STA137-PROJECT

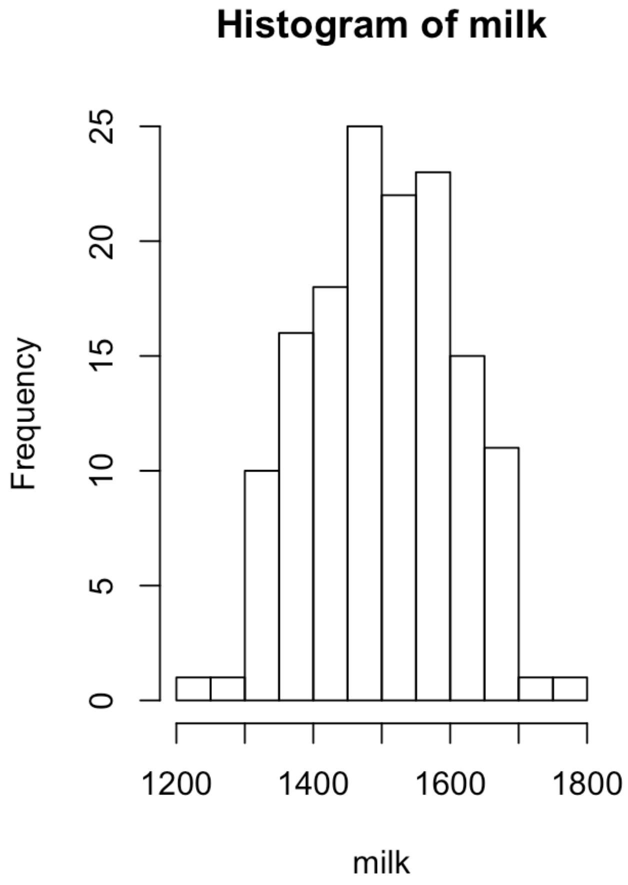
Description of the data

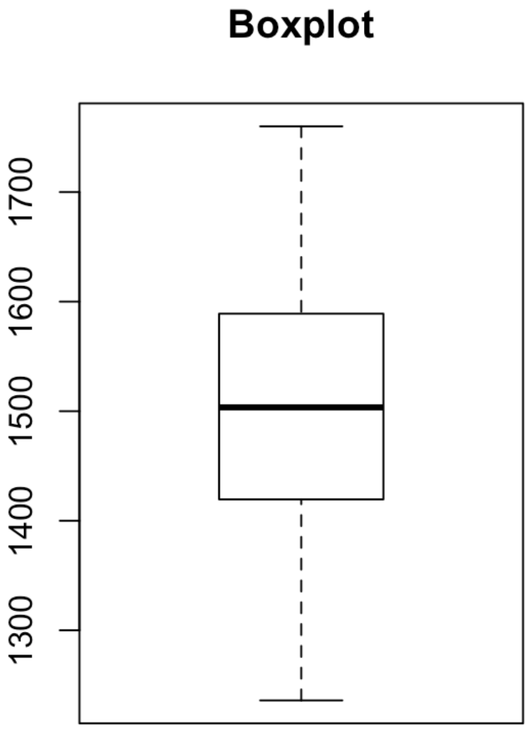
The dataset of milk from Time Series Analysis(TSA) package measures the average monthly milk production per cow in the United States. The data is monthly from January 1994 to December 2005. From the plot of the data, we can see the plot is increasing constantly, which means that there is an apparent linear trend. Since there is no variation goes up gradually, we don’t need transformaton for this dataset. From the histogram and boxplot below, we can also observe that there no outlier. The plot of the data is shown below.

```
require(TSA)
```

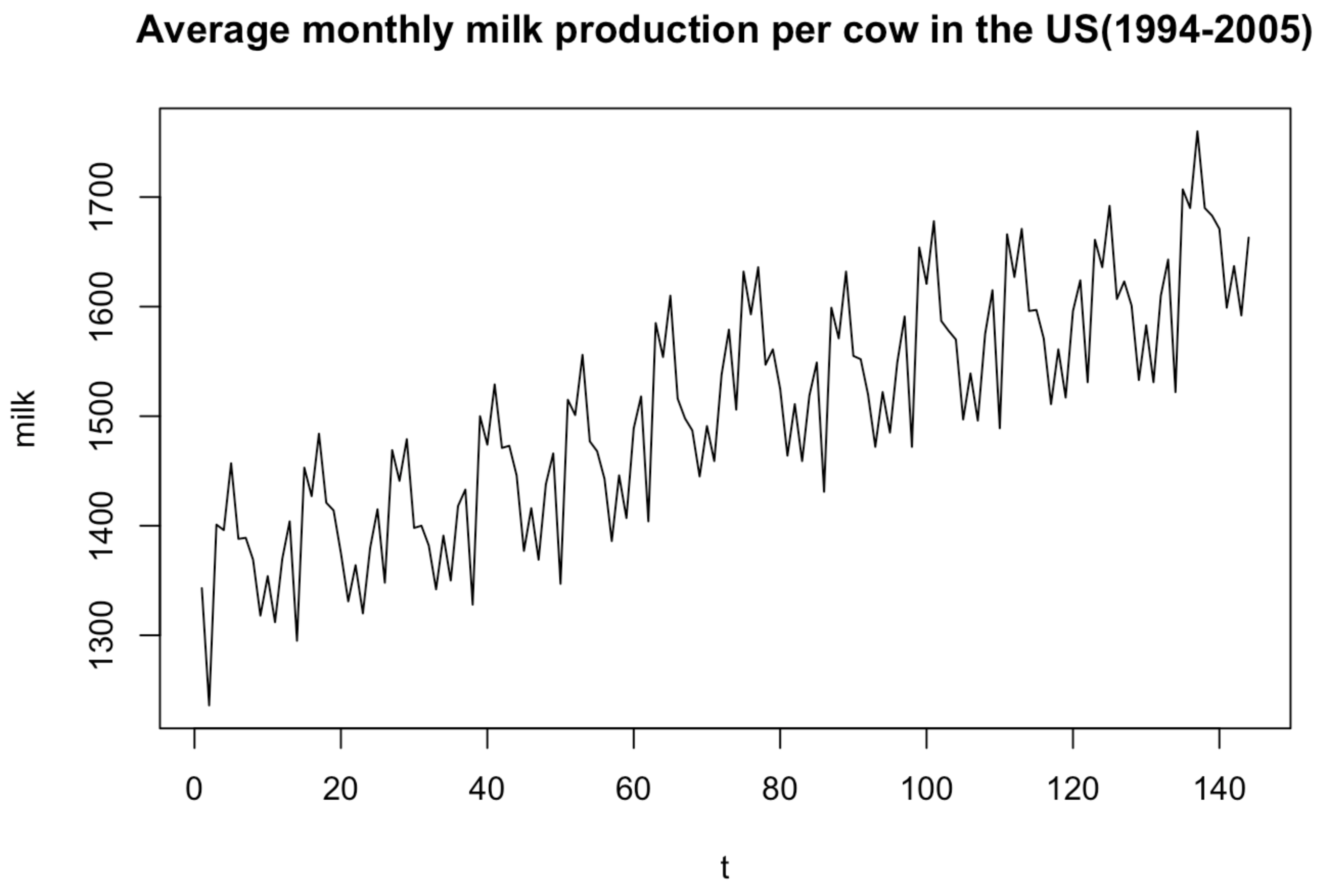
```
## Loading required package: TSA
## Loading required package: leaps
## Loading required package: locfit
## locfit 1.5-9.1    2013-03-22
## Loading required package: mgcv
## Loading required package: nlme
## This is mgcv 1.8-7. For overview type 'help("mgcv-package")'.
## Loading required package: tseries
##
## Attaching package: 'TSA'
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
##
## The following object is masked from 'package:utils':
##
##     tar
```

```
data(milk)

#Histogram and boxplot to check outlier
par(mfrow=c(1,2))
hist(milk)
boxplot(milk, main="Boxplot")
```





```
milk = as.vector(milk)
time = as.vector(time(milk))
n = length(milk)
t = 1:n

#plot of the data
par(mfrow=c(1,1))
plot(t, milk , type="l", main="Average monthly milk production per cow in the U
S(1994-2005)", ylab= "milk")
```



Deterministic components

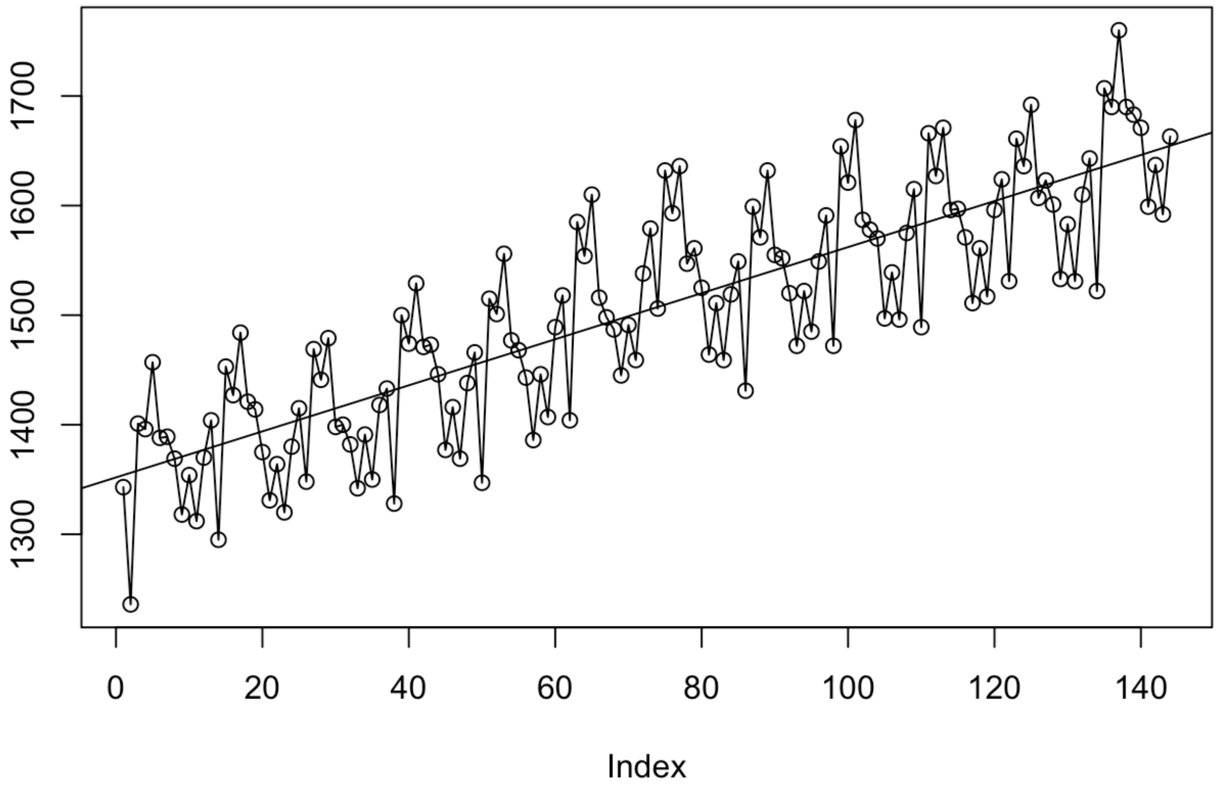
After plotting the data, the data suggests that it is a realization from the model:

Xt = mt + st + Yt ; where
mt is the trend component
st is the seasonal component
and Yt is a noise component

```
#fit an abline
summary(fit <- lm(milk~time(milk))) # regress on time
```

```
##
## Call:
## lm(formula = milk ~ time(milk))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -120.092  -39.831    2.425   37.871  122.388
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1351.8855     9.7711  138.36   <2e-16 ***
## time(milk)     2.1030     0.1169   17.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.32 on 142 degrees of freedom
## Multiple R-squared:  0.695,  Adjusted R-squared:  0.6928
## F-statistic: 323.5 on 1 and 142 DF,  p-value: < 2.2e-16
```

```
plot(milk, type="o", ylab="")
abline(fit)
```

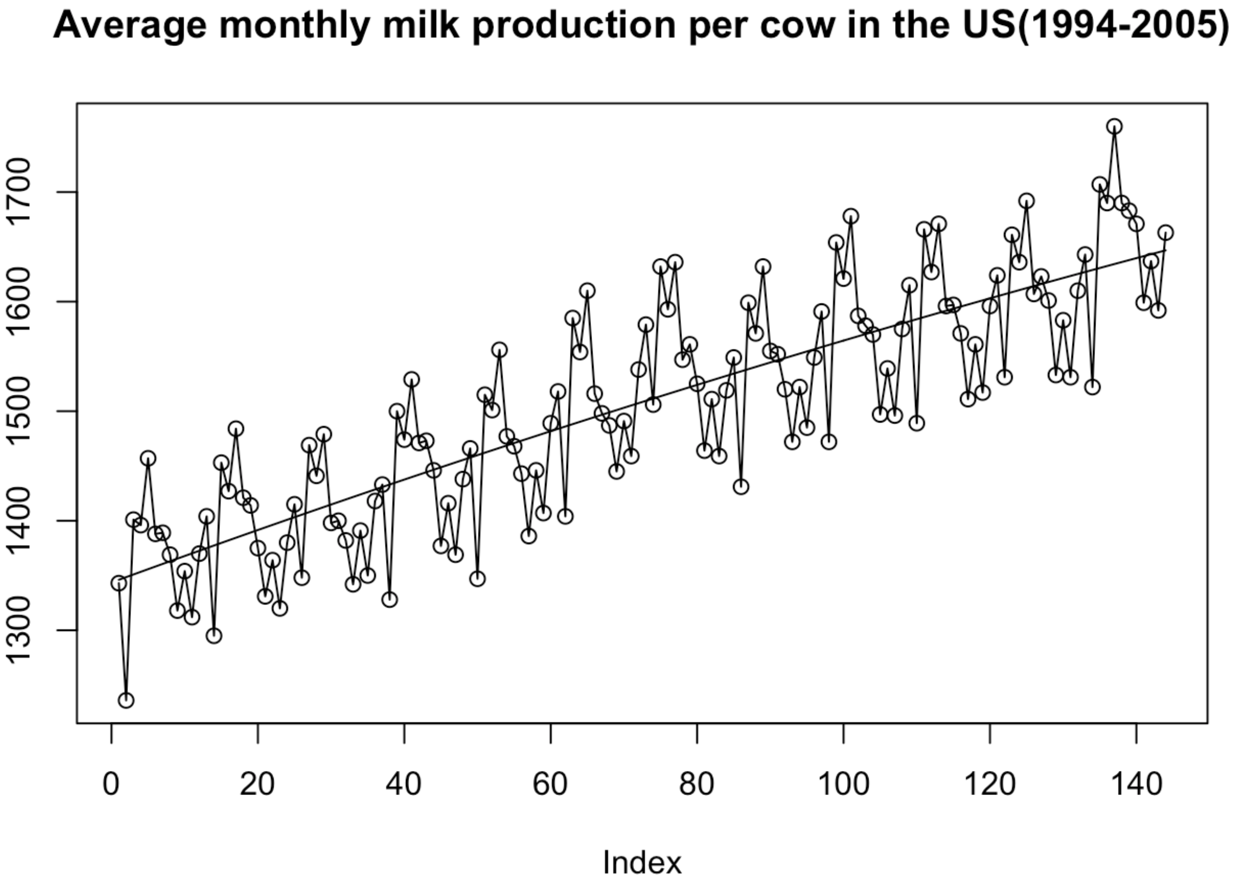


Since the abline gives a good fit, we choose to use the polynomial regression method to remove the trend.

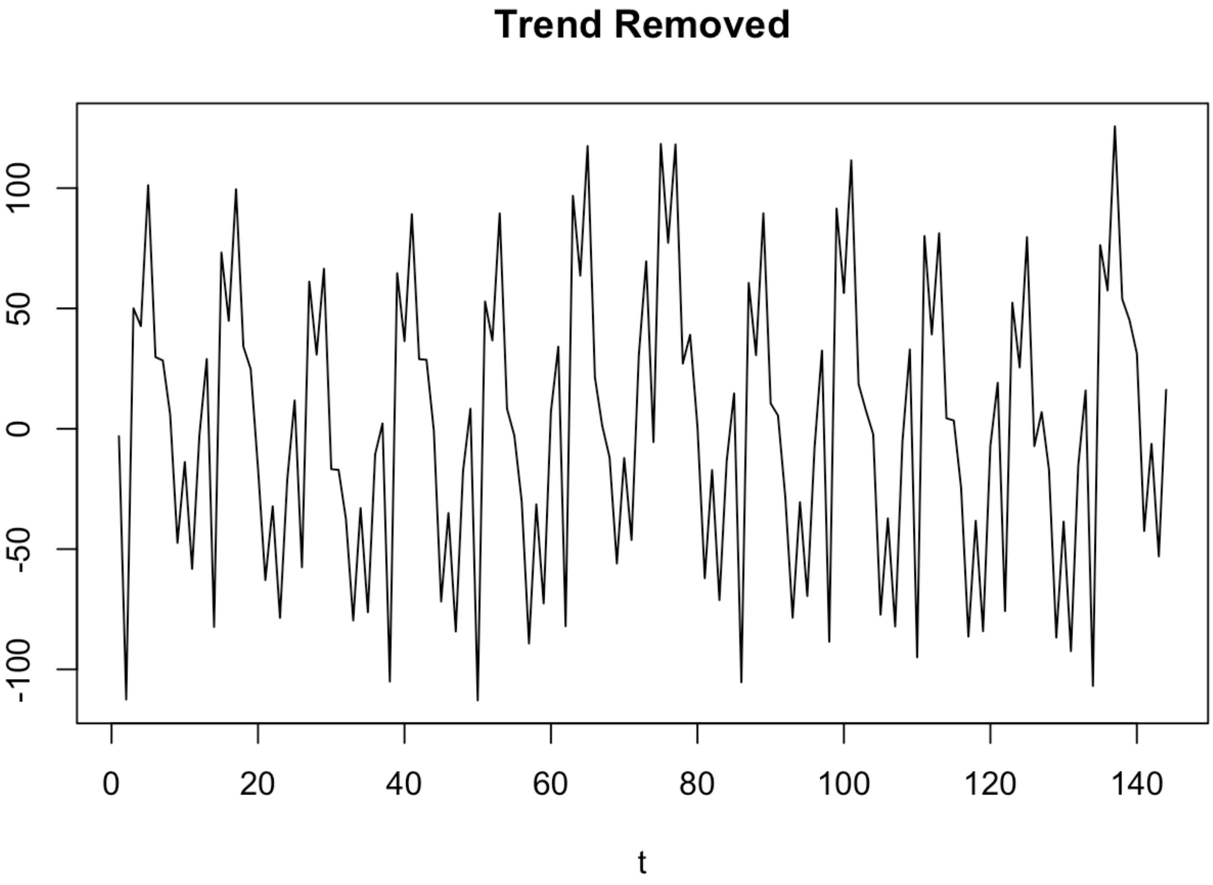
```
#remove trend
t2 = t^2
summary(fit <- lm(milk ~ t+t2 ))
```

```
## 
## Call:
## lm(formula = milk ~ t + t2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -112.890  -38.329    1.108   36.494  125.683 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.344e+03  1.481e+01  90.733  < 2e-16 ***
## t            2.442e+00  4.715e-01   5.178 7.57e-07 ***
## t2          -2.336e-03  3.150e-03  -0.742     0.46    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 58.41 on 141 degrees of freedom
## Multiple R-squared:  0.6962, Adjusted R-squared:  0.6918 
## F-statistic: 161.5 on 2 and 141 DF,  p-value: < 2.2e-16
```

```
#fitting an polynomial regression and plot
plot(milk, type="o", main = "Average monthly milk production per cow in the US
(1994-2005)", ylab="")
yhat = fitted(fit)
lines(as.numeric(t),yhat)
```

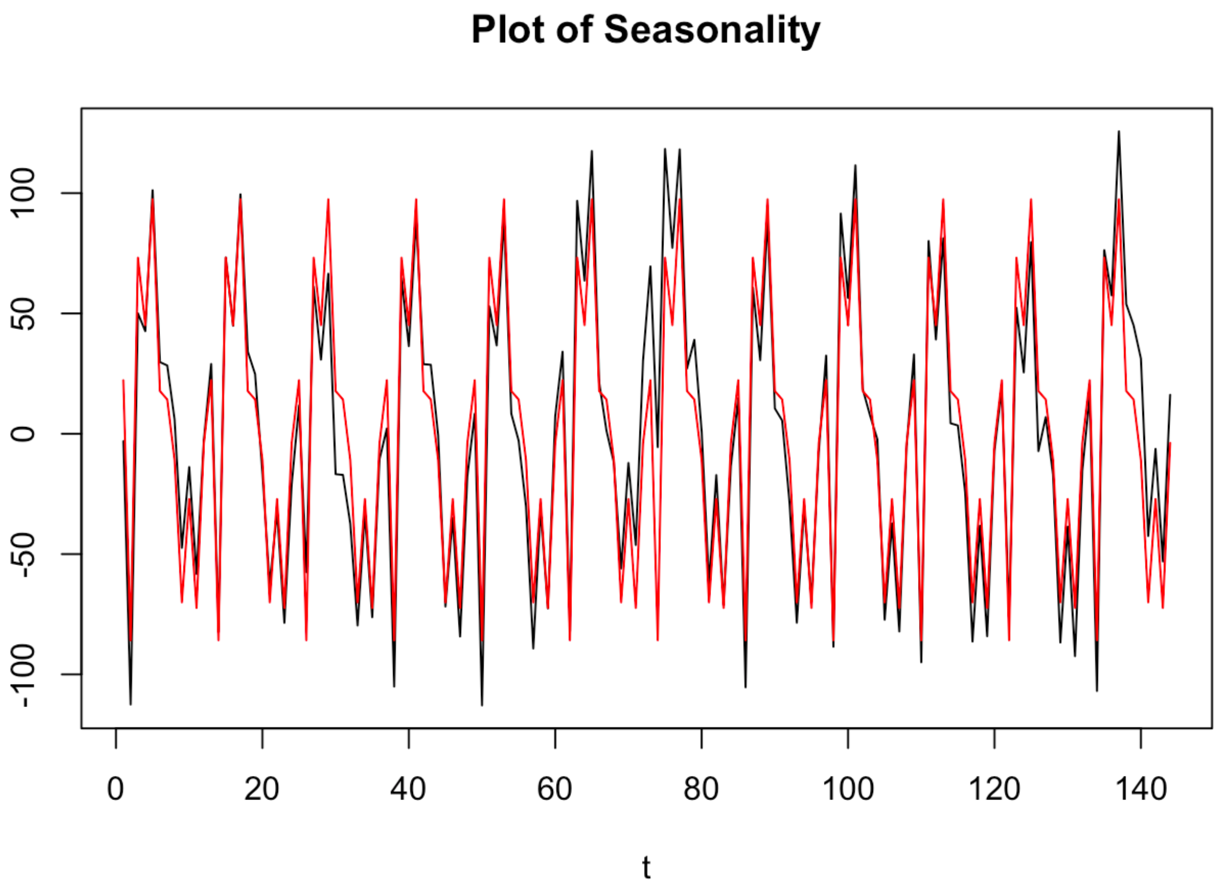


```
# plot residuals after trend is removed
y = residuals(fit)
plot(t,y, type="l", main="Trend Removed", ylab="")
```

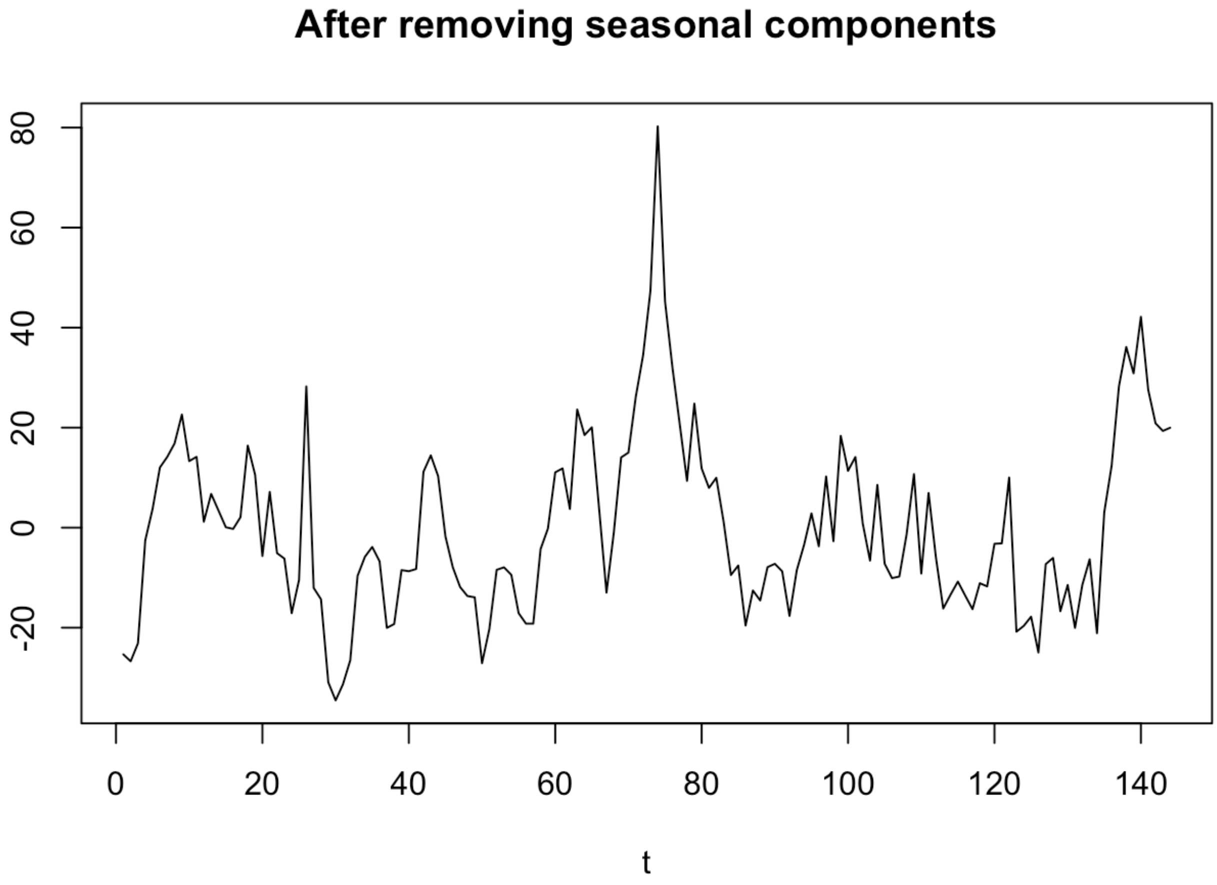


From the residual plot, we can see that there is an apperent seasonal component because the data would increase in March and would peak in May every year. The period d is 12 because the observed data is measured monthly.

```r
#sum of harmonic

n = length(t)
t = 1:length(y)
t = (t) / n

d = 12 #data is monthly
n.harm = 6 #set to [d/2]
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)
fit2 = lm(y~., data=dat)

# 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", trace = F)

#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="black",main = "Plot of Seasonality", ylab="")
lines(t, fitted(fit.back), col="red")
```


Plot of Seasonality

```
#plot the residuals after seasonal component is removed
ts.plot(residuals(fit.back), main="After removing seasonal components", ylab =
"", xlab="t")
```



```
#Check if it's stationary by using adf and kpss tests
kpss.test(residuals(fit.back))
```

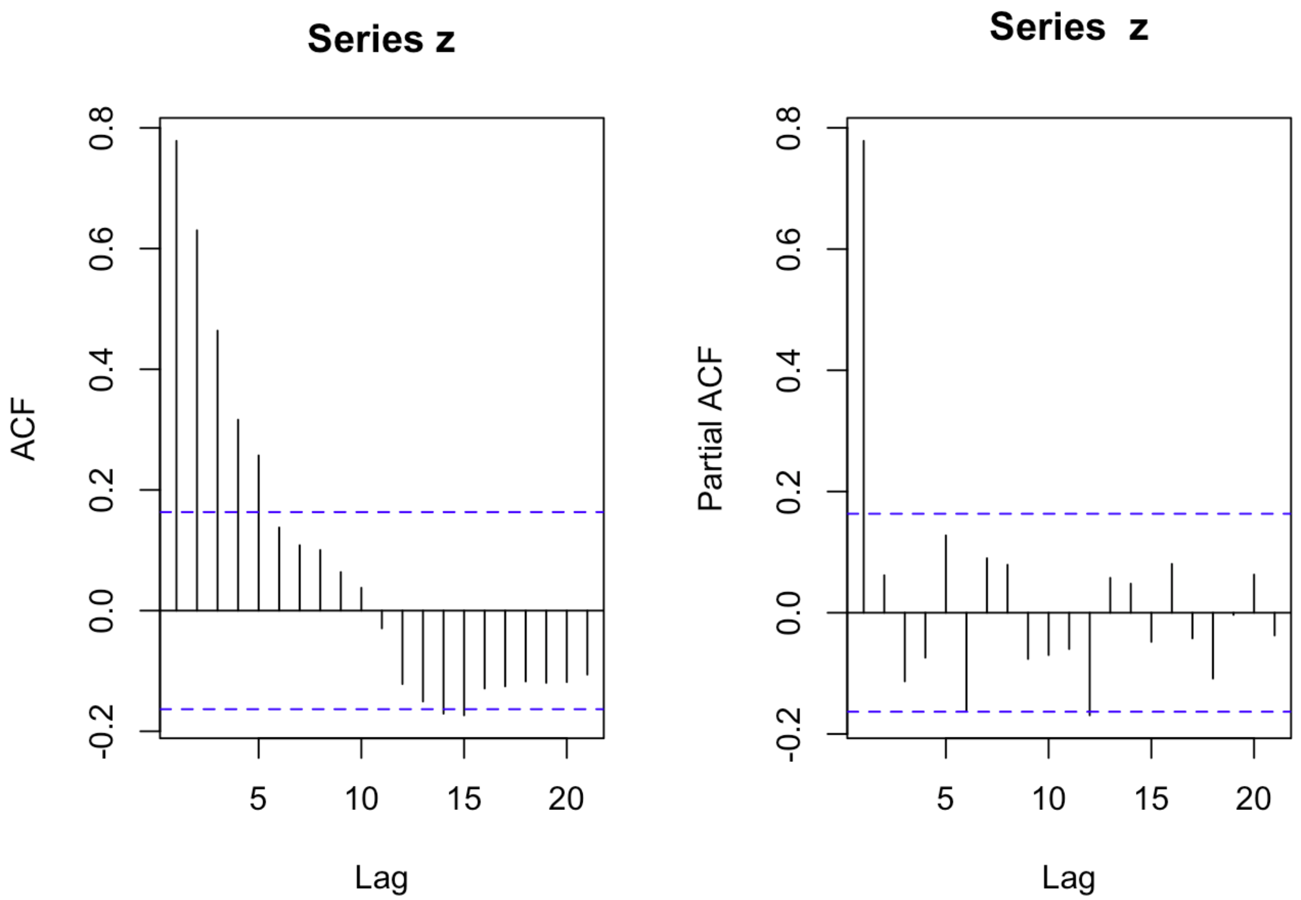
```
## Warning in kpss.test(residuals(fit.back)): p-value greater than printed p-
## value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  residuals(fit.back)
## KPSS Level = 0.18427, Truncation lag parameter = 2, p-value = 0.1
```

we used the sum of harmonic method to remove the seasonality, which appears that it does not have to be any obvious trend left. The residuals look stationary after removing the trend and seasonal components. Then, we double checked in the kpss tests, and the test indicates that the data is stationary.

Time series model

```
#Plot the ACF & PACF
z = residuals(fit.back)
par(mfrow=c(1,2))
acf(z)
pacf(z)
```



The plot of ACF seems trail-off and PACF cuts off at lag 1. Therefore, i believe an AR(1) model may fit well.

```
#Find the best model using an AIC criteria
require(forecast)
```

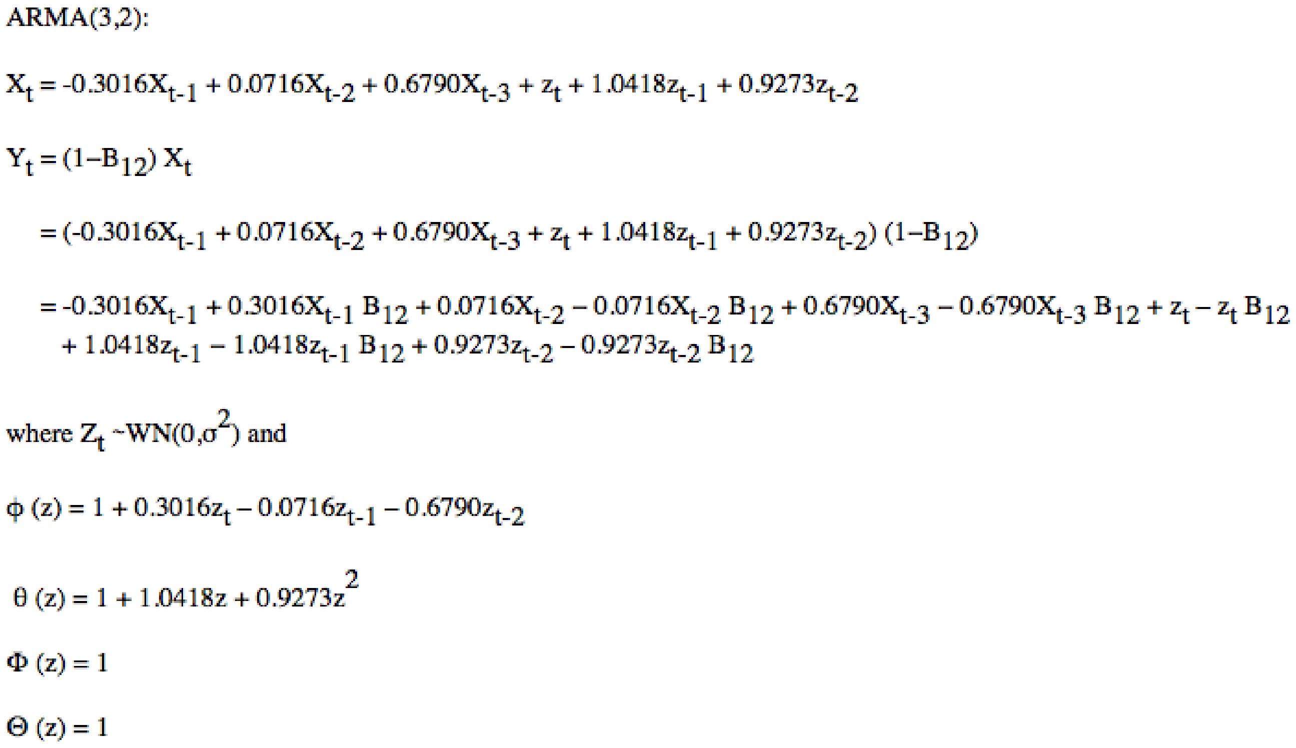
```
## Loading required package: forecast
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
##
## Attaching package: 'timeDate'
##
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
##
## This is forecast 6.2
##
##
## Attaching package: 'forecast'
##
## The following objects are masked from 'package:TSA':
##
##     fitted.Arima, plot.Arima
##
## The following object is masked from 'package:nlme':
##
##     getResponse
```

```
fit3 = auto.arima(z, stepwise=F, trace = F, allowmean = F)

summary(fit3)
```

```
## Series: z
## ARIMA(3,0,2) with zero mean
##
## Coefficients:
##           ar1     ar2     ar3     ma1     ma2
##       -0.3016  0.0716  0.6790  1.0418  0.9273
## s.e.   0.0909  0.0796  0.0803  0.0627  0.0451
##
## sigma^2 estimated as 107.7:  log likelihood=-542.33
## AIC=1096.67   AICc=1097.28   BIC=1114.48
##
## Training set error measures:
##                     ME     RMSE     MAE       MPE     MAPE      MASE
## Training set 0.1835227 10.37809 8.03856 -29.33434 143.1593 0.9201077
##                    ACF1
## Training set 0.04904249
```

By using the auto.arima, we found that the best model is ARIMA(3,0,2) with zero mean.

The expression for ARIMA(3,0,2) is:



ARMA(3,2):

$X_t = -0.3016X_{t-1} + 0.0716X_{t-2} + 0.6790X_{t-3} + z_t + 1.0418z_{t-1} + 0.9273z_{t-2}$

$Y_t = (1-B_{12})\, X_t$

$= (-0.3016X_{t-1} + 0.0716X_{t-2} + 0.6790X_{t-3} + z_t + 1.0418z_{t-1} + 0.9273z_{t-2})\, (1-B_{12})$

$= -0.3016X_{t-1} + 0.3016X_{t-1}\, B_{12} + 0.0716X_{t-2} - 0.0716X_{t-2}\, B_{12} + 0.6790X_{t-3} - 0.6790X_{t-3}\, B_{12} + z_t - z_t\, B_{12}$
$+ 1.0418z_{t-1} - 1.0418z_{t-1}\, B_{12} + 0.9273z_{t-2} - 0.9273z_{t-2}\, B_{12}$

where Z_t ~WN$(0,\sigma^2)$ and

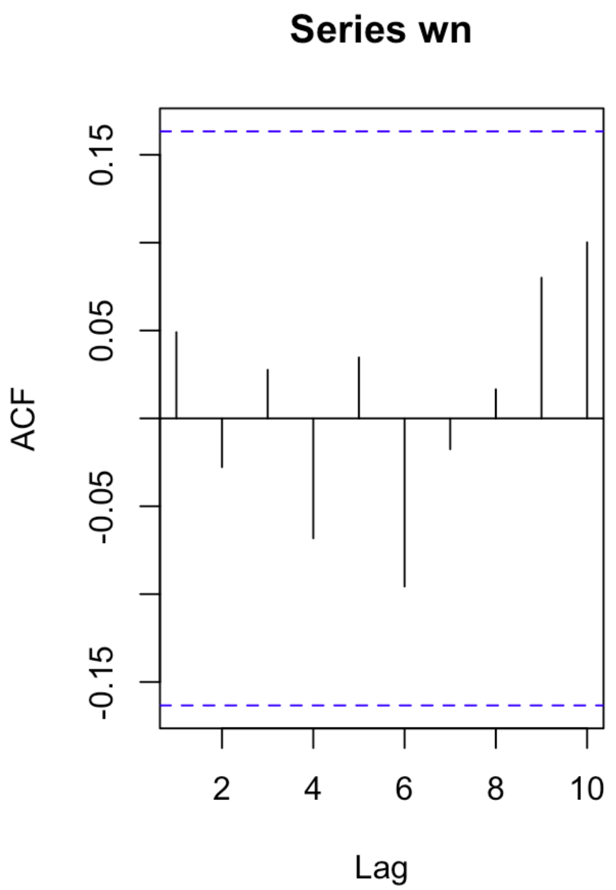
$\phi\,(z) = 1 + 0.3016z_t - 0.0716z_{t-1} - 0.6790z_{t-2}$

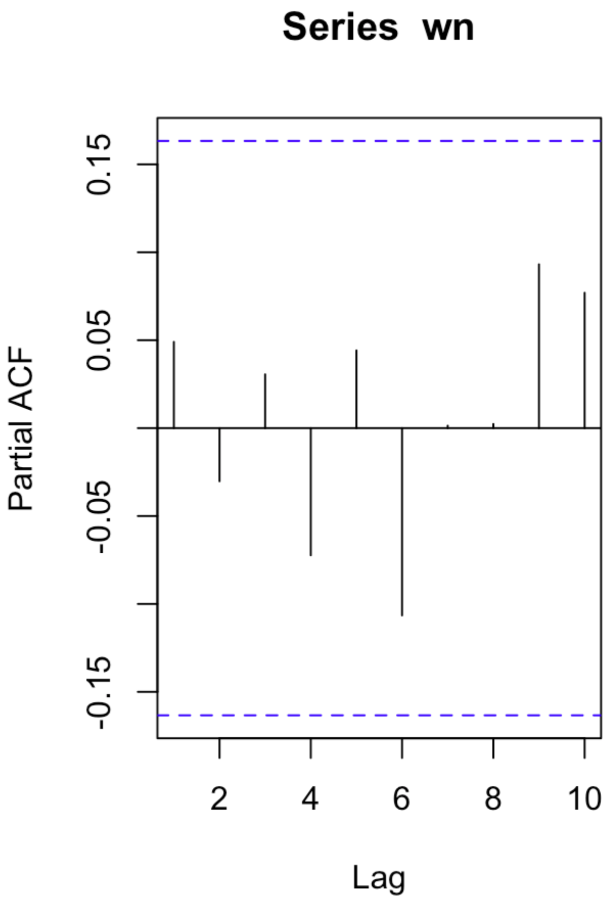
$\theta\,(z) = 1 + 1.0418z + 0.9273z^2$

$\Phi\,(z) = 1$

$\Theta\,(z) = 1$

```
#Plot the residuals after the model is fitted
wn = resid(fit3)
par(mfrow=c(1,2))
acf(wn, lag.max=10)
pacf(wn, lag.max=10)
```





```
#Ljung-Box test for white noise
Box.test(wn, lag = 24, type = "Ljung-Box") #lag=2d
```

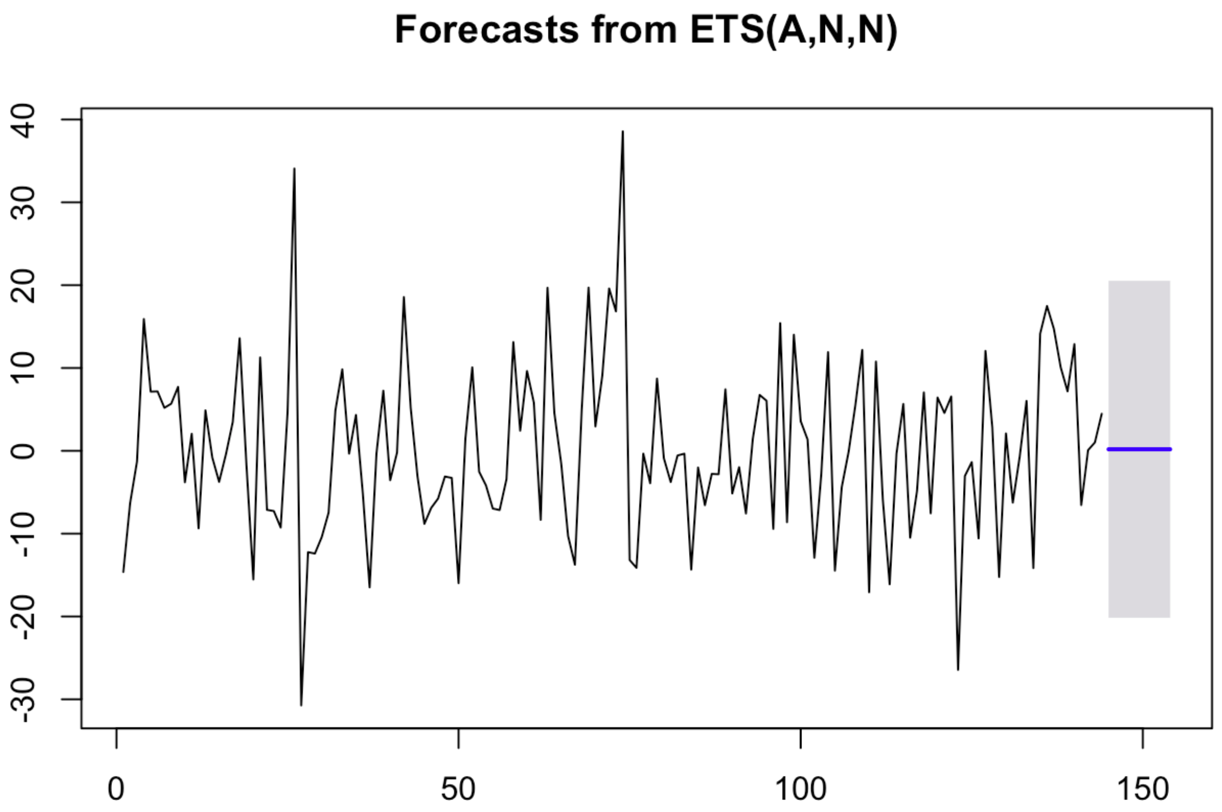
```
##
##  Box-Ljung test
##
## data:  wn
## X-squared = 22.287, df = 24, p-value = 0.5621
```

From the plots, theres no significant lag that are outside of the boundaries. We conducted the Ljung-Box test to get the p-value to determine if there is any dependent structure remaining. From the Box-Ljung test, the p-value is 0.5621.Since the p-value is large, we accept the hypothesis and conclude that there is no dependence structure remaining. It also tells us that the remaining residuals are white noise.

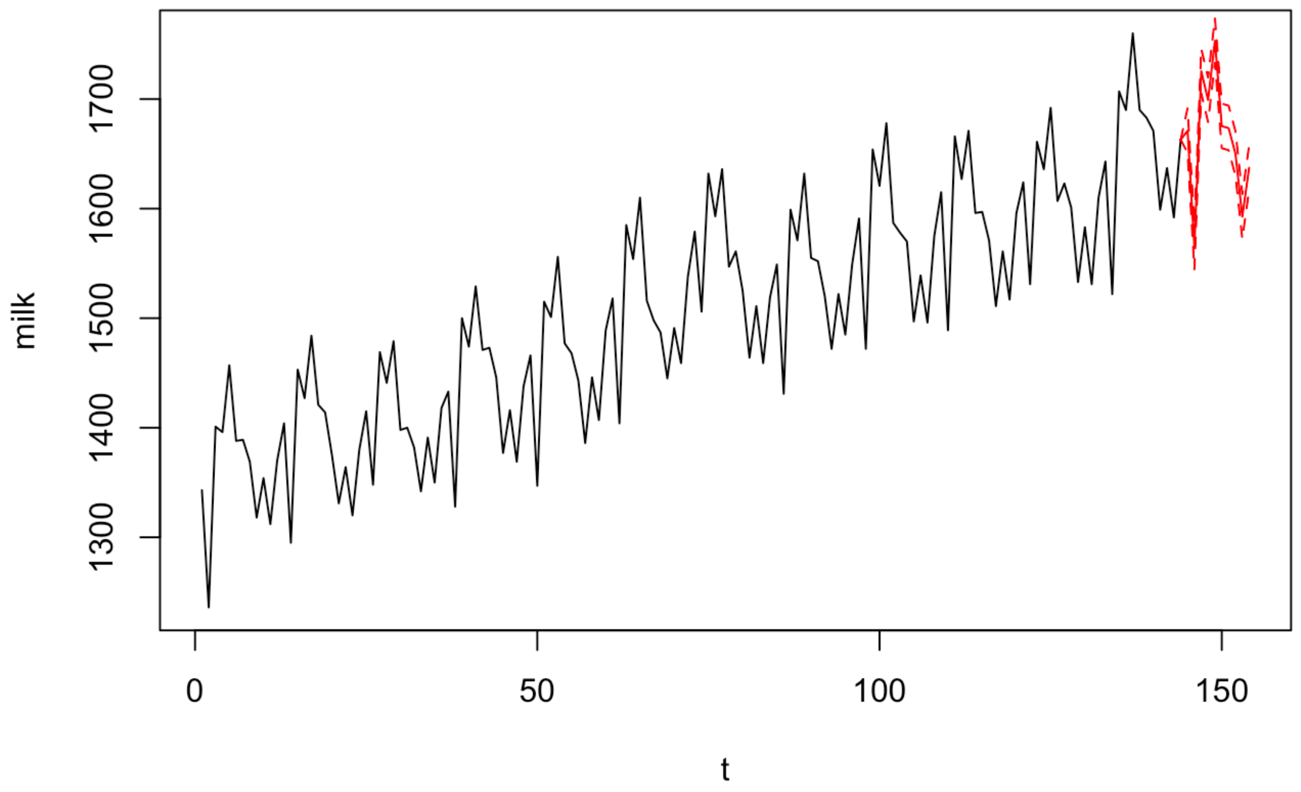
Forecasting

Since Yt(hat) is stationary, then we can utilize its properties along with the trend component and the seasonal component to forecast the future values of Xt.

```
#forecast the noise
fc = forecast(wn, h=10, level = .95)

#plot the noise forecast
plot(fc)
```



```
#forecast the seasonal component and noise
season.fc = fit.back$fitted.values[1:10]+fc$mean

#forecast the trend
trend.fc = predict(fit, newdata = data.frame(t=145:154, t2 = (145:154)^2))

#add the seasonal and noise forecasts
x.hat = season.fc+trend.fc

plot(t, milk, xlim = c(1,154), type="l")
lines(144:154, c(milk[144], x.hat), col="red")
#add the forecast intervals
lines(144:154, c(milk[144], x.hat+fc$lower), col="red", lty=2)
lines(144:154, c(milk[144], x.hat+fc$upper), col="red", lty=2)

#the actual values
lines(144:154, milk[144:154], col="blue")
```



The value of the forecasts are:

```
x.hat
```

```
## Time Series:
## Start = 145
## End = 154
## Frequency = 1
##        1        2        3        4        5        6        7        8
## 1671.022 1564.741 1725.460 1699.179 1753.231 1675.284 1673.503 1650.138
##        9       10
## 1592.690 1637.326
```

We think that the forecast values are accurate because all of the values fall within the boxplot we ploted in the beginning of the report. And the values in March and May also shows the seasonal trend as what we observed before.