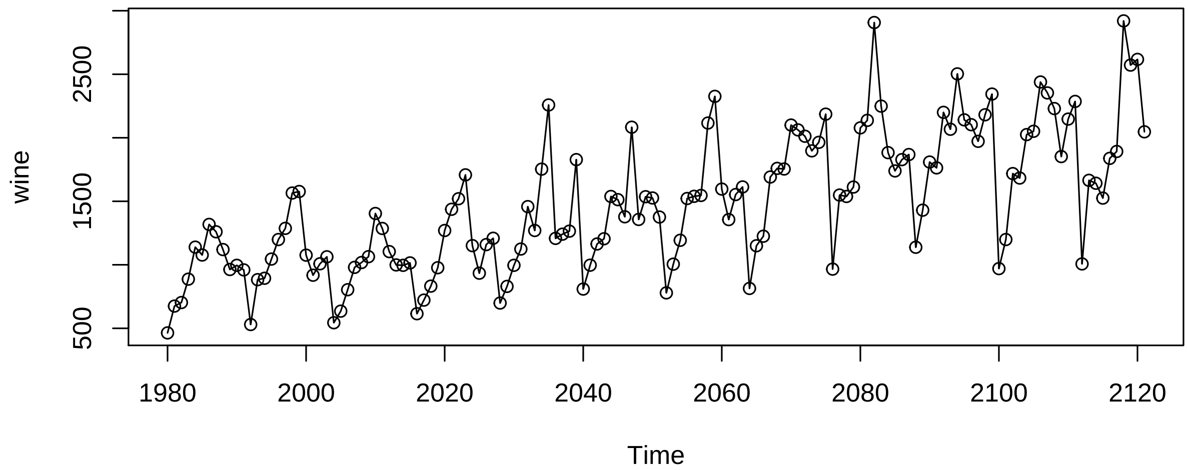
InClass-A3 Sample Analysis

Contents

Here is the code to load the data from the web.

```
source('https://nmimoto.github.io/R/TS-00.txt')

D  <- read.csv("https://nmimoto.github.io/datasets/wine.csv")
D1   <- ts(D, start=c(1980,1), freq=1)
plot(D1, type='o')
```



Now your “D1” in R contains monthly wine sales in Australia.

Preliminary Analysis

1. Does “D1” look like stationary time series?

State your graphical observations and conclusions drawn from p-values form Stationarity.tests().

```
Stationarity.tests(D1)
```

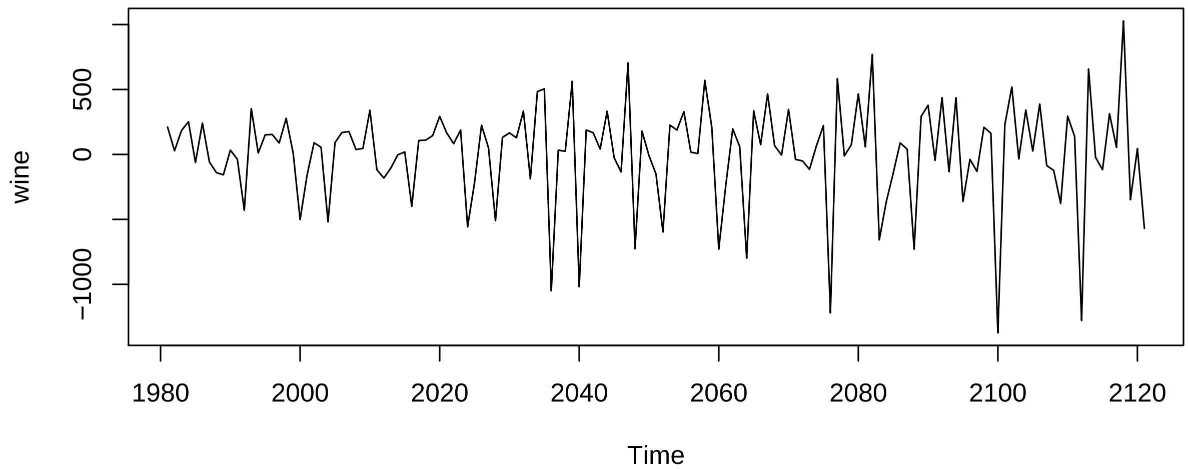
```
## Warning in adf.test(A): p-value smaller than printed p-value

## Warning in pp.test(A): p-value smaller than printed p-value

## Warning in kpss.test(A): p-value smaller than printed p-value

##         KPSS  ADF   PP
## p-val: 0.01 0.01 0.01
```

```
## KPSS and ADF conflicting
```

2. Take difference of D1 using diff(),

plot it and check the stationarity. State your graphical observations and conclusions drawn from p-values form Stationarity.tests().

```
plot(diff(D1))
```




```r
Stationarity.tests(diff(D1))
```

```
## Warning in adf.test(A): p-value smaller than printed p-value
```

```
## Warning in pp.test(A): p-value smaller than printed p-value
```

```
## Warning in kpss.test(A): p-value greater than printed p-value
```

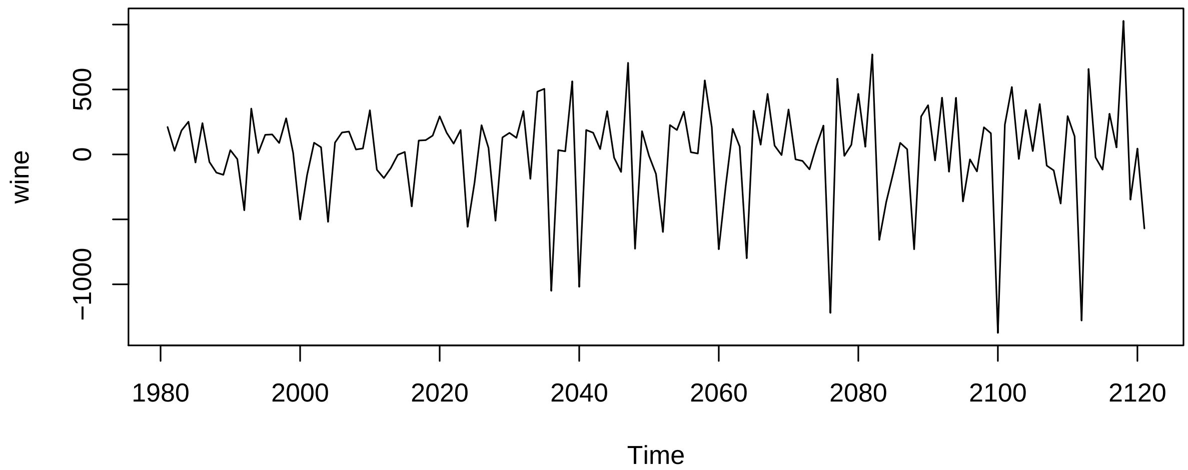
```
##         KPSS  ADF   PP
## p-val:   0.1 0.01 0.01
```

```r
##  d=1 is Stationary.
```

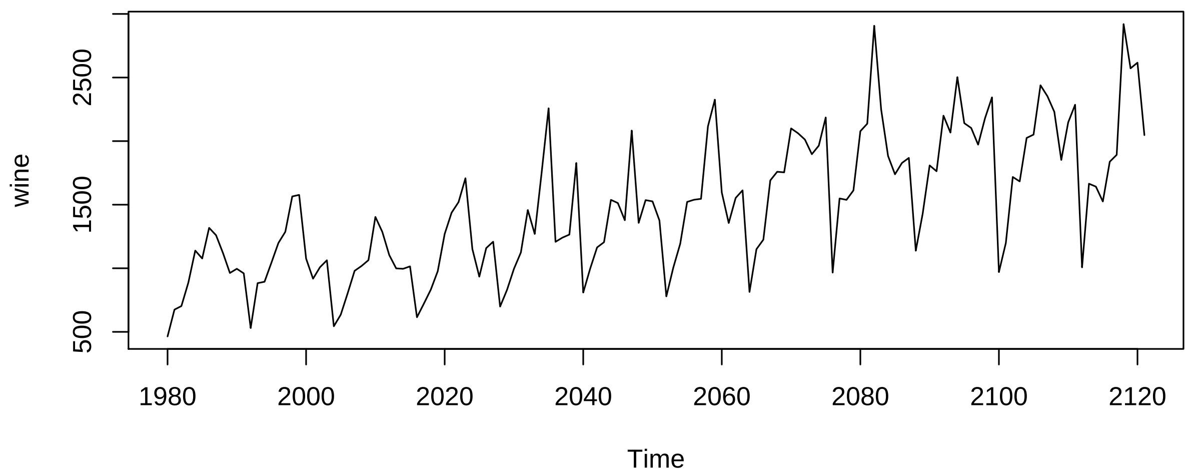
3. Should we take any transformation before differencing?

Why or why not? If yes, use Box-Cox power transformation. Pick your value of lambda.

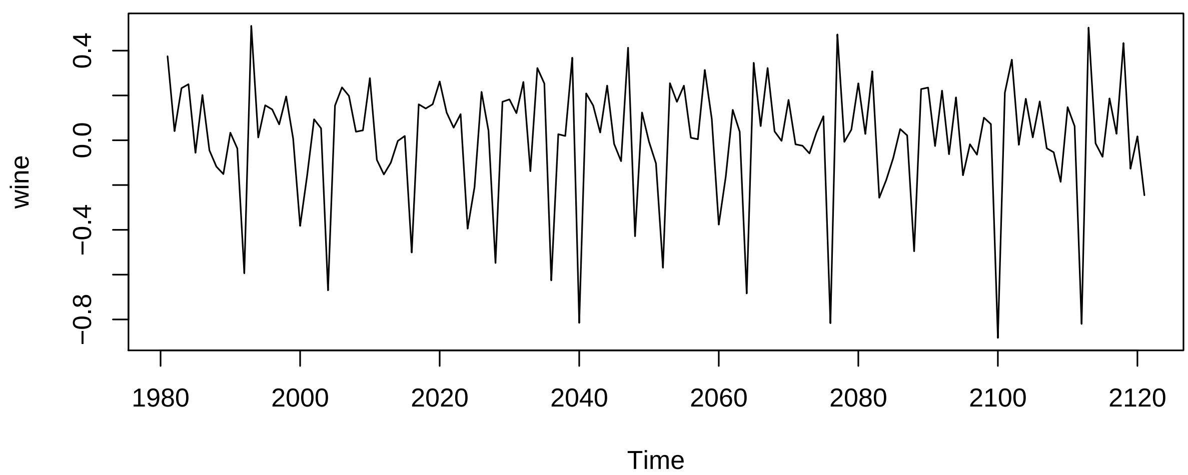
```r
plot(diff(D1))
```




```
plot((D1))
```




```
plot(diff(log(D1)))
```




```
## Even though D1 passed the stationarity tests, it looks like it
## has increasing variance problem.
## This is probably due to D1 increasing as time goes on.
## Taking a log will solve this problem.
##
## Set lambda=0.
```

For all problems below lambda=0.

ARIMA(d=1) analysis

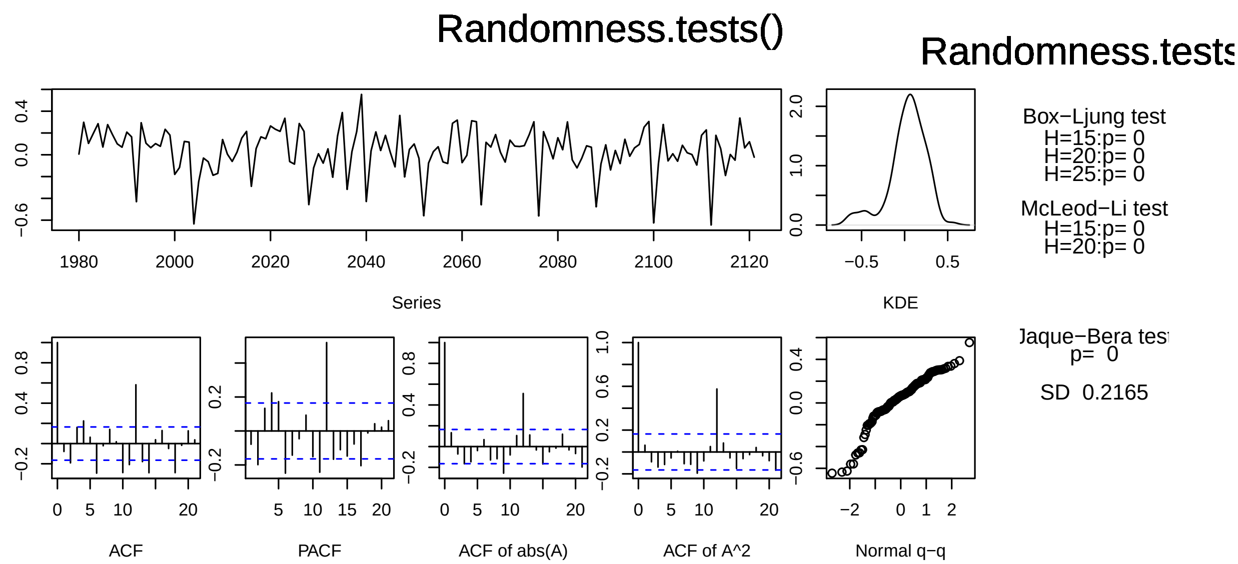
4. Use auto.arima() function to find best ARIMA(p,d,q)

model with constraint that d=1. What is the suggested model? (use stepwise=FALSE, approximation=FALSE option.) Does it pass the residual test for model adequacy? Copy and Paste the output from auto.arima() and Randomness.tests(). (not the plot)

```
Fit01 <- auto.arima(D1, d=1, lambda=0, stepwise=FALSE, approximation=FALSE)
Fit01
```

```
## Series: D1
## ARIMA(3,1,2)
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2
##       1.2830  -0.3135  -0.2752  -1.8894  0.9566
## s.e.  0.1063   0.1474   0.0929   0.1135  0.1178
##
## sigma^2 estimated as 0.04984:  log likelihood=11.51
## AIC=-11.02   AICc=-10.39   BIC=6.68
```

```r
Randomness.tests(Fit01$residuals)
```




```
##   B-L test H0: the series is uncorrelated
##   M-L test H0: the square of the series is uncorrelated
##   J-B test H0: the series came from Normal distribution
##   SD         : Standard Deviation of the series

##      BL15 BL20 BL25 ML15 ML20 JB    SD
## [1,]    0    0    0    0    0  0 0.216
```

```r
## ARIMA(3,1,2) is suggested.
##
## Residual analysis does not show adequate fit. P-value for Ljung-Box tests are
## too low, indicating there's still autocorrelation left in the residual.
##
## Acf plot confirms this.
```

5. Now we search for better ARIMA model without the guidance of AICc.

Start with ARIMA(15, 1, 15) with the drift model, use Arima() function to estimate parameters. Reduce p and/or q if the last parameter in AR or MA is not significant. Stop if the LAST parameter of both AR and MA term is significant. Remove the drift if not significant.
What is your final model? Compare AICc of your final model to the ones you got from (4). Which one is lower? Why did this model was not suggested in (4)? Does this final model pass the residual adequacy test? (Only include the output of your final model. Model pram + Residual p-values.)

```r
#- If you start removing AR(15) first

  Arima(D1, lambda=0, order=c(15,1,15), include.drift=TRUE)
    #AR15 not significant.
```

```r
  #Arima(D1, lambda=0, order=c(14,1,15), include.drift=TRUE)
    #This gives estimation error. use CSS.

  Arima(D1, lambda=0, order=c(14,1,15), include.drift=TRUE, method="CSS")
    #MA15 not sig

  Arima(D1, lambda=0, order=c(14,1,14), include.drift=TRUE)
    #AR14 not sig

  Arima(D1, lambda=0, order=c(13,1,14), include.drift=TRUE)
    #AR13, MA14 both not sig
    # a) remove AR13    b) remove MA14

  Arima(D1, lambda=0, order=c(12,1,14), include.drift=TRUE)
    #a) MA14 not sig

  Arima(D1, lambda=0, order=c(13,1,13), include.drift=TRUE)
    #b) AR13 not sig
```

```r
  Fit05 <- Arima(D1, lambda=0, order=c(12,1,13), include.drift=TRUE)
  Fit05
    #a) b) MA13 barely sig

  Randomness.tests(Fit05$residuals)



#- If you start removing MA(15) first

  Fit05b <- Arima(D1, lambda=0, order=c(15,1,13), include.drift=TRUE)
  Fit05b
    #AR15 not significant.

  Randomness.tests(Fit05b$residuals)


##  Depending of what you remove first you end up with either
##  ARIMA(12,1,13) with drift, or ARIMA(15,1,13) with drift.
##  (lambda is set to 0).
##
##  I will use ARIMA(12,1,13) with drift to answer questions below.
##
##  AICc of Fit05 is -146.11.  AICc of Fit01 is -10.39.  Based on AICc,
##  ARIMA(12,1,13) should have been reported by auto.arima(), but
##  was not looked at, because maximum p and q of the default setting is 5.
##
##  The model fit of both ARIMA(12,1,13) with drift and
##  ARIMA(15,1,13) with drift is adequate by residual analysis.
##
```

ARIMA(d=0) with Linear Trend analysis

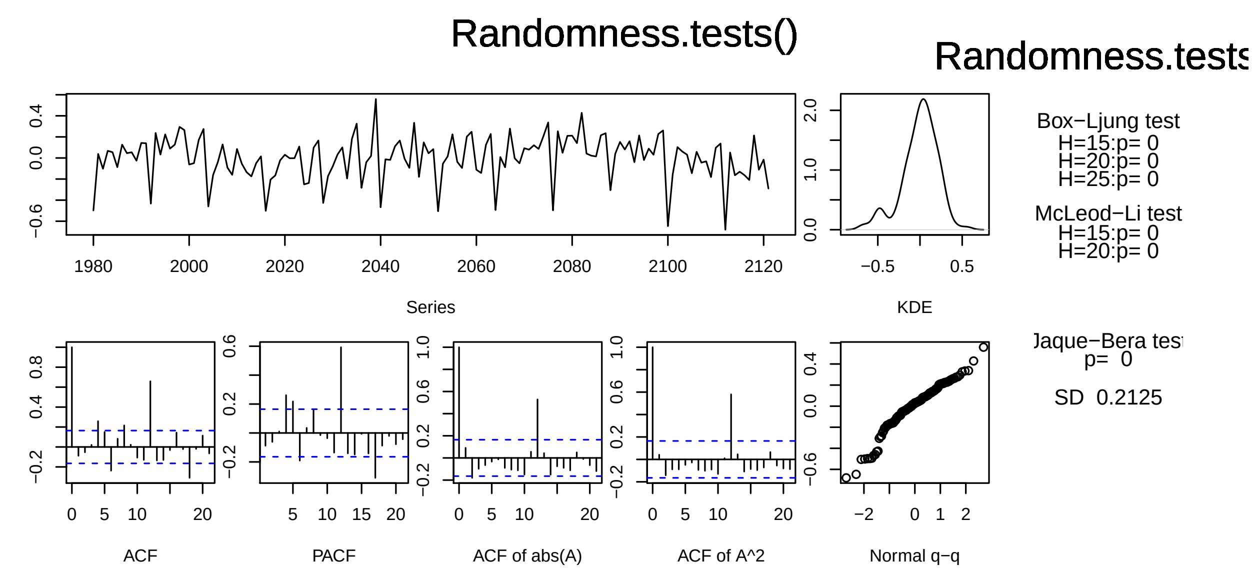
6. Another model we can fit this data is d=0 with linear trend model.

Use auto.arima() with d=0 and xreg=time(D1) option to find best ARMA(p,q) model to go on top of the linear trend. Don't forget to use the same lambda as before. What is your linear trend model? Does this final model pass the residual adequacy test?
(Model pram + Residual p-values.)

```
Fit06 <- auto.arima(D1, stepwise=FALSE, approximation=FALSE,
                                        lambda=0, xreg=time(D1))
Fit06
```

```
## Series: D1
## Regression with ARIMA(4,0,1) errors
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1  intercept    xreg
##       1.0148  -0.2605  0.1093  -0.3121  -0.6627    -6.1249  0.0065
## s.e.  0.0999   0.1211  0.1176   0.0837   0.0734     0.7148  0.0004
##
## sigma^2 estimated as 0.04715:  log likelihood=18.35
## AIC=-20.69   AICc=-19.61   BIC=2.95
```

```
Randomness.tests(Fit06$residuals)
```




```
##    B-L test H0: the series is uncorrelated
##    M-L test H0: the square of the series is uncorrelated
##    J-B test H0: the series came from Normal distribution
##    SD         : Standard Deviation of the series
```

```
##      BL15 BL20 BL25 ML15 ML20 JB    SD
## [1,]    0    0    0    0    0  0 0.212
```

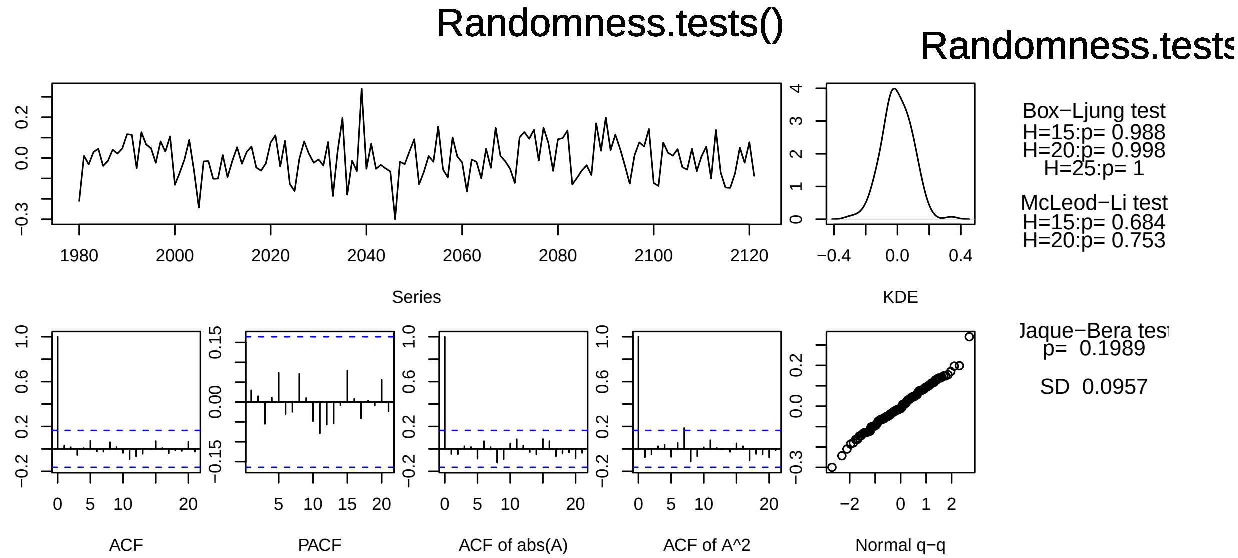
```r
## Model fit is not adequate. P-value is too low for all
## L-B test.
## There's significant autocorrelation at lag 12.
## Should try ARMA with higher p,q like we did in #5.


  Fit06b <- Arima(D1, order=c(15,0,15), xreg=time(D1), lambda=0)
  Fit06b
```

```
## Series: D1
## Regression with ARIMA(15,0,15) errors
## Box Cox transformation: lambda= 0
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##           ar1     ar2     ar3     ar4      ar5      ar6      ar7      ar8     ar9
##       -0.4765  0.7913  0.5179  0.0028  -0.0316  -0.0511  -0.0146  -0.0411  -0.041
## s.e.      NaN  0.1081  0.1210  0.0196   0.0208   0.0094   0.0223   0.0207     NaN
##          ar10    ar11    ar12    ar13     ar14     ar15     ma1      ma2      ma3
##       -0.0240  0.0006  0.9728  0.4459  -0.8184  -0.5375  0.5869  -0.6199  -0.4733
## s.e.   0.0234  0.0195  0.0152     NaN   0.1054   0.1101     NaN   0.1573   0.1992
##           ma4    ma5     ma6      ma7     ma8     ma9    ma10     ma11     ma12
##       -0.1190  0.044  0.0293  -0.1085  0.0821  0.0420  0.1038  -0.0402  -0.9426
## s.e.   0.1041  0.088  0.1449   0.1027  0.1189  0.1167  0.1104   0.1028   0.1050
##         ma13    ma14    ma15  intercept    xreg
##       -0.577  0.5308  0.4618    -6.7277  0.0068
## s.e.     NaN  0.1523  0.1204     0.4158  0.0002
##
## sigma^2 estimated as 0.01175:  log likelihood=113.56
## AIC=-161.11   AICc=-140.33   BIC=-63.57
```

```r
  Randomness.tests(Fit06b$residuals)
```




```
##   B-L test H0: the series is uncorrelated
##   M-L test H0: the square of the series is uncorrelated
##   J-B test H0: the series came from Normal distribution
##   SD         : Standard Deviation of the series

##       BL15  BL20 BL25  ML15  ML20    JB    SD
## [1,] 0.988 0.998    1 0.684 0.753 0.199 0.096
```

```
## Model fit is now adequate. AR15 and MA15 both significant.
## Slope and intercept basically unchanged from before.
```

7. Is the slope estimate you get in (6) consistent with

the drift term you had in (5)?

```
Fit05
```

```
## Series: D1
## ARIMA(12,1,13) with drift
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.129  -0.1408  -0.1249  -0.1166  -0.1477  -0.1350  -0.1231  -0.1540
## s.e.   0.113   0.1182   0.1164   0.1273   0.1094   0.1196   0.1077   0.1189
##          ar9     ar10     ar11     ar12      ma1     ma2      ma3      ma4     ma5
##      -0.1318  -0.1439  -0.1240   0.8481  -0.686  0.0612  -0.0590  -0.0427  0.1771
## s.e.  0.1150   0.1220   0.1177   0.1140   0.152  0.1148   0.1249   0.1682  0.1550
##          ma6     ma7     ma8      ma9    ma10     ma11     ma12     ma13   drift
##      -0.1860  0.0520  0.0851  -0.0852  0.1419  -0.1909  -0.6793   0.4119  0.0063
## s.e.  0.1453  0.1125  0.1637   0.1361  0.1148   0.1400   0.2134   0.1866  0.0005
```

```
##
## sigma^2 estimated as 0.01263:  log likelihood=106.74
## AIC=-159.49   AICc=-146.11   BIC=-79.87
```

```
Fit06b
```

```
## Series: D1
## Regression with ARIMA(15,0,15) errors
## Box Cox transformation: lambda= 0
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##            ar1     ar2     ar3     ar4      ar5      ar6      ar7      ar8     ar9
##        -0.4765  0.7913  0.5179  0.0028  -0.0316  -0.0511  -0.0146  -0.0411  -0.041
## s.e.       NaN  0.1081  0.1210  0.0196   0.0208   0.0094   0.0223   0.0207     NaN
##           ar10    ar11    ar12    ar13     ar14     ar15      ma1      ma2     ma3
##        -0.0240  0.0006  0.9728  0.4459  -0.8184  -0.5375   0.5869  -0.6199  -0.4733
## s.e.    0.0234  0.0195  0.0152     NaN   0.1054   0.1101      NaN   0.1573   0.1992
##            ma4     ma5     ma6      ma7      ma8      ma9     ma10     ma11     ma12
##        -0.1190  0.044  0.0293  -0.1085   0.0821   0.0420   0.1038  -0.0402  -0.9426
## s.e.    0.1041  0.088  0.1449   0.1027   0.1189   0.1167   0.1104   0.1028   0.1050
##           ma13    ma14    ma15  intercept    xreg
##         -0.577  0.5308  0.4618    -6.7277  0.0068
## s.e.       NaN  0.1523  0.1204     0.4158  0.0002
##
## sigma^2 estimated as 0.01175:  log likelihood=113.56
## AIC=-161.11   AICc=-140.33   BIC=-63.57
```

```
## drift term from Fit05 is 0.0061.
## xreg (slope) term from Fit06b is 0.0068.
## They are close and therefore consistent.
## At this point, we can not decide on which model is more
## plausible.  We will do more test on #8.
```

8. Use the following code to fit the regression line outside

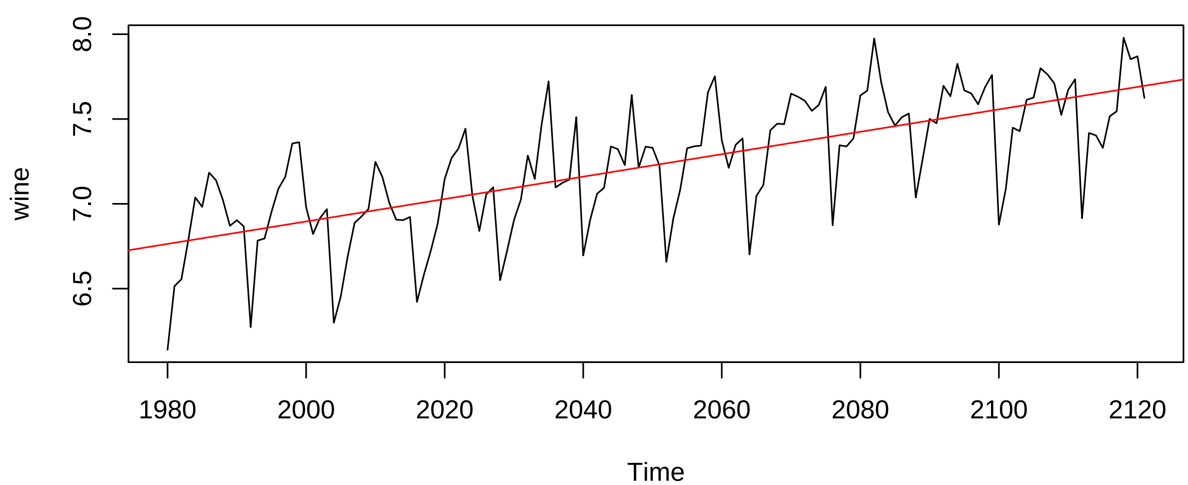
of auto.arima(), and test the regression residuals for stationarity. Is the estimate consistent with (6)? (Replace lambda with your lambda)

```
  D2 <- BoxCox(D1, lambda=0)
  Reg <- lm(D2~time(D2))
  summary(Reg)
```

```
##
## Call:
## lm(formula = D2 ~ time(D2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.72153 -0.11939  0.03217  0.17556  0.59503
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.3264336  1.1271356  -5.613 1.03e-07 ***
## time(D2)     0.0066111  0.0005496  12.029  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2684 on 140 degrees of freedom
## Multiple R-squared:  0.5083, Adjusted R-squared:  0.5048
## F-statistic: 144.7 on 1 and 140 DF,  p-value: < 2.2e-16
```

```
plot(D2)
abline(Reg, col="red")
```




```
Stationarity.tests(Reg$residuals)
```

```
## Warning in adf.test(A): p-value smaller than printed p-value

## Warning in pp.test(A): p-value smaller than printed p-value

## Warning in kpss.test(A): p-value greater than printed p-value

##        KPSS  ADF   PP
## p-val:  0.1 0.01 0.01
```

```
## Large, small, small p-values unaniously indicates that
## the Regression residuals are stationary.
## This means that model in (#5) is not a good model(necessary) for the data,
## even though ARMA residual analysis looked good on #5.
```

```r
## We should be modeling this data with
##         Yt = Linear Trend + ARMA error
## as in #6.
```

Compare the two model

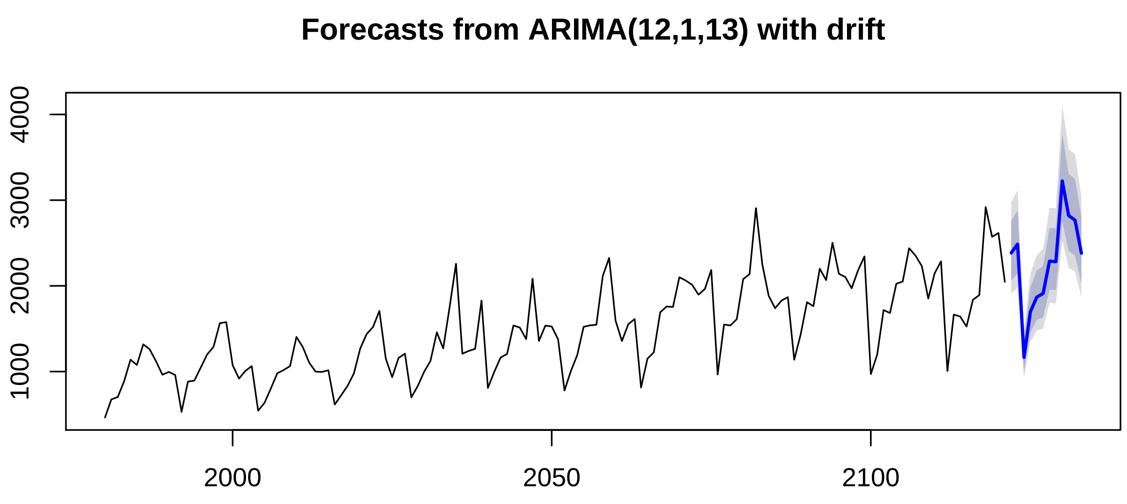
9. Perform 12-step forecast using the model from (5).

What is the 95% PI for the next observation? Include the numbers here.

```r
forecast(Fit05, 12)
```

```
##      Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## 2122       2383.929 2061.805 2756.380 1909.2958 2976.553
## 2123       2486.541 2144.839 2882.681 1983.3926 3117.328
## 2124       1165.406 1001.926 1355.560  924.8835 1468.478
## 2125       1696.449 1456.237 1976.285 1343.1679 2142.650
## 2126       1868.304 1602.821 2177.761 1477.9128 2361.817
## 2127       1911.142 1633.666 2235.748 1503.4791 2429.342
## 2128       2288.551 1956.208 2677.355 1800.2840 2909.244
## 2129       2282.224 1949.905 2671.180 1794.0473 2903.238
## 2130       3223.748 2751.203 3777.456 2529.7749 4108.093
## 2131       2819.906 2406.211 3304.728 2212.3798 3594.261
## 2132       2766.423 2356.006 3248.334 2163.9995 3536.551
## 2133       2380.877 2027.637 2795.656 1862.3804 3043.726
```

```r
plot(forecast(Fit05, 12))
```



```
##  Older version of the question wrongly said CI instead of PI.
##  We are getting Prediction Interval for next 12 observations.

##  Assuming Normality, 95% PI for next obs is
```

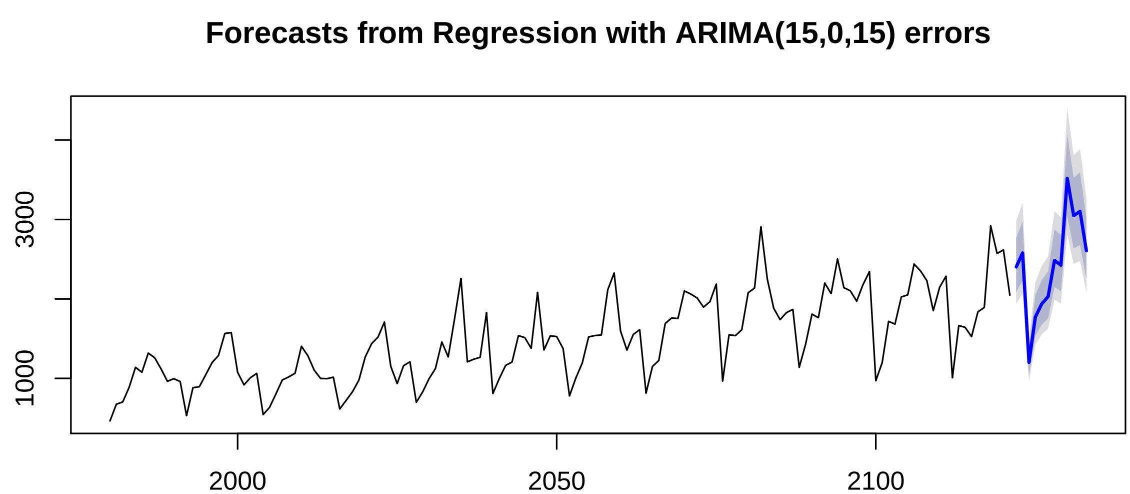
10. Perform 12-step forecast using the model from (6).

What is the 95% PI for the next observation? Include the numbers here.
(Remember that your forecast() needs xreg. See slide (6-5))

```
h=12
forecast(Fit06b, h, xreg=last(time(D1))+(1:h)/frequency(D1))
```

```
##      Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## 2122       2402.444 2085.221 2767.926 1934.6183 2983.398
## 2123       2578.940 2236.241 2974.156 2073.6653 3207.330
## 2124       1201.127 1040.651 1386.351  964.5702 1495.699
## 2125       1772.715 1535.119 2047.084 1422.5199 2209.121
## 2126       1937.720 1678.018 2237.614 1554.9418 2414.725
## 2127       2030.426 1755.982 2347.762 1626.0520 2535.361
## 2128       2485.653 2149.599 2874.244 1990.5051 3103.972
## 2129       2424.606 2096.848 2803.596 1941.6790 3027.644
## 2130       3518.213 3041.743 4069.319 2816.2208 4395.189
## 2131       3047.903 2634.624 3526.012 2439.0393 3808.760
## 2132       3102.146 2678.221 3593.172 2477.7894 3883.828
## 2133       2605.651 2247.112 3021.398 2077.7380 3267.697
```

```
plot(forecast(Fit06b, h, xreg=last(time(D1))+(1:h)/frequency(D1)))
```



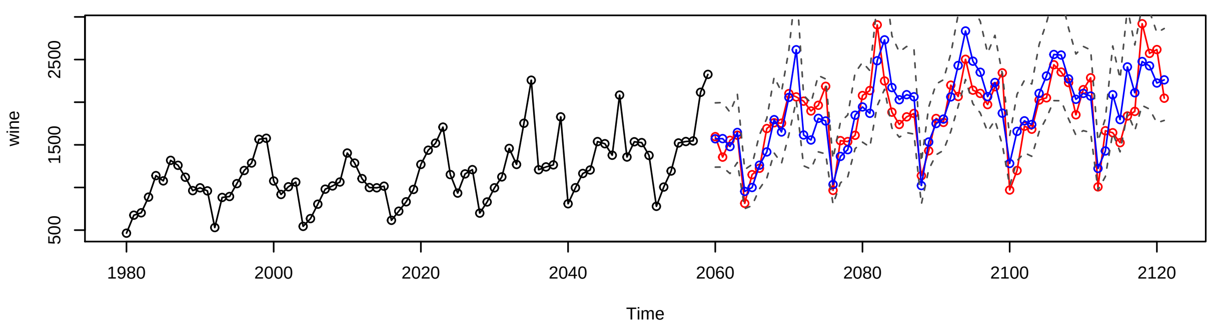
```
## Assuming Normality, 95% PI for next obs is
```

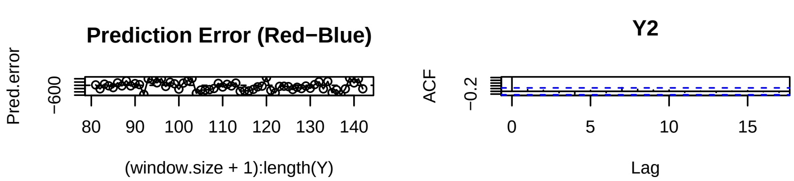
11. Perform Rolling 1-step prediction of last 42 observations

retrospectively using model from (5). Report prediction rMSE. Compare that with sigma-hat from the model.

```
#- Set options
Y <- D1                       # Original data
window.size <- 80            # Window size for estimation
Arima.order <- c(12,1,13)    # Arima(p,d,q) order
pred.plot   <- TRUE          # do you want plot at end?

#- set Arima() options:
include.mean  = FALSE
include.drift = TRUE
lambda    = 0                 # NULL=no transformaton. 0=Log
xreg      = NULL              # NULL=no xreg. TRUE=Linear Trend is present
seasonal = c(0, 0, 0)        # seasonal component

#- then use the function
Rolling1step.forecast(Y, window.size, Arima.order, pred.plot,
                        include.mean, include.drift, lambda, xreg, seasonal)
```

```
##
## Last  62  obs fit retrospectively
##     with Rolling 1-step prediction
##   Average prediction error:  -44.8632
##   root Mean Squared Error:   261.1243
```






```
##       mean pred error     rMSE
## [1,]         -44.8632 261.1243
```

```r
#Rolling1step.forecast.old(Y, window.size, Arima.order, pred.plot,
#                          include.mean, include.drift, lambda, xreg, seasonal)
```

```r
# Transforming Sigma Back

D2     <- BoxCox(D1, lambda=0)
D.base <- D2[length(D1)-(42:1)]  # last 42 obs in D1
Y  <- rnorm(1000*42, 0, 0.1120268)     # simulating normal data with same theoretical model rMSE
Y2 <- Y+rep(D.base, times=1000)                    # add
X  <- InvBoxCox(Y2, lambda = 0) #inverse transform
sd(X)
```

```
## [1] 501.1282
```

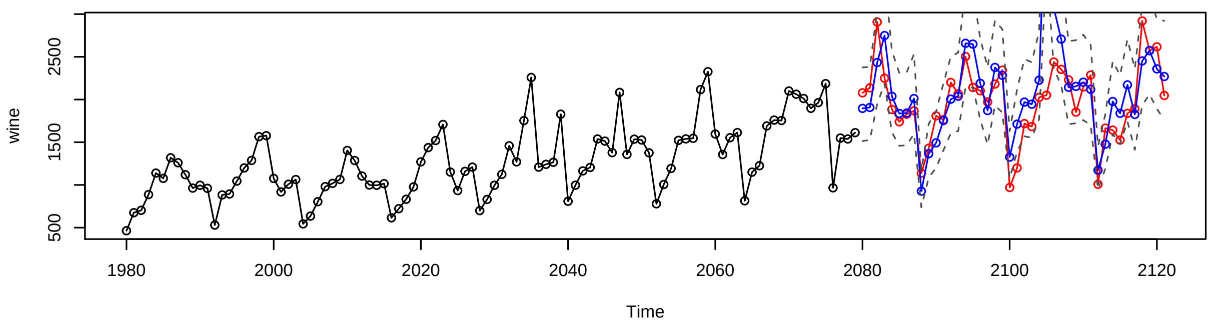
12. Perform Rolling 1-step prediction of last 42 observations

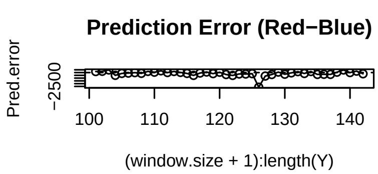
retrospectively using model from (6). Report prediction rMSE. Compare that with sigma-hat from the model.

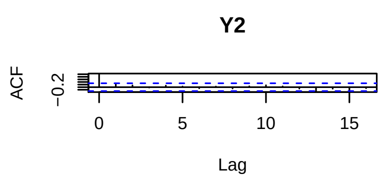
```r
#- Set options
Y <- D1                      # Original data
window.size <- 100           # Window size for estimation
Arima.order <- c(15,0,15)    # Arima(p,d,q) order
pred.plot   <- TRUE          # do you want plot at end?

#- set Arima() options:
include.mean  = TRUE         #
include.drift = FALSE        #
lambda   = 0                 # NULL=no transformaton. 0=Log
xreg     = TRUE              # NULL=no xreg. TRUE=Linear Trend is present
seasonal = c(0, 0, 0)        # seasonal component

#- then use the function
Rolling1step.forecast(Y, window.size, Arima.order, pred.plot,
                        include.mean, include.drift, lambda, xreg, seasonal)
```

```
##     i= 18   MLE-CSS failed.  Using CSS.

## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): MA part
## of model is not invertible

##     i= 26   MLE-CSS failed.  Using CSS.

## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): MA part
## of model is not invertible

##     i= 27   MLE-CSS failed.  Using CSS.

## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): MA part
## of model is not invertible
```

```
##     i= 38   MLE-CSS failed.  Using CSS.

## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): MA part
## of model is not invertible

##     i= 39   MLE-CSS failed.  Using CSS.

## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): MA part
## of model is not invertible

##     i= 42   MLE-CSS failed.  Using CSS.

## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): MA part
## of model is not invertible

##
## Last  42  obs fit retrospectively
##     with Rolling 1-step prediction
##   Average prediction error:  -135.2106
##   root Mean Squared Error:   488.934
```








```
##      mean pred error    rMSE
## [1,]       -135.2106 488.934
```

13. Which model do you like better and why? Model from (5) or (6)?

Write down your mathematical model, and list estimates for all parameters. You can type using following notation. (you may not need to use some of them)

```r
## Both models were fitting and adequate,
## but in #8, regression residuals were stationary.
## This indicates Linear Trend + ARMA has no problem fitting/explaining
## the data, and therefore, no need for ARIMA with d=1.
## Model from #6 is my best model.

Fit06b
```

```
## Series: D1
## Regression with ARIMA(15,0,15) errors
## Box Cox transformation: lambda= 0
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##            ar1     ar2     ar3     ar4      ar5      ar6      ar7
##        -0.4765  0.7913  0.5179  0.0028  -0.0316  -0.0511  -0.0146
## s.e.       NaN  0.1081  0.1210  0.0196   0.0208   0.0094   0.0223
##            ar8     ar9     ar10    ar11    ar12    ar13     ar14
##        -0.0411  -0.041  -0.0240  0.0006  0.9728  0.4459  -0.8184
## s.e.    0.0207     NaN   0.0234  0.0195  0.0152     NaN   0.1054
##           ar15     ma1      ma2      ma3      ma4    ma5     ma6
##        -0.5375  0.5869  -0.6199  -0.4733  -0.1190  0.044  0.0293
## s.e.    0.1101     NaN   0.1573   0.1992   0.1041  0.088  0.1449
##            ma7     ma8     ma9    ma10     ma11     ma12     ma13
##        -0.1085  0.0821  0.0420  0.1038  -0.0402  -0.9426  -0.577
## s.e.    0.1027  0.1189  0.1167  0.1104   0.1028   0.1050     NaN
##          ma14    ma15  intercept    xreg
##        0.5308  0.4618    -6.7277  0.0068
## s.e.   0.1523  0.1204     0.4158  0.0002
##
## sigma^2 estimated as 0.01175:  log likelihood=113.56
## AIC=-161.11   AICc=-140.33   BIC=-63.57
```

- Mathematical expression:

$$Y_t = \text{observation}$$
$$Y_t = a + bt + X_t$$
$$X_t \text{ is ARIMA(15,0,15)}$$

$$X_t = phi_1 X_{t-1} + phi_1 X_{t-1} + phi_2 X_{t-2} + \cdots + phi_{15} X_{t-15} + e_t + theta_1 e_{t-1} + cdots + theta_{15} e_{t-15}$$

$$e_t \sim WN(0, \sigma^2)$$

TS Class Webpage – R resource page