Tiffany Chen

Un Leong

Soloman Wong

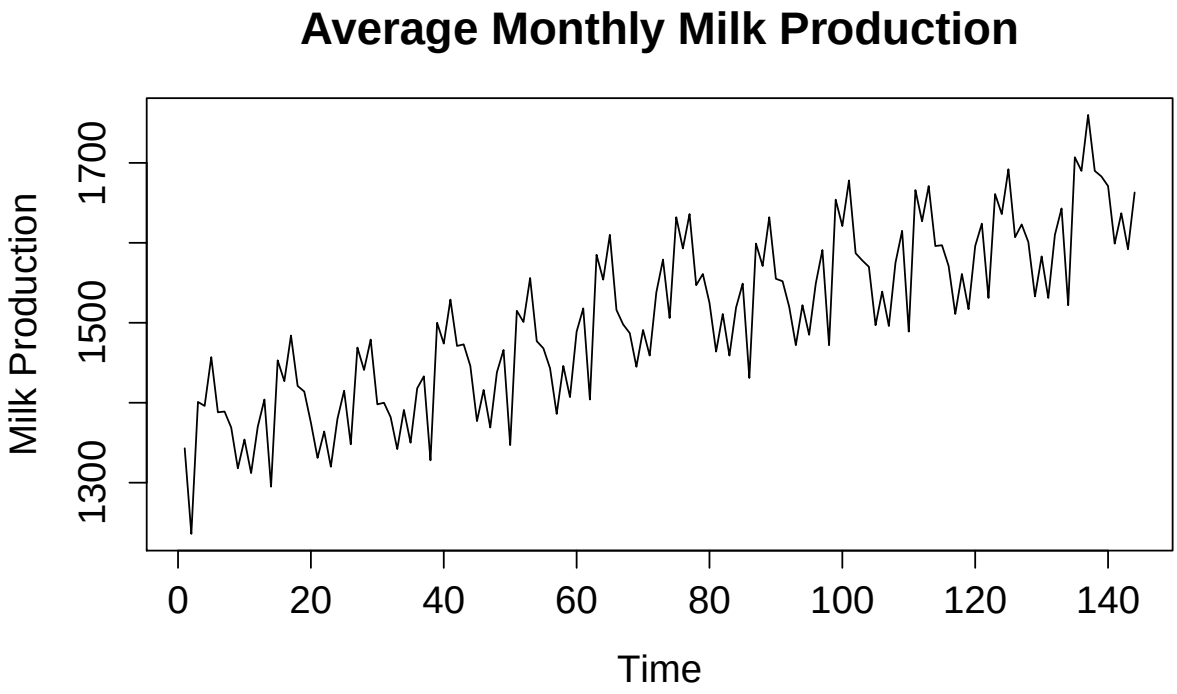
Section A02

March 8, 2016

STA 137 Project

I. Description of the Data

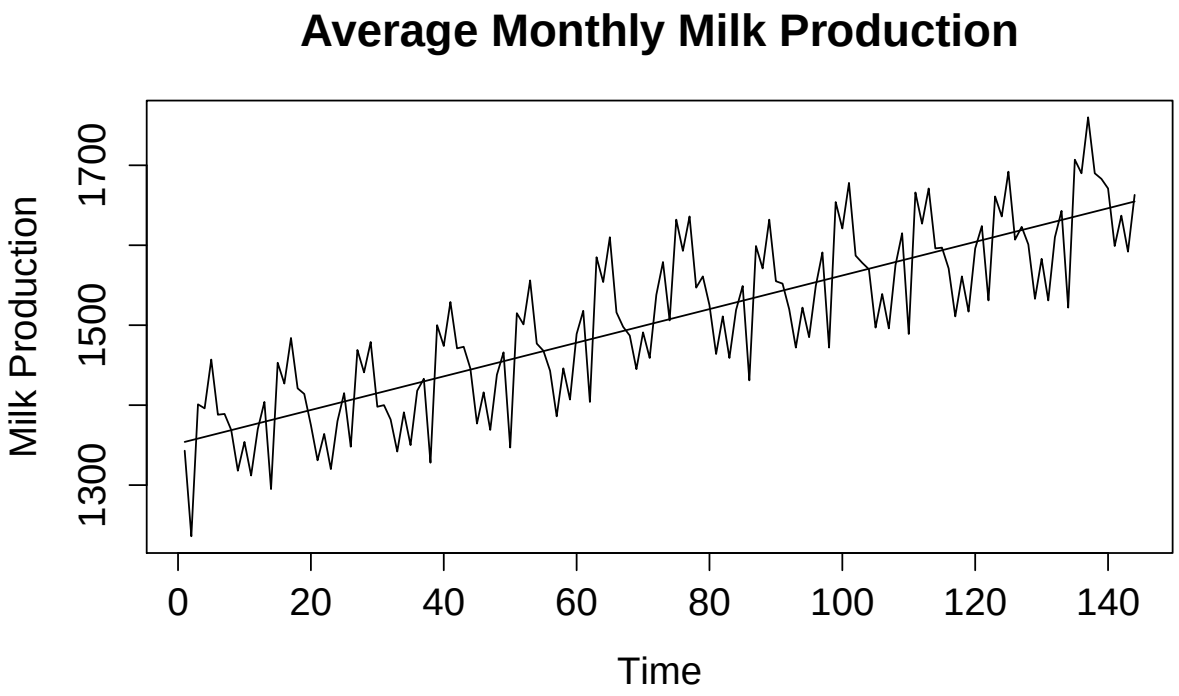
The data that we chose to use is the Milk data set from the data provided by Professor Patrick. The Milk data set is from the TSA library in R which consists of the average monthly milk production per cow in the United States from January 1994 to December 2005. The plot of the data looks like:



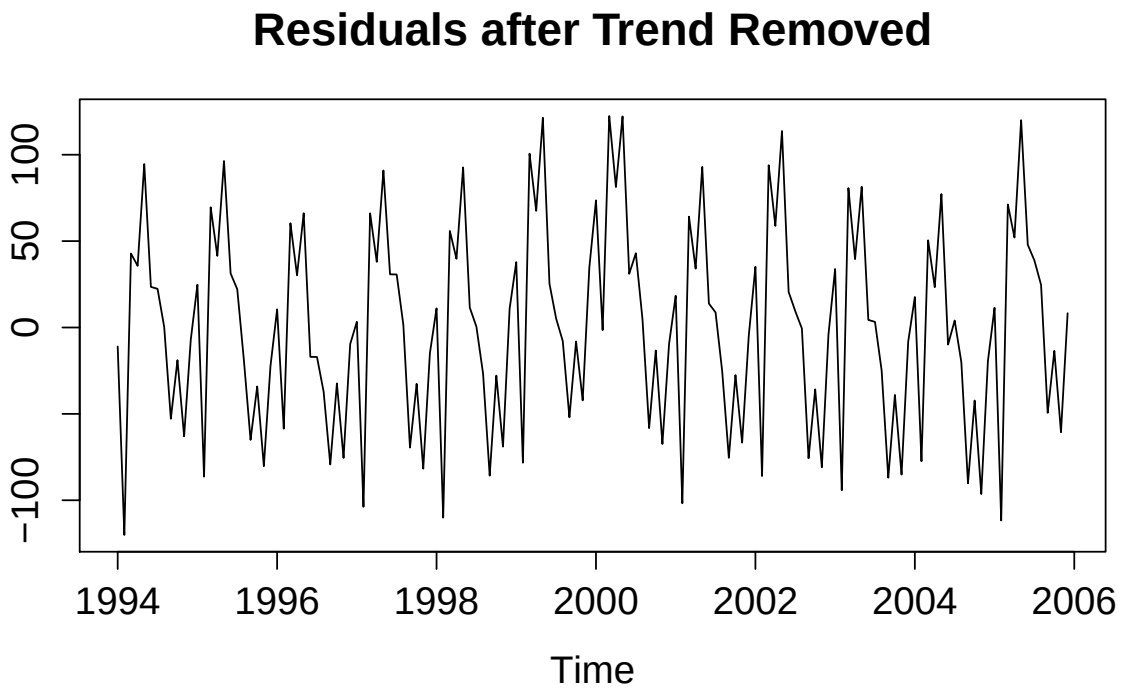
There seems to be a positive linear trend and a seasonal component where the height of milk production is during the summer and the low period is during the winter. There does not seem to be any obvious outliers or the need to transform the data.

II. Deterministic Components

There seems to be a linear positive trend and a seasonal component in the data. We first removed the trend with a polynomial of order one. We checked to see whether an order one polynomial is good enough by running a summary on the trend. Both the intercept and t are significant so we know that order one polynomial is good enough. The plot of the data with the trend fitted on is:

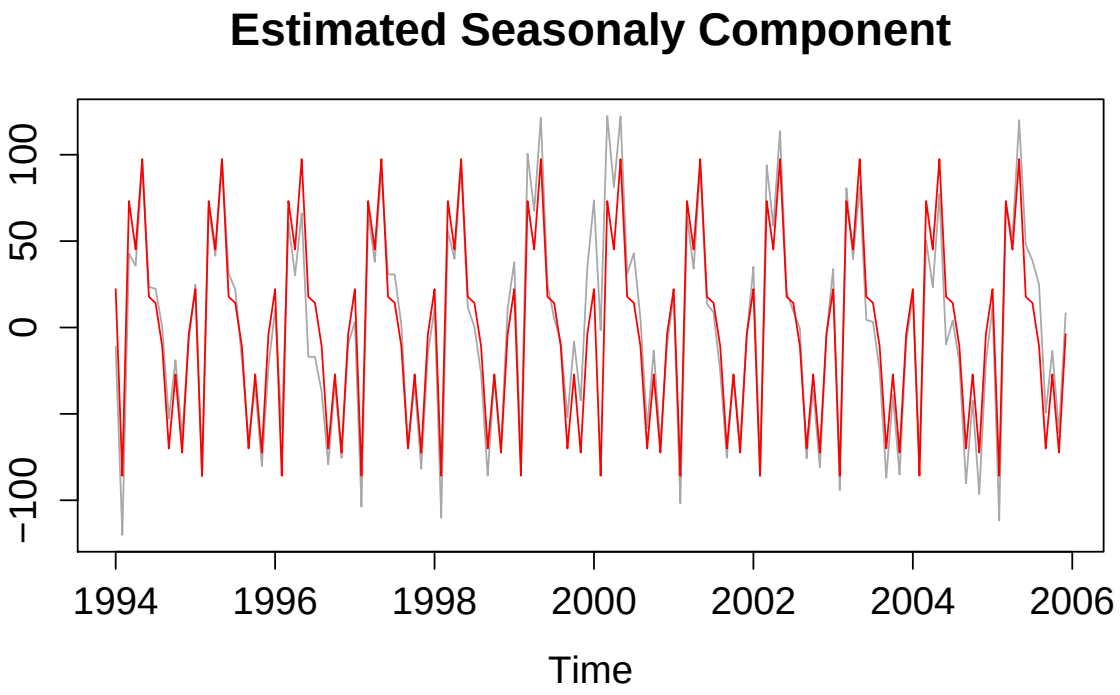


The plot of the residuals after the trend is removed by a polynomial of order one is:

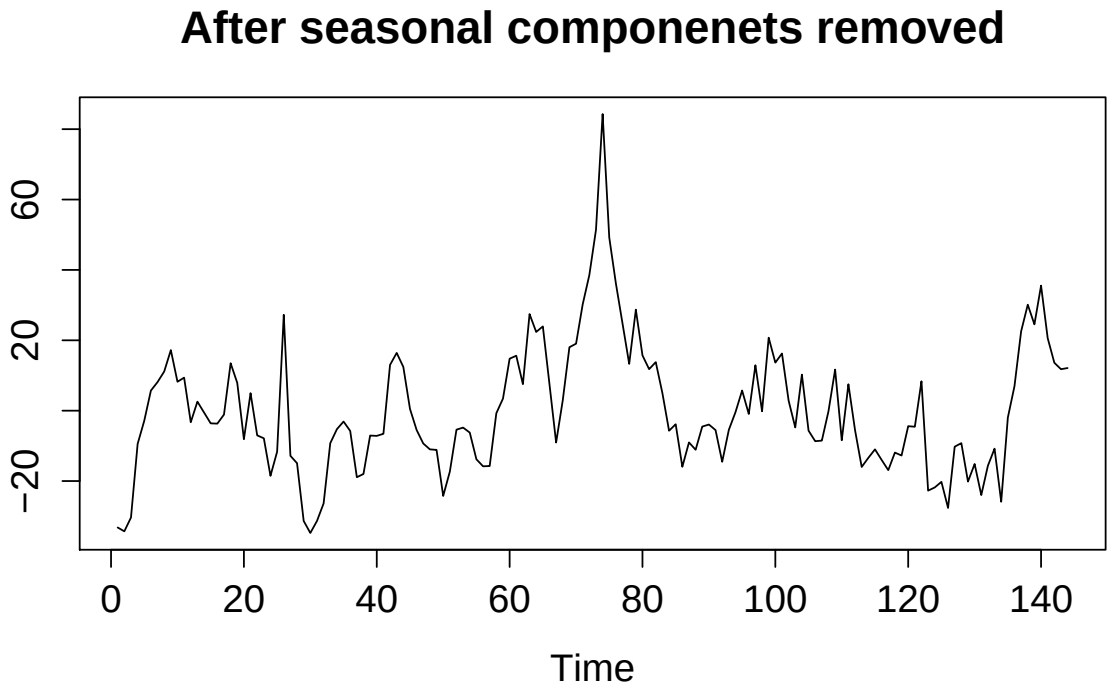


The residuals after the trend has been removed looks fairly stationary. The mean and variance are constant over time.

We removed the seasonal component using the sum of harmonics method. We used the residuals left over from removing the trend by the polynomial of order one method. Our data has 144 observations and there are 12 cycles so the d we chose is 12. The estimated seasonal component is:



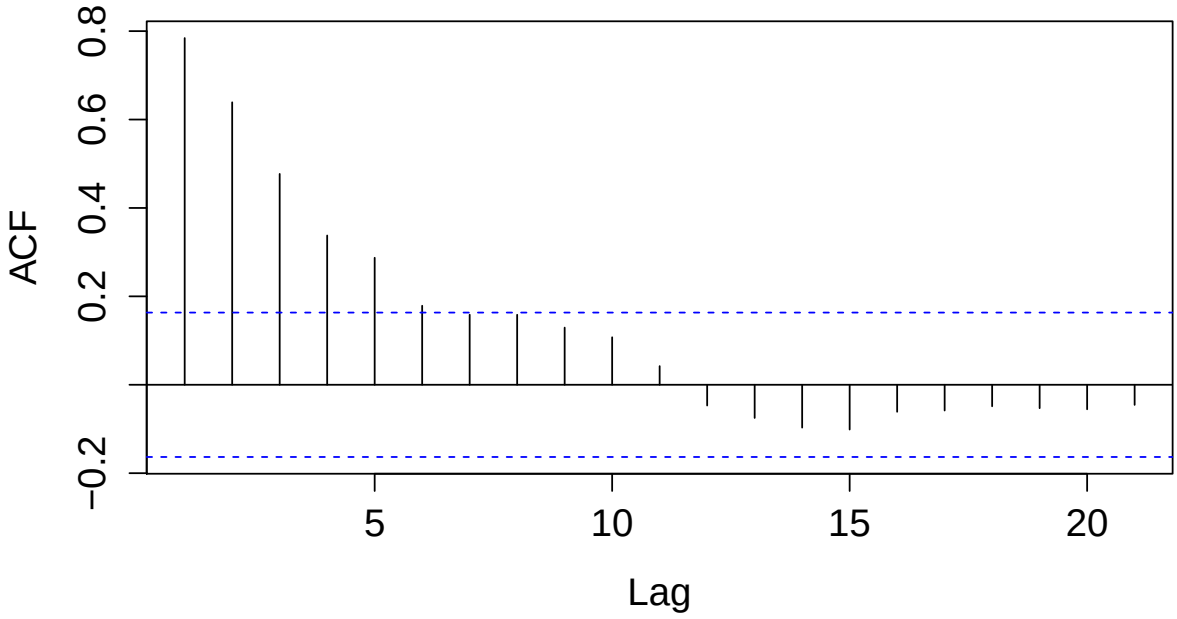
The residuals after the trend has been removed by the polynomial of order one and the seasonal component by the sum of harmonics is:

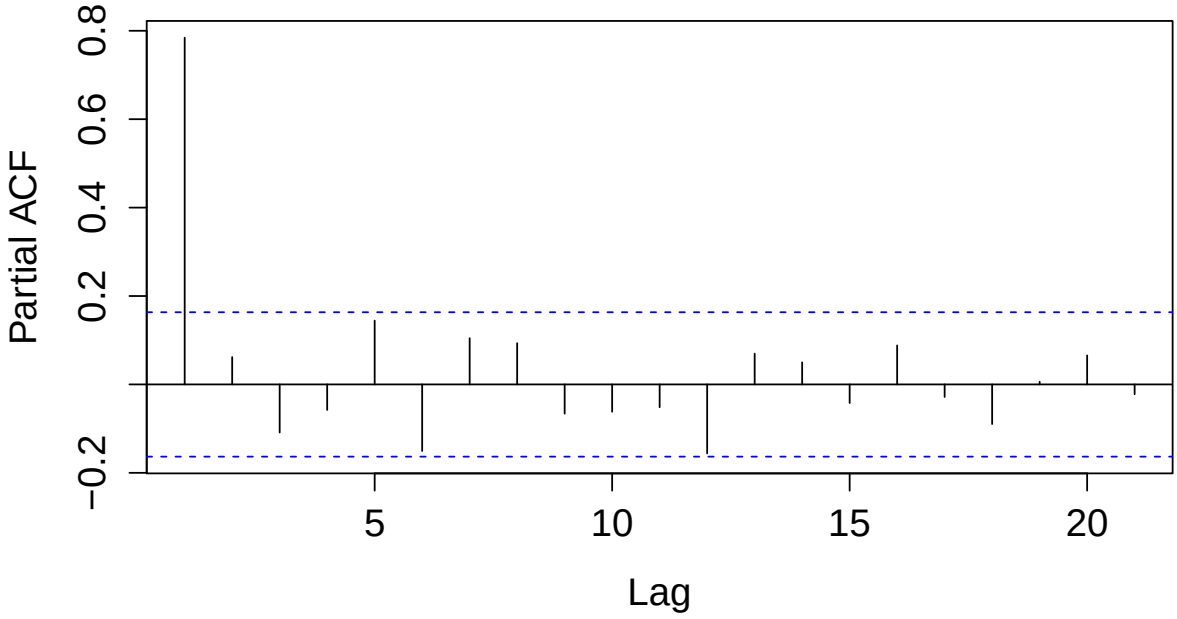


Most of the residuals look like random noise now, however there is a spike at t = 70. These two methods of removing the deterministic part of the data looks good.

Time Series Model

These are the ACF and PACF plots our of residuals



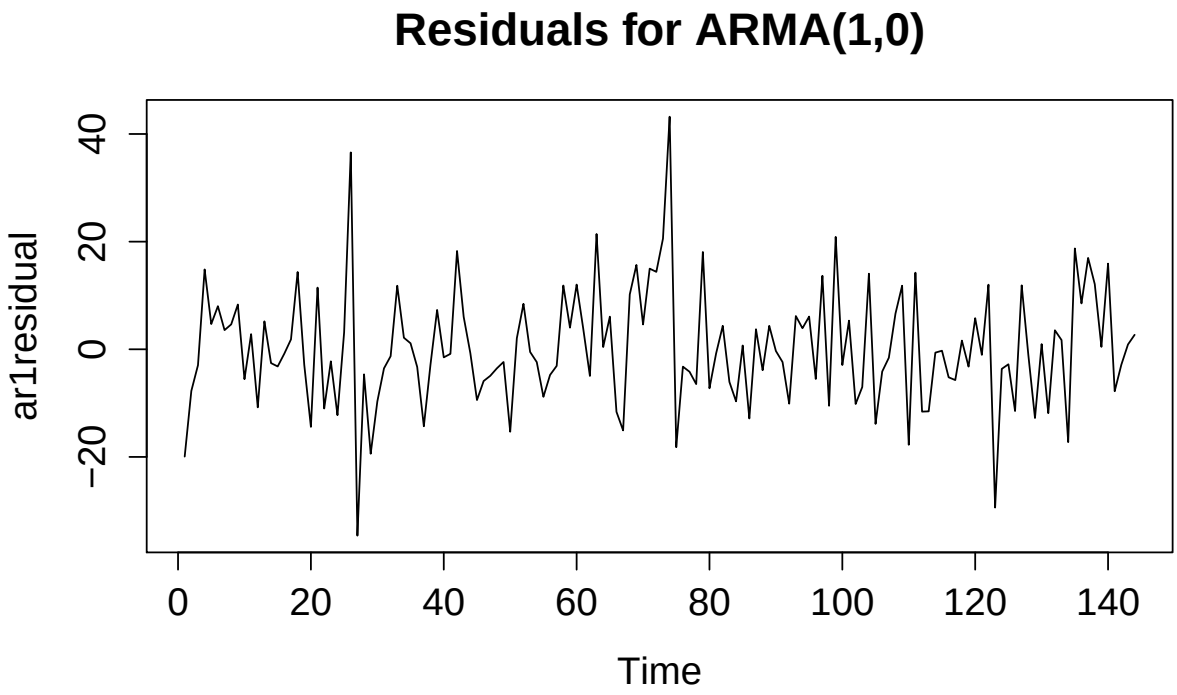


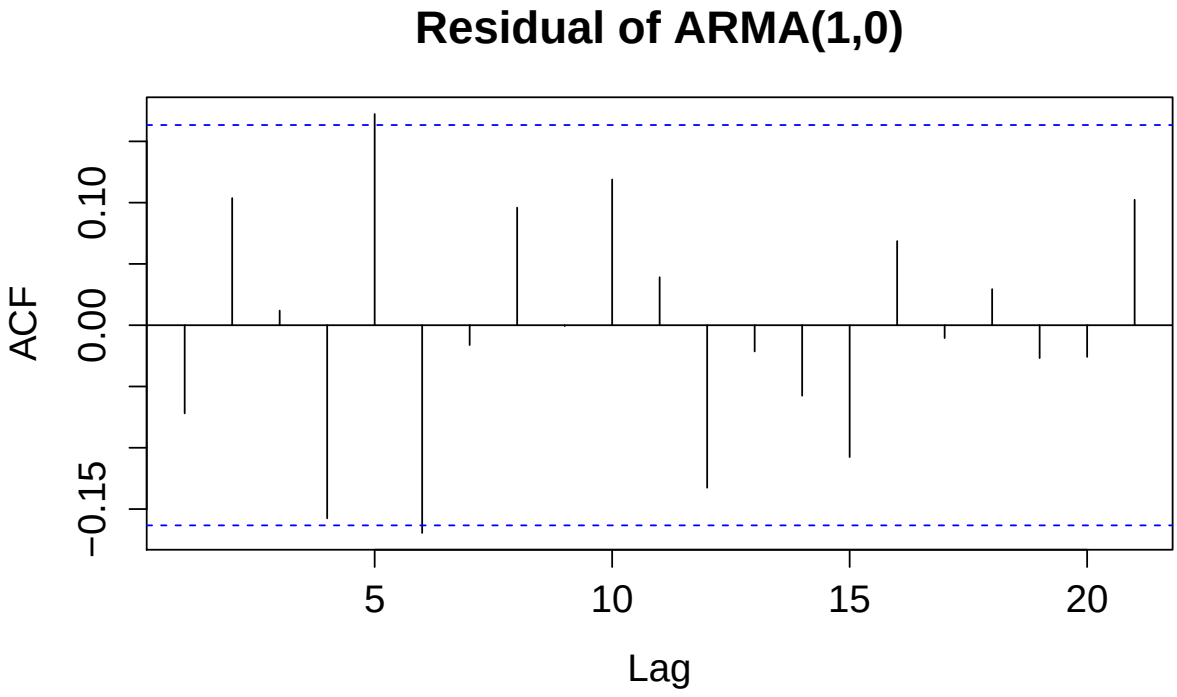
According to the ACF and PACF graphs, a possible model for this dataset is an ARMA(1,0) model since the ACF trials off and the PACF is not significant after 1. We should check the AIC for this model and the

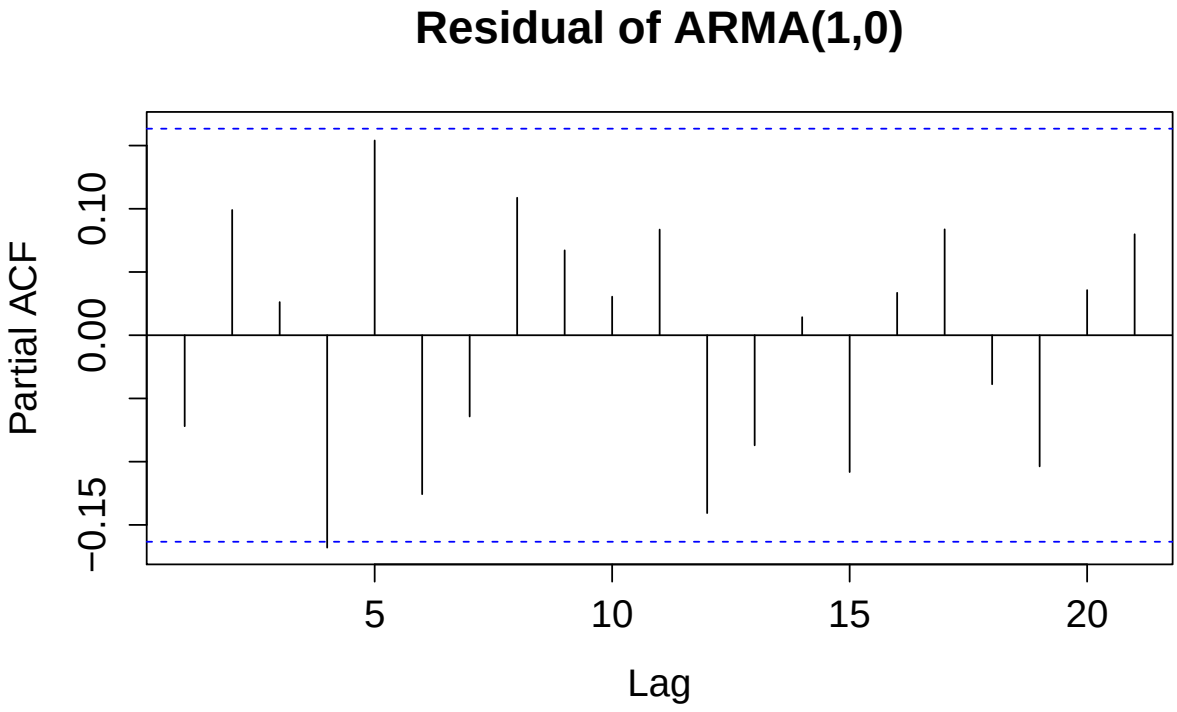
residuals. The AIC for the ARMA(1) model using auto.arima with stepwise = True is 1103.14. However, after using the auto.arima function in R with stepwise = False, the model that it found was an ARMA(3,2) model:

$$X_t - 0.2954X_{t-1} + 0.0817X_{t-2} + 0.6871X_{t-3} = Z_t + 1.0431Z_{t-1} + 0.9271Z_{t-2}$$

We decided to go with the ARMA(1,0) model instead because based on the ACF and PACF plots, an ARMA(1,0) model looks more appropriate. The residuals of the ARMA(1,0), ACF and PACF plots are below:







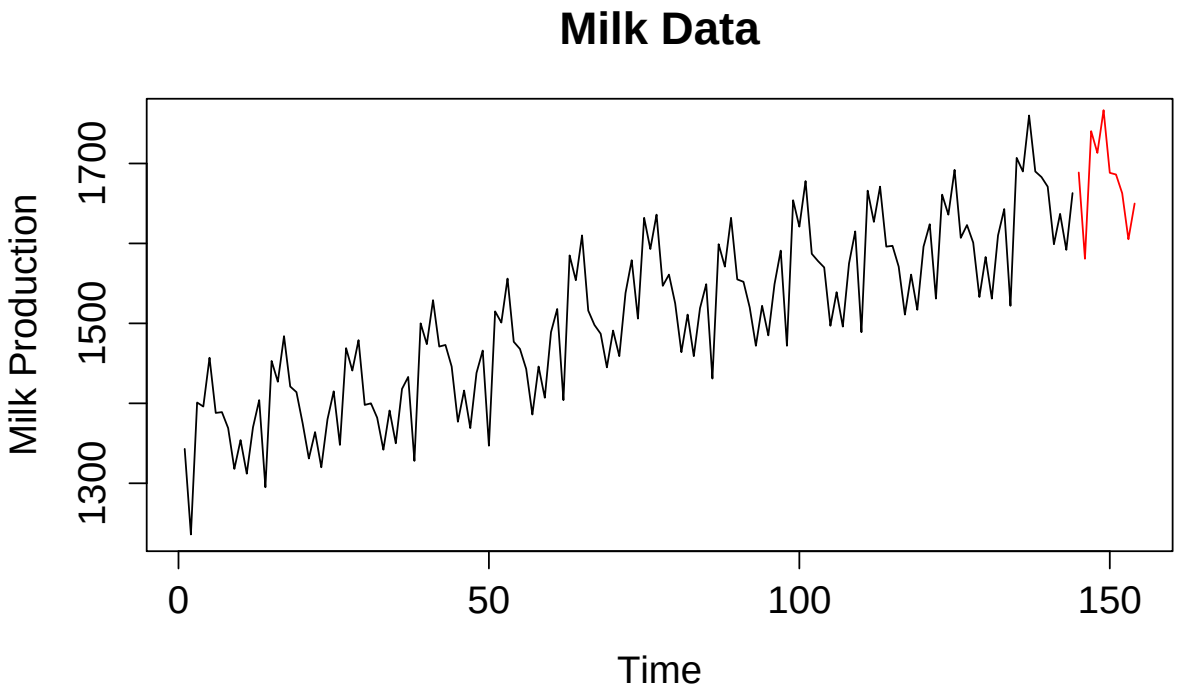
There seems to be some significant lags in both the ACF and PACF plots such as at lag 4 on the ACF plot and lag 3 on the PACF plot so we did the Ljung-Box test. The p-value came out to be 0.09 with lag 24

since there is a seasonal component and the d is 12. Since the p-value is larger than the 0.05 level of significance we are testing against, we fail to reject the null hypothesis and conclude that there is no dependence in the residuals. The plots of the ACF and PACF of the residuals under the ARMA(1,0) model also concludes that there is no dependence structure between them. The final model we will use is the ARMA(1,0) model:

$$X_t + 0.7996X_{t-1} = Z_t$$

Forecast

To forecast the next ten points, we first forecasted the residuals then the seasonal component, and then the trend component. The forecasted next ten points are below:



The forecast for the next ten points are 1688.725, 1580.867, 1740.398, 1713.24, 1766.664, 1688.287, 1686.235, 1662.728, 1605.239, and 1649.914. The forecast for the next ten points look pretty good. The forecast has the same seasonal form as the previous points and the trend is still linear and positive.

Appendix

```
# STA 137
# Project
# milk dataset
# load the library
library(TSA)
# Part 1
# look at its description
?milk
# Average monthly milk production per cow in the US, 01/1994 - 12/2005 data(milk)
x = milk n =length(x)
t = as.vector(1:n) x = as.vector(x)
# plot the data
plot(t,x, type = "l", xlab = "Time", ylab = "Milk Production", main = "Average Monthly Milk
Production")
# doesn't seem to have any obvious outliers # the variances looks somewhat constant
# Part 2
# remove the trend
# have the trend of order 1
trend.fit = lm(x t)
# see if an order 1 polynomial is good enough
summary(trend.fit)
# the intercept and the t are both significant so a polynomial of order
# 1 is good enough to remove the trend
# plot the data with the trend line fitted
plot(t, x, type="l", ylab="Milk Production", xlab = "Time", main = "Average Monthly Milk
Production")
lines(t,fitted(trend.fit))
# plot residuals after trend is removed
y = residuals(trend.fit)
plot(t,y, type="l", main = "Residuals after Trend Removed", ylab="", xlab = "Time")
```

```
# the residuals look pretty stationary
# remove the seasonality component
# try the sum of harmonics method
# y is the residual after the trend has been removed by the polynomial of order 1
plot(t,y, type = "l", main = "Residuals after Trend Removed", ylab = "")
# use t that is in the interval [0,1]
t = 1:length(y)
n = length(t)
t = (t) / length(t)
# d is 12
# make matrix of the harmonics
n.harm = 6 # set to [d/2]
d = 12 # number of time points in each season
harm = matrix(nrow=length(t), ncol=2*n.harm)
for(i in 1:n.harm){
harm[,i*2-1] = sin(n/d * i *2*pi*t)
harm[,i*2] = cos(n/d * i *2*pi*t)
}
colnames(harm)= paste0(c("sin", "cos"), rep(1:n.harm, each = 2))
# fit on all of the sines and cosines
dat = data.frame(y, harm)
fit = lm(y ., data=dat)
summary(fit)
# setup the full model and the model with only an intercept
full = lm(y .,data=dat)
reduced = lm(y 1, data=dat)
# stepwise regression starting with the full model
fit.back = step( full, scope = formula(reduced), direction = "both")
# get back the original t so that we can plot over this range
t = as.vector(time(milk))
# plot the estimated seasonal components
```

```
plot(t,y, type="l", col="darkgrey", ylab="", xlab = "Time", main = "Estimated Seasonaly Component")
lines(t, fitted(fit.back), col="red")
# plot the residuals after seasonal component is removed
ts.plot(residuals(fit.back), main ="After seasonal componenets removed", ylab = "", xlab = "Time")
# Part 3 # Time Series Model
# plot of the residuals without the deterministic parts
ts.plot(residuals(fit.back), main = "Residual after removal of Deterministic Parts" , ylab = "", xlab =
"Time")
# plot the ACF and PACF for the residuals after the trend has been removed
# by the polynomial of order 1 and the seasonality removed by the sum of
# harmonics
acf(residuals(fit.back), main = "")
pacf(residuals(fit.back), main = "")
# from acf and pacf, we should consider the model that best fits
# it is an ARMA(1,0) or an ARMA(1,1)
# use the AIC criteria to test a ARMA(1,0) model
ar1 = arima(residuals(fit.back), order = c(1,0,0), include.mean = FALSE); ar1 # the AIC is 1103.14
# plot the residuals after model is fitted
ar1residual = residuals(ar1)
plot(ar1residual, main = "Residuals for ARMA(1,0)")
# check to see whether the residuals are independent
# plot the acf and pacf of the residuals
acf(ar1residual, main = "Residual of ARMA(1,0)")
pacf(ar1residual, main = "Residual of ARMA(1,0)")
# since both the acf and pacf of the residuals are within the confidence bound,
# we can conclude that the residuals are white noise
# do the Ljung and Box test # the lag is the min(2d,n/5)
Box.test(ar1residual, type = "Ljung-Box", lag = 24)
# p-value is 0.09039 so we fail to reject the null so the residuals are independent
# Forecasting
# forecast next ten points and give the values of the forecast and
```

```
# make an inference on them
# load the forecase package library(forecast)
# forecast for n + 1, n + 2 ,..., n + 10 for the residuals
forc = forecast(ar1, 10)
# plot the forecasted residuals
plot(forc)
# forecast for the seasonal component for the next 10 points
season.f = fitted(fit.back)[1:10]
# forecast for the trend component for the next 10 points
trend.f = predict(trend.fit, newdata = data.frame(t = 145:154))
# put the forecast for the trend, seasonal, and residual together
fc = trend.f + season.f + forc$mean
# plot the forecast for all three parts
plot(fc)
# graph the original data plus the forecasted points in red
ts.plot(x, xlim = c(1,154), main = "Milk Data", ylab = "Milk Production") lines(145:154, fc, col = "red")
```