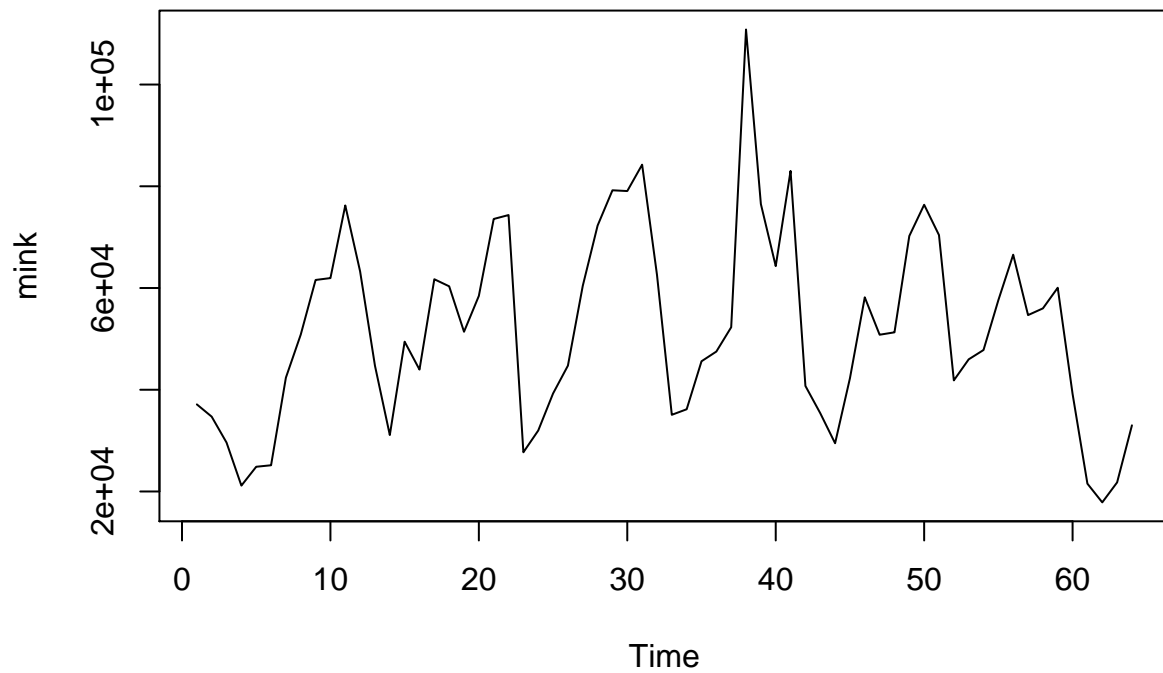


Take-Home Final Exam

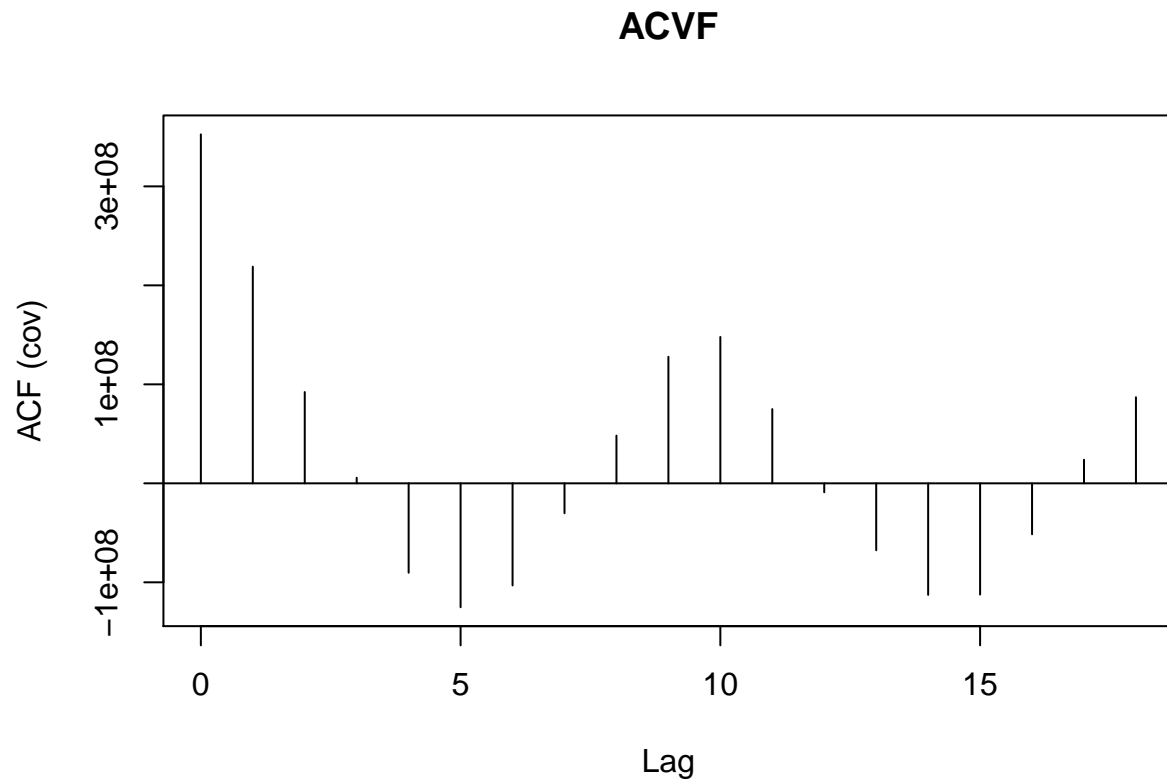
Steve Harms

November 29, 2018

```
library(itsmr)
#read in the data
mink <- ts(read.table("apph.tsm"))
#plot the original time series
ts.plot(mink)
```



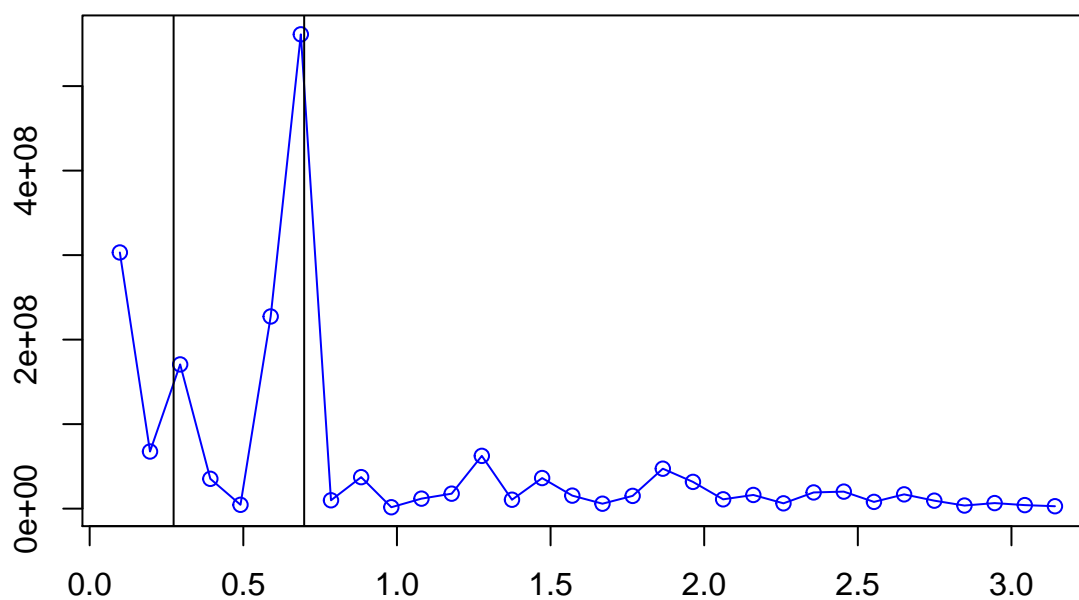
```
#take log of the original series
minkl <- Resid(mink,M=c("log"))
#plot the ACVF, ACF and PACF
acf(minkl, type="covariance", main = "ACVF")
```



I took the logarithm of the original data, just like we did with the lynx data in class. This seems reasonable, as the trappings will depend on population size, thus it may be better to change the scale. The un-transformed time series looks to have a seasonal component based on the ACF, and the PACF indicates we may not need more than $AR(1)$.

```
#Plot the periodogram to examine the spectral density  
periodogram(mink1) + abline(v=2*pi/9)+ abline(v=2*pi/23)
```

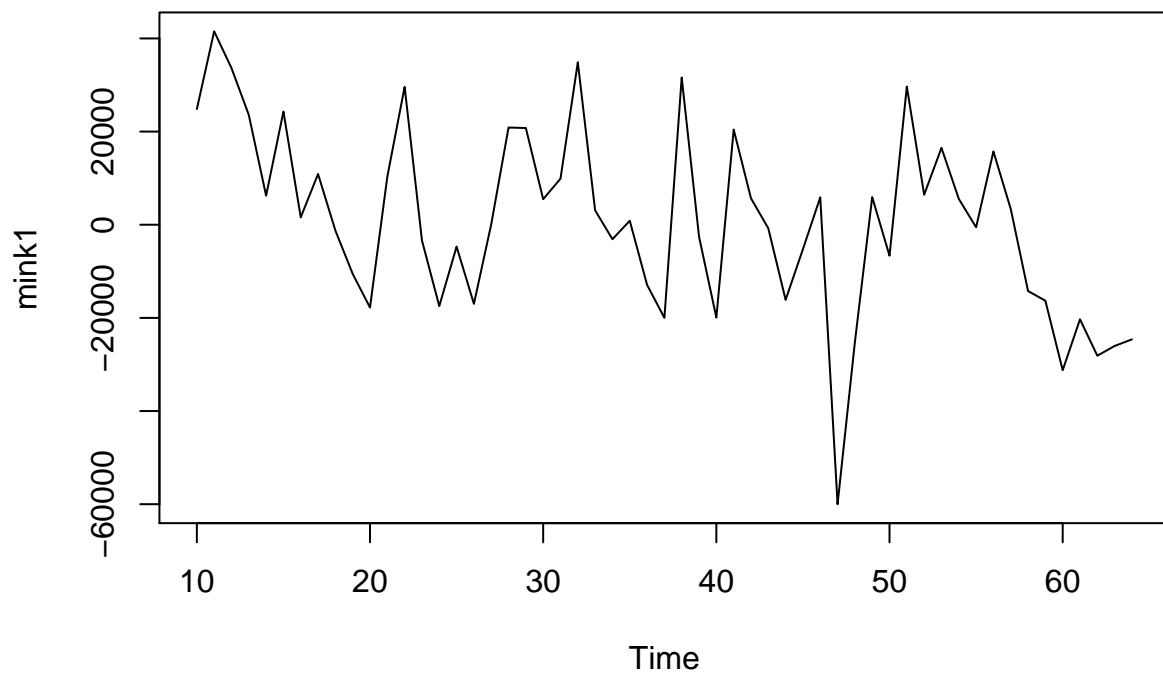
Periodogram/2pi



```
## numeric(0)
```

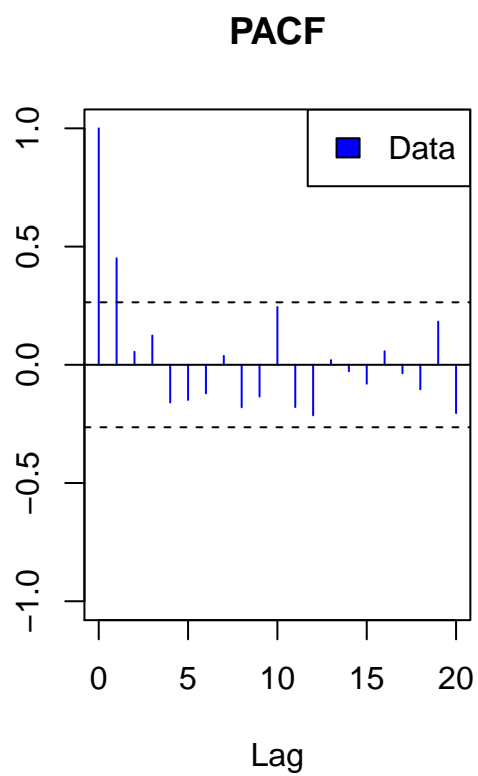
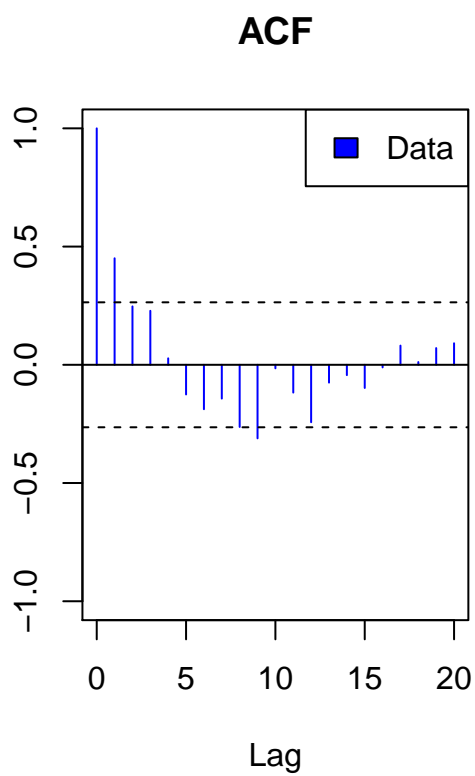
The periodogram looks we have spikes at $2\pi/9$ and $2\pi/23$, indicating that periods of 9 and 23 might be our seasonal periods to investigate.

```
#take seasonal differences with d=9  
mink1 <- diff(mink1,lag=9)  
#plot the new residuals of the time series  
ts.plot(mink1)
```



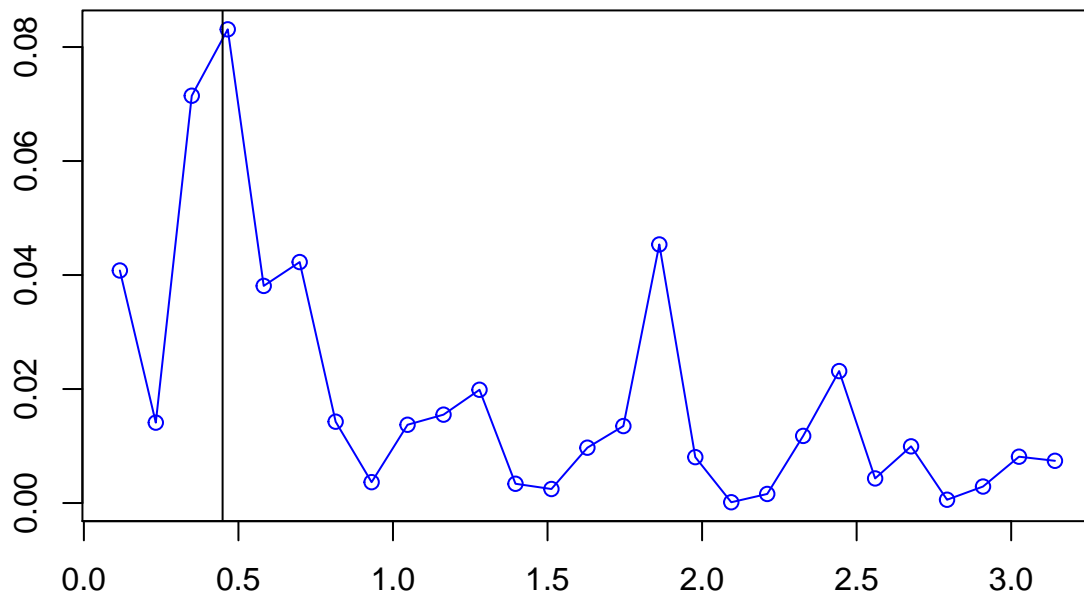
The new plot looks to have a clear downward trend, I just use a linear trend.

```
#get residuals from a model with log transformation, d=9 differencing, and linear trend
mink1t <- Resid(mink, M=c("log","diff",9,"trend",1))
#ACF and PACF of our new residuals
plota(mink1t, h=20)
```



```
#Periodogram of the new residuals  
periodogram(mink1t) + abline(v=2*pi/14)
```

Periodogram/2pi

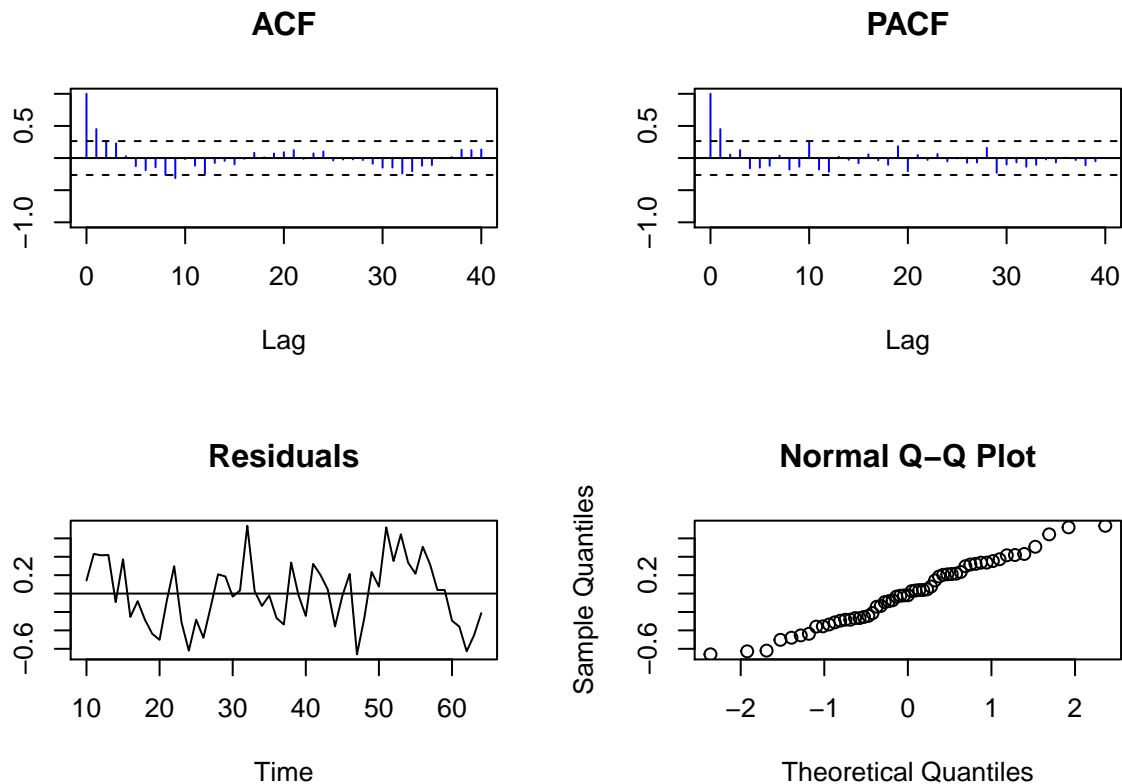


```
## numeric(0)
```

After taking seasonal difference operator $d=9$, the periodogram still indicates a spike at $2\pi/14$, i.e., period = 14. This matches what we had above ($9+14=23$, the other spike in the original periodogram). Now we are met with a problem: Do we take differences again, this time with $d=14$, or do we move on?. The ACF and PACF look somewhat reasonable, it appears an $AR(1)$ model might do quite well. However, the p -value for the McLeod-Li Statistic is small, indicating that we do not have Gaussian white noise in our residuals:

```
#test the residuals for white noise / stationarity
test(Resid(mink, M=c("log","diff",9,"trend",1)))
```

```
## Null hypothesis: Residuals are iid noise.
## Test      Distribution Statistic  p-value
## Ljung-Box Q      Q ~ chisq(20)    42.74    0.0022 *
## McLeod-Li Q      Q ~ chisq(20)    16.69    0.6729
## Turning points T  (T-35.3)/3.1 ~ N(0,1)    32     0.2784
## Diff signs S      (S-27)/2.2 ~ N(0,1)    23     0.0641
## Rank P          (P-742.5)/68.9 ~ N(0,1)   744    0.9826
```



The new PACF indicates we may not need many AR terms in our final model after differencing. The ACF and PACF look reasonable, so it's time to look at possible ARMA models for the residuals. I am going to resist using another difference operator with $d=14$ to avoid overfitting, although we may run into trouble later. Here are some options for ARMA models:

```
#estimate some models
arma11 <- arma(mink1t, p=1, q=1)
arma10 <- arma(mink1t, p=1, q=0)
arma01<- arma(mink1t, p=0, q=1)
arma00 <- arma(mink1t, p=0, q=0)

arma11$aic; arma10$aic; arma01$aic; arma00$aic

## [1] 32.54644
## [1] 30.72121
## [1] 32.59049
## [1] 40.9627
```

The best model is the $ARMA(1,0)$ model based on AIC, however all of the models considered above are very close and thus are competitive. Using any of them to forecast leads to very similar predictions. The forecast is largely determined by the seasonal and trend structure, not the small scale MA and AR terms.

Estimates and Confidence Intervals

```
#we can use autofit() to have ITSM pick the best model for us based on AIC
#turns out to be ARMA(1,0)
bestmodel <- autofit(mink1t, p=0:3, q=0:3)
```

ϕ

```
bestmodel$phi
```

```
## [1] 0.4470151
```

```
c(bestmodel$phi - qnorm(.975)*bestmodel$se.phi, bestmodel$phi + qnorm(.975)*bestmodel$se.phi)
```

```
## [1] 0.2134469 0.6805834
```

θ

```
bestmodel$theta
```

```
## [1] 0
```

```
c(bestmodel$theta - qnorm(.975)*bestmodel$se.theta, bestmodel$theta + qnorm(.975)*bestmodel$se.theta)
```

```
## [1] 0 0
```

White Noise variance σ^2

```
bestmodel$sigma2
```

```
## [1] 0.09439325
```

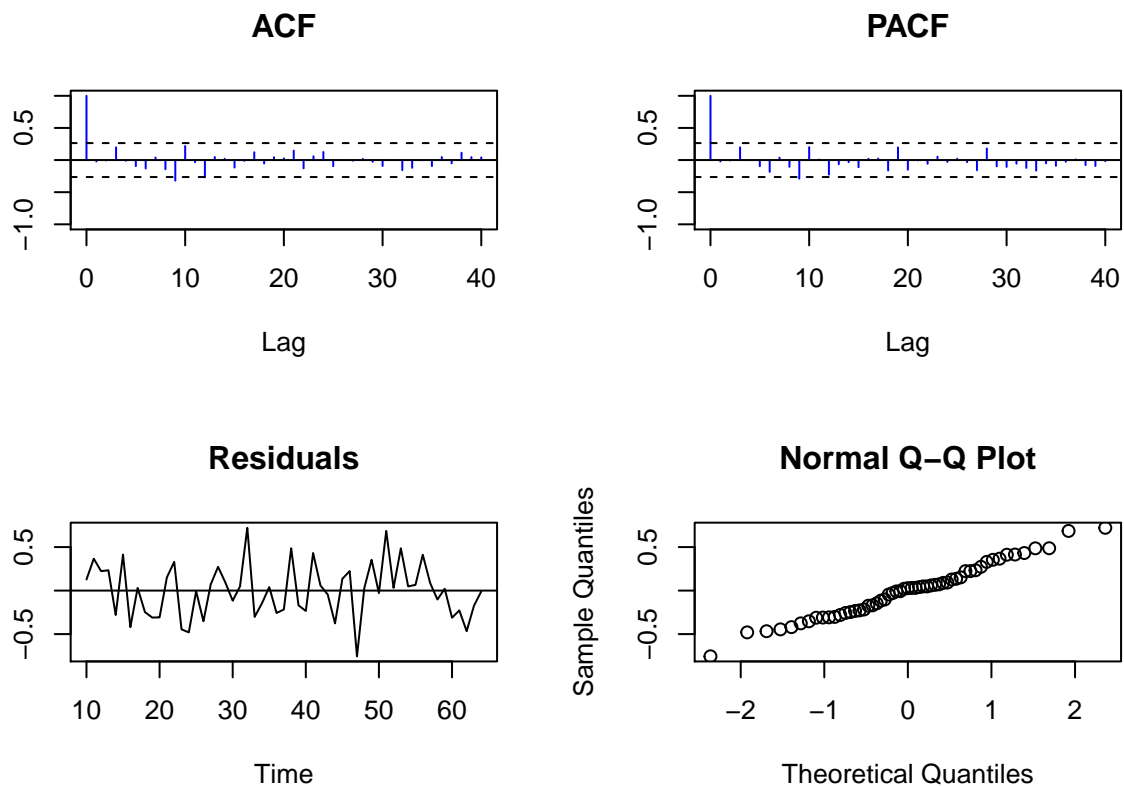
Checking for White Noise in the residuals

The p -values are large for all of the tests except for 1, and the ACF and PACF look OK. There are no obvious missing AR terms according to the PACF, and the ACF has a slight pattern but only 2 of the 40 lags (95%) are outside of the Bartlett bounds, which is about what we would expect. The biggest concern is that the McLeod-Li statistic is large enough to reject the null hypothesis, indicating we don't have Gaussian white noise. Based on the other tests, though, I think we're ok. Once again, I'm resisting the temptation to go back and more terms to make it "perfect", even though it looks like we could improve the model. Similar results for the other ARMA models considered above.

```
test(Resid(mink,M=c("log","diff",9,"trend",1), a=bestmodel))
```

```
## Null hypothesis: Residuals are iid noise.
```

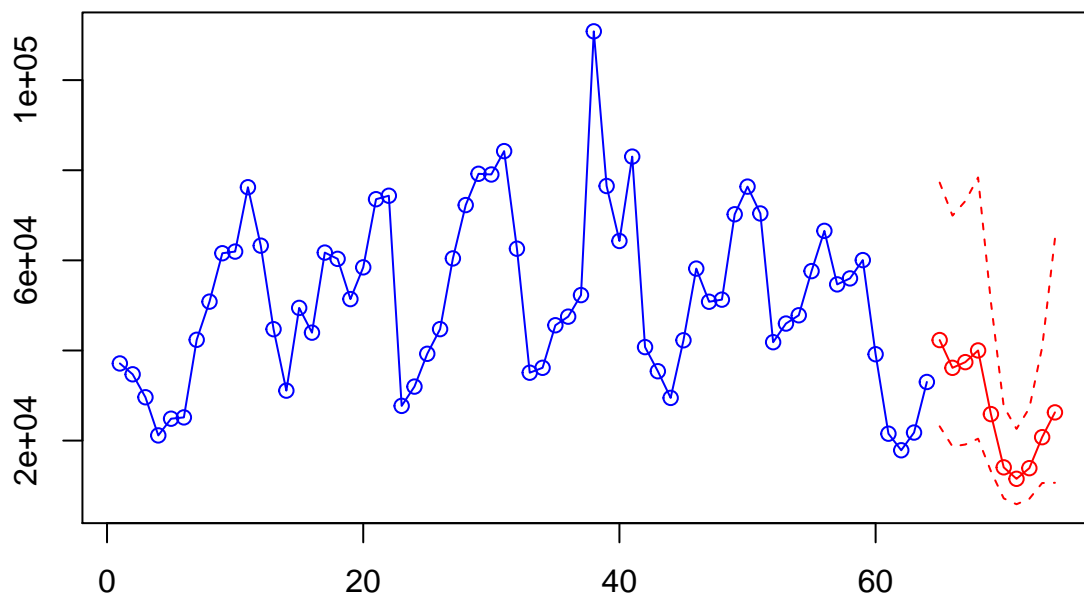
## Test	Distribution	Statistic	p-value
## Ljung-Box Q	Q ~ chisq(20)	24.12	0.237
## McLeod-Li Q	Q ~ chisq(20)	35.81	0.0162 *
## Turning points T	(T-35.3)/3.1 ~ N(0,1)	36	0.8284
## Diff signs S	(S-27)/2.2 ~ N(0,1)	29	0.3545
## Rank P	(P-742.5)/68.9 ~ N(0,1)	727	0.8219



Finally, a forecast of our model.

```
forecast(mink,M=c("log","diff",9,"trend",1), a=bestmodel, h=10)
```

##	Step	Prediction	Lower Bound	Upper Bound
##	1	42312.77	23171.17	77267.17
##	2	36155.09	18694.16	69925.09
##	3	37411.64	19134.43	73147.23
##	4	40012.81	20420.78	78401.75
##	5	25876.23	13200.42	50724.09
##	6	14068.6	7176.294	27580.44
##	7	11523.87	5878.147	22592.09
##	8	13881.81	7080.868	27214.82
##	9	20758.02	10588.3	40695.43
##	10	26262.21	10640.1	64821.13



What would it have looked like if we had done the $d=14$ differencing as considered above? The model fits better, but the prediction/interval is awful for the best ARMA model:

```
ma1model <- autofit(Resid(mink,M=c("log","diff",9,"trend",1,"diff",14)), p=0:5, q=0:10)
forecast(mink, M=c("log","diff",9,"trend",1,"diff",14), a=ma1model, h=10)
```

##	Step	Prediction	Lower Bound	Upper Bound
##	1	85045.77	36361.57	198912.8
##	2	51713.31	20309.99	131672.4
##	3	72454.47	27980.09	187620.9
##	4	57300.62	22050.71	148900.5
##	5	32972.14	12679.2	85743.72
##	6	24050.21	9246.915	62551.96
##	7	16260.2	6251.58	42292.36
##	8	14849.4	5709.13	38623.17
##	9	22195.02	8533.278	57729.14
##	10	40461.47	11256.09	145444

