

# MA 4780 Excam 2

Benjamin Hendrick

April 16, 2016

## Question 1

### Part A

Given the ARIMA model

$$Y_t = 0.7Y_{t-1} + Y_{t-12} - 0.7Y_{t-13} + e_t - 0.9e_{t-1} - 0.1e_{t-2}$$

- The process written in backshift notation is  $Y_t(1 - 0.7B)(1 - B^{12}) = e_t(1 - 0.9B - 0.1B^2)$
- The given model has  $p = 1, d = 0, q = 1, P = 1, D = 1, Q = 0$ , and  $s = 12 \rightarrow ARIMA(1, 0, 1) \times (1, 1, 0)_{12}$ . The model has parameters  $\phi_1 = 0.7, \Phi = -1, \theta_1 = -0.9, \theta_2 = -0.1, \Theta = 0$
- The roots of  $\psi(z) = (1 - 0.7z)(1 - z^{12}) = 0$  include -1 and 1, which are outside of the unit circle. Therefore, the model is **not stationary or causal**. The roots of  $\Theta(B) = (1 - 0.9B - 0.1B^2)$  include -1 and 1, which are outside of the unit circle. Therefore, the model is **not invertible**.
- The behavior of the forecast value for the  $ARIMA(1, 0, 1) \times (1, 1, 0)_{12}$  would be be annually seasonal (with peaks and valleys), but the mean would not increase over time. Because the seasonality is AR, the trend would not decrease. The behavior of the prediction limits would increase and the limits grow farther apart.

### Part B

Given the ARIMA model

$$Y_t = -0.2Y_{t-1} + e_t - 0.5e_{t-4}$$

- The process written in backshift notation is  $Y_t(1 + 0.2B) = e_t(1 - 0.5B^4)$
- The given model has  $p = 1, d = 0, q = 1, P = 0, D = 0, Q = 1$ , and  $s = 12 \rightarrow ARIMA(1, 0, 1) \times (0, 0, 1)_4$ . The model has parameters  $\phi_1 = -0.2, \Phi = 0, \theta_1 = 1, \Theta = -0.5$ .
- The roots of  $\psi(z) = (1 + 0.2z) = 0$  is -5, which is outside of the unit circle. Therefore the model is **not stationary or causal**. The roots of  $\Theta(B) = (1 - 0.5B^4) = 0$  include 2, which is outside of the unit circle. Therefore, the model is **not invertible**.
- The behavior of the forecast value for the  $ARIMA(1, 0, 1) \times (0, 0, 1)_{12}$  would be be annually seasonal (with peaks and valleys), but the mean would not increase over time. Because the seasonality is MA, the trend decrease and increase over time. The behavior of the prediction limits would be constant.

### Part C

Given the ARIMA model

$$Y_t = 0.2Y_{t-1} - 0.8Y_{t-2} + e_t - 0.2e_{t-1}$$

- i. The process written in backshift notation is  $Y_t(1 - 0.2B + 0.8B^2) = e_t(1 - 0.2B)$ .
- ii. The given model has  $p = 2, d = 0, q = 1, P = 0, D = 0, Q = 0$  and no seasonality, so  $s = 0 \rightarrow ARIMA(2, 0, 1) \times (0, 0, 0)_0$ . This model has parameters  $\phi_1 = 0.2, \phi_2 = -0.8, \Phi = 0, \theta_1 = -0.2, \Theta = 0$ .
- iii. The roots of  $\psi(z) = 1 - 0.2z + 0.8^2 = 0$  are  $z \rightarrow 0.125 \pm 1.11102i$ , which is on the unit circle. therefore, the model **is** *stationary* and *causal*. The root of  $\Theta(B) = (1 - 0.2B) = 0$  is 5. Therefore, the model is **not** *invertible*.
- iv. The behavior of the forecast value for the  $ARIMA(1, 0, 1) \times (0, 0, 0)_0$  would be be annually seasonal (with small peaks and valleys), but the mean would not increase over time. There would be no seasonality. The behavior of the prediction limits would be constant.

## Question 2

### Part A

- i. I would suggest an ARIMA model with random walk because there are a lot of big jumps between lags and no apparent seasonality. There might be a slight deterministic trend to the model because it increases over time.
- ii. I would determine my model by (1) checking the behavior of the ACF and PACF plots, (2) check the EACF correlation matrix, and (3) consult a BIC matrix plot. After narrowing down the top candidate models, I would then use the `arima` function in R to model the different ARIMA models of the data. Then I would select the model with the smallest AIC value.

### Part B

- i. I would suggest an ARIMA with MA seasonality because the data does not increase or decrease and the seasonality does not increase or decrease.
- ii. I would select my model in the same way as in Part A.

## Question 3

Load the data.

```
q3.data <- read.table("~/GitHub/MA-4780/Exam 2/Exam2-1.txt", quote="\"", comment.char="")
q3.ts <- ts(data = q3.data, start = 1970, end = 1979, frequency = 52)
```

### Part A

The autocorrelations at lag 1 and lag 2 of the sample ACF are  $r_1 = 0.772$  and  $r_2 = 0.681$ . The autocorrelation, meaning the correlation of the data with itself at another time, is decreasing. This can be used to detect non-randomness in the data. Therefore, for lag 1 and lag 2, we can say that there is about 77% and 68% non-randomness at these lags, respectively.

```
acf(q3.data, plot=FALSE)[2:3]
```

```
##
## Autocorrelations of series 'q3.data', by lag
##
```

```
##      2      3
## 0.772 0.681
```

The partial autocorrelations at lag 1 and lag 2 of the sample PACF are  $\hat{\phi}_{11} = 0.438$  and  $\hat{\phi}_{22} = 0.025$ . The partial autocorrelations found are the autocorrelations of themselves (e.g. the autocorrelation between 1 and 1).

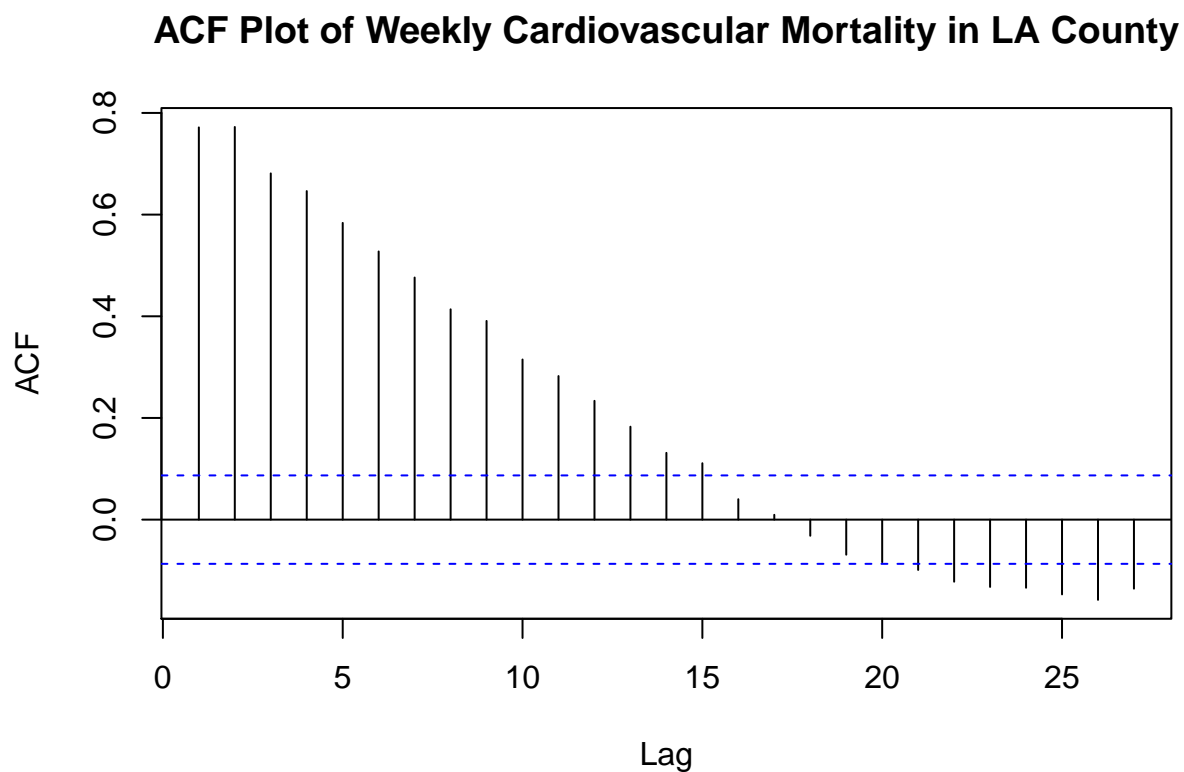
```
pacf(q3.data, plot=FALSE)[2:3]
```

```
##
## Partial autocorrelations of series 'q3.data', by lag
##
##      2      3
## 0.438 0.025
```

## Part B

The ACF plot:

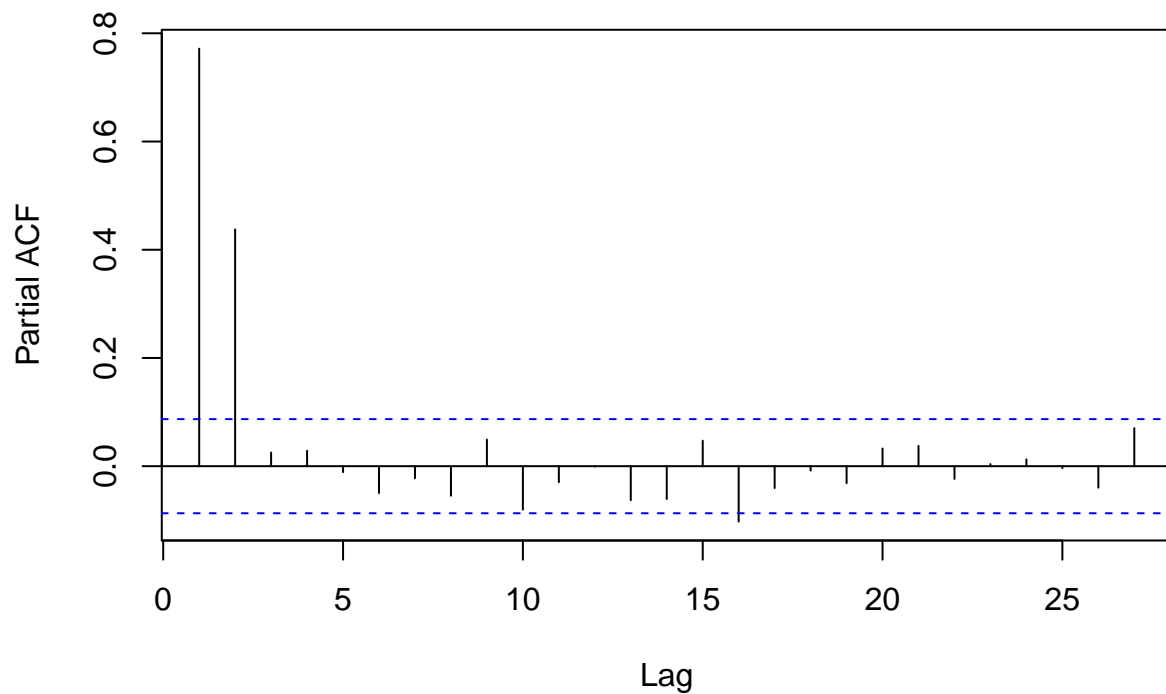
```
acf(q3.data, main = "ACF Plot of Weekly Cardiovascular Mortality in LA County")
```



The PACF plot:

```
pacf(q3.data, main = "PACF Plot of Weekly Cardiovascular Mortality in LA County")
```

## PACF Plot of Weekly Cardiovascular Mortality in LA County



Use an EACF matrix and BIC matrix to help determine the orders of  $p$  and  $q$  of an  $ARMA(p, q)$  model.

```
eacf(q3.ts)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x x x
## 1 x x x o o o o o x x o o o o
## 2 o o o o o o o o x o o o o o
## 3 x o o o o o o o x o o o o o
## 4 x x o o o o o o o o o o o o
## 5 x x o o o o o o o o o o o o
## 6 x x x o o o o o o o o o o o
## 7 x x x o x o x o o o o o o o
```

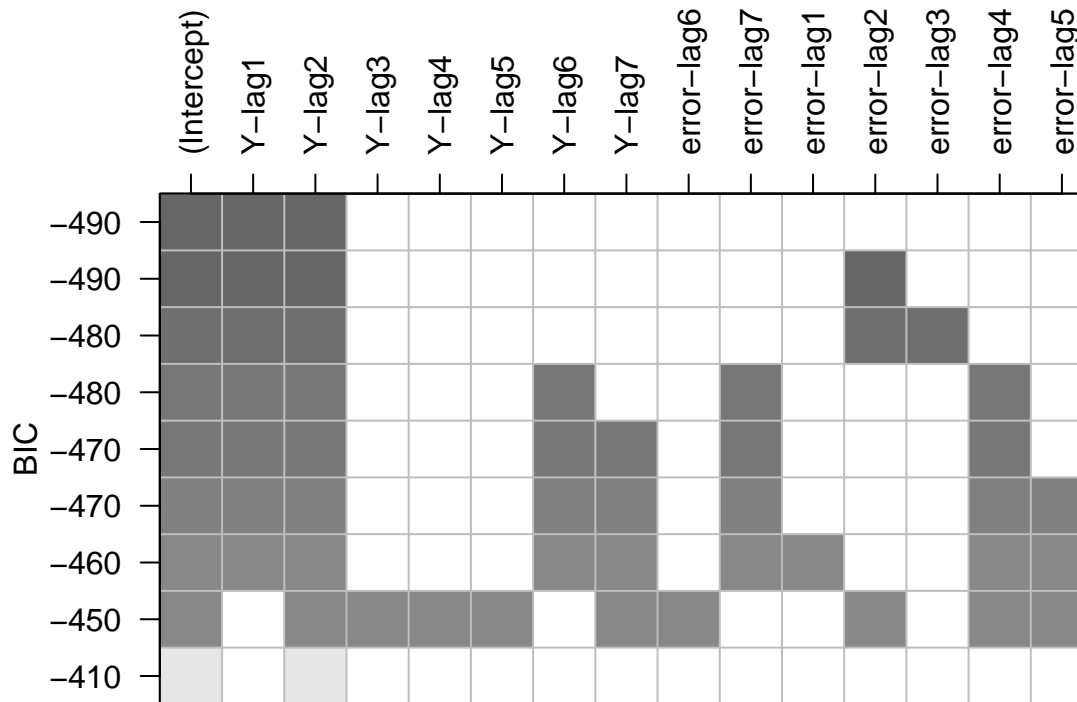
The EACF matrix suggests using an AR(2) process to model the data.

Now plot a BIC matrix to help confirm/determine the model selection.

```
res <- armasubsets(y=q3.ts, nar=7, nma = 7)
```

```
## Reordering variables and trying again:
```

```
plot(res)
```



The BIC plot suggest an AR(2) process with a BIC of -490. This was the top model selected by the `armasubsets` function. Another top selection was an ARMA(2,2) model.

Model the AR(2) process and ARMA(2,2) process for the data with the `arima` function to compare the AIC values. Use maximum likelihood (`method = "ML"`) to create the model estimates.

```
ar2 <- arima(q3.ts, method = "ML", order = c(2,0,0))
arma22 <- arima(q3.ts, method = "ML", order = c(2,0,2))
```

The AIC value for the AR(2) model is 2984.0546577. The AIC value for the ARMA(2,2) model is 2987.7352529. I will select the AR(2) model as the best model because it has the lowest AIC value and is less complex as the ARMA(2,2) model.

## Part C

Use the modeled AR(2) model from Part B to determine the point estimate of  $\theta_0, \phi_1, \phi_2$ , and  $\sigma_e^2$ .

```
ar2

##
## Call:
## arima(x = q3.ts, order = c(2, 0, 0), method = "ML")
##
## Coefficients:
##          ar1      ar2  intercept
##         0.4219  0.4497   89.6431
## s.e.    0.0411  0.0413    2.0357
##
## sigma^2 estimated as 33.42:  log likelihood = -1489.03,  aic = 2984.05
```

From the above output, we can determine that  $\theta_0 = 89.6430549$ ,  $\phi_1 = 0.4218777$ ,  $\phi_2 = 0.4496655$ , and  $\sigma_e^2 = 33.4170903$ .

## Part D

Using R, a theoretical confidence interval can be made for the AR(2) process where  $\phi_1 = 0.4$ . The AR(2) process `ar2` will be used.

```
phi1 <- ar2$coef[1]
se <- sqrt((1-phi1^2)/100)
ciu <- phi1 + 2*se
cil <- phi1 - 2*se
```

The confidence interval for the hypothesis  $\phi_1 = 0.4$  is (0.2405471, 0.6032082). Because 0.4 is inside of the confidence interval, we fail to reject the null hypothesis  $\phi_1 = 0.4$  and conclude that the hypothesis is compatible with the data.

## Part E

The `predict` function in R predicts the 1-step and 2-step ahead MMSE forecasts.

```
ar2.pred <- predict(ar2, n.ahead = 2)
ar2.pred$pred
```

```
## Time Series:
## Start = c(1979, 2)
## End = c(1979, 3)
## Frequency = 52
## [1] 97.14380 95.45011
```

The predicted 1-step and 2-step values are 97.1437968 and 95.4501098, respectively. The following code creates the lower and upper prediction interval bounds.

```
lower.pi <- ar2.pred$pred - qnorm(0.9756, 0, 1) * ar2.pred$se
upper.pi <- ar2.pred$pred + qnorm(0.9756, 0, 1) * ar2.pred$se
```

The 95% prediction interval for the 1-step prediction is (85.7537799, 108.5338137). The 95% prediction interval for the 2-step prediction is (83.0879753, 107.8122444).

## Question 4

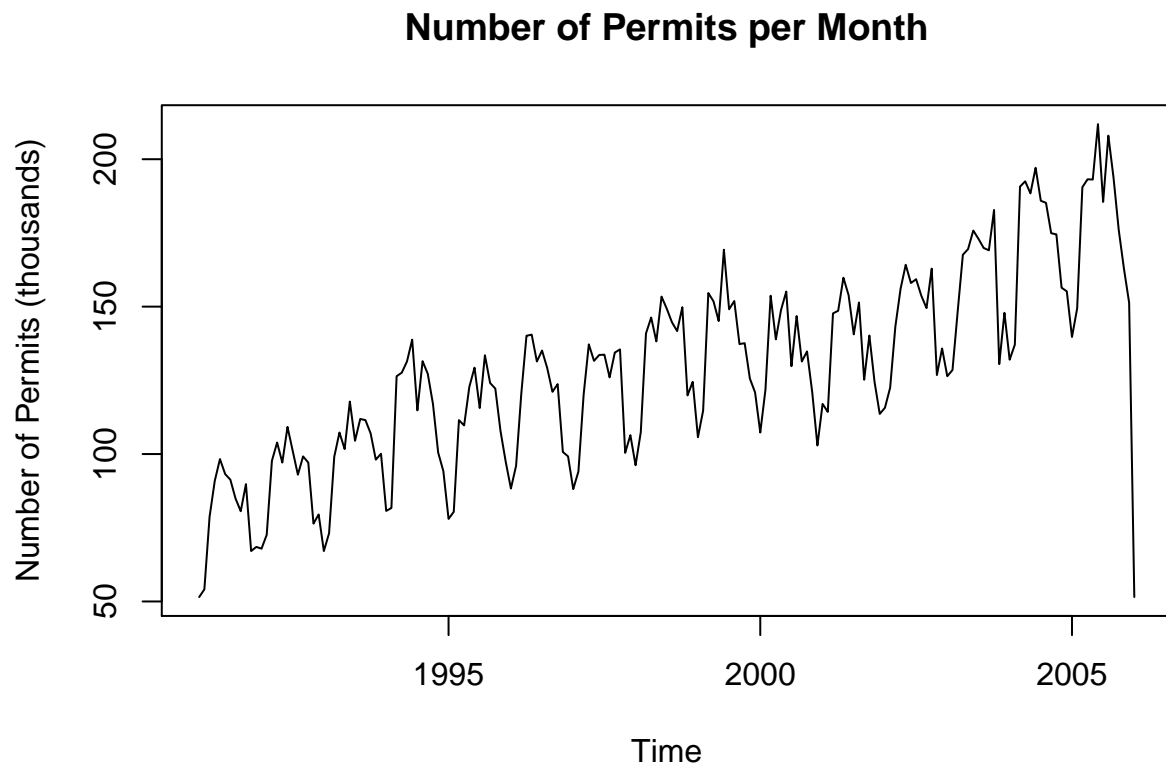
Load the data.

```
permit <- read.csv("~/Downloads/Exam1-permit.csv", header=FALSE)
permit.ts <- ts(permit[2], start = 1991, end = 2006, frequency = 12)
```

## Part A

- i. Begin by plotting the data.

```
plot(permit.ts, xlab = "Time", ylab = "Number of Permits (thousands)", main = "Number of Permits per Month")
```



Take the difference to remove the upward trend. Take the seasonal difference of the newly transformed data to remove the seasonality.

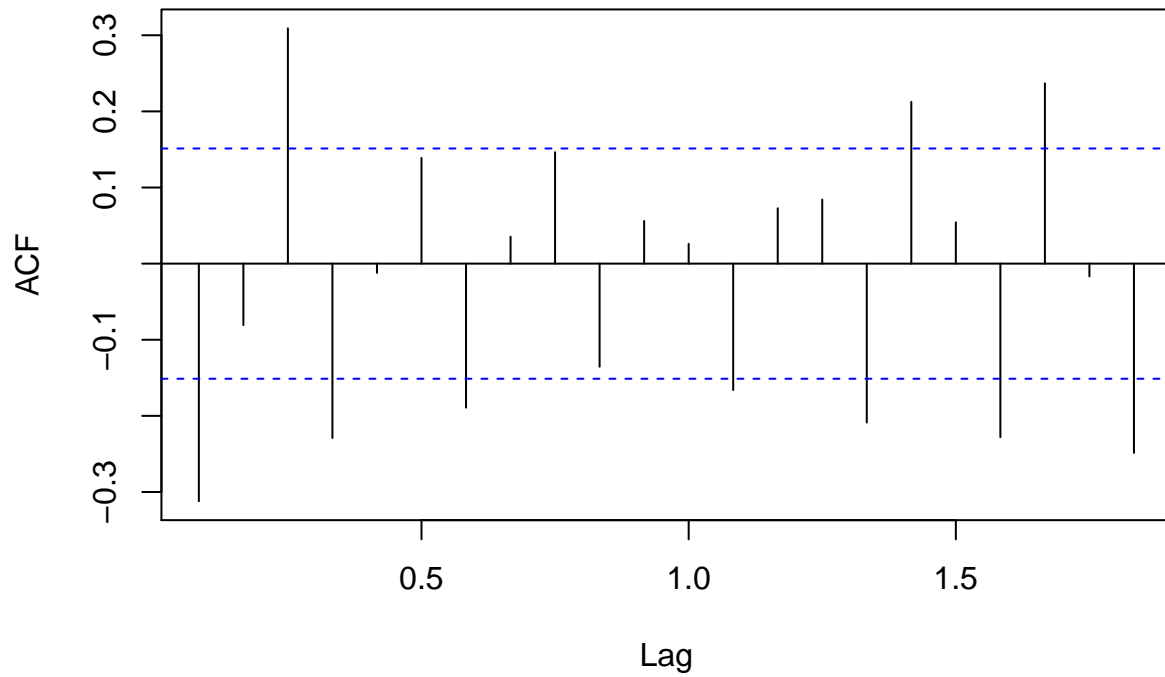
```
permit.sdiff <- diff(diff(permit.ts),12)
```

There is obvious seasonality.

Plot the ACF of the data.

```
acf(permit.sdiff, main = "ACF of Permit Data")
```

### ACF of Permit Data

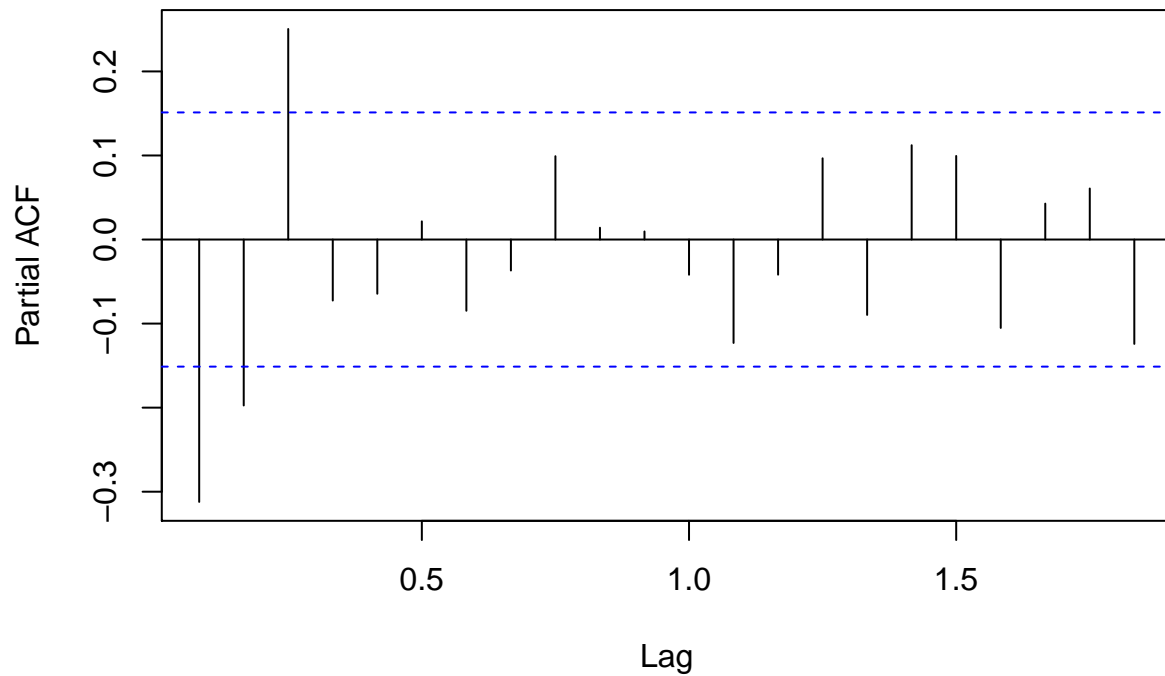


From the ACF, the data appear to shape an ARMA model.

Plot the PACF of the data.

```
pacf(permit.sdiff, main = "PACF of Permit Data")
```

### PACF of Permit Data





From the PACF, the data appear to shape an ARMA or MA model.

Make the EACF matrix.

```
eacf(permit.sdiff)
```

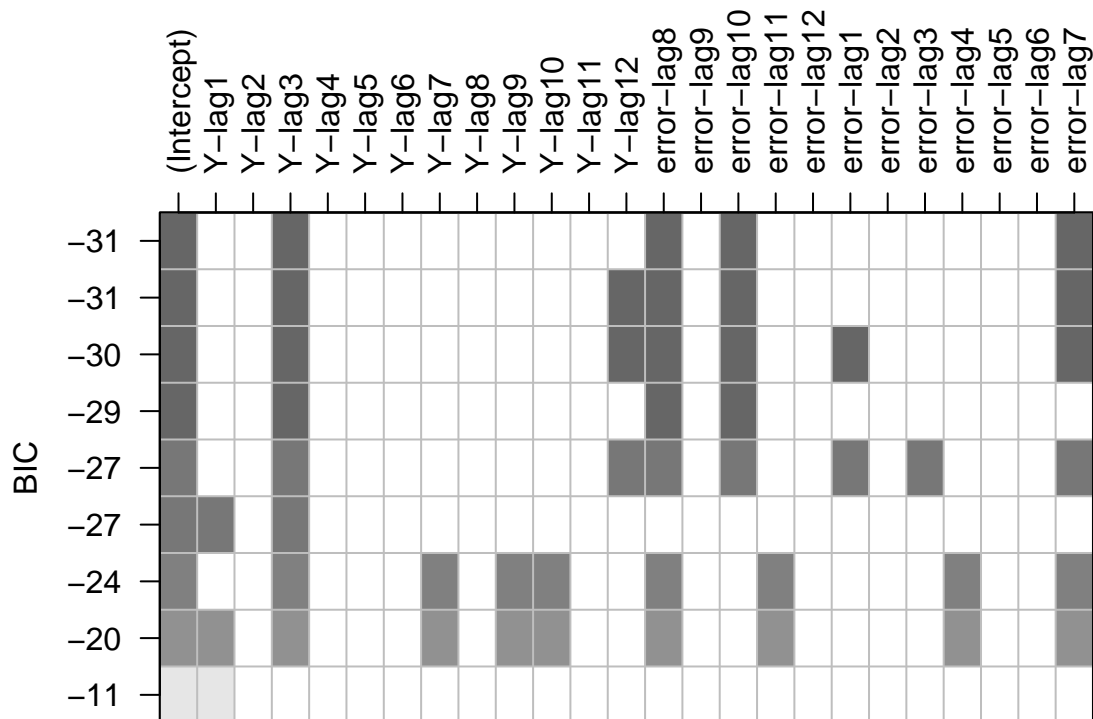
```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o x x o o x o o o o o x o
## 1 x o x x o o x o x o o o x x
## 2 x o o o o o o o o o o x o o
## 3 o o x o o o o o o o x o o o
## 4 x x x o o o o o o o o o o
## 5 o x x o o o x o o o o x o o
## 6 o x x x o x o o o o o x o x
## 7 x x x x o x o o o o o x o x
```

Create a BIC plot to help determine the ideal model.

```
permit.res <- armasubsets(y=permit.sdiff, nar=12, nma = 12)
```

```
## Reordering variables and trying again:
```

```
plot(permit.res)
```



From the BIC plot, the data will be modeled by an  $ARIMA(2, 1, 0) \times (1, 1, 1)_{12}$  seasonal process. This is the least complex model with the lowest BIC.

Create the chosen model.

```
permit.model <- arima(permit.sdiff, method = "ML", order = c(2,0,0),
                      seasonal = list(order = c(1,0,1), period = 12))
```

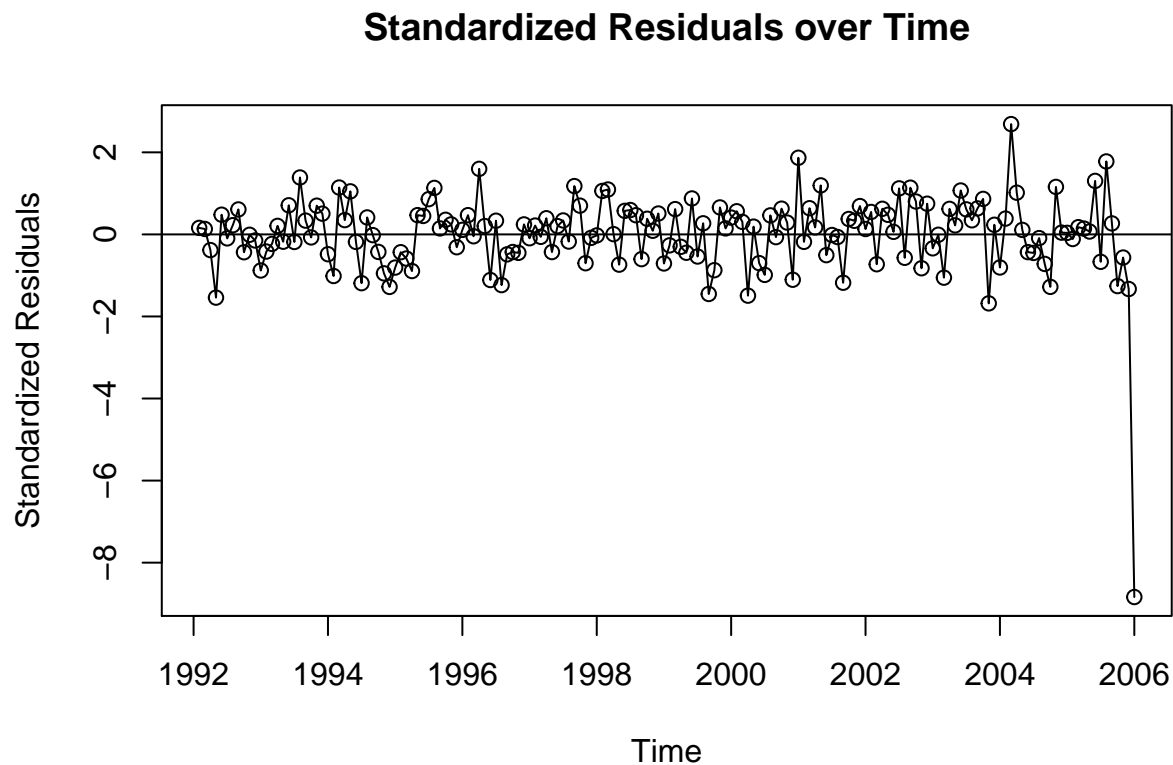
ii. The chosen model formula is  $Y_t = -0.7240Y_{t-1} - 0.4503Y_{t-2} + 0.2660Y_{t-12} - e_{t-12}$ .

```
permit.model
```

```
##
## Call:
## arima(x = permit.sdiff, order = c(2, 0, 0), seasonal = list(order = c(1, 0,
##      1), period = 12), method = "ML")
##
## Coefficients:
##      ar1      ar2      sar1      sma1  intercept
##    -0.6579 -0.4609  0.3725 -0.8452    -0.1723
## s.e.   0.0987   0.0992  0.1380   0.0993     0.1466
##
## sigma^2 estimated as 103:  log likelihood = -631.87,  aic = 1273.73
```

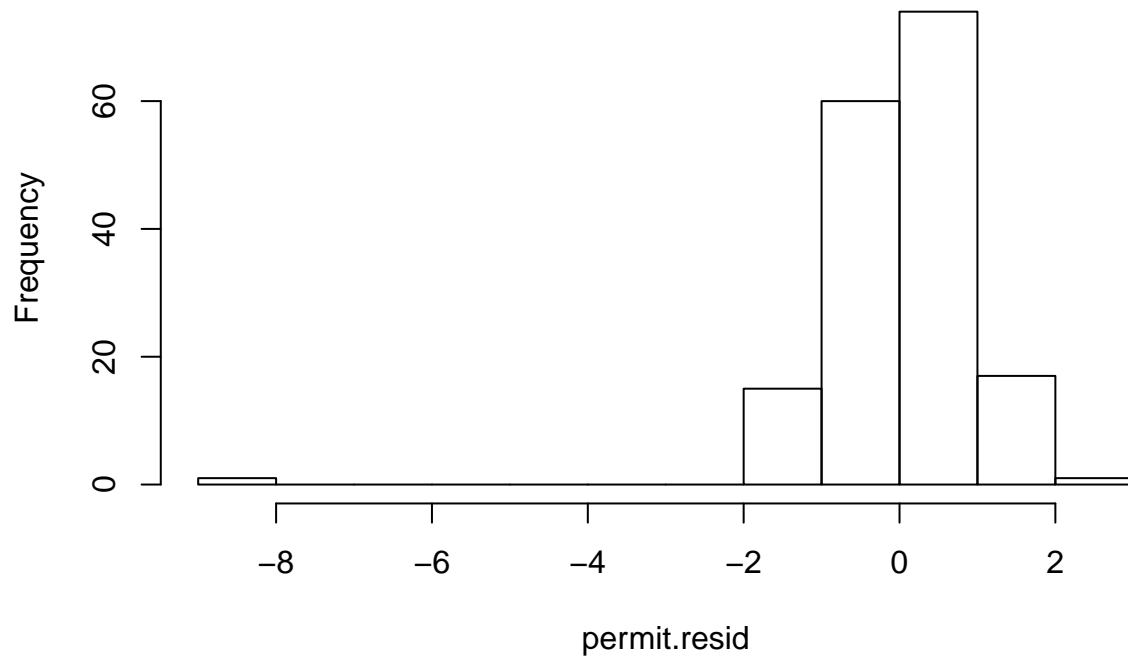
iii. Graphically check for normality assumption with a plot of the residuals, histogram, and Q-Q plot.

```
permit.resid <- rstandard(permit.model)
plot(permit.resid, type = "o", ylab = "Standardized Residuals", main = "Standardized Residuals over Time",
     abline(h=0))
```



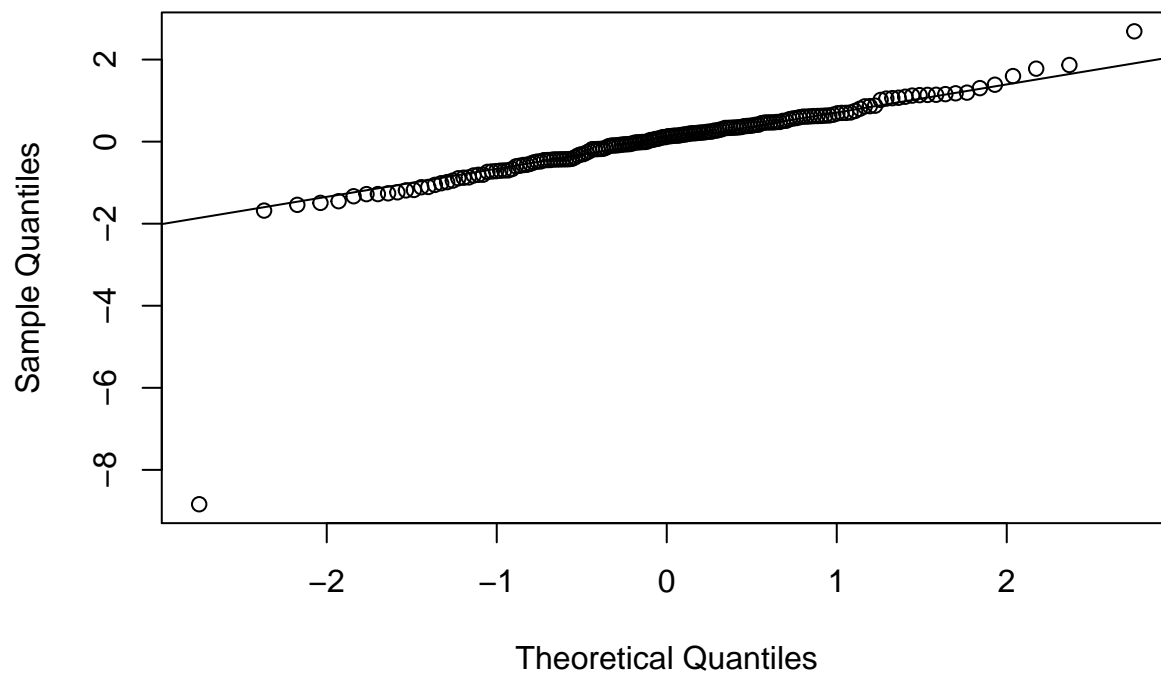
```
hist(permit.resid, main = "Histogram of Standardized Residuals")
```

**Histogram of Standardized Residuals**



```
qqnorm(permit.resid)  
qqline(permit.resid)
```

**Normal Q-Q Plot**



By the residual plot, histogram, and Q-Q plot, the residuals all appear to be normally distributed. A Shapiro-Wilk Normality Test will statistically determine normality.

```
shapiro.test(permit.resid)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: permit.resid  
## W = 0.74149, p-value = 7.055e-16
```

Because the p-value of the Shapiro-Wilk test is so high, the null hypothesis is unable to be rejected and normality is assumed.

Test the independence assumption with a Runs test.

```
runs(permit.resid)
```

```
## $pvalue  
## [1] 0.559  
##  
## $observed.runs  
## [1] 80  
##  
## $expected.runs  
## [1] 84.2381  
##  
## $n1  
## [1] 76  
##  
## $n2  
## [1] 92  
##  
## $k  
## [1] 0
```

Because the p-value of the Runs test is so high, the null hypothesis is unable to be reject and independence is assumed.

## Part B

Load the forecasting data.

```
permit.forecast <- read.table("~/Downloads/Exam2-2-forecast.csv", quote="", comment.char="")  
permit.forecast.ts <- ts(permit.forecast, start = 2006, frequency = 12)
```

- i. Plot the forecast of the seasonal  $ARIMA(2,0,0) \times (1,0,1)_{12}$  model.

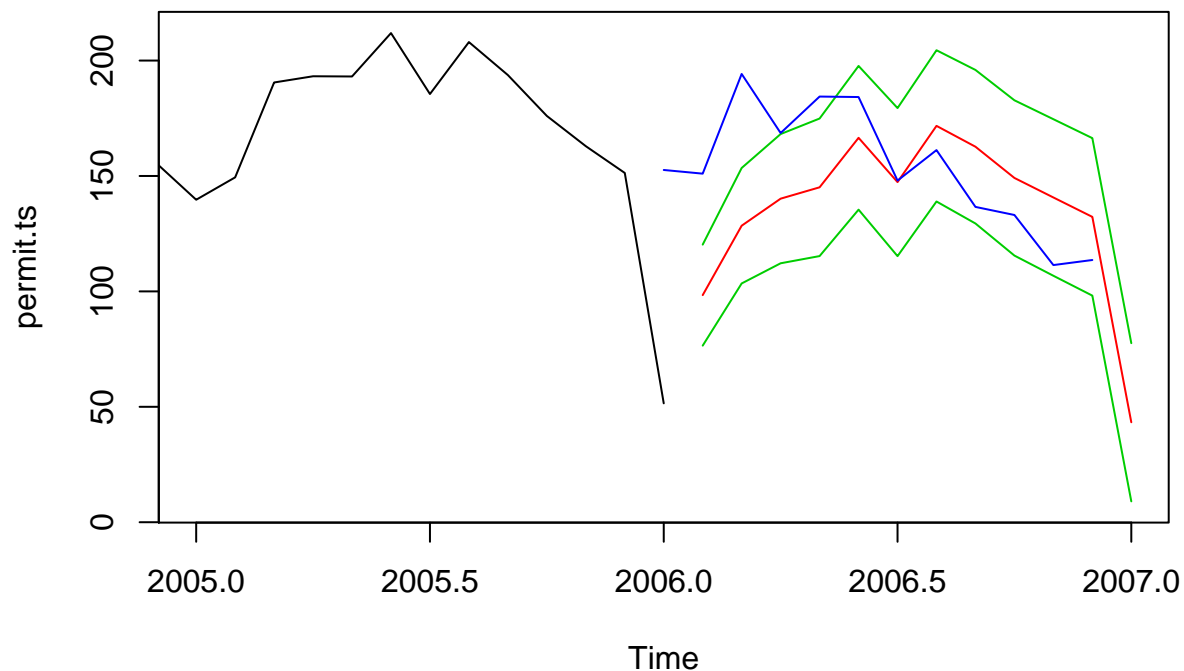
```

model <- arima(permit.ts, method = "ML", order = c(2,0,0),
               seasonal = list(order = c(1,0,1), period = 12))
predx <- predict(model,n.ahead=12)
pr <- predx$pred
uci <- pr+2*predx$se
lci <- pr-2*predx$se

ymin <- min(c(as.vector(lci),permit.ts))-1
ymax <- max(c(as.vector(uci),permit.ts))+1

plot(permit.ts,xlim = c(2005,2007),ylim=c(ymin,ymax))
lines(pr,col=2)
lines(uci,col=3)
lines(lci,col=3)
lines(permit.forecast.ts, col = 4)

```



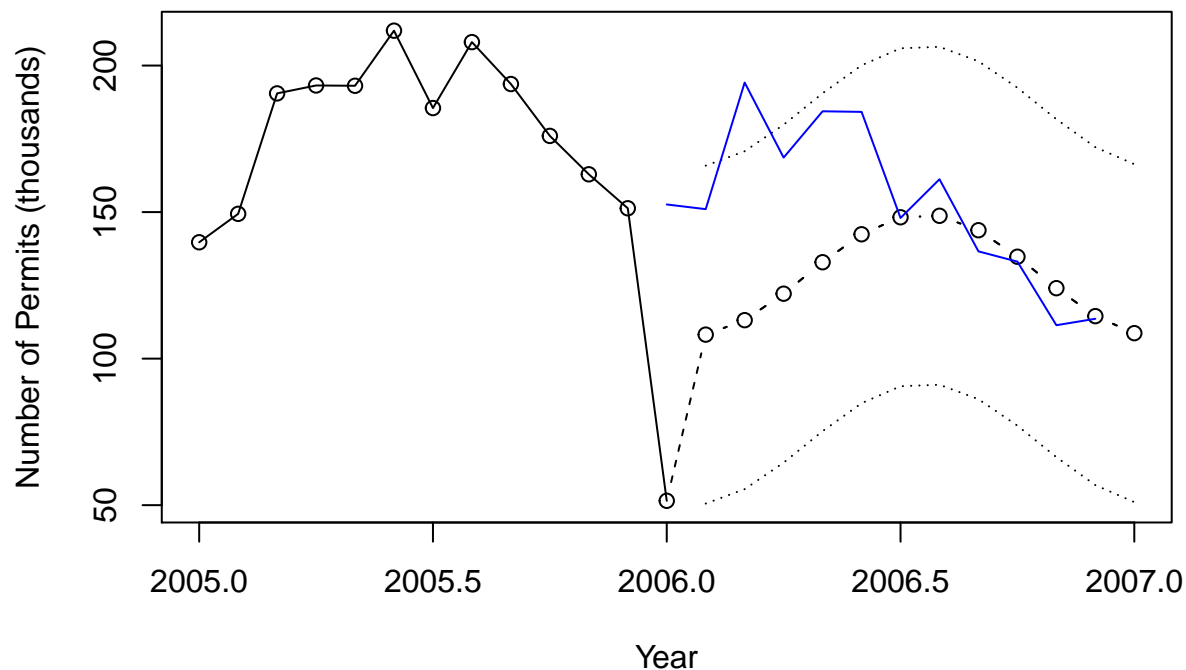
The red line of the forecast is my prediction. The blue line is the data `permit.forecast.ts`. The green lines are the 95% prediction intervals. The given forecast data fell outside of the prediction interval after six months in 2006. That said, it isn't too far out. Otherwise, it stayed inside out the interval.

- ii. Plot the cosine trend + linear trend + white noise (iid noise) model against the given forecast data.

```

har <- harmonic(permit.ts, 1)
permit.m5 <- arima(permit.ts, order = c(0,0,0), xreg = har)
newhar <- harmonic(ts(rep(1,12), start = c(2006,1), frequency = 12), 1)
plot(permit.m5, n.ahead = 12, n1= c(2005,1), newxreg = newhar,
     type = "b", ylab = "Number of Permits (thousands)", xlab = "Year")
lines(permit.forecast.ts, col = 4)

```



The predictions are not similar for the first six months of 2006. The second six months of 2006 are actually quite similar. The given forecast data seems to fall in the majority of the prediction interval.)

Compute the sum of squares and forecast errors of the two models.

```


arma.sse <- sum((permit.forecast.ts-pr)^2)
arma.fe <- sum(permit.forecast.ts-pr)
cosine.sse <- sum((permit.forecast.ts-predict(permit.m5, newxreg = newhar)$pred)^2)
cosine.fe <- sum((permit.forecast.ts-predict(permit.m5, newxreg = newhar)$pred))


```

The following table compares the sum of squares and forecast error for the two models.

Model	Sum of Squares Error	Forecast Error
ARIMA	9901.891	-235.9471
Cosine	13102.8	293.52