“Time series analysis”

author: “Uranie and Appanna” date: “10/06/2021”

I. Non seasonal data (vanilla)

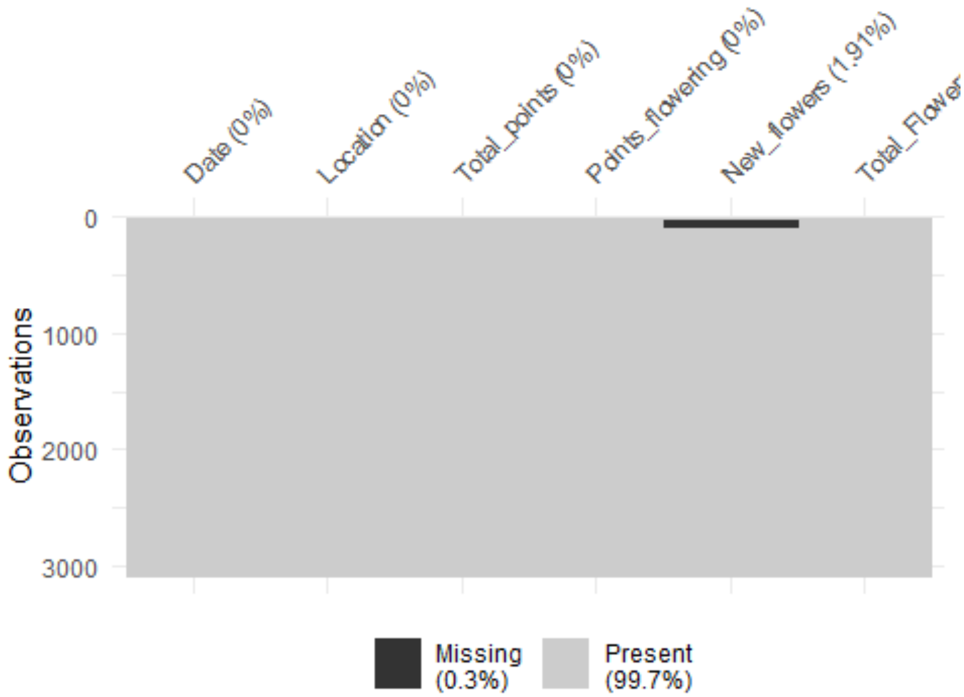
The dataset contains information about the number of new flowers which are flowering at different locations within the vanilla cultivation, for the different dates. We will attempt to model the data so that we can make predictions about the expected total number of new flowers per day. The data describes the number of new flowers for only 1 flowering season.

1. Characterisation of the data

```
##        Date Location Total_points Points_flowering New_flowers
Total_Flowers
## 1 12 Mar 20    3*2ag            8                            0
## 2 12 Mar 20     3*3g            2                           NA
## 3 12 Mar 20     4*3a            9                            1
## 4 12 Mar 20     4*2a            8                           NA
## 5 12 Mar 20     5*2a            1                           NA
## 6 12 Mar 20    5*2ag            3                           NA
```

Structure of the data

```
## Rows: 3,091
## Columns: 6
## $ Date             <fct> 12 Mar 20, 12 Mar 20, 12 Mar 20, 12 Mar 20, 12
Mar...
## $ Location         <fct> 3*2ag, 3*3g, 4*3a, 4*2a, 5*2a, 5*2ag, 4*3a2g,
5*4a...
## $ Total_points     <fct> 8, 2, 9, 8, 1, 3, 10, 9, 5, 1, 5, 5, 1, 6, 2, 3,
2...
## $ Points_flowering <fct> , , , , , , , , , , , , , , , , , , , , , , , , , ,
## $ New_flowers      <dbl> 0, NA, 1, NA, NA, NA, NA, 1, NA, NA, NA, NA, NA,
N...
## $ Total_Flowers    <fct> , , , , , , , , , , , , , , , , , , , , , , , , , ,
```



Data cleaning

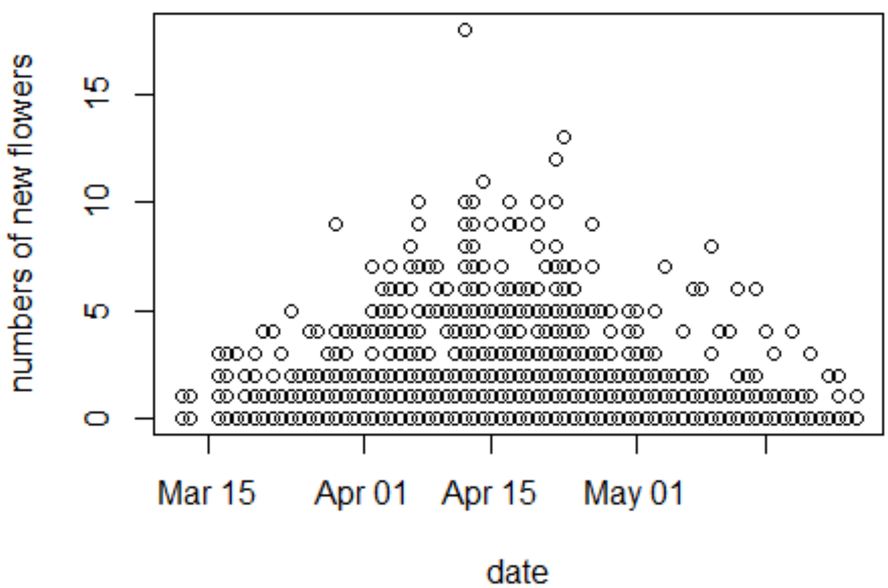
The dates were transformed into "date" format which first were in "character" format. Only the 'date' and 'new_flowers' column need to be considered for the time series analysis. The missing values in the 'new_flowers' column were due to the fact that there were no flowers at that 'location' for that day. These missing values were replaced with 0.

```
## [1] "data.frame"

##     new_date      date location total_points points_flowering new_flowers
## 1 2020-03-12 12 Mar 20    3*2ag            8                           0
## 2 2020-03-12 12 Mar 20     3*3g            2                           0
## 3 2020-03-12 12 Mar 20     4*3a            9                           1
## 4 2020-03-12 12 Mar 20     4*2a            8                           0
## 5 2020-03-12 12 Mar 20     5*2a            1                           0
## 6 2020-03-12 12 Mar 20    5*2ag            3                           0
##   total_flowers
## 1
## 2
## 3
## 4
## 5
## 6
```

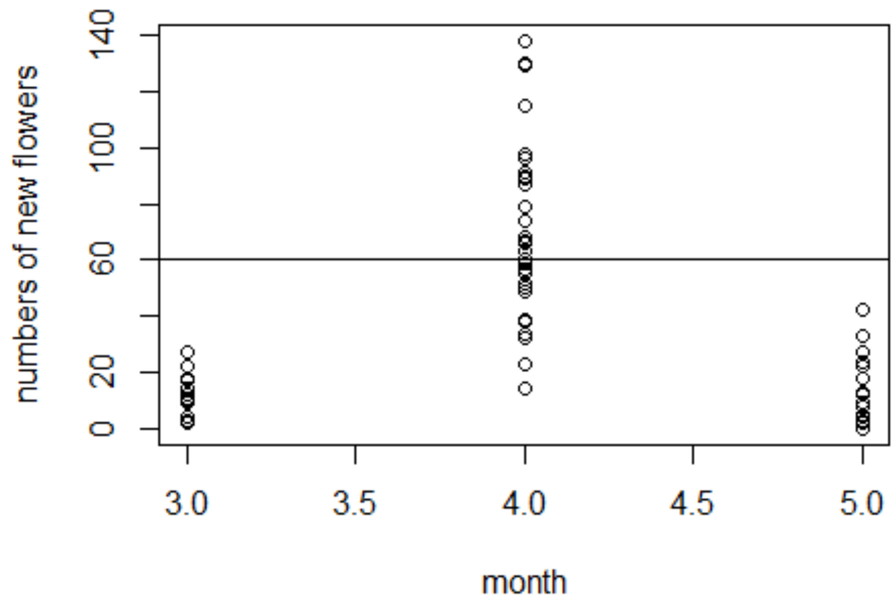
Data was aggregated to have one value per date.

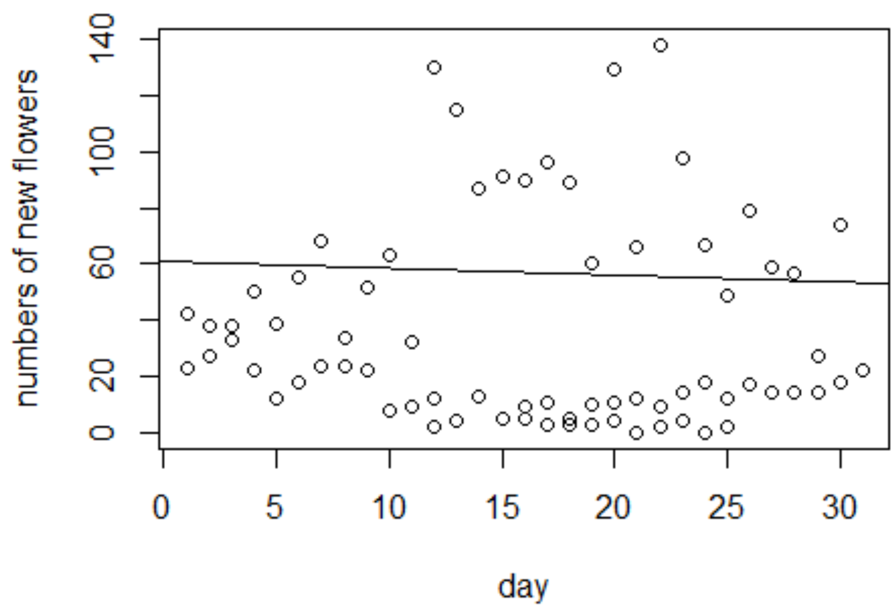
```
## # A tibble: 6 x 2
##   new_date   new_flowers
##   <date>           <dbl>
## 1 2020-03-12           2
## 2 2020-03-13           4
## 3 2020-03-16           9
## 4 2020-03-17          11
## 5 2020-03-18           3
## 6 2020-03-19          10
```

2. Linear regression



To be able to fit the linear regression the data was split into two features : day and month






```
##
## Call:
## lm(formula = new_flowers ~ day + month, data = vanilla_lm)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40.57 -26.10 -15.59  23.66 103.16
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  61.2692    29.9571   2.045   0.0446 *
## day          -0.2492     0.5543  -0.450   0.6544
## month        -5.2370     6.0473  -0.866   0.3894
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.85 on 70 degrees of freedom
## Multiple R-squared:  0.01075,    Adjusted R-squared:  -0.01752
## F-statistic: 0.3802 on 2 and 70 DF,  p-value: 0.6851
```

From the regression results, judging from the high p-value, we cannot reject the null hypothesis for the dependency of new flowers, on the date column. This is not unusual because the number of new flowers naturally tend to a normal distribution as with many other phenomenon in nature.

Fitting of the model

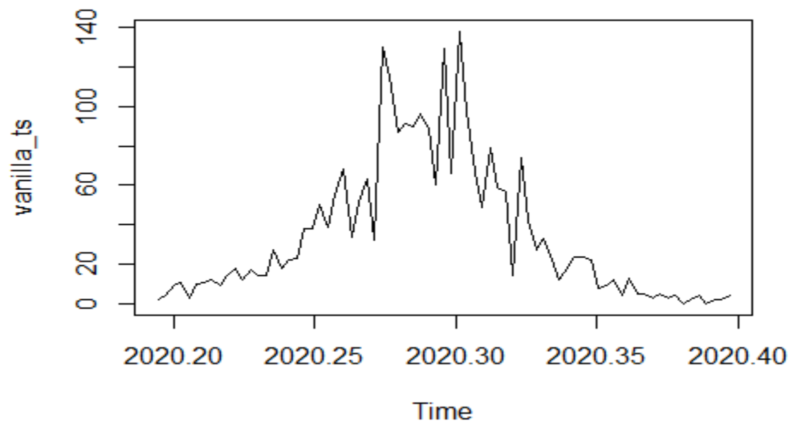
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_lm)
## W = 0.84505, p-value = 3.013e-07

Residuals not normally distributed, p.value < 0.05
```

SSE

```
## [1] 89972.27
```

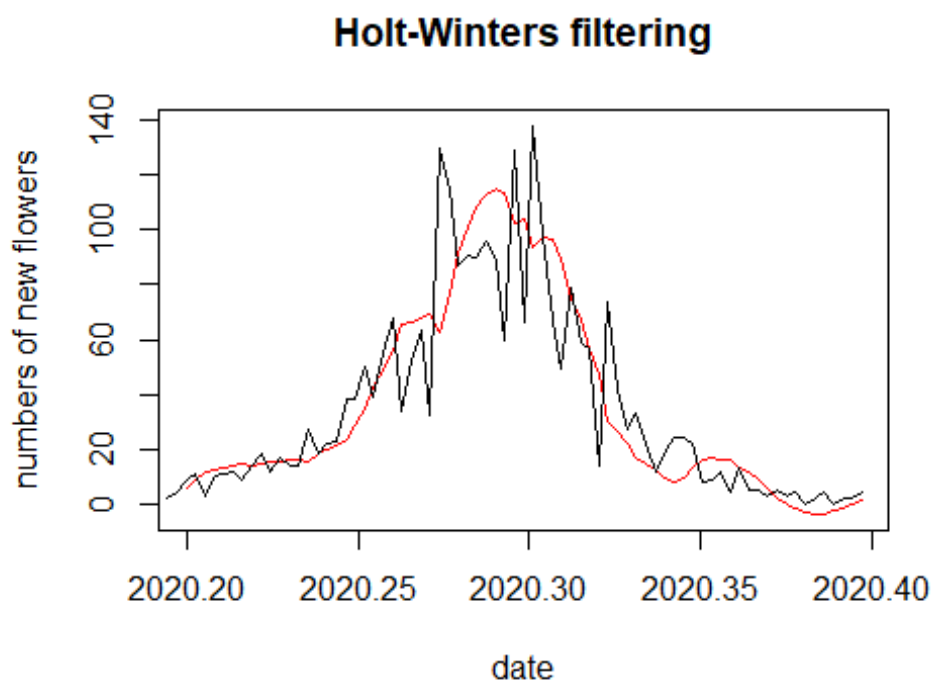
3. Holt-Winter

First we will visualize the distribution of the data.



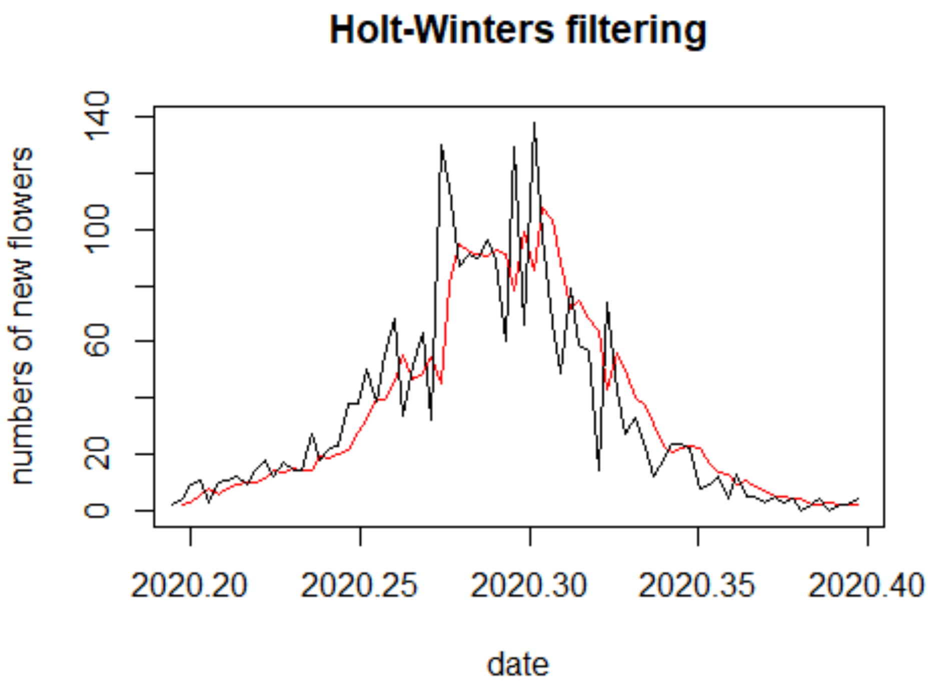
In the first plot, we will set gamma to be False because there is no seasonality. However, the model assumes there is a trend which is represented in the beta component.

```
model_ht <- HoltWinters(vanilla_ts, gamma = FALSE)
```



In the second plot, we will set both beta and gamma to false. This makes the model perform "exponential smoothing" because it assumes that the data has no trend or seasonality.

```
model_ht1 <- HoltWinters(vanilla_ts,beta = FALSE, gamma = FALSE)
```



SSE

The SSE for both the plots are displayed below. We can see that the first model, where we assumed that there is trend but no seasonality, performs better as evidenced by the relatively lower value for sum of squared error.

```
## [1] 25417.17

## [1] 27607.85
```

4. Auto ARIMA

Seasonal set to False

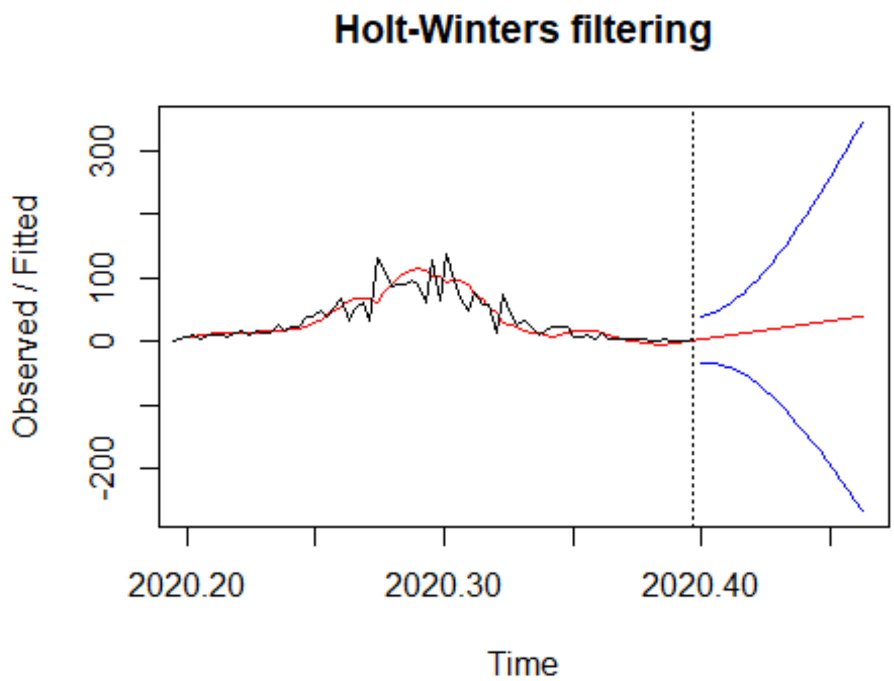
```
model_auto_arima <- auto.arima(vanilla_ts, seasonal = FALSE)
```

SSE

```
## [1] 26505.93
```

5. Result

The best model is Holt-Winter which assumes the presence of trend (beta) and no seasonality (gamma).



Confidence interval

```
## Time Series:
## Start = 2020.39972621492
## End = 2020.46269678303
## Frequency = 365.25
##                 fit       upr        lwr
## 2020.400  3.247624  40.07272  -33.57747
## 2020.402  4.828049  42.65685  -33.00075
## 2020.405  6.408474  46.40361  -33.58666
## 2020.408  7.988899  51.57022  -35.59242
## 2020.411  9.569324  58.22777  -39.08913
## 2020.413 11.149749  66.30422  -44.00473
## 2020.416 12.730174  75.65799  -50.19764
## 2020.419 14.310599  86.13533  -57.51414
## 2020.422 15.891024  97.59860  -65.81655
## 2020.424 17.471449 109.93431  -74.99142
## 2020.427 19.051874 123.05196  -84.94821
## 2020.430 20.632299 136.87989  -95.61529
## 2020.433 22.212724 151.36109 -106.93565
## 2020.435 23.793149 166.44965 -118.86336
## 2020.438 25.373574 182.10806 -131.36091
## 2020.441 26.953999 198.30518 -144.39718
## 2020.444 28.534424 215.01480 -157.94595
## 2020.446 30.114849 232.21451 -171.98481
## 2020.449 31.695274 249.88491 -186.49436
## 2020.452 33.275698 268.00902 -201.45762
```

```
## 2020.454 34.856123 286.57179 -216.85955
## 2020.457 36.436548 305.55980 -232.68670
## 2020.460 38.016973 324.96093 -248.92699
## 2020.463 39.597398 344.76420 -265.56940
```

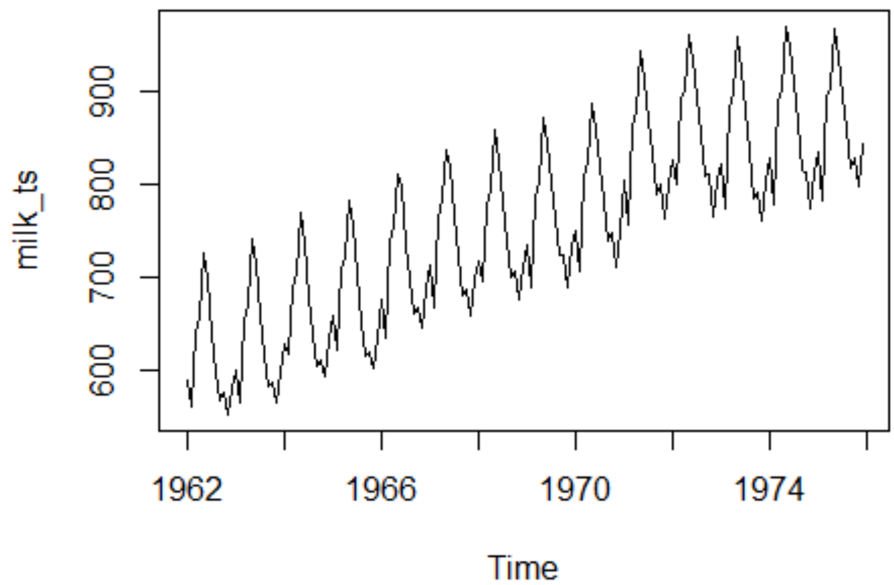
II. Seasonal data (milk production)

We decided to use another dataset with seasonal data, to explore different approaches which we may have missed in the previous dataset.

1. Characterisation of the data

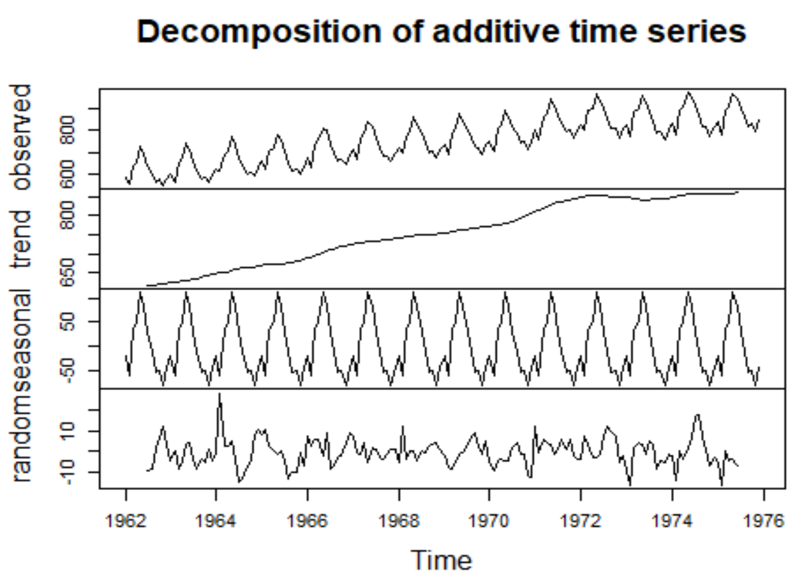
The data set we have used is for milk production per month from 1962 to 1975. As we can see in the visualization, the data is additive seasonal, and has a positive linear trend.

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1962 589 561 640 656 727 697 640 599 568 577 553 582
## 1963 600 566 653 673 742 716 660 617 583 587 565 598
## 1964 628 618 688 705 770 736 678 639 604 611 594 634
## 1965 658 622 709 722 782 756 702 653 615 621 602 635
## 1966 677 635 736 755 811 798 735 697 661 667 645 688
## 1967 713 667 762 784 837 817 767 722 681 687 660 698
## 1968 717 696 775 796 858 826 783 740 701 706 677 711
## 1969 734 690 785 805 871 845 801 764 725 723 690 734
## 1970 750 707 807 824 886 859 819 783 740 747 711 751
## 1971 804 756 860 878 942 913 869 834 790 800 763 800
## 1972 826 799 890 900 961 935 894 855 809 810 766 805
## 1973 821 773 883 898 957 924 881 837 784 791 760 802
## 1974 828 778 889 902 969 947 908 867 815 812 773 813
## 1975 834 782 892 903 966 937 896 858 817 827 797 843
```



2. Decompose time series

For this type of data we can use the classic decomposition method which uses the moving average. We notice that the systemic elements of trend and season have been separated properly and the randomness does not appear to have any trend.



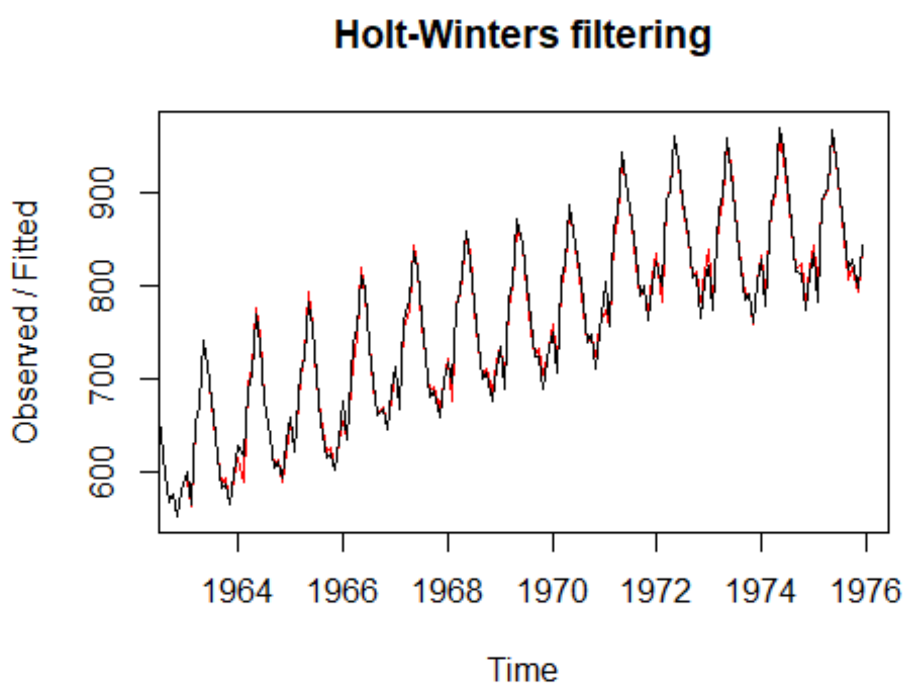
3. Holt-Winter

Since the series has a trend and is also seasonal, both beta and gamma will not be set to False. We can experiment in tuning these parameters manually as well.

a. Model 1

First we will check the result of Holt Winters model with the default values of the model.

```
model_hw <- HoltWinters(milk_ts)
```



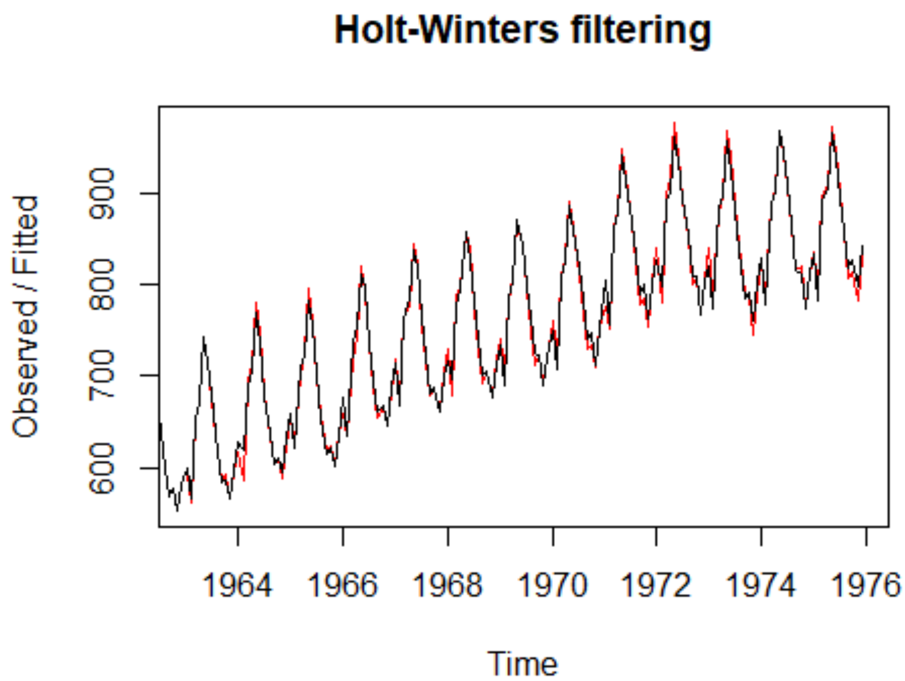
SSE

```
## [1] 10534.58
```

b. Model 2 : Multiplicative

Although we are aware (visually) that the series has a seasonal component which is additive in nature, we want to test how the model performs if we set the seasonality as multiplicative.

```
model_hw_2 <- HoltWinters(milk_ts, seasonal = "multiplicative")
```



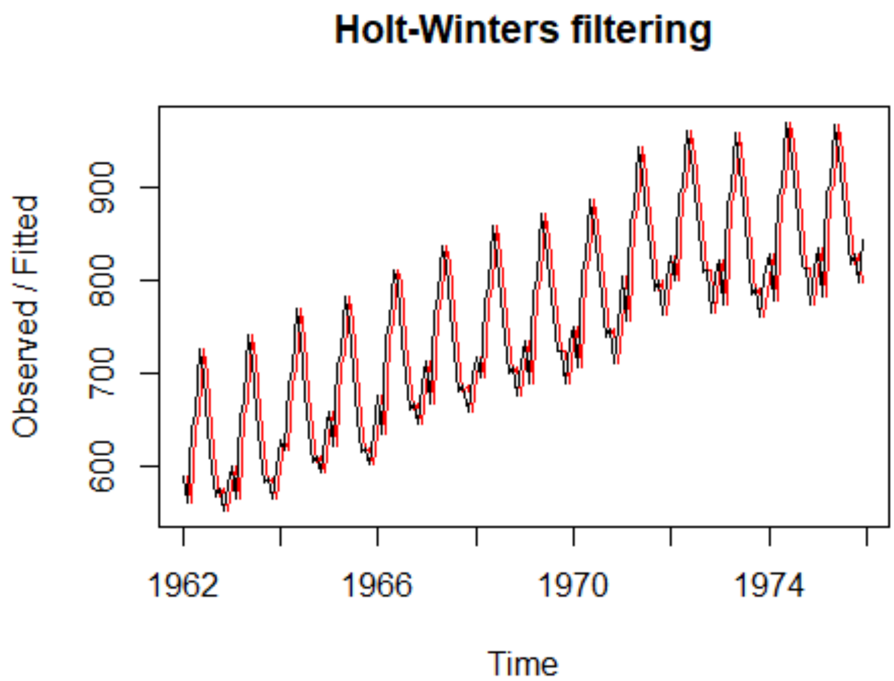
SSE

```
## [1] 11967.11
```

c. Model 3 : Exponential smoothing

Next we can set beta and gamma to false so that the model does not include a trend component and hence defaults to exponential smoothing of the levels. Typically, the exponential smoothing model assumes that the series does not have trend or seasonality and these components have to be removed before running the model and added back again. We can observe that the model responds late to the changes in the trend because it relies only on the past values.

```
model_hw_3 <-  HoltWinters(milk_ts, beta=FALSE, gamma = FALSE)
```



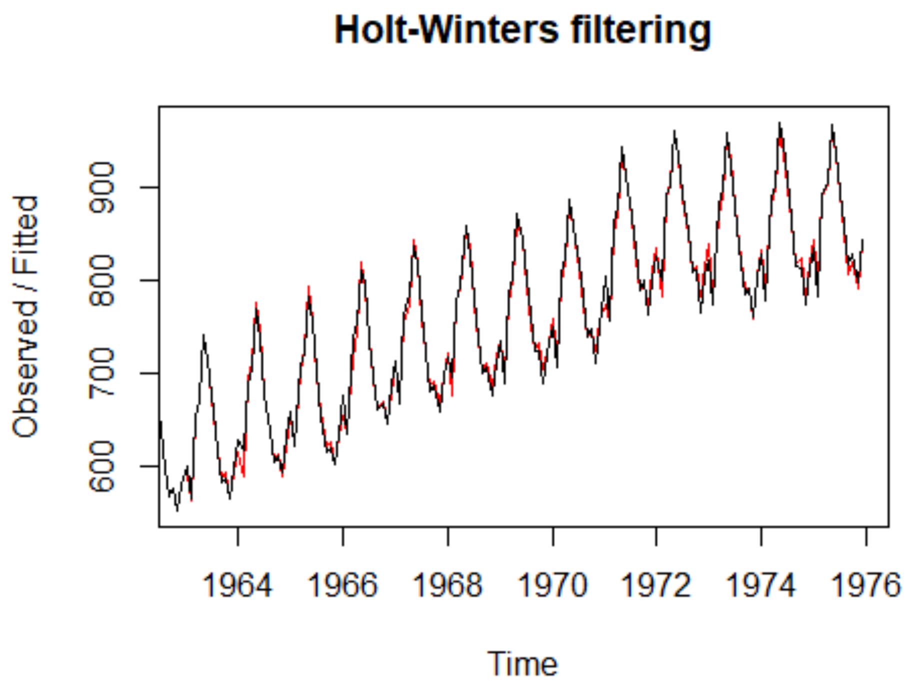
SSE

```
## [1] 343033.1
```

b. Model 4 : Manual tuning

We also tried to manually tune the parameters for beta and gamma to get the lowest values. We achieved a very close SSE value to the default parameters of the Holt Winters model by setting beta to 0 and gamma to 0.8. This can be translated to the idea that the model is set to learn the trend by accounting for more past values, and the seasonality by accounting for the most recent past values.

```
model_hwt <- HoltWinters(milk_ts, beta = 0, gamma =0.8)
```



SSE

```
## [1] 10536.39
```

4. Auto ARIMA

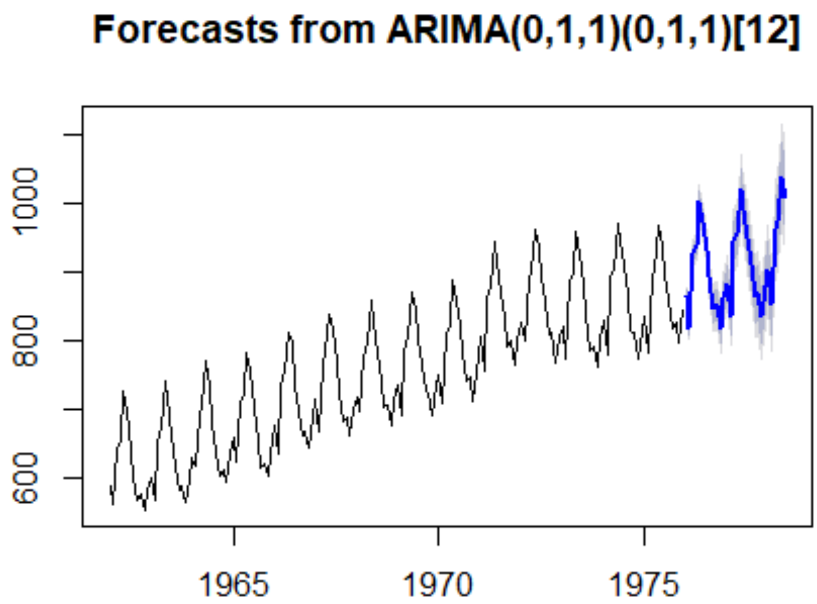
Finally we use Auto ARIMA to check if the SSE is better than our manual selections of models. Seasonal parameter is set to True. We can observe that this method provides the best result for this dataset.

```
model_auto_arima <- auto.arima(milk_ts, seasonal = TRUE)
```

```
## [1] 8173.711
```

5. Prediction

The best model is Auto ARIMA with a lower SSE



Confidence interval

```
prediction_arima

##          Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## Jan 1976       864.9773 855.6103  874.3443 850.6517  879.3029
## Feb 1976       817.7493 805.8719  829.6267 799.5843  835.9142
## Mar 1976       924.4056 910.4626  938.3485 903.0817  945.7295
## Apr 1976       937.4836 921.7439  953.2233 913.4118  961.5554
## May 1976      1000.6235 983.2721 1017.9749 974.0868 1027.1601
## Jun 1976       973.2165 954.3909  992.0420 944.4252 1002.0077
## Jul 1976       931.8501 911.6576  952.0426 900.9684  962.7318
## Aug 1976       892.2597 870.7873  913.7322 859.4204  925.0991
## Sep 1976       846.3679 823.6875  869.0483 811.6812  881.0545
## Oct 1976       851.5326 827.7055  875.3597 815.0921  887.9731
## Nov 1976       817.4931 792.5719  842.4143 779.3795  855.6068
## Dec 1976       859.7534 833.7842  885.7225 820.0370  899.4698
## Jan 1977       882.8150 854.6706  910.9593 839.7719  925.8581
## Feb 1977       835.5870 805.6961  865.4779 789.8728  881.3012
## Mar 1977       942.2433 910.7024  973.7842 894.0057  990.4809
## Apr 1977       955.3213 922.2126  988.4301 904.6859 1005.9568
## May 1977      1018.4612 983.8555 1053.0668 965.5364 1071.3859
## Jun 1977       991.0542 955.0138 1027.0946 935.9351 1046.1732
## Jul 1977       949.6878 912.2676  987.1080 892.4585 1006.9171
## Aug 1977       910.0975 871.3465  948.8484 850.8330  969.3619
## Sep 1977       864.2056 824.1682  904.2430 802.9736  925.4375
## Oct 1977       869.3703 828.0865  910.6541 806.2321  932.5085
## Nov 1977       835.3308 792.8371  877.8245 770.3423  900.3194
```

```
## Dec 1977       877.5911 833.9210  921.2612 810.8034  944.3787
## Jan 1978       900.6527 854.9096  946.3957 830.6947  970.6106
## Feb 1978       853.4247 805.9157  900.9337 780.7660  926.0834
## Mar 1978       960.0810 910.8694 1009.2926 884.8183 1035.3437
## Apr 1978       973.1590 922.3017 1024.0163 895.3795 1050.9385
## May 1978      1036.2989 983.8475 1088.7502 956.0815 1116.5163
## Jun 1978      1008.8919 954.8935 1062.8902 926.3085 1091.4752
```