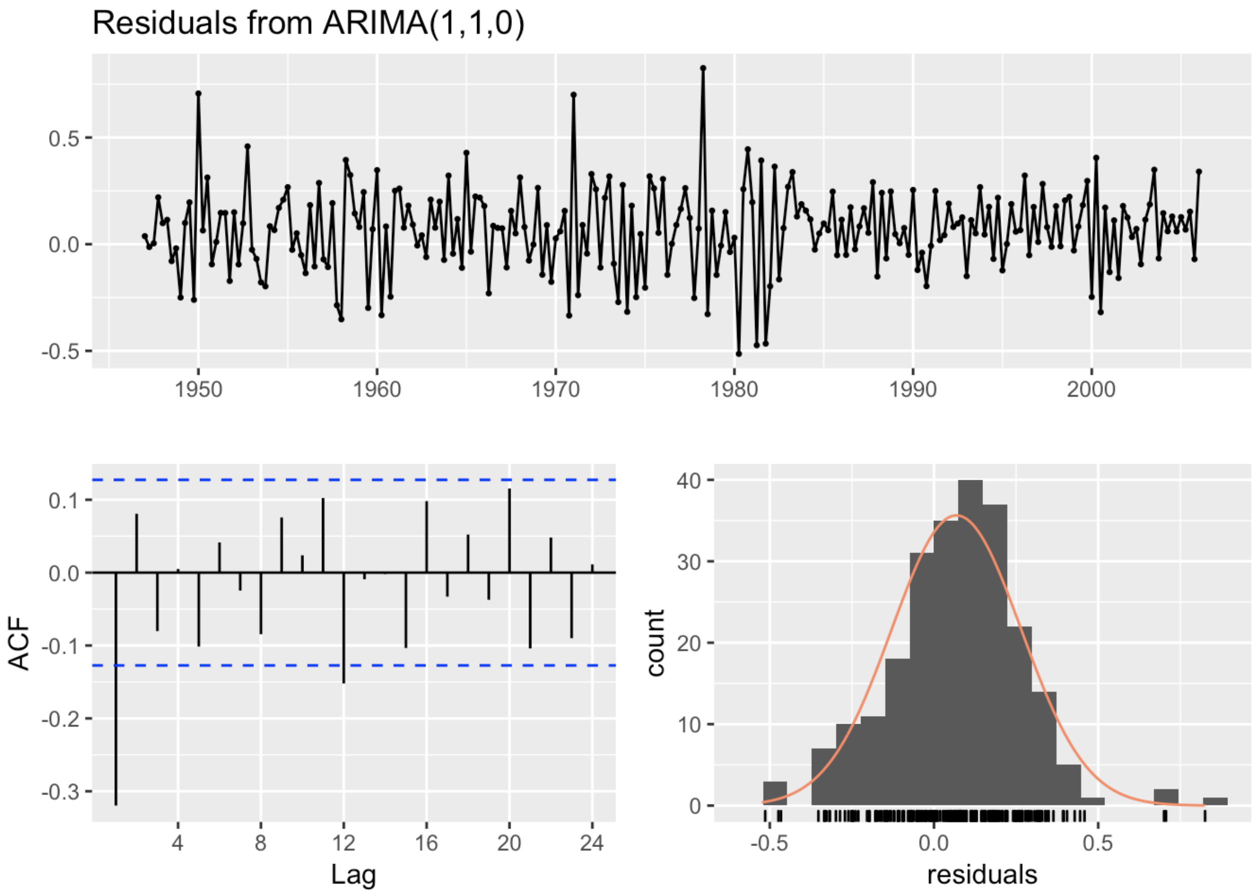
```
accuracy(arima_usgdp_210)
```

```
##                    ME     RMSE      MAE       MPE      MAPE      MASE
## Training set 11.06831 42.21158 32.00633 0.2244843 0.7393035 0.1808912
##                    ACF1
## Training set -0.1210488
```

```
accuracy(arima_usgdp_210_true)
```

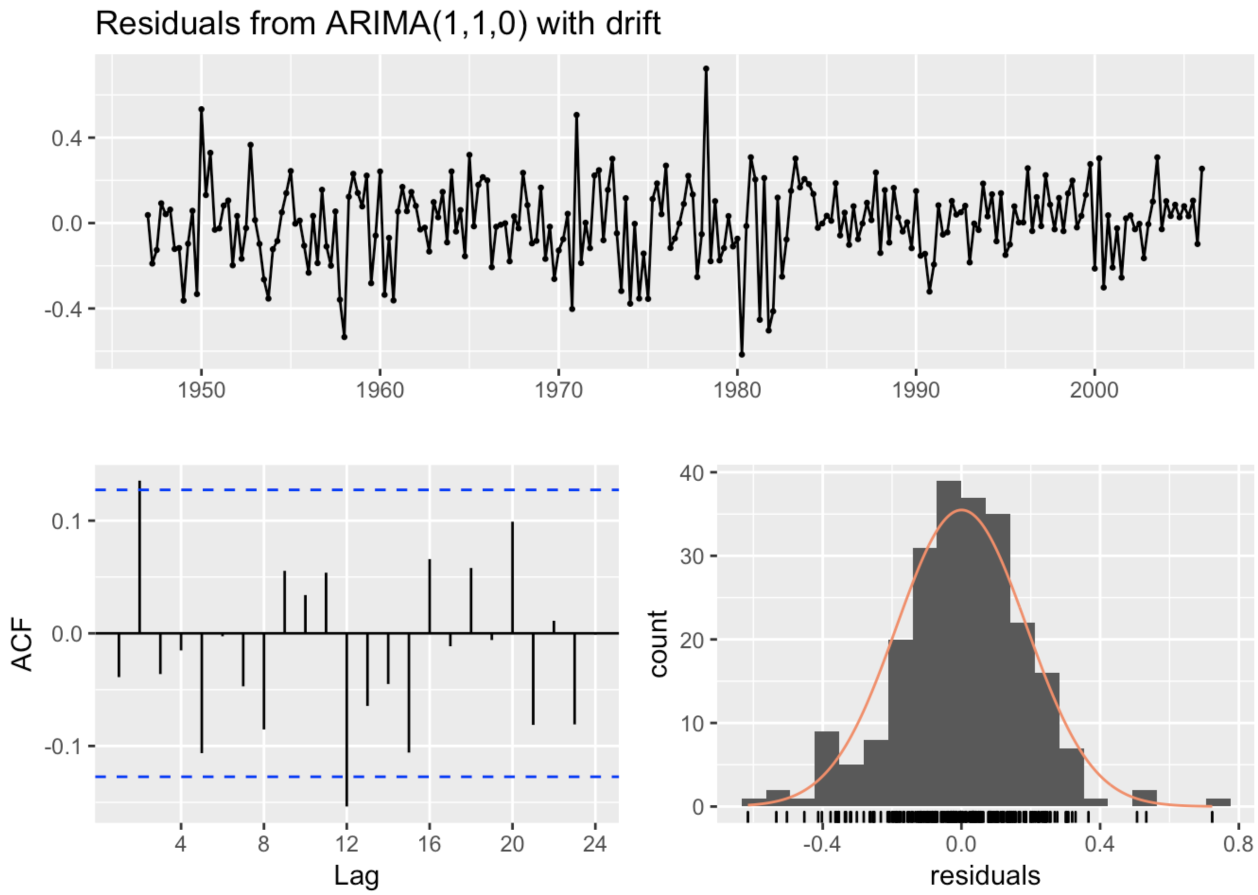
```
##                    ME     RMSE      MAE         MPE      MAPE      MASE
## Training set 1.195275 39.2224 29.29521 -0.01363259 0.6863491 0.1655687
##                     ACF1
## Training set -0.03824844
```

```
# from the above models we can see that  when we include drife we are getting
lower RMSE and MAE.

checkresiduals(arima_usgdp_110)
```



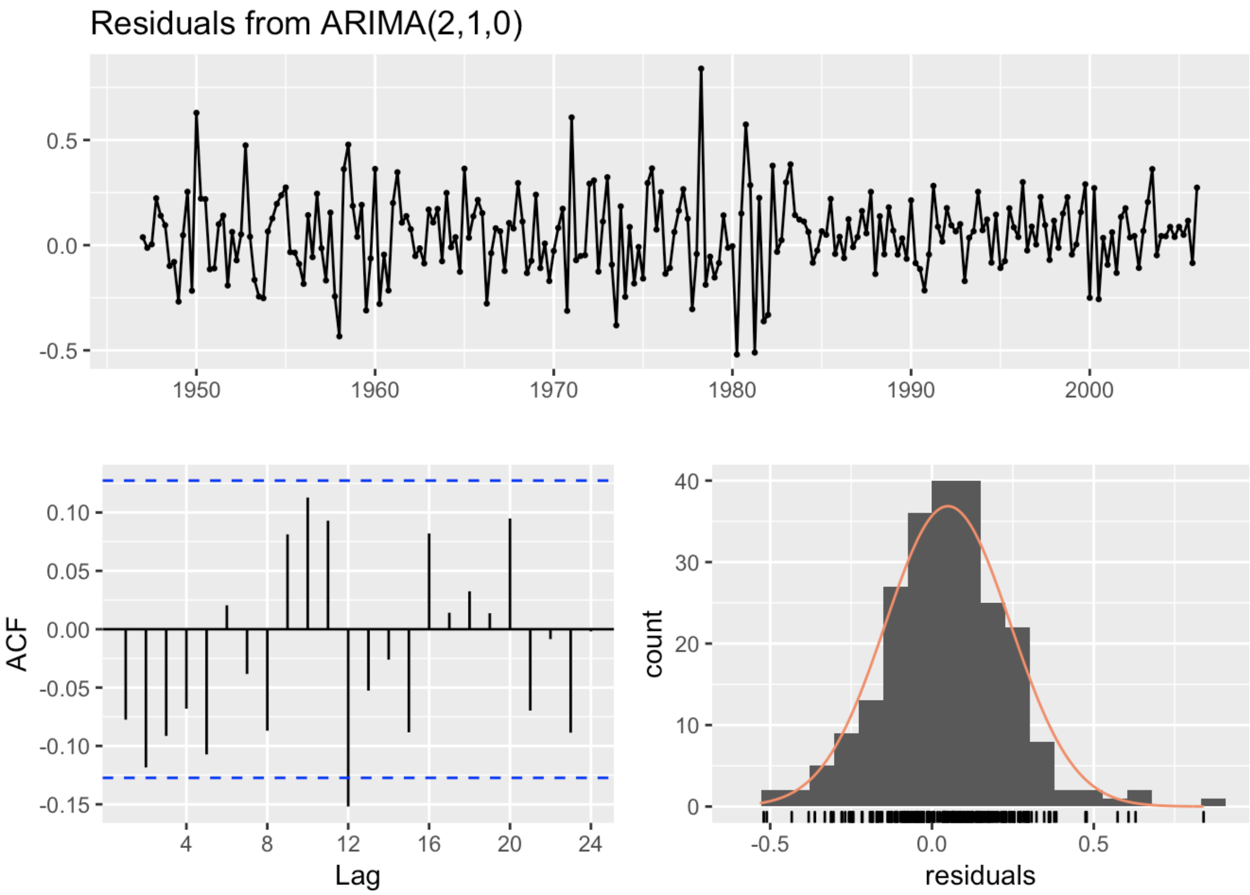

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)
## Q* = 32.515, df = 7, p-value = 3.259e-05
##
## Model df: 1.   Total lags used: 8
```

```
checkresiduals(arima_usgdp_110_true)
```



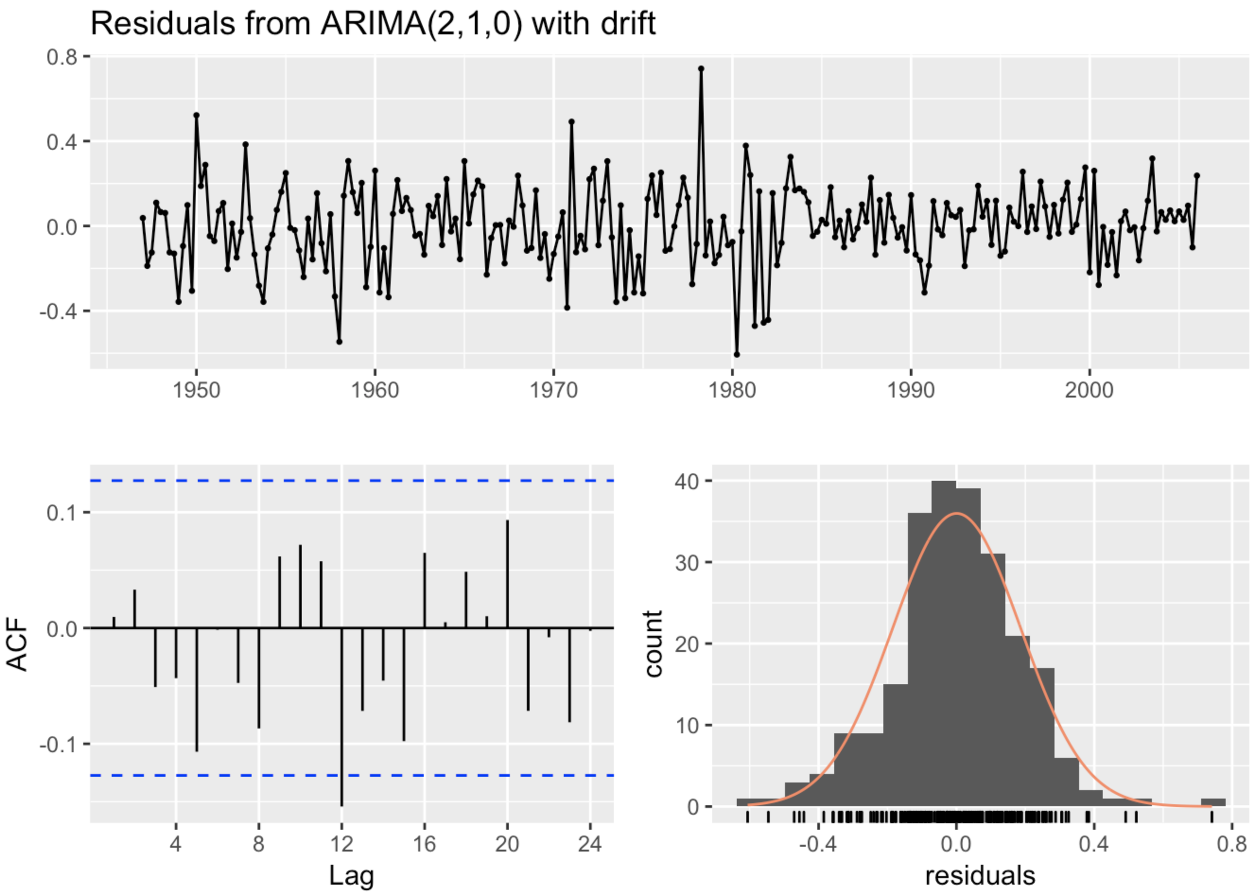

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0) with drift
## Q* = 10.274, df = 6, p-value = 0.1136
##
## Model df: 2.   Total lags used: 8
```

```
checkresiduals(arima_usgdp_210)
```



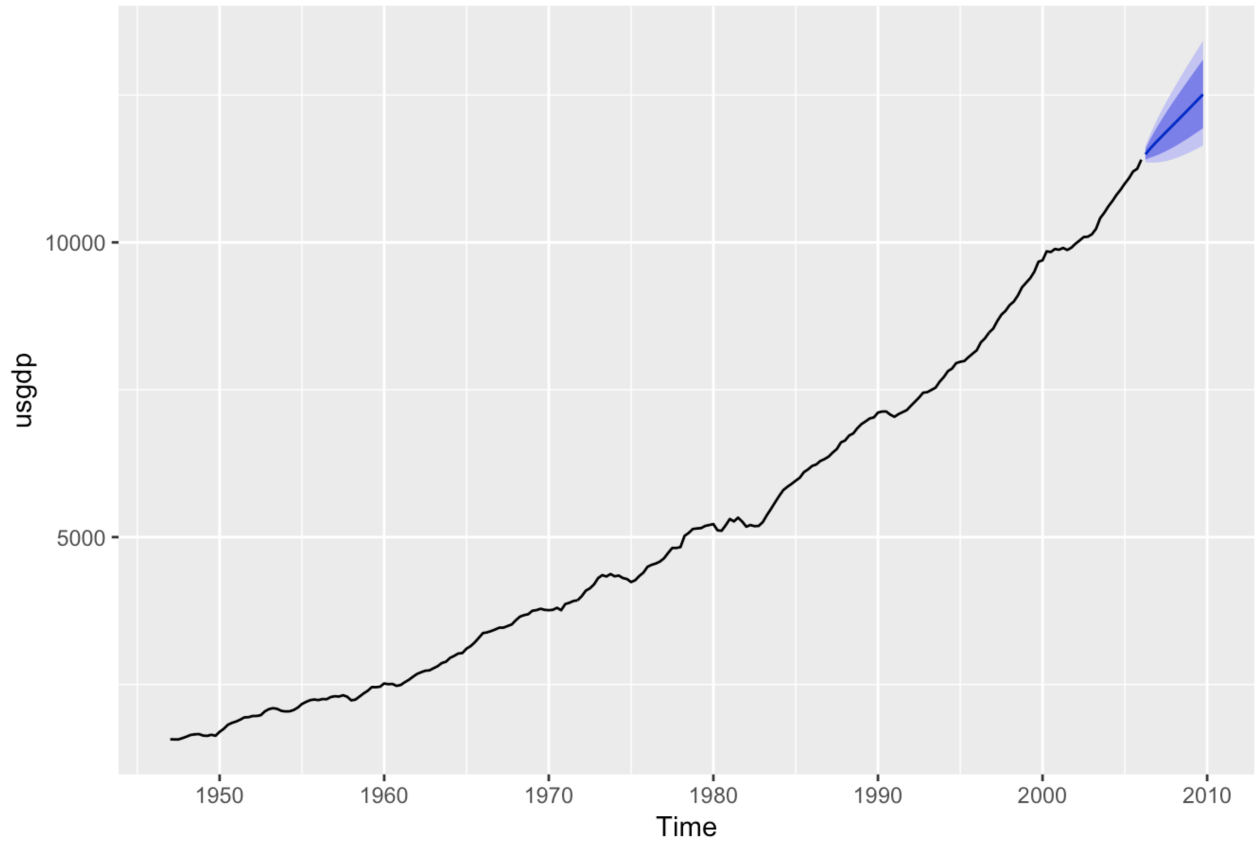

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)
## Q* = 13.115, df = 6, p-value = 0.04125
##
## Model df: 2.   Total lags used: 8
```

```
checkresiduals(arima_usgdp_210_true)
```

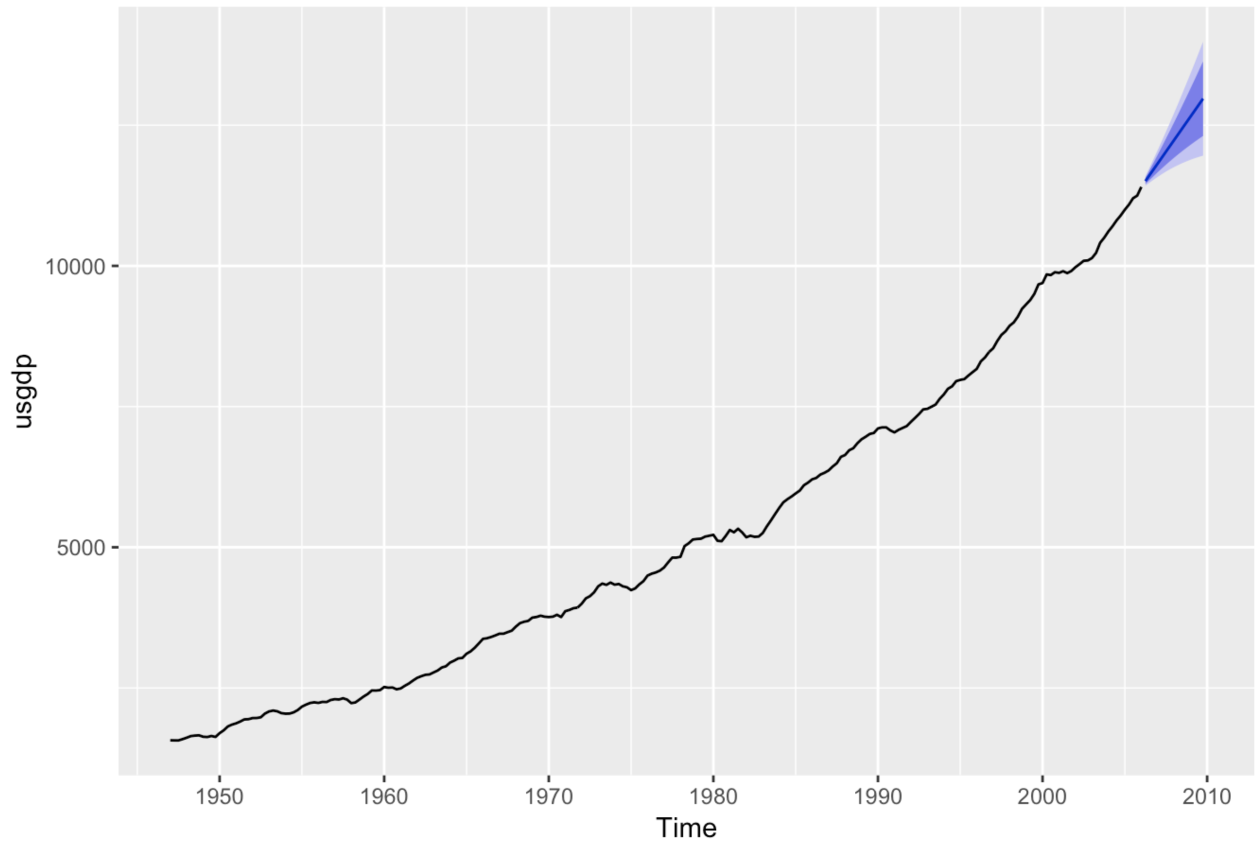



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0) with drift
## Q* = 6.5772, df = 5, p-value = 0.254
##
## Model df: 3.   Total lags used: 8
```

```
# we can infer that all the residuals are white noise.
# we are getting better accuracy for arima_usdp_210_true model because of low
MAE, RMSE.
# so arima_usdp_210_true gave better results than rest of the models.


# e)
autoplot(usgdp)+autolayer(forecast(arima_usgdp_210_true,h=15))
```




```
autoplot(usgdp)+ autolayer(forecast(ets(usgdp),h=15))
```



```
# both model look almost same, from the model we can see that ets is more line
ar and where as forecasting using arima model is taking consideration of lags


##for understanding the results
train <- usgdp[1:230]
test  <- usgdp[230:237]


arima_usgdp_210_true_train <- Arima( train, lambda = lambdaa, order = c(2, 1,
0), include.drift = TRUE)
forecast(arima_usgdp_210_true_train,h=7)
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 231       10781.66 10694.68 10869.08 10648.82 10915.53
## 232       10853.26 10711.47 10996.23 10636.89 11072.39
## 233       10921.70 10729.74 11115.83 10628.99 11219.48
## 234       10988.71 10752.48 11228.21 10628.74 11356.32
## 235       11055.11 10779.22 11335.42 10634.96 11485.62
## 236       11121.30 10809.48 11438.77 10646.68 11609.12
## 237       11187.53 10842.70 11539.23 10662.91 11728.21
```

```
forecast(ets(usgdp),h=7)
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2006 Q2       11508.06 11454.38 11561.75 11425.96 11590.16
## 2006 Q3       11612.53 11525.42 11699.64 11479.31 11745.76
## 2006 Q4       11717.00 11596.31 11837.69 11532.42 11901.58
## 2007 Q1       11821.47 11665.72 11977.22 11583.27 12059.67
## 2007 Q2       11925.94 11733.29 12118.59 11631.30 12220.57
## 2007 Q3       12030.41 11798.94 12261.87 11676.41 12384.40
## 2007 Q4       12134.87 11862.68 12407.07 11718.58 12551.17
```

```
# from the above results we can say that arima model performed better than et
s.
```

```
# 3)

##a)

summary(books)
```

```
##    Paperback       Hardcover
##  Min.   :111.0   Min.   :128.0
##  1st Qu.:167.2   1st Qu.:170.5
##  Median :189.0   Median :200.5
##  Mean   :186.4   Mean   :198.8
##  3rd Qu.:207.2   3rd Qu.:222.0
##  Max.   :247.0   Max.   :283.0
```

```
paperback = books[,"Paperback"]
Hardcover = books[,"Hardcover"]

# forecast for four days paperback:
results_paperback_holt <- holt(paperback, h = 4)
results_paperback_holt
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       209.4668 166.6035 252.3301 143.9130 275.0205
## 32       210.7177 167.8544 253.5811 145.1640 276.2715
## 33       211.9687 169.1054 254.8320 146.4149 277.5225
## 34       213.2197 170.3564 256.0830 147.6659 278.7735
```

```
# forecast for four days Hardcover:
results_hardcover_holt <- holt(Hardcover, h = 4)
results_hardcover_holt
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       250.1739 212.7390 287.6087 192.9222 307.4256
## 32       253.4765 216.0416 290.9113 196.2248 310.7282
## 33       256.7791 219.3442 294.2140 199.5274 314.0308
## 34       260.0817 222.6468 297.5166 202.8300 317.3334
```

```
## b)
results_paperback_ses <- ses(paperback, h= 4)
results_paperback_ses
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       207.1097 162.4882 251.7311 138.8670 275.3523
## 32       207.1097 161.8589 252.3604 137.9046 276.3147
## 33       207.1097 161.2382 252.9811 136.9554 277.2639
## 34       207.1097 160.6259 253.5935 136.0188 278.2005
```

```
results_hardcover_ses <- ses(Hardcover, h= 4)
results_hardcover_ses
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       239.5601 197.2026 281.9176 174.7799 304.3403
## 32       239.5601 194.9788 284.1414 171.3788 307.7414
## 33       239.5601 192.8607 286.2595 168.1396 310.9806
## 34       239.5601 190.8347 288.2855 165.0410 314.0792
```

```
accuracy(results_hardcover_holt)
```

```
##                      ME     RMSE      MAE       MPE    MAPE      MASE
## Training set -0.1357882 27.19358 23.15557 -2.114792 12.1626 0.6908555
##                     ACF1
## Training set -0.03245186
```

```
accuracy(results_hardcover_ses)
```

```
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 9.166735 31.93101 26.77319 2.636189 13.39487 0.7987887
##                    ACF1
## Training set -0.1417763
```

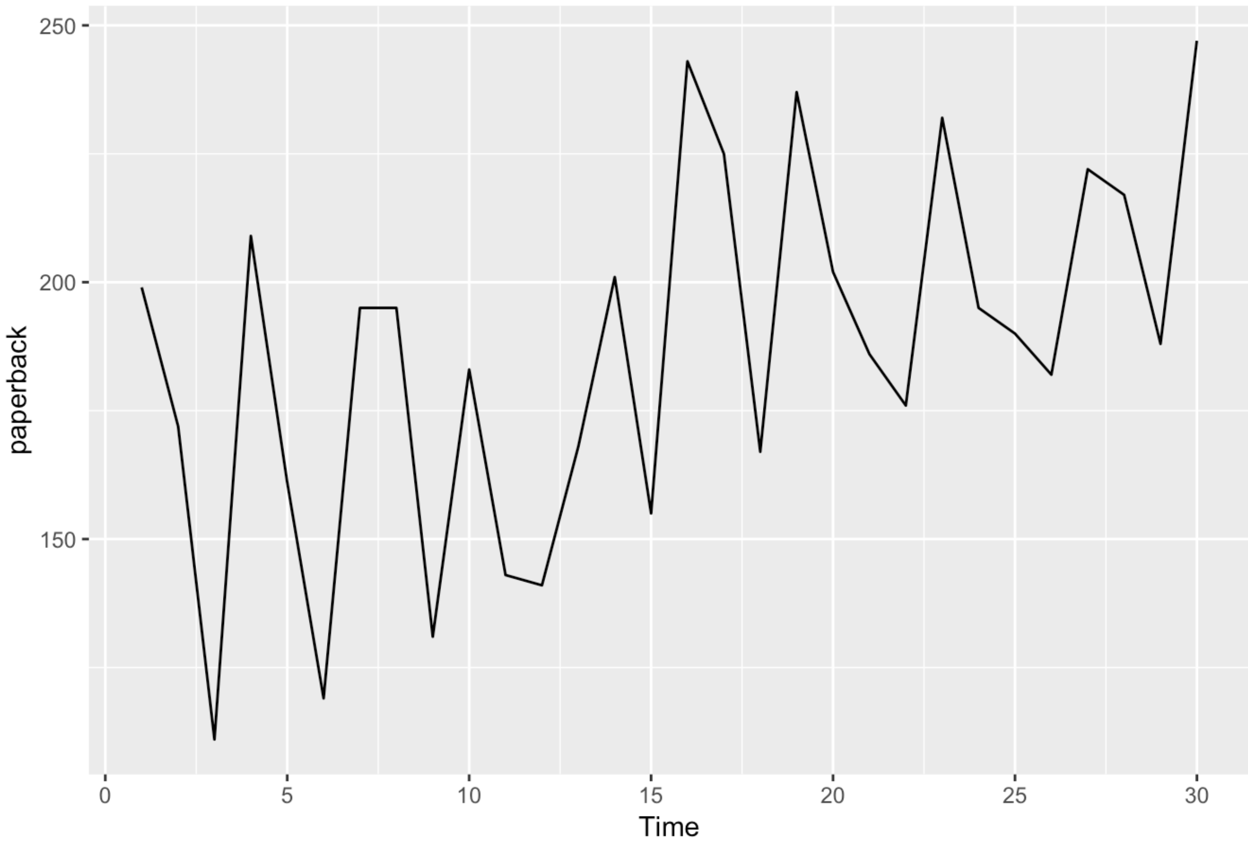
```
accuracy(results_paperback_ses)
```

```
##                    ME     RMSE     MAE       MPE     MAPE      MASE
## Training set 7.175981 33.63769 27.8431 0.4736071 15.57784 0.7021303
##                    ACF1
## Training set -0.2117522
```

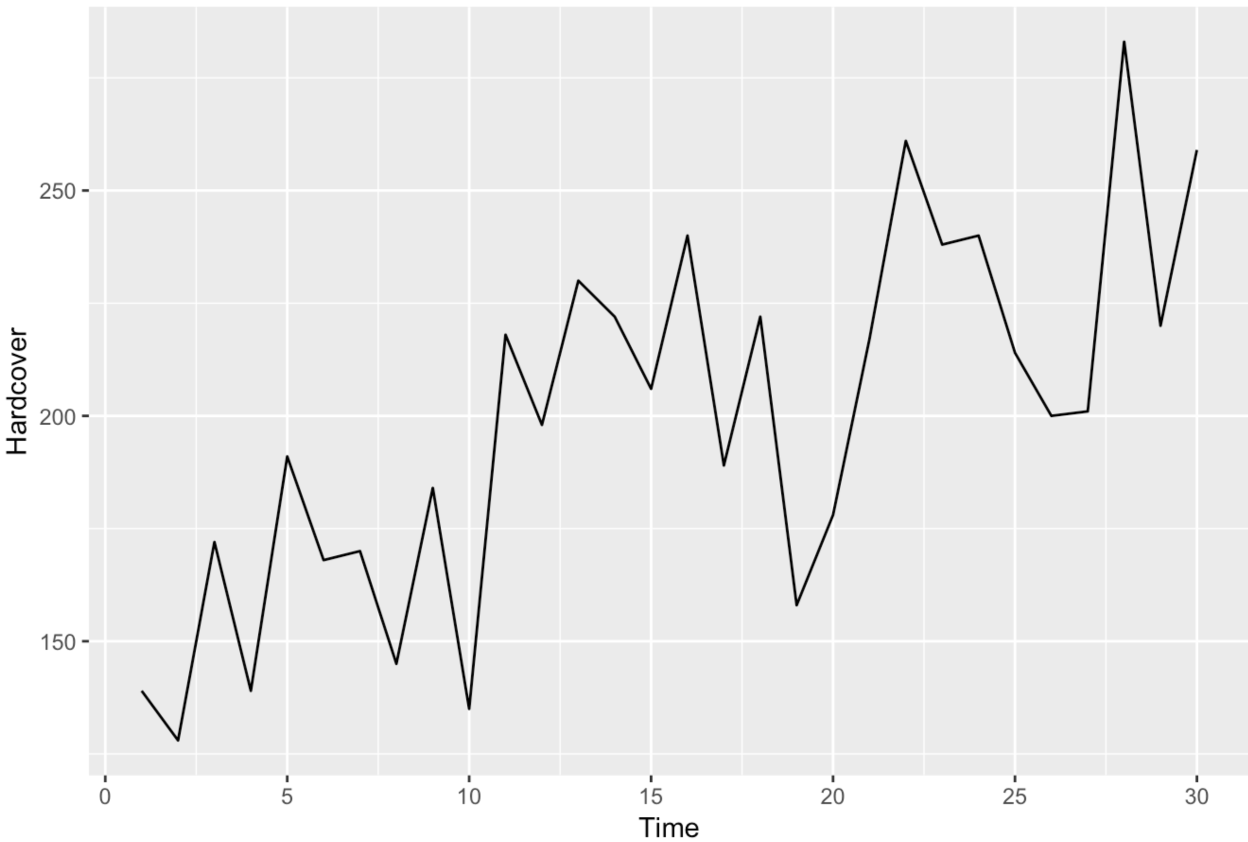
```
accuracy(results_paperback_holt)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -3.717178 31.13692 26.18083 -5.508526 15.58354 0.6602122
##                    ACF1
## Training set -0.1750792
```

```
#plots
autoplot(paperback)
```




```
autoplot(Hardcover)
```



```
## from the above results we see that holt's method has less RMSE for both har
dcover and paperback.
##  from the plot we can that data cannot be approximated to a linear model. F
rom the literature of holt's method we see that it uses 2 paramenters atleast
one for trend smooth and second for overall smoothing. since there is only one
parameter we prefer to take simple expoential smoothing.



## c)
## from the above results we see that holt's method has less RMSE for both har
dcover and paperback.
##  from the plot we can that data cannot be approximated to a linear model. F
rom the literature of holt's method we see that it uses 2 paramenters atleast
one for trend smooth and second for overall smoothing. since there is only one
parameter we prefer to take simple expoential smoothing.
## RMSE, MAE, MPE is lower for holt's method. Among the series paperback and h
ardcover, hardcover had least RMSE MAE and MPE. From this inference we can con
clude that hardcover timeseries did best among all.

## d)
# first forecast method for each series used was the holt's method

acc_hardcover_holt <- accuracy(results_hardcover_holt)
acc_hardcover_ses <- accuracy(results_hardcover_ses)
acc_paperback_holt <- accuracy(results_paperback_ses)
acc_paperback_ses <-accuracy(results_paperback_holt)

acc_hardcover_holt[,"RMSE"]
```

```
## [1] 27.19358
```

```
##Formula based - paperback method:holt
paperback_holt_upper <- results_paperback_holt$upper[1, "95%"]
paperback_holt_upper
```

```
##      95%
## 275.0205
```

```
paperback_holt_lower <- results_paperback_holt$lower[1, "95%"]
paperback_holt_lower
```

```
##     95%
## 143.913
```

```
##calculation
upper = results_paperback_holt$mean[1] + (1.96*(accuracy(results_paperback_hol
t)[,"RMSE"]))
upper
```

```
## [1] 270.4951
```

```
lower = results_paperback_holt$mean[1] - (1.96*(accuracy(results_paperback_hol
t)[,"RMSE"]))
lower
```

```
## [1] 148.4384
```

```
#Fomula based - hardcover method: holt
hardcover_holt_upper <- results_hardcover_holt$upper[1, "95%"]
hardcover_holt_upper
```

```
##      95%
## 307.4256
```

```
hardcover_holt_lower <- results_hardcover_holt$lower[1, "95%"]
hardcover_holt_lower
```

```
##      95%
## 192.9222
```

```
# calculation
upper = results_hardcover_holt$mean[1] + (1.96*(accuracy(results_hardcover_hol
t)[,"RMSE"]))
upper
```

```
## [1] 303.4733
```

```
lower = results_hardcover_holt$mean[1] - (1.96*(accuracy(results_hardcover_hol
t)[,"RMSE"]))
lower
```

```
## [1] 196.8745
```

```
# In holts method upper and lower limits are almost same to that of formula to
calculated.


##Formula based - paperback method:ses
paperback_ses_upper <- results_paperback_ses$upper[1, "95%"]
paperback_ses_upper
```

```
##      95%
## 275.3523
```

```
paperback_ses_lower <- results_paperback_ses$lower[1, "95%"]
paperback_ses_lower
```

```
##     95%
## 138.867
```

```
##calculation
upper = results_paperback_ses$mean[1] + (1.96*(accuracy(results_paperback_se
s)[,"RMSE"]))
upper
```

```
## [1] 273.0395
```

```
lower = results_paperback_ses$mean[1] - (1.96*(accuracy(results_paperback_se
s)[,"RMSE"]))
lower
```

```
## [1] 141.1798
```

```
#Fomula based - hardcover method: holt
hardcover_ses_upper <- results_hardcover_ses$upper[1, "95%"]
hardcover_ses_upper
```

```
##      95%
## 304.3403
```

```
hardcover_ses_lower <- results_hardcover_ses$lower[1, "95%"]
hardcover_ses_lower
```

```
##      95%
## 174.7799
```

```
# calculation
upper = results_hardcover_ses$mean[1] + (1.96*(accuracy(results_hardcover_se
s)[,"RMSE"]))
upper
```

```
## [1] 302.1449
```

```
lower = results_hardcover_ses$mean[1] - (1.96*(accuracy(results_hardcover_se
s)[,"RMSE"]))
lower
```

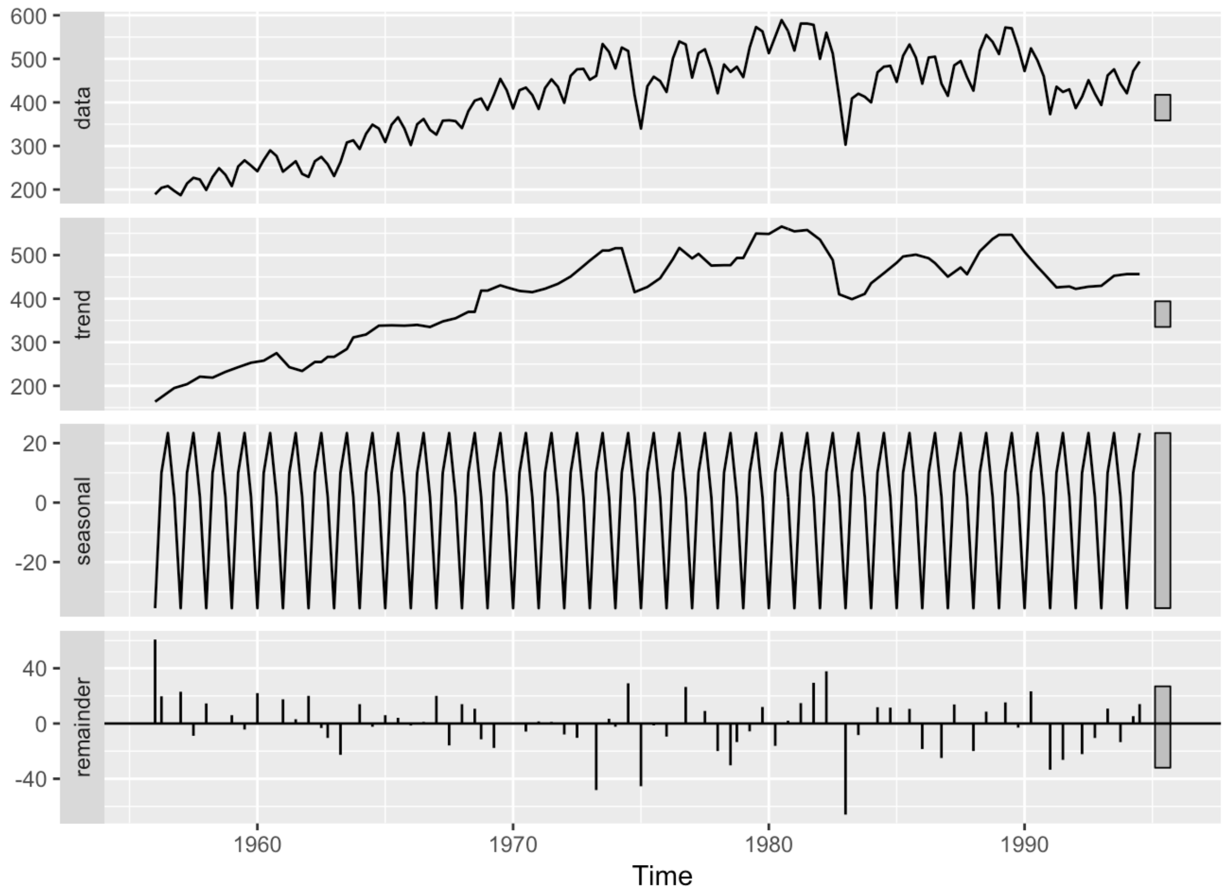
```
## [1] 176.9753
```

```
# In ses method upper and lower limits are almost same to that of formula to c
alculated.

# The intervals produced by ses and holt are almost equal.
```

```
## a)
summary(bricksq)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   187.0   338.5   428.0   408.8   490.5   589.0
```

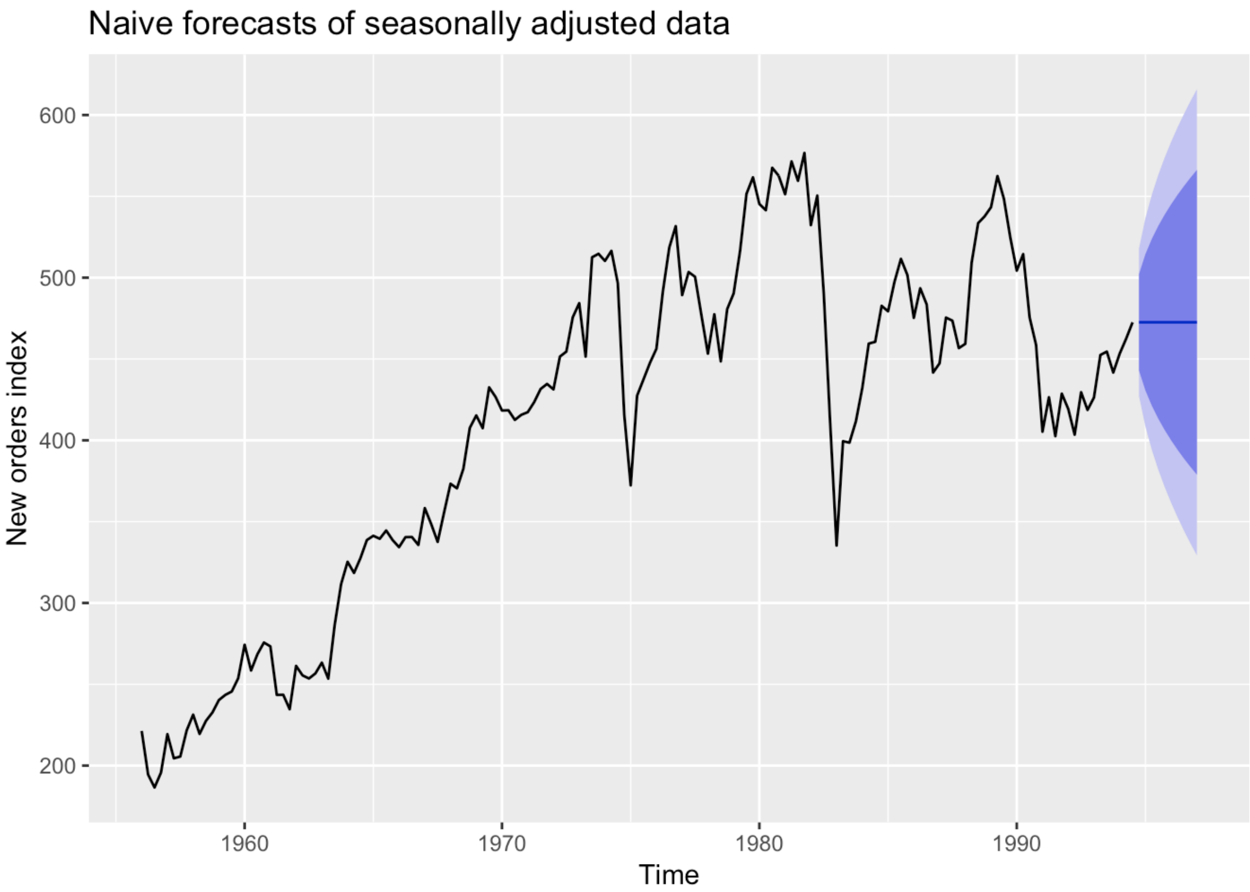
```
 bricksq%>%
  stl(t.window=4, s.window="periodic", robust=TRUE) %>%
  autoplot()
```



```
# Trend cycle  increases over time and becomes after a certain level (1980).
# seasonal component remains constant over time.
# remainder are low.


#fix_chang_stl <-  bricksq%>%
  # stl(t.window=4, s.window="periodic", robust=TRUE) %>%
  # autoplot()

 #fixed seasonlity
fixed_sea_stl <- bricksq%>%
   stl(s.window="periodic", robust=TRUE)

 #changing seasonality

changing_sea_stl <- bricksq%>%
   stl(s.window=4, robust=TRUE)


#b)
autoplot(bricksq, series ="data") +autolayer(seasadj(fixed_sea_stl),series = "
seasonally_adjusted_data")
```


```
autoplot(bricksq, series ="data") +autolayer(seasadj(changing_sea_stl),series
= "seasonally_adjusted_data")
```


```
#autoplot(bricksq, series ="data") + autolayer(seasadj(fixed_sea_stl), series
= "seasonally_adjusted_data")


#c)

 fit <- stl(bricksq, t.window=10, s.window="periodic",
            robust=TRUE)
 fit %>% seasadj() %>% naive() %>%
   autoplot() + ylab("New orders index") +
   ggtitle("Naive forecasts of seasonally adjusted data")
```




```
# The upper and lower limits of the prediction intervals on the seasonally adj
usted data are “reseasonalised” by adding in the forecasts of the seasonal com
ponent.
```