ECON 430 Homework 3

Gefei Zhao

2020/11/24

Contents

1. Introduction

The dataset from FRED provides data of alcohol sales in U.S. from January 1992 to September 2020.

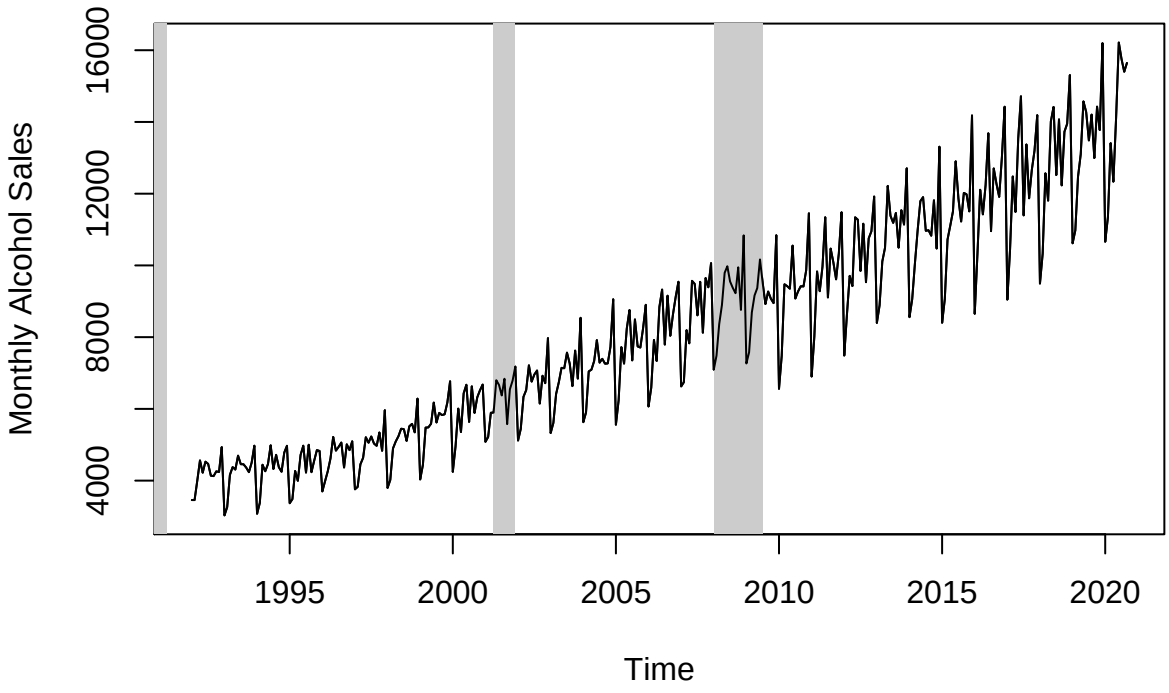
Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
alcohol	345	8,221.270	3,159.680	3,031	5,353	10,616	16,215

2. Results

2.1 Modeling and Forecasting Trend

(a) Time Series Plot



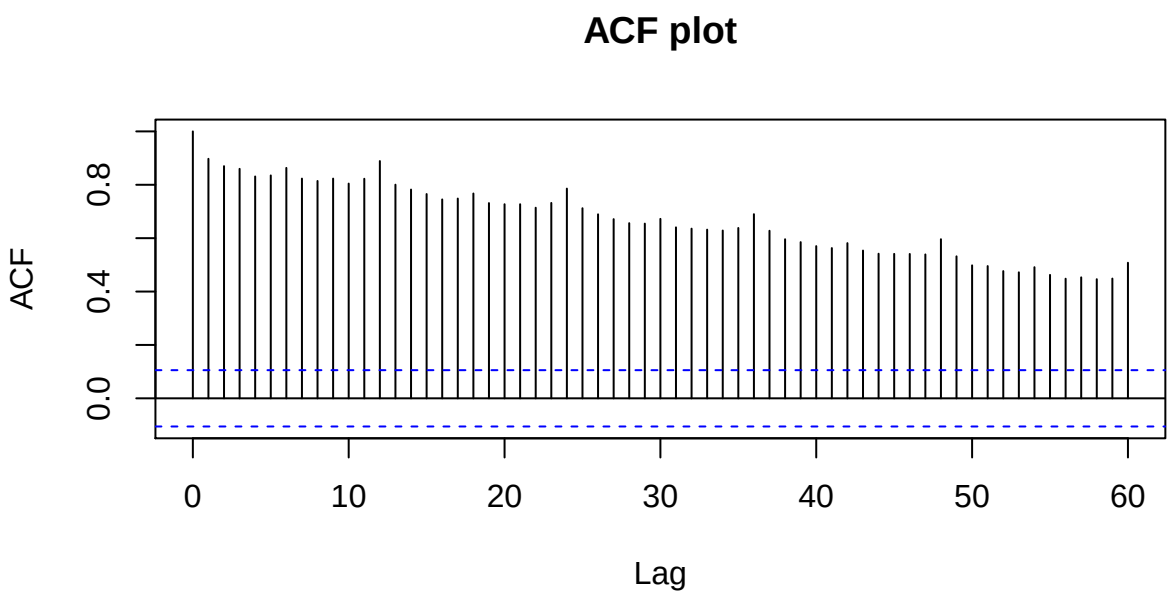
The times series have significant trend and seasonality.

(b) Covariance Stationary

The time series is not covariance stationary. Its mean and variance are not constant but increasing over time. It has significant trend and seasonality and not a mean-reverting pattern.

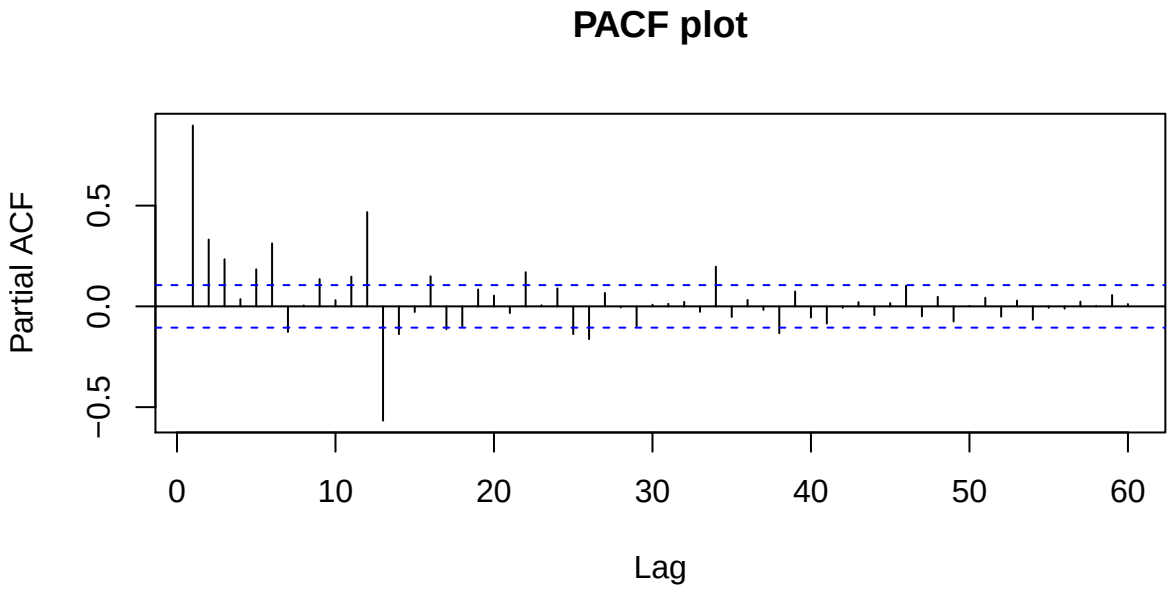
(c) ACF and PACF

- ACF Plot



Autocorrelations are large, so there is high dependence on how alcohol sales have changed overtime. Besides, there is an sudden increase for every 12 lag operator, indicating the time series exist highly seasonality.

- PACF Plot

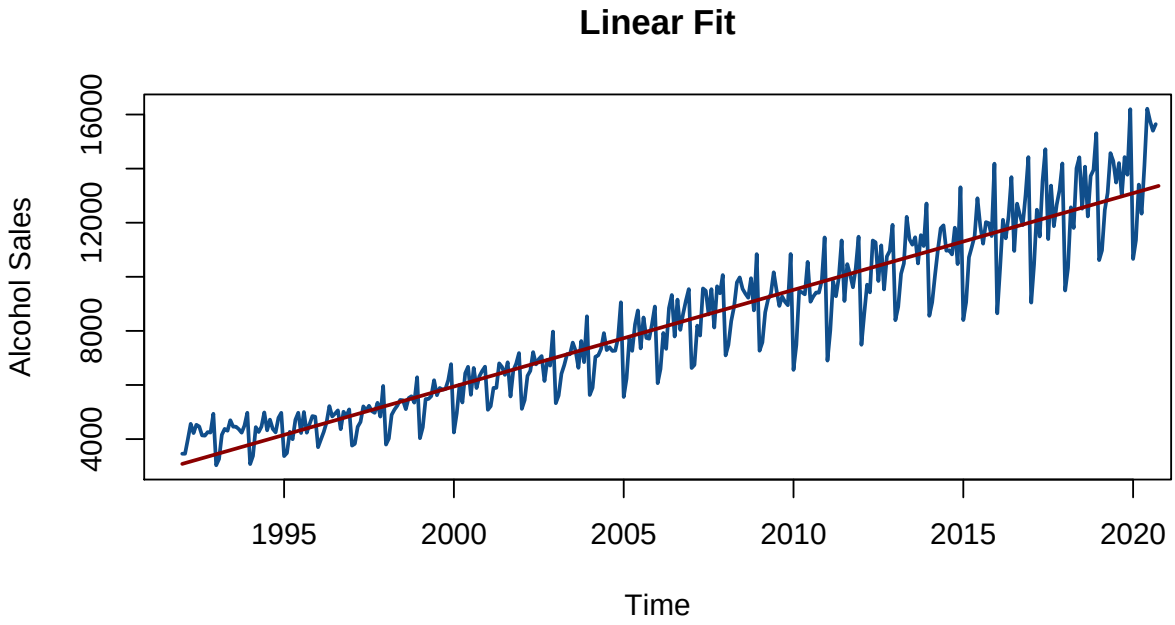


The partial autocorraltion is significant large for past 12 observations and declining to significant zero for observations lagged more than 38. It suggests that the partial autocorrelation is high with the data in last 1 year oberservation. It has less partial corrlation with data in last 2-3 years and no significant partial autocorraltion with data over 3 years.

(d) Fitting Linear and Nonlinear Models

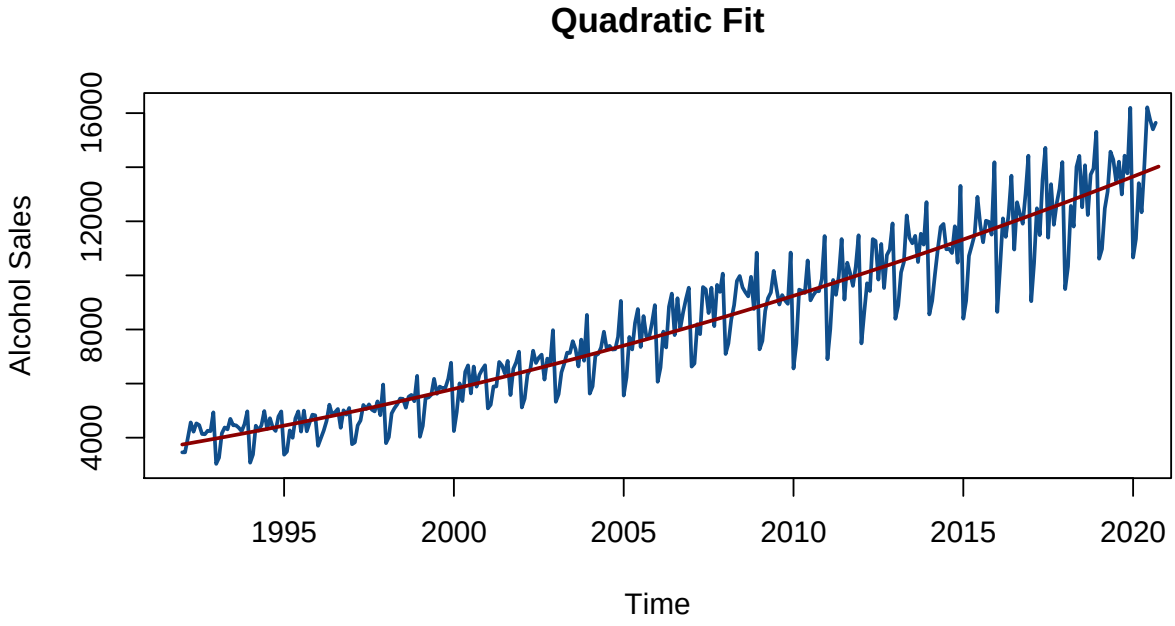
Linear Fit

$$Model: y_t = \beta_0 + \beta_1 TIME$$



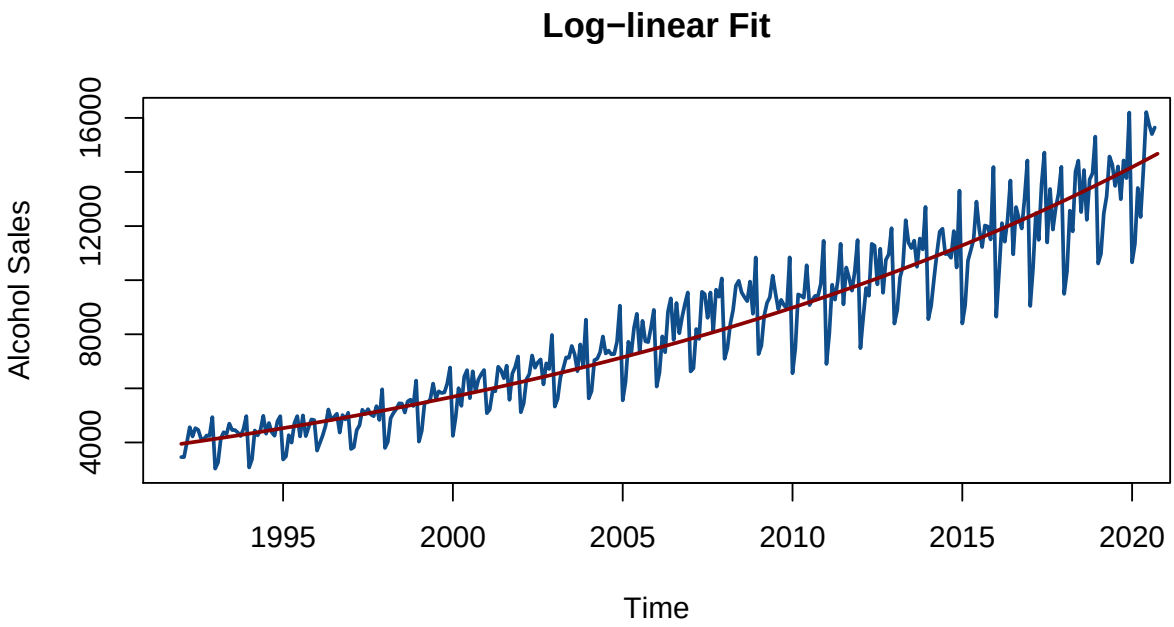
Quadratic Fit

$$Model: y_t = \beta_0 + \beta_1 TIME + \beta_2 TIME^2$$



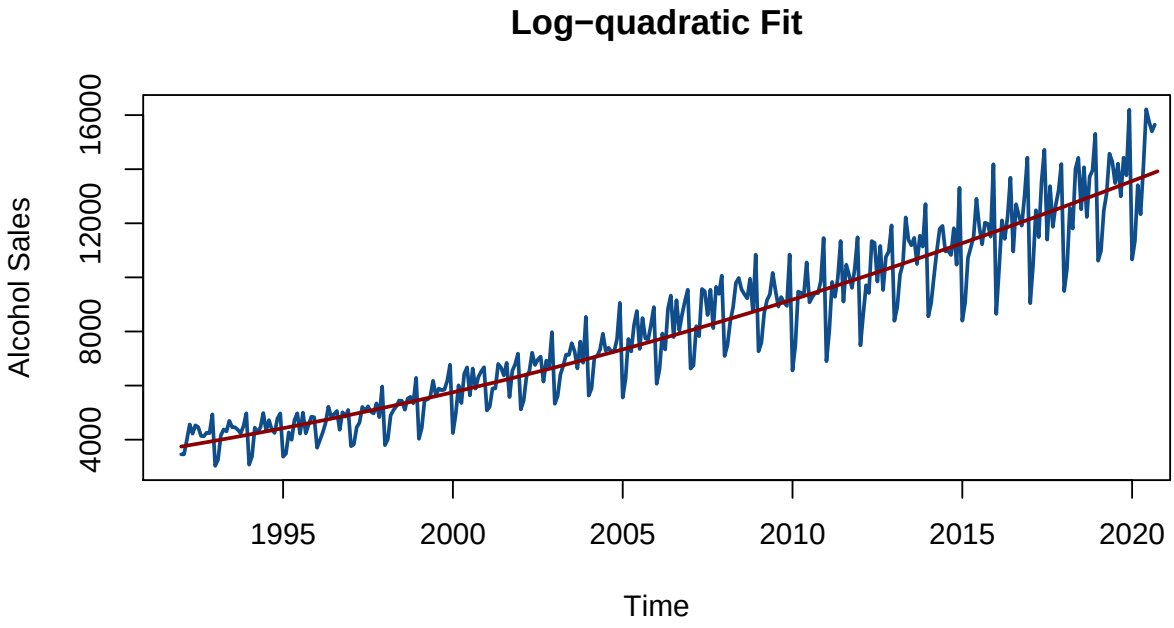
Log-linear Fit

$$Model : log(y_t) = \beta_0 + \beta_1 TIME$$



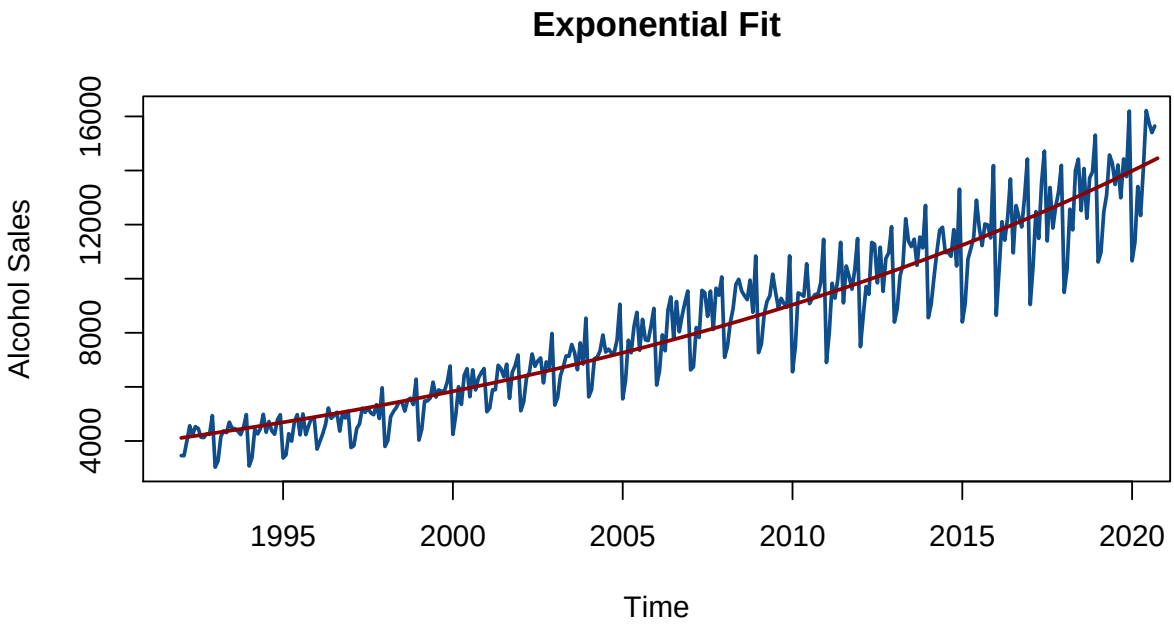
Log-quadratic Fit

$$Model : log(y_t) = \beta_0 + \beta_1 TIME + \beta_2 TIME^2$$



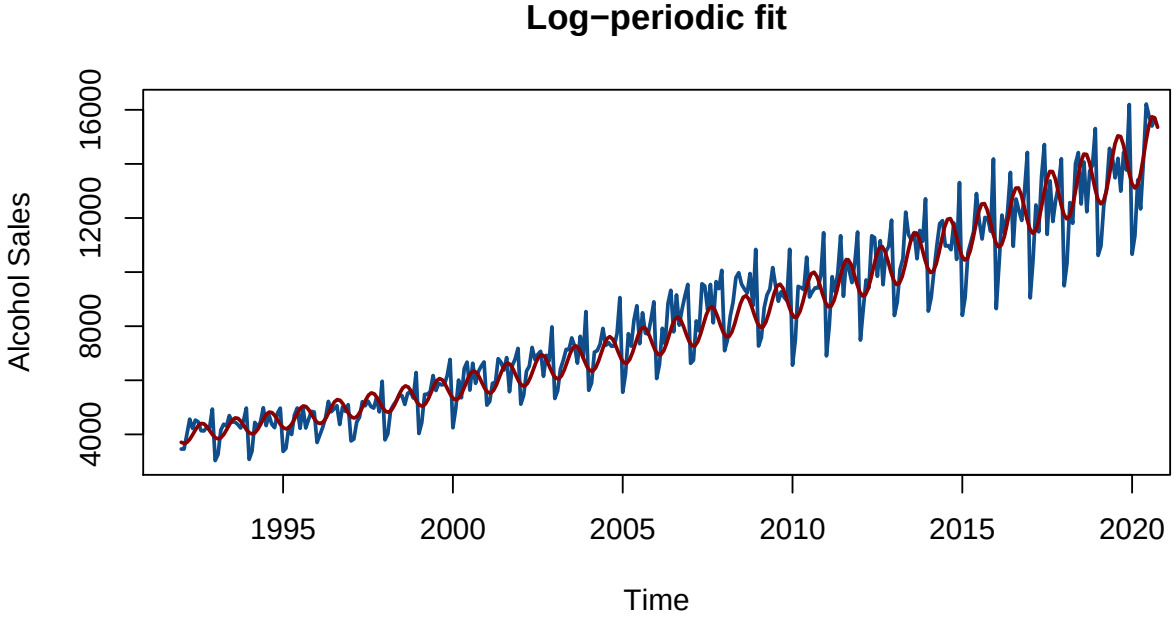
Exponential Fit

$$Model : y_t = e^{(\beta_0 + \beta_1 TIME)}$$



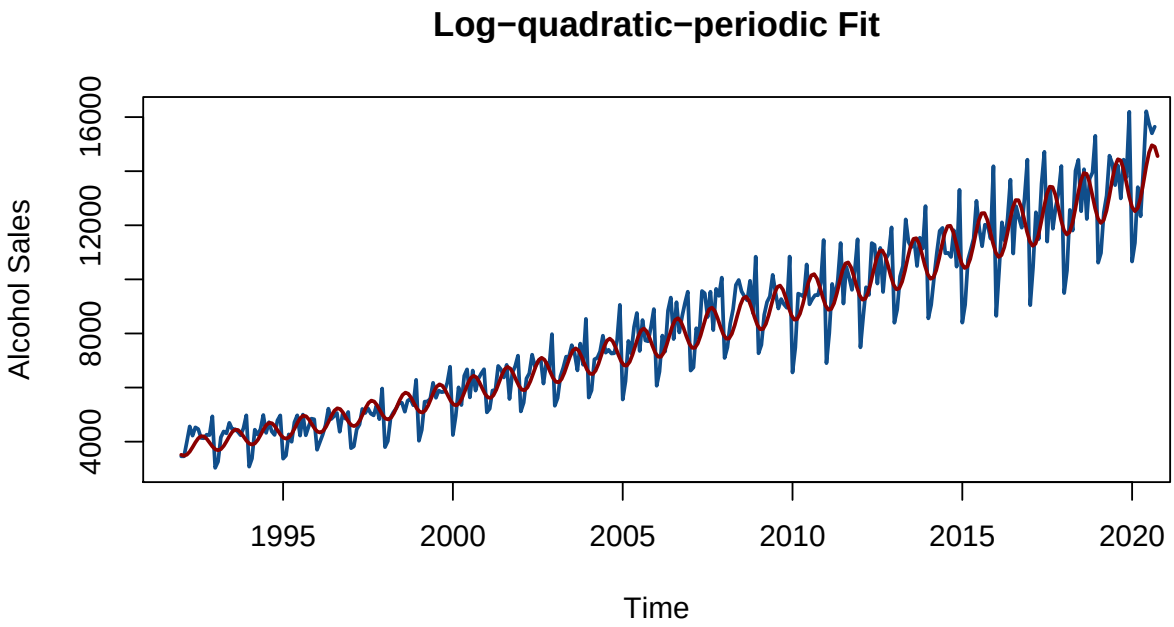
Log-periodic fit

$$Model : log(y_t) = \beta_0 + \beta_1 TIME + \beta_3 sin(2\pi TIME) + \beta_4 cos(2\pi TIME)$$

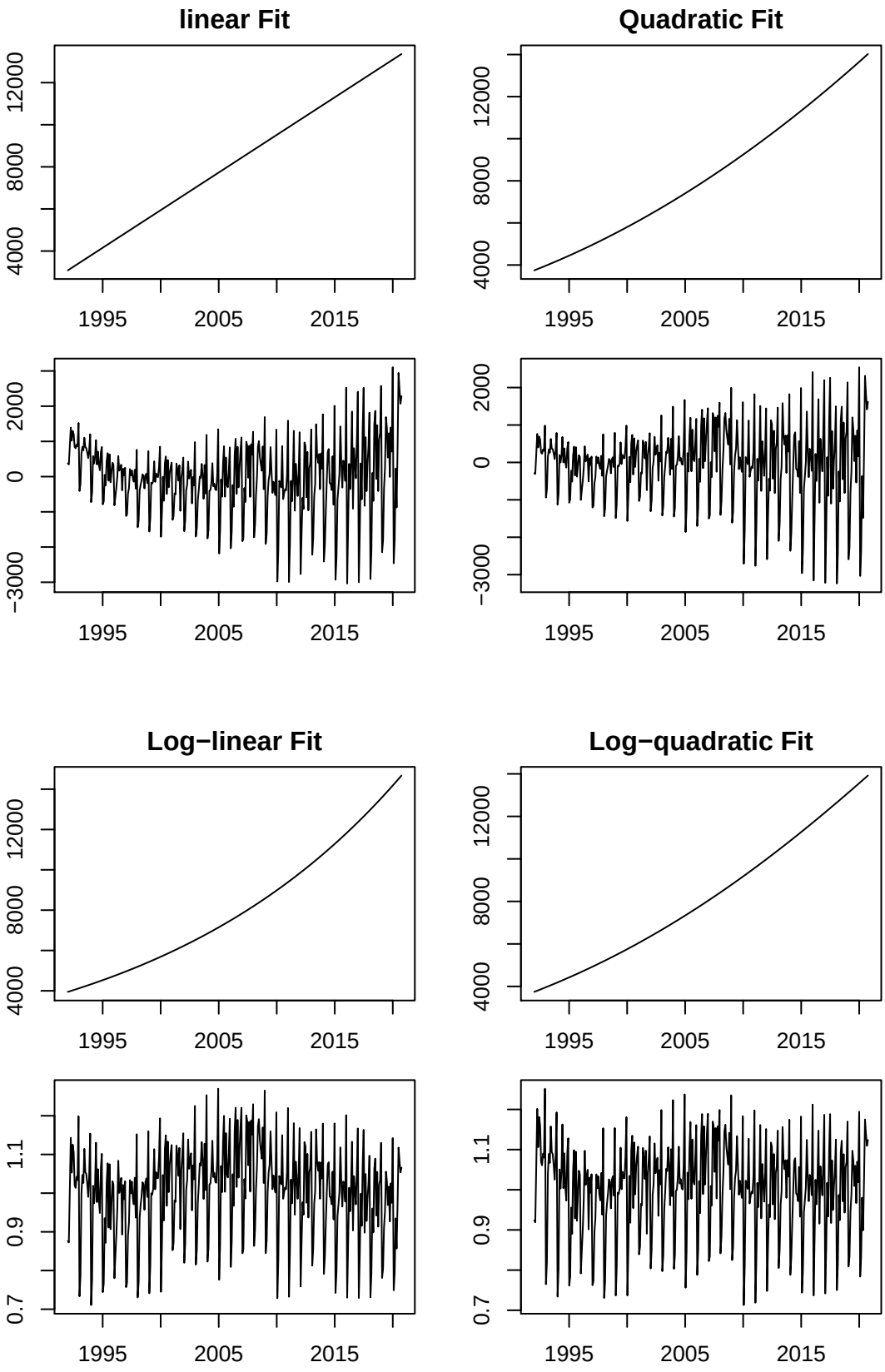


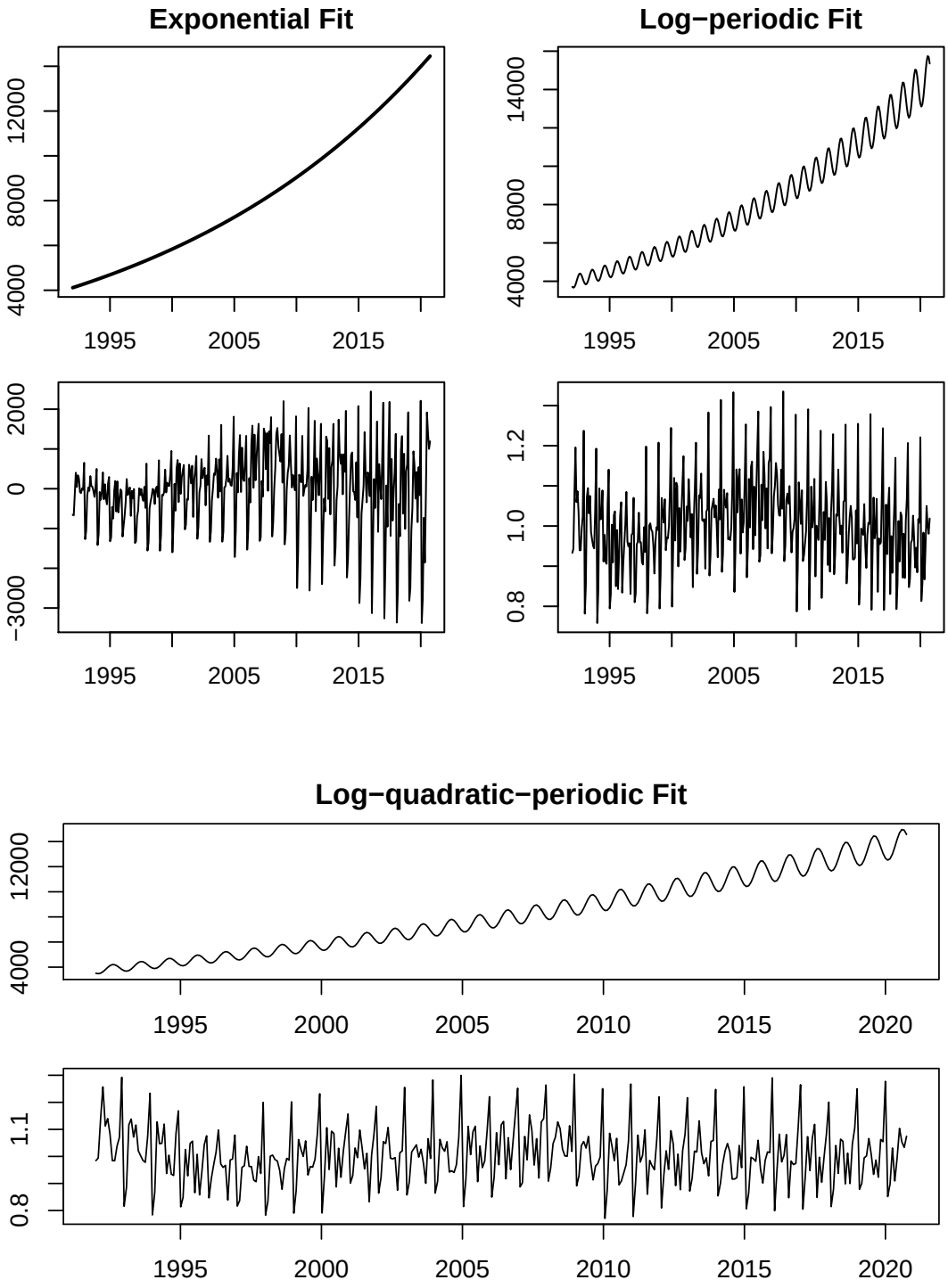
Log-quadratic-periodic Fit

$$Model : log(y_t) = \beta_0 + \beta_1 TIME + \beta_1 TIME^2 + \beta_4 sin(2\pi TIME) + \beta_5 cos(2\pi TIME)$$



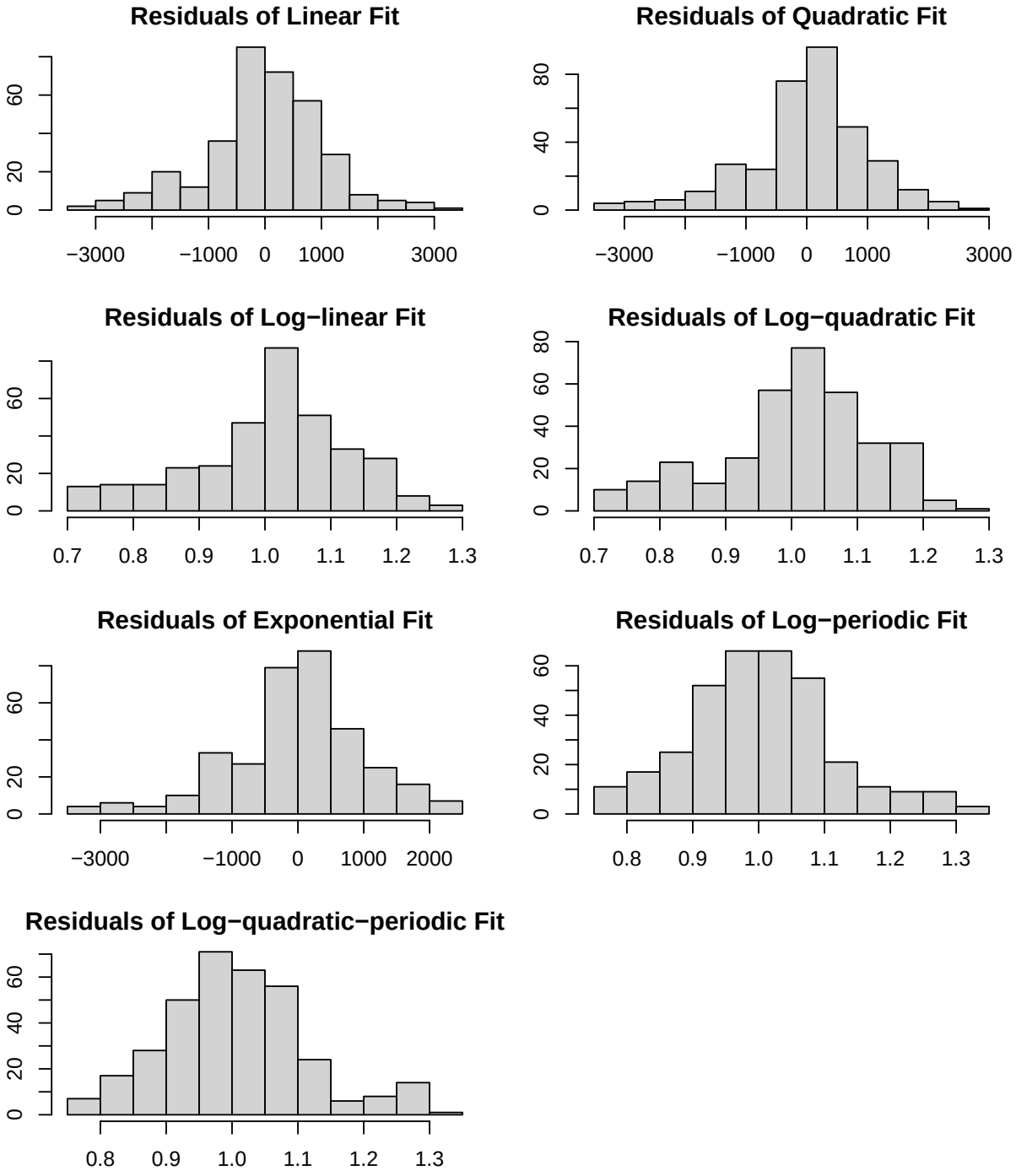
(e) Residuals vs. Fitted Values





The residual of linear, quadratic and exponential fit have non-constant mean and increasing variance as years approach 2020. Compared with above models, log-linear, log-quadratic and log-period fit make the spread of residual more even with time but still have fluctuating mean around zero. These models not good enougth tp fit the trend. The log-quadratic-period fit has a almost constant mean around zero and a even spread. It is perfect to measure the trend in our time series.

(f) Histograms of Residuals



The histograms of residuals suggest the same conclusion with the residual plot that the log-quadratic-periodic model fits the trend best. Compared to distribution of residuals of other model, log-quadratic-period and log-period fits have more normal distributed residuals.

(g) Diagnostic Statistics

Table 2:

	Dependent variable:		
	ts		log(ts)
	Linear	Quadratic	Log-linear
t	357.474***	−19,048.380***	0.046***
	(6.808)	(3,521.995)	(0.001)
I(t^2)		4.836***	
		(0.878)	
Constant	−709,005.400***	18,758,366.000***	−82.690***
	(13,659.700)	(3,533,179.000)	(1.600)
Observations	345	345	345
R^2	0.889	0.898	0.905
Adjusted R^2	0.889	0.898	0.905
Residual Std. Error	1,052.546 (df = 343)	1,010.199 (df = 342)	0.123 (df = 343)
F Statistic	2,757.007*** (df = 1; 343)	1,511.676*** (df = 2; 342)	3,279.262*** (df = 1; 343)

Table 3:

	Dependent variable:		
		log(ts)	
	Log-quadratic	Log-periodic	Log-quadratic-periodic
t	1.586***	0.046***	1.596***
	(0.422)	(0.001)	(0.373)
I(t^2)	−0.0004***		−0.0004***
	(0.0001)		(0.0001)
sin_t		−0.048***	−0.047***
		(0.008)	(0.008)
cos_t		−0.065***	−0.065***
		(0.008)	(0.008)
Constant	−1,628.100***	−82.366***	−1,637.562***
	(423.689)	(1.421)	(374.041)
Observations	345	345	345
R^2	0.909	0.926	0.929
Adjusted R^2	0.908	0.925	0.929
Residual Std. Error	0.121 (df = 342)	0.109 (df = 341)	0.107 (df = 340)
F Statistic	1,705.102*** (df = 2; 342)	1,418.403*** (df = 3; 341)	1,118.936*** (df = 4; 340)
Note:			*p<0.1; **p<0.05; ***p<0.01

For all above linear fits, the estimates of coefficients are statistical significant at 99% level of confidence. As for linear fits without periodic terms, the log-quadratic model performs good than other model because of larger adjusted R^2. From the aspect of economic meaning of the model, we know that taking log on alcohol sales would make the regression more stationary. And there exist quadratic relationship that the sales of alcohol increases at an decreasing rate with years.

After adding the periodic terms into log-quadratic model, R^2 fo the models get larger so that we can fit and predict better.

Let look at the performance of non-linear model.

```
##
## Formula: ts ~ exp(a + b * t)
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## a -7.873e+01  1.765e+00  -44.62   <2e-16 ***
## b  4.370e-02  8.771e-04   49.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1026 on 343 degrees of freedom
##
## Number of iterations to convergence: 21
## Achieved convergence tolerance: 1.49e-08
```

The estimates of coefficients are statistical significant but the RSE is large. Therefore, exponential model not fit the trend well.

We can also look at the MSE of the models.

```
##                            MSE
## linear_fit        1.107853e+06
## quad_fit          1.020502e+06
## log_fit           1.520140e-02
## logQuad_fit       1.467496e-02
## exp_fit           1.045775e+06
## logPeriod_fit     1.198019e-02
## logQuadPeriod_fit 1.143405e-02
```

The result agrees with our analysis above. The log-quadratic-period moel has the smallest MSE.

(h) Trend Model Selection

```
##                          AIC       BIC
## linear_fit        5784.7490 5796.2796
## quad_fit          5757.4074 5772.7816
## log_fit           -461.2351 -449.7044
## logQuad_fit       -472.4018 -457.0277
## exp_fit           5766.8605 5778.3911
## logPeriod_fit     -541.4086 -522.1908
## logQuadPeriod_fit -556.5191 -533.4578
```

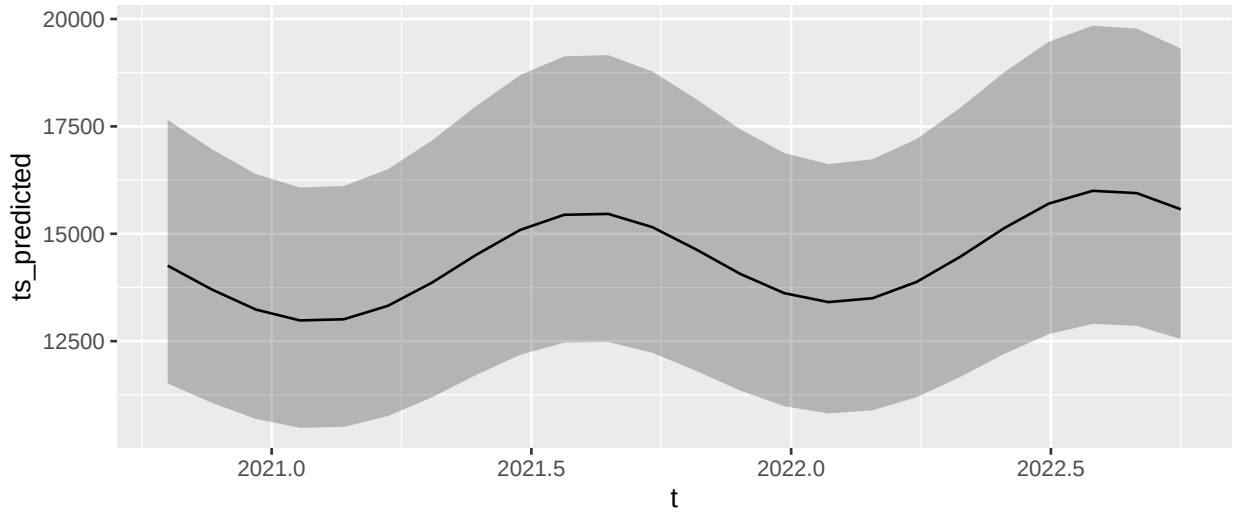
Both AIC and BIC select the log-quadratic-period model as the best-fitted model of the trend.

(i) Forecast

We can forecast 24-steps ahead.

```
##    ts_predicted     pi_h     pi_l
## 1      14256.69 17653.05 11513.77
## 2      13703.34 16969.37 11065.90
## 3      13236.08 16391.98 10687.78
## 4      12982.05 16078.36 10482.01
## 5      13009.45 16113.10 10503.61
## 6      13322.97 16502.15 10756.26
## 7      13862.81 17171.58 11191.60
## 8      14506.94 17970.36 11711.02
## 9      15088.80 18692.30 12179.99
## 10     15441.91 19131.23 12464.04
## 11     15462.33 19158.37 12479.33
## 12     15155.21 18779.90 12230.12
## 13     14632.57 18134.29 11807.02
## 14     14064.26 17431.80 11347.28
## 15     13616.37 16878.16 10984.94
## 16     13408.81 16622.09 10816.71
## 17     13498.75 16734.62 10888.59
## 18     13877.36 17204.98 11193.34
## 19     14469.07 17939.65 11669.90
## 20     15135.90 18767.71 12206.89
## 21     15700.93 19469.94 12661.52
## 22     15999.94 19842.76 12901.33
## 23     15946.08 19778.36 12856.34
## 24     15569.47 19313.82 12551.03
```

Intuitively, the plot of predicted values and respective prediction interval is shown below.



2.2 Modeling and Forcasting Seasonality

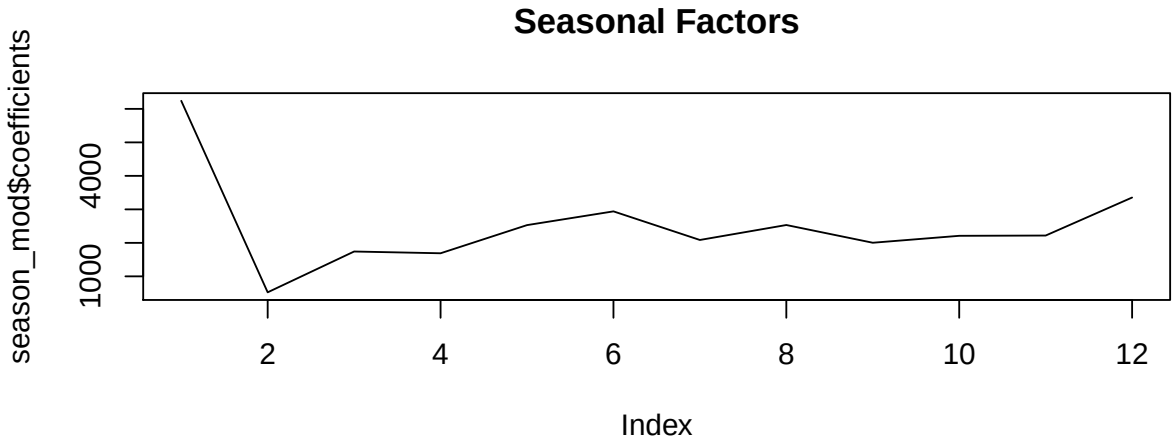
(a) Seasonality Test

Table 4:

	Dependent variable:
	ts
seasonFeb	523.483
	(808.273)
seasonMar	1,741.931**
	(808.273)
seasonApr	1,688.966**
	(808.273)
seasonMay	2,529.897***
	(808.273)
seasonJun	2,939.828***
	(808.273)
seasonJul	2,086.448**
	(808.273)
seasonAug	2,532.483***
	(808.273)
seasonSep	2,003.000**
	(808.273)
seasonOct	2,210.389***
	(815.457)
seasonNov	2,218.353***
	(815.457)
seasonDec	3,353.175***
	(815.457)
Constant	6,240.897***
	(571.535)
Observations	345
R^2	0.081
Adjusted R^2	0.051
Residual Std. Error	3,077.810 (df = 333)
F Statistic	2.686*** (df = 11; 333)
Note:	*p<0.1; **p<0.05; ***p<0.01

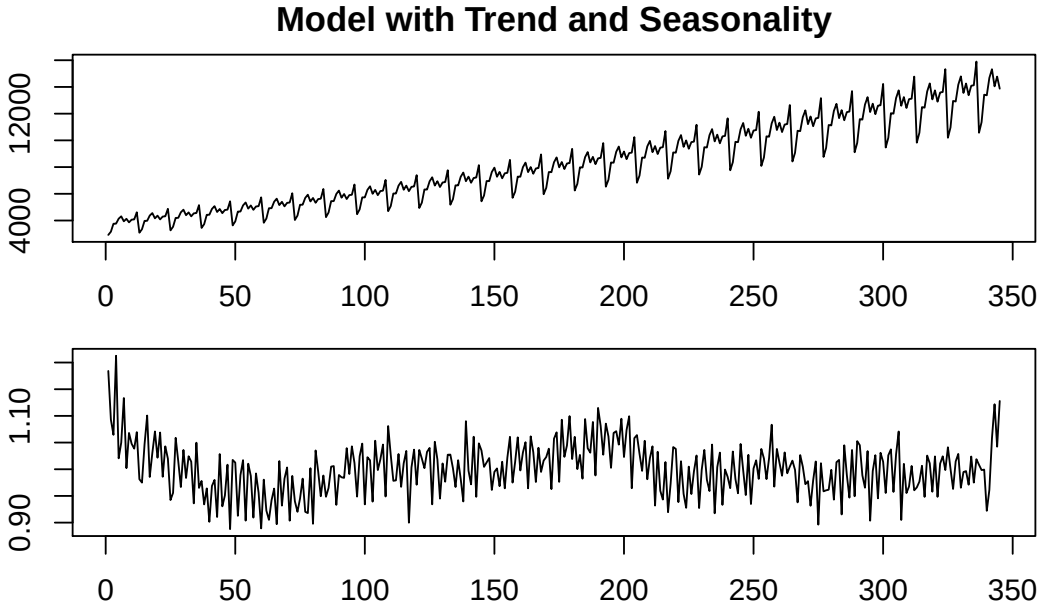
Except for February, the seasonality is statistically significant, indicating our model has seasonality.

(b) Seasonal Factors



The seasonal factors suggest that alcohol sell bad in February then raise to two peak in June and August. It might because people prefer to drink in hot summer instead of cold winter time. And alcohol sales are also good in December because people would like to drink on Christmas and New year.

(c) Model with Trend and Seasonality



The residuals of full model have a non-constant mean and even more volatile than the trend model in part 1, suggesting that there are other dynamic in the times series we do not capture.

(d) Results of Full Model

Table 5:

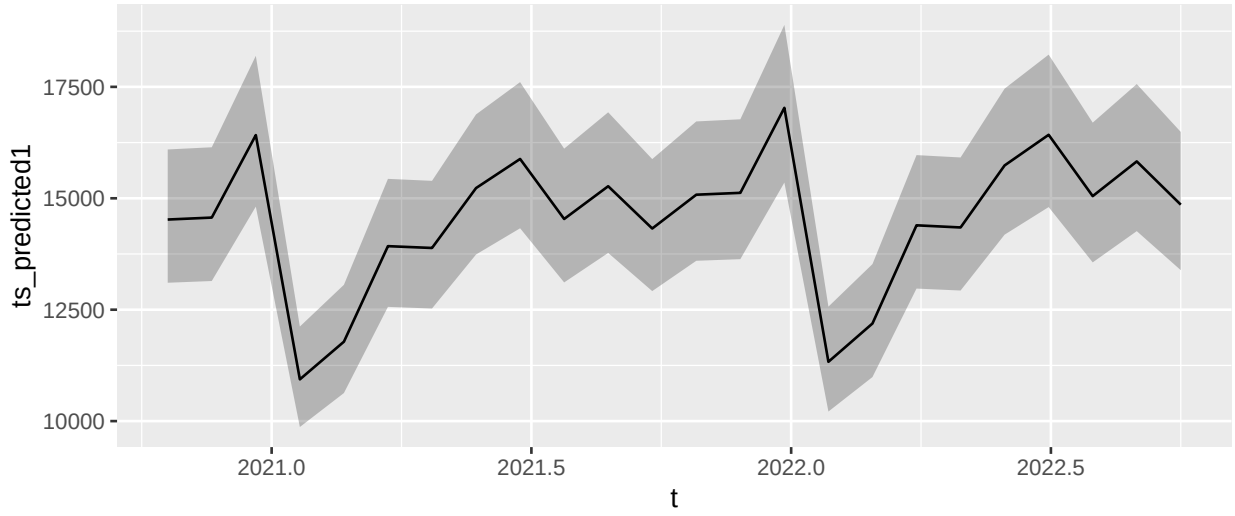
	Dependent variable:
	log(ts)
t	1.490***
	(0.177)
I(t^2)	−0.0004***
	(0.00004)
I(sin(2 ∗pi ∗t))	0.010
	(0.026)
I(cos(2 ∗pi ∗t))	0.019
	(0.026)
seasonFeb	0.073***
	(0.019)
seasonMar	0.244***
	(0.029)
seasonApr	0.248***
	(0.038)
seasonMay	0.349***
	(0.046)
seasonJun	0.397***
	(0.051)
seasonJul	0.309***
	(0.053)
seasonAug	0.354***
	(0.051)
seasonSep	0.279***
	(0.046)
seasonOct	0.317***
	(0.038)
seasonNov	0.306***
	(0.029)
seasonDec	0.414***
	(0.019)
Constant	−1,532.073***
	(177.764)
Observations	345
R^2	0.985
Adjusted R^2	0.984
Residual Std. Error	0.051 (df = 329)
F Statistic	1,403.919*** (df = 15; 329)

The regession result told us why the model performs not satisfying. It is because when we adding seasonal dummies into model, the period terms are unsignificant and the estimate of quadratic term is really small. So we fail to capture the whole dynamic of trend in the data. However, the MSE equals to 0.00257452 which is small and R^2 equals to 0.984 which is large, indicating our model fits good.

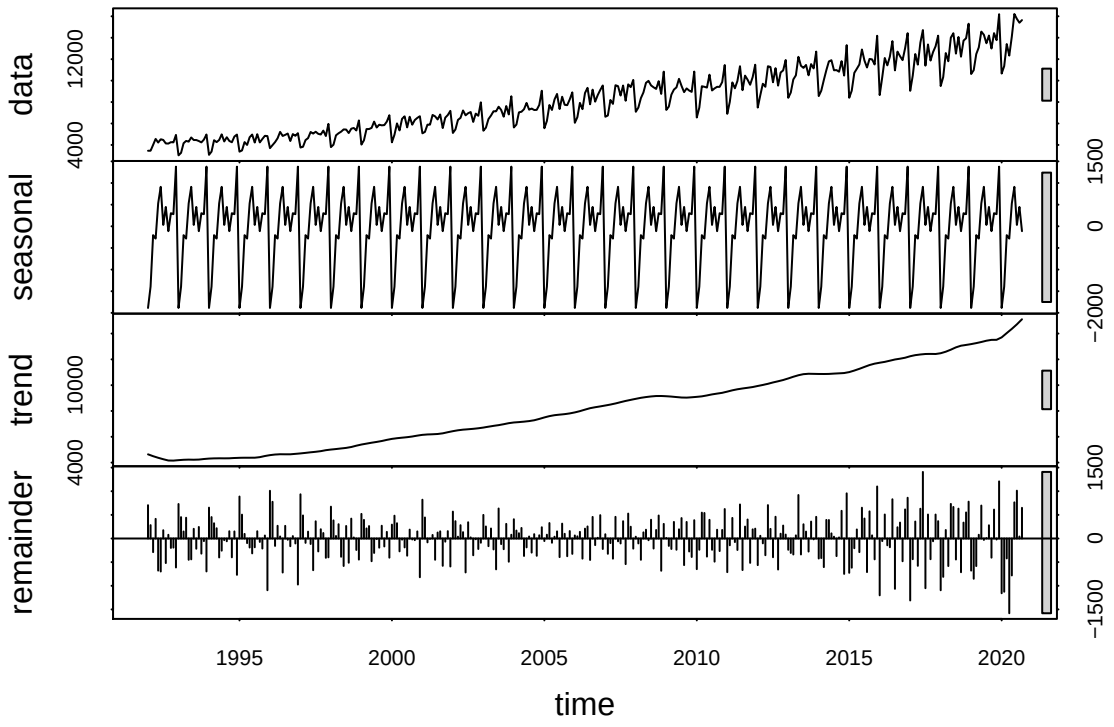
(e) Forcast

```
##    ts_predicted1    pi_h1     pi_l1
## 1       14523.17 16095.50 13104.437
## 2       14566.96 16144.79 13143.322
## 3       16417.94 18197.17 14812.674
## 4       10937.20 12122.31  9867.955
## 5       11782.08 13059.30 10629.769
## 6       13925.63 15435.82 12563.192
## 7       13885.24 15391.58 12526.332
## 8       15229.37 16882.04 13738.489
## 9       15882.43 17606.53 14327.167
## 10      14535.71 16114.23 13111.816
## 11      15270.23 16929.41 13773.652
## 12      14322.21 15879.54 12917.607
## 13      15080.38 16724.76 13597.683
## 14      15122.21 16772.82 13634.036
## 15      17029.50 18890.04 15352.217
## 16      11330.81 12569.13 10214.490
## 17      12190.50 13523.65 10988.781
## 18      14393.93 15968.89 12974.301
## 19      14345.51 15915.88 12930.082
## 20      15737.07 17460.39 14183.837
## 21      16424.43 18223.65 14802.850
## 22      15048.52 16697.84 13562.107
## 23      15826.99 17563.10 14262.493
## 24      14856.77 16488.51 13386.515
```

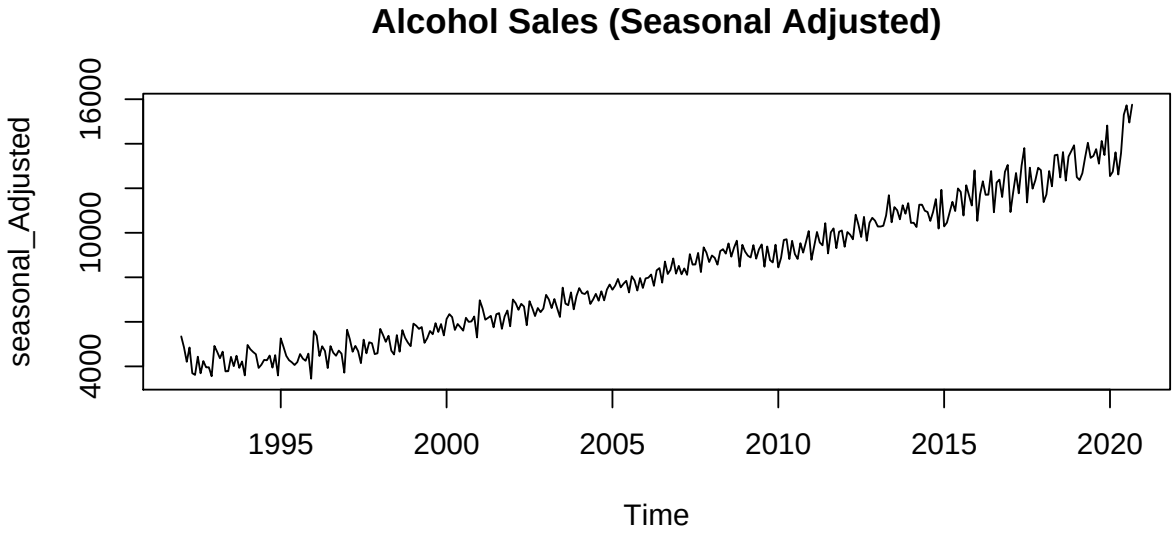
Intuitively, the plot of predicted values and respective prediction interval is shown below.



(f) De-seasonality



After decomposing the times series, we can see the seasonal fluctuation do no vary much with time, we can choose addictive adjustment to remove seasonality.



After the adjustment of seasonality, the trend is a linear model adding cycles. Obviously, the periodic terms in the former model are not appropriate because their effects are overlapped with the seasonal dummies.

3. Conclusions and Future Work

Based on the full model, we find that the times series can be decomposed to a linear trend and additive seasonality and cycles. The ACF plot suggests the time series have strong autocorrelation so that we can further remove the seasonality and trend, then use AR model to fit and forecast the time series.

References

- Data is from FRED
- Professor Rojas's Class Slides and Codes
- TA Onyambu's Code
- Forcast package
- Debugging for the Function Forecast

R Source Code

```
# ---------------[0] PREPARATION ---------------
# fix default of language to English
Sys.setenv(LANGUAGE = "en")

# load libraries
library(tseries)
library(stargazer)
library(forecast)

# set working document
setwd('C:/Users/Gefei Zhao/Desktop/UCLA/430/Homework/Homework 3-20201117')

# import training data
df <- read.table("S4248SM144NCEN.csv", sep = ",", head = T)

# rename the variable
names(df) <- c("date", "alcohol")


# ---------------[1] SUMMARY OF TIME SERIES ---------------
# summary statistics
stargazer(df)

# convert data to time series formate
ts <- ts(df$alcohol, start = c(1992, 1), frequency = 12)
t <- seq(1992, 2020.75, length = length(ts))
alco_df <- cbind.data.frame(t, ts)

plot(ts, xlab = "Time", ylab = "Monthly Alcohol Sales")
tis::nberShade() # add recession bands
lines(ts)


# ACF
```

```r
acf(df$alcohol, lag = 60, main = "ACF plot")
## Notice: Do not use the data of time series format to do the ACF and PACF plot,
## the x axis which is lag operater would be non-integer

# PACF
pacf(df$alcohol, lag = 60, main = "PACF plot")


# ---------------[2] BUILT MODELS TO FIT TREND ---------------
# linear
linear_fit <- lm(ts ~ t)

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Linear Fit")
lines(t, linear_fit$fit, col = "darkred", lwd = 2, type = "l")

# quadratic
quad_fit <- lm(ts ~ t + I(t^2))

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Quadratic Fit")
lines(t, quad_fit$fit, col = "darkred", lwd = 2, type = "l")

# log-linear
log_fit <- lm(log(ts) ~ t)

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Log-linear Fit")
lines(t, exp(log_fit$fit), col = "darkred", lwd = 2, type = "l")

# log-quadratic
logQuad_fit <- lm(log(ts) ~ t + I(t^2))

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Log-quadratic Fit")
lines(t, exp(logQuad_fit$fit), col = "darkred", lwd = 2, type = "l")

# exponential
exp_fit <- minpack.lm::nlsLM(ts ~ exp(a + b * t), start = list(a = 0, b = 0))

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Exponential Fit")
lines(t, predict(exp_fit, list(x = t)), col = "darkred", lwd = 2)

# log-periodic
sint <- sin(2 * pi * t)
cost <- cos(2 * pi * t)
logPeriod_fit <- lm(log(ts) ~ t + sint + cost)

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Log-periodic fit")
lines(t, exp(logPeriod_fit$fit), col = "darkred", lwd = 2, type = "l")
```

```r
# log-quadratic-periodic
logQuadPeriod_fit <- lm(log(ts) ~ t + I(t^2) + I(sin(2 * pi * t)) + I(cos(2 * pi * t)))

plot(ts, ylab = "Alcohol Sales", xlab = "Time", lwd = 2,
     col = "dodgerblue4", xlim = c(1992, 2020), main = "Log-quadratic-periodic Fit")
lines(t, exp(logQuadPeriod_fit$fit), col = "darkred", lwd = 2, type = "l")


# ---------------[3] ANALYSIS OF THE RESULTS OF MODELS---------------
# residual plots vs. fitted values
par(mfcol = c(2, 2), mar = c (1, 2, 2, 2), oma = c(2, 2, 1.5, 2))

plot(t, linear_fit$fit, ylab = "Alcohol Sales", xlab = "Time", type = "l", main = "linear Fit")
plot(t, linear_fit$residuals, ylab="Residuals", ,xlab="Time", type='l')

plot(t, quad_fit$fit, ylab = "Alcohol Sales", xlab = "Time", type = "l", main = "Quadratic Fit")
plot(t, quad_fit$residuals, ylab="Residuals", ,xlab="Time", type='l')

plot(t, exp(log_fit$fit), ylab = "Alcohol Sales", xlab = "Time", type = "l", main = "Log-linear Fit")
plot(t, exp(log_fit$residuals), ylab="Residuals", ,xlab="Time", type='l')

plot(t, exp(logQuad_fit$fit), ylab = "Alcohol Sales", xlab = "Time", type = "l", main = "Log-quadratic
plot(t, exp(logQuad_fit$residuals), ylab="Residuals", ,xlab="Time", type='l')

plot(t, predict(exp_fit, list(x = t)), lwd = 2, ylab = "Alcohol Sales", xlab = "Time",
     type = "l", main = "Exponential Fit")
plot(t, residuals(exp_fit), ylab="Residuals", ,xlab="Time", type='l')

plot(t, exp(logPeriod_fit$fit), ylab = "alcohol Sales", xlab = "Time", type = "l", main = "Log-periodic
plot(t, exp(logPeriod_fit$residuals), ylab="Residuals", ,xlab="Time", type='l')

par(mfcol = c(2,1), mar = c (1, 2, 2, 2), oma = c(2, 2, 1.5, 2))
plot(t, exp(logQuadPeriod_fit$fit), ylab = "alcohol Sales", xlab = "Time", type = "l", main = "Log-quadi
plot(t, exp(logQuadPeriod_fit$residuals), ylab="Residuals", ,xlab="Time", type='l')


# histograms of residuals for each model
par(mfrow = c(2,2), mar = c (3, 2, 2, 2))
hist(linear_fit$resid, main = "Residuals of Linear Fit")
hist(quad_fit$resid, main = "Residuals of Quadratic Fit")
hist(exp(log_fit$resid), main = "Residuals of Log-linear Fit")
hist(exp(logQuad_fit$resid), main = "Residuals of Log-quadratic Fit")
hist(residuals(exp_fit), main = "Residuals of Exponential Fit")
hist(exp(logPeriod_fit$resid), main = "Residuals of Log-periodic Fit")
hist(exp(logQuadPeriod_fit$resid), main = "Residuals of Log-quadratic-periodic Fit")

# results of model
stargazer(linear_fit, quad_fit, log_fit)
stargazer(logQuad_fit, logPeriod_fit, logQuadPeriod_fit)
summary(exp_fit)

# MSE
models = list(linear_fit = linear_fit, quad_fit = quad_fit, log_fit=log_fit, logQuad_fit = logQuad_fit,
```

```
              exp_fit = exp_fit, logPeriod_fit = logPeriod_fit, logQuadPeriod_fit = logQuadPeriod_fit)

data.frame(MSE = sapply(models, function(x) sum(residuals(x)^2) / (345 - length(x$coeff))))



# ---------------[4] MODEL SELECTION ---------------
# AIC & BIC
t(sapply(models, function(x) c(AIC = AIC(x), BIC = BIC(x))))


# ---------------[5] FORECAST WITH TREND MODEL ---------------
# set up newdata to forecast 24 step ahead
tn <- data.frame(t = seq(2020.8, 2022.75, length = 24))

f1 <- forecast(logQuadPeriod_fit, newdata = tn, h = 24, level = 0.95) #log(ts)

# remeber to take exp() of the results for model use log(y)
ts_predicted <- exp(f1$mean)
pi_h <- exp(f1$upper)
pi_l <- exp(f1$lower)
forecast_df <- data.frame(cbind(t = tn$t, ts_predicted, pi_h, pi_l))
forecast_df[, -1]

# plot the forecast value and prediction interval
ggplot(forecast_df, aes(x = t)) + # time for prediction
         geom_ribbon(aes(ymin = pi_l, ymax = pi_h), alpha = 0.3) +  # prediction intervals
         geom_line(aes(y = ts_predicted)) # predicted values



# ---------------[6] TEST AND CONSTRUCT SEASONALITY ---------------

# test for seasonality
season <- factor(months(df[,'date'], abbreviate = TRUE), month.abb)
stargazer(season_mod <- lm(ts ~ season))


# plot the seasonal factors
plot(season_mod$coefficients, type = "l", main = "Seasonal Factors")


# ---------------[7] FORECAST WITH FULL MODEL ---------------
mod <- lm(log(ts) ~ t + I(t^2) + I(sin(2 * pi * t)) + I(cos(2 * pi * t)) + season)

par(mfrow = c(2,1), mar = c (1, 2, 2, 2), oma = c(2, 2, 1.5, 2))
plot(exp(mod$fitted.values), xlab = "Time", ylab = "log(Alcohol Sales)",
     main = "Model with Trend and Seasonality", type = "l")
plot(exp(mod$residuals), xlab = "Time", type = "l")

stargazer(mod)
mse <- sum(mod$residuals^2)/(345-length(mod$coefficients))
```

```r
# ---------------[8] MODEL WITH TREND AND SEASONALITY ---------------
# forecast with full model ahead of 24 steps
f2 <- forecast(tslm(log(ts) ~ t + I(t^2) + I(sin(2 * pi * t)) + I(cos(2 * pi * t)) + season),
         newdata = tn, h = 24, level = 0.95)

ts_predicted1 <- exp(f2$mean)
pi_h1 <- exp(f2$upper)
pi_l1 <- exp(f2$lower)
forecast_df1 <- data.frame(cbind(t = tn$t, ts_predicted1, pi_h1, pi_l1))
forecast_df1[, -1]

ggplot(forecast_df1, aes(x = t)) + # time for prediction
         geom_ribbon(aes(ymin = pi_l1, ymax = pi_h1), alpha = 0.3) +  # prediction intervals
         geom_line(aes(y = ts_predicted1)) # predicted values


# ---------------[9] REMOVE SEASONALITY ---------------
# decompose the model
plot(mod1 <- stl(log(ts),s.window="periodic"))


# remove seasonality
seasonal <- mod1$time.series[, 1]
seasonal_Adjusted <- ts - seasonal

plot(seasonal_Adjusted, main = "Alcohol Sales (Seasonal Adjusted)")
```