Time Series Analysis and Forecasting

Matthew Juan z5259434

Contents

1 Introduction to Time Series'

Time series are found all over the place and are extremely useful to analyse and more importantly, forecast. There are many situations where you want to predict past your last data point. Use cases include:

- Traders predicting the stock market so they know when to buy or sell
- Data centre operators identifying when a critical system may go down or is acting irregularly
- Retail store managers knowing how to maximise what stock they order to maximise profit and minimise items that won't sell

Unfortunatly, just like humans, it is impossible to predict the future. However, by analysing the time series, it is possible to formulate statistical properites that can be used to give a rough estimate of where the data is heading.

2 Properties of Time Series Data

A time series can fundamentally be thought of as a random variable consisting of a signal and noise. This signal and noise are formed by 3 components which are core to understanding time series', trend, seasonality, and residual. Understanding their role and what they represent as well as some other key definitions are needed before diving in to analysing time series data.

2.1 Trend

The first component is trend. Trend can be viewed as how the data is changing overall. Is the data increasing or decreasing.

2.2 Seasonality

Seasonality refers to a regular occuring pattern that emerges within a given period. E.g peaks or valleys in the data during certain months or hours in the day.

2.3 Residual

Residual, also commonly referred to as errors, are the parts left over after fitting a model to the data. In many time series models, it is defined as the difference between the actual observation and the predicted value.

2.4 Additive vs Multiplicative

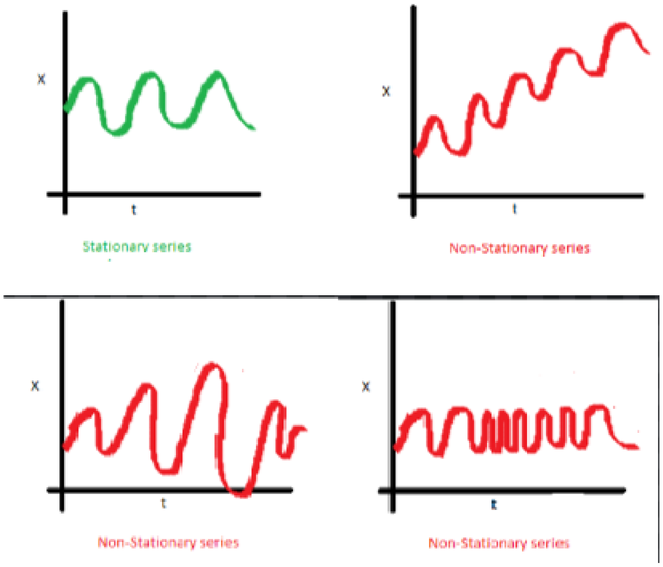
The trend and seasonality in a time series can be classified as being additive or multiplicative. In additive time series', the data exhibits a general upwards or downwards trend but does so at a constant rate. Peaks and valleys in your data are roughly the same size. In contrast, multiplicative time series data exhibits peaks or valleys that are amplified within the seasonal activity. The data becomes exaggerated and the difference between peaks at the start are very different than at the tail.

2.5 Stationarity

Before going into analysis, the time series needs to be stationary. A random variable that forms a time series is classified stationary if it's probabilty distribution is constant throughout the full range of times. Formally, if T is the set of all times in your time series of length n, a random process X is stationary if the joint probability distribution at times $t_1, ..., t_n \in T$ and $t_1 + \Delta, ..., t_n + \Delta \in T,\ \Delta \in \mathbb{R}$ are equal. That is

$$F_X(x_{t_1}, ..., x_{t_n}) = F_X(x_{t_1+\Delta}, ..., x_{t_n+\Delta})\ \forall x_{t_1}, ..., x_{t_n} \in T$$

Intuitively, this means a stationary time series has constant propertues such as mean, variance, covariance, etc. Below shows the differences between stationary and non-stationary processes.



Stationary series

Non-Stationary series

Non-Stationary series

Non-Stationary series

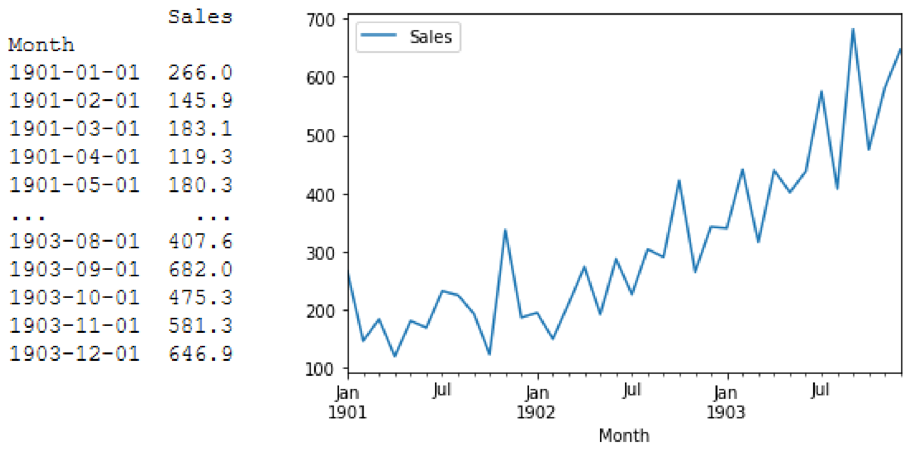
Figure 1

2.5.1 How to test for stationarity

To determine whether a time series is stationary or not, the Augmented Dickey Fuller(ADF) test can be performed. ADF performs a hypothesis test and upon observing p-value $p < \alpha$ where α is commonly 0.05, the hypothesis is rejected and the data is considered to be stationary.

2.5.2 Methods of making a time series stationary

If you have a non-stationary time series, there are a few ways to make it stationary. Let's use the Shampoo Sales dataset which describes the monthly number of sales of shampoo over a 3 year period. The data here has a distinct multiplicative increasing trend and is clearly non stationary. Using the ADF test, the p-value comes out to $1 > 0.05$ so there is weak evidence against the null hypothesis so the data is non-stationary.



```
            Sales
Month
1901-01-01  266.0
1901-02-01  145.9
1901-03-01  183.1
1901-04-01  119.3
1901-05-01  180.3
...           ...
1903-08-01  407.6
1903-09-01  682.0
1903-10-01  475.3
1903-11-01  581.3
1903-12-01  646.9
```

Figure 2

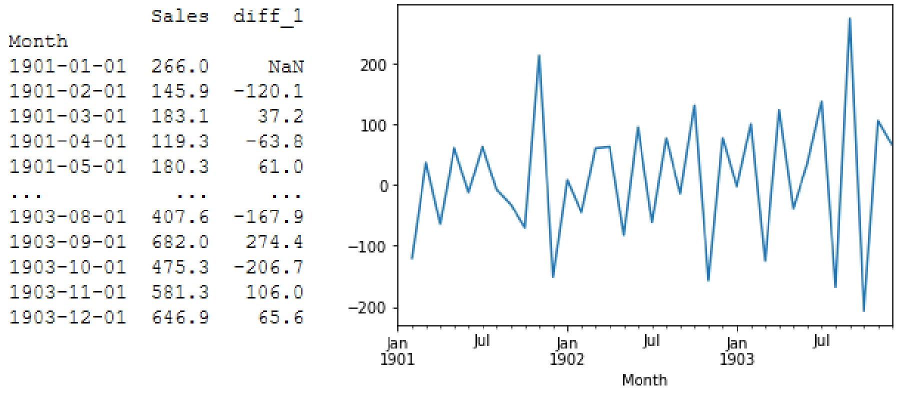
The most common is differencing. Differencing involves creating a new dataset where the values at time t is equal to the difference between the original value at time t and original value at $t-1$. So if Y is the new dataset and y is the original data, then

$$Y_t = y_t - y_{t-1}.$$

Sometimes the differenced data will not make the data stationary enough and it may be necessary to difference the data again known as 2nd order differencing. This process is applicable to higher orders Note, the 2nd order difference is not $Y_t = y_t - y_{t-2}$ but it is the first differnce of the first difference so it comes out to

$$Y_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}).$$

The p-value from the ADF test is about $1.8 \times 10^{-10} < 0.05$ so the data is now stationary.



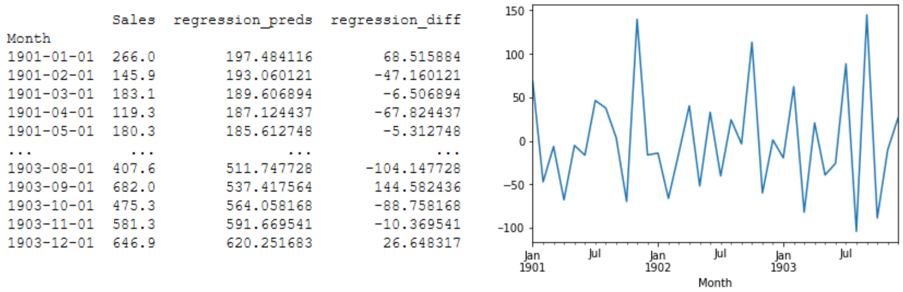
```
             Sales  diff_1
Month
1901-01-01   266.0     NaN
1901-02-01   145.9  -120.1
1901-03-01   183.1    37.2
1901-04-01   119.3   -63.8
1901-05-01   180.3    61.0
...            ...     ...
1903-08-01   407.6  -167.9
1903-09-01   682.0   274.4
1903-10-01   475.3  -206.7
1903-11-01   581.3   106.0
1903-12-01   646.9    65.6
```

Figure 3

Another common technique is to detrend by model fitting. You simply fit a regression model, such as linear or exponential, and create a new dataset where each value at time t is equal to the difference between the original value and the predicted value. When fitted with a 2nd order polynomial regression model, the resulting differences produced a p-value of about $3.4 \times 10^{-10} < 0.05$ so again, the data is now stationary.

$$Y_t = y_t - pred_t.$$

Again, the data is much more stationary than the original.



```
             Sales  regression_preds  regression_diff
Month
1901-01-01   266.0        197.484116        68.515884
1901-02-01   145.9        193.060121       -47.160121
1901-03-01   183.1        189.606894        -6.506894
1901-04-01   119.3        187.124437       -67.824437
1901-05-01   180.3        185.612748        -5.312748
...            ...               ...              ...
1903-08-01   407.6        511.747728      -104.147728
1903-09-01   682.0        537.417564       144.582436
1903-10-01   475.3        564.058168       -88.758168
1903-11-01   581.3        591.669541       -10.369541
1903-12-01   646.9        620.251683        26.648317
```

Figure 4

There are various other techniques that can be performed and combined with one another to make your data more stationary. If your data has a seasonal pattern to it, seasonal differencing can be performed where you take the difference between t and $t - \rho$ where ρ is the period of your data. Non-linear transformations your dataset such as logging can also prove effective at making data stationary.

3 Non-Seasonal Forecasting Models

There are many types of forecasting models but this paper will look two types. ARIMA (Auto-Regressive Integrated Moving Average) and Exponential Smoothing models. For these non-seasonal models, we will use the Shampoo Sales dataset shown previously.

3.1 Exponential Smoothing

As the name suggests, Exponential Smoothing models are essentially weighted averages of past observations which decay exponentially over time. This type allows for a reliable and quick way to forecast for a wide array of time series'. This section of the paper will go into an implementation of simple exponential smoothing, Holt's linear trend method and the damped trend method.

3.1.1 Simple Exponential Smoothing

This method of forecasting is extremely simple and is useful when there is no clear trend or seasonal pattern in the data. As mentioned, this method uses a weighted average where the weights decrease exponentially the further in the past the associated observation is. Therefore, for a time series with T datapoints, we can define such an forecast equation as:

$$\hat{y}_{T+1} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \alpha(1-\alpha)^3 y_{T-3} + ...,$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter and $\hat{y}_T$ is the forecasted value at time $T+1$. This equation can be simplified by using the forecasted value at time T to predict the value at time $T+1$ resulting in

$$\hat{y}_{T+1} = \alpha y_T + (1-\alpha)\hat{y}_T.$$

This method can also be expressed in component form. The only component with single exponential smoothing is the level, l_t. In component form, simple exponential smoothing is shown by:

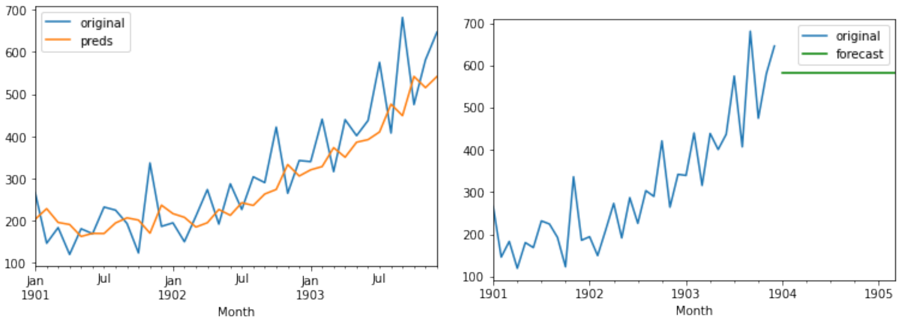
$$\begin{array}{ll} \text{Level equation} & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1}), \qquad 0 \leq \alpha \leq 1 \\ \text{Forecast equation} & \hat{y}_{t+h} = \ell_t \end{array}$$

The forecasting equation produces a "flat" forecast equal to the last level component since setting $t = T$, we get

$$\hat{y}_{T+h} = \ell_T, \qquad h = 1, 2, ...$$

Let's see how this is applied to the shampoo dataset. Firstly, we need to estimate values for α as well as the initial level ℓ_0. By minimising the SSE (sum of squared errors), the estimated parameters are $\alpha = 0.40$ and $\ell_0 = 202.78$. The table below shows the calculations using these parameters.

Time(t)	Observation(y_t)	Level(ℓ_t)	Forecast($\hat{y}_t$)
0		202.78	
1	266.0	228.03	202.78
2	145.9	195.23	228.03
3	183.1	190.38	195.23
4	119.3	161.99	190.38
$\vdots$	$\vdots$	$\vdots$	$\vdots$
33	682.0	541.92	448.78
34	475.3	515.32	541.92
35	581.3	541.67	515.32
36	646.9	583.70	541.67
h			$\hat{y}_{T+h}$
1			583.70
2			583.70
3			583.70
4			583.70
5			583.70



(a) Fitted values

(b) Forecasted values

Figure 5: Fitted and forecasted values using simple exponential smoothing

3.1.2 Trend Methods

This section will show Holt's linear trend method and Gardner & McKenzie's damped trend method which extend simple exponential smoothing by incorporating the trend component.

Holt's Linear Trend Method

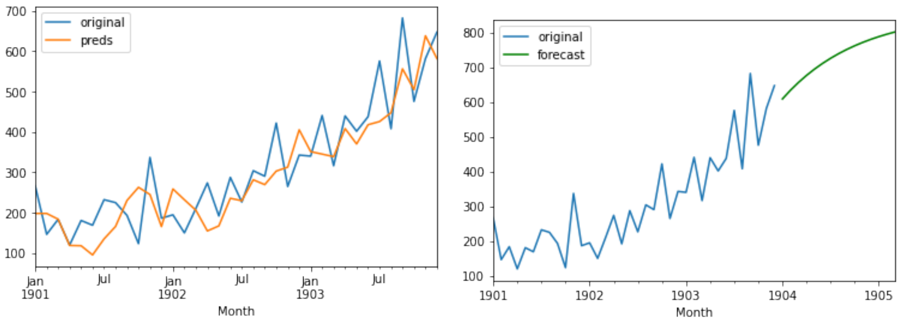
This method now has three equations, a level smoothing equation, a trend smoothing equation, and a forecast equation:

$$\begin{aligned} \text{Forecast equation} \qquad \hat{y}_{t+h|t} &= \ell_t + hb_t \\ \text{Level equation} \qquad \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} \qquad b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}, \end{aligned}$$

where $0 \leq \alpha, \beta \leq 1$. By including the trend, forecasts are no longer flat but a linear function of h. As such, the forecasted value at time $T+h$ is equal to the last level plus h multiplied by the last calculated trend value.

There are now two smoothing parameters α and β as well as two initial states ℓ_0 and b_0 which are estimated to $\alpha = 0.49, \beta = 0.27, \ell_0 = 262.93, b_0 = -18.01$.

Time(t)	Observation(y_t)	Level(ℓ_t)	Slope(b_t)	Forecast($\hat{y}_t$)
0		262.93	-18.01	
1	266.0	255.19	-15.23	244.92
2	145.9	194.14	-27.65	244.92
3	183.1	174.58	-25.46	239.96
4	119.3	134.60	-29.40	166.489
⋮	⋮	⋮	⋮	⋮
33	682.0	583.60	39.51	540.92
34	475.3	551.11	19.99	490.15
35	581.3	576.07	21.33	623.11
36	646.9	621.51	27.87	571.10
h				$\hat{y}_{T+h}$
1				649.38
2				677.25
3				705.12
4				732.99
5				760.86



(a) Fitted values

(b) Forecasted values $\phi = 0.9$

Figure 6: Fitted and forecasted values using Holt linear trend method

Damped Trend Method

Holt's Linear method show a constant increasing or decreasing trend which extends indefinitely into the future. However, in reality this method can be quite inaccurate for extended forecasts. Gardner & McKenzie introduced a dampening parameter ϕ, where $0 \leq \phi \leq 1$. This parameter dampens the trend to forecast a flat line after some time.

The component form of this method is:

$$\begin{aligned} \text{Forecast equation} \qquad \hat{y}_{t+h|t} &= \ell_t + (\phi + \phi^2 + ... + \phi^h) b_t \\ \text{Level equation} \qquad \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \\ \text{Trend equation} \qquad b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)\phi b_{t-1}. \end{aligned}$$

When $\phi = 1$, this method produces Holt's linear method.

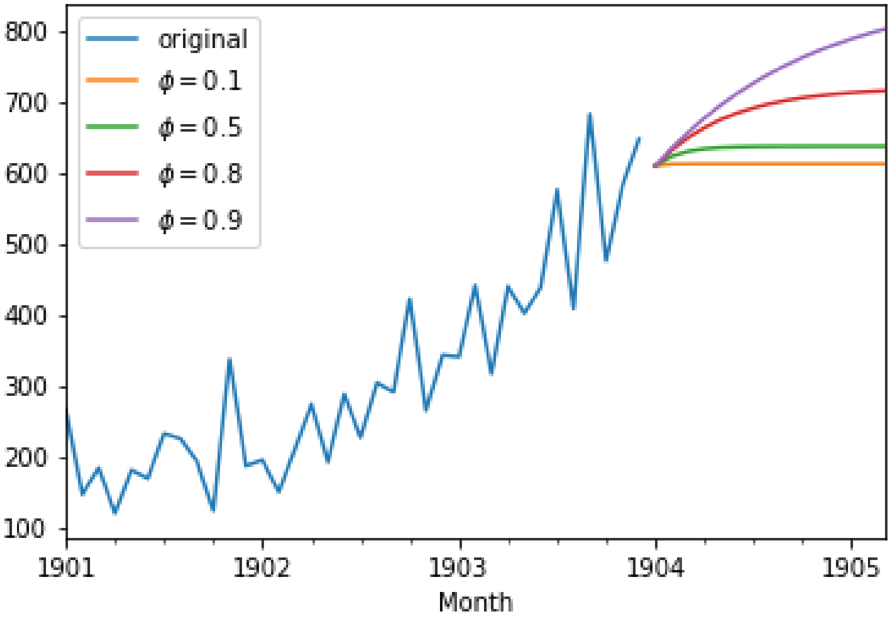


Figure 7: Forecasts for different values of ϕ

3.2 ARIMA

ARIMA is an extension of the ARMA or Box-Jenkin model popularised by George E. P. Box and Gwilym Jenkins in 1970. The main difference is that ARIMA simply allows for an extra parameter d which defines how much differencing should be done before fitting the model, whereas ARMA assumes the data is stationary already. ARIMA is actually a combination of three components, an autoregressive model (AR), a moving averages model (MA), and an integrated (I) part denoting the previously mentioned differencing. We will dive into what makes up AR and MA models and how ARIMA is able to effectively combine the two. As previously discussed, a random variable in the form of a time series can be viewed as being a combination of signal and noise. The goal of ARIMA is to filter the noise and extrapolate the signal.

3.2.1 Backshift Notation

Before we jump into defining what an ARIMA model is, there is a useful backshift operator B denoting the time series lags,

$$By_t = y_{t-1}.$$

B effectively shifts that data back one time period. So naturally, two applications of B shifts the data back two time periods.

$$\begin{aligned} B(By_t) &= B^2 y_t \\ &= y_{t-2}. \end{aligned}$$

If we describe how differencing works using this notation it comes out to

$$\begin{aligned} Y_t &= y_t - y_{t-1} \\ &= y_t - By_t \\ &= (1-B)y_t. \end{aligned}$$

You will start to see a pattern when you take higher order differences. For second order,

$$\begin{aligned} Y_t &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= (y_t - 2By_t + B^2 y_t) \\ &= (1-B)^2 y_t. \end{aligned}$$

In general, the d'th order of differencing is defined as

$$Y_t = (1-B)^d y_t.$$

This notation is great as we can use B is an algebraic manner which greatly simplifies some upcoming formulas.

3.2.2 Autoregressive Models

The term *autoregressive* defines that the model uses a linear combination of previous values of a variable and learned predictors. This differs from a typical regression model which predicts based on a linear combination of the input features and learned predictors. Autoregressive models has one parameter p which defines the order of the model. This parameter defines how many previous values influence the one being predicted. An AR(1) model is simply defines a model in which the predicted value is equal to some linear combination of the value at the previous time step

$$y_t = \mu + \phi_1 y_{t-1} + \varepsilon_t,$$

where μ is the average period to period change. An AR(2) model uses the previous two timesteps as so on. Thus, an AR(p) model can be generalised to

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p}$$

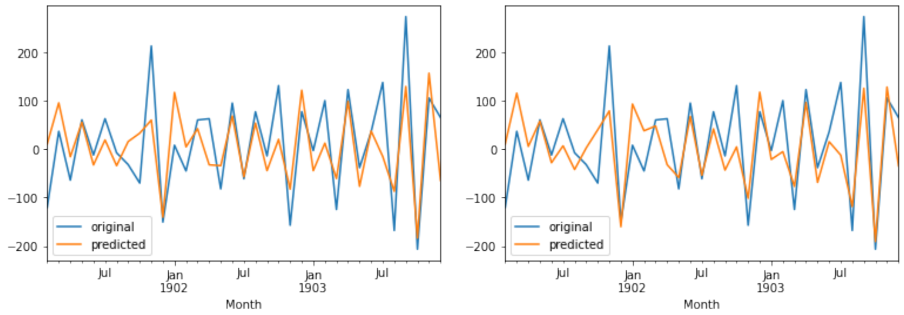
or in equivalent backshift notation

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p) y_t = \mu.$$

We can see that we are using previous known values as the features to predict the future.

Let's see how this works for the first differenced data. Firstly, the lags are calculated and then we estimate parameters using these lags and our original time series values. We simply do a dot product between the coefficients and add it to our value for μ to retrieve our predictions.

This is the resulting predictions of a few AR(p) models.



(a) AR(1)

(b) AR(2)

Figure 8: Two types of AR models. AR(1) with estimated $\phi_1 = -0.70868438$. AR(2) with estimated $\phi_1 = -0.9237595, \phi_2 = -0.26627682$.

Observe that there are subtle changes between the two. If we kept on increasing the AR term, our model would fit this data extremely well. In machine learning, this is commonly known as overfitting and future forecasts can be very inaccurate.

3.2.3 Moving Average Models

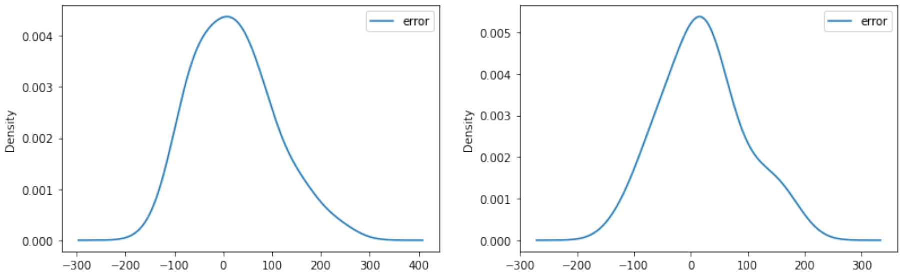
In a sort of similar manner, Moving Average (MA) models use the errors of past forecasts to predict. An MA model instead has parameter q so an MA(q) model can be written similarly as

$$y_t = \mu + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + ... + \theta_q\varepsilon_{t-q}$$

or in equivalent backshift notation

$$y_t = \mu + (\theta_1 B + \theta_2 B^2 + ... + \theta_q B^q)\varepsilon_t.$$

If your data is properly stationary, the error terms produced should form white noise in the form of a normal distribution with 0 mean and θ^2 variance.



(a) MA(1)

(b) MA(2)

Figure 9: Error terms for MA(1) with estimated $\theta_1 = -0.58305195$ and MA(2) with estimated $\theta_1 = -0.71206191, \theta_2 = 0.28793809$. Both generally look normally distributed with 0 mean.

3.2.4 Putting It All Together

When we combine differencing, an AR, and an MA model, we obtain an ARIMA model. It's general equation is as follows

$$y'_t = \mu + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \theta_1 \varepsilon'_{t-1} + ... + \theta_q \varepsilon'_{t-q}$$

or in equivalent backshift notation

$$(1 - \phi_1 B - ... - \phi_p B^p)(1 - B)^d y_t = \mu + (\theta_1 B + ... + \theta_q B^q)\varepsilon_t.$$

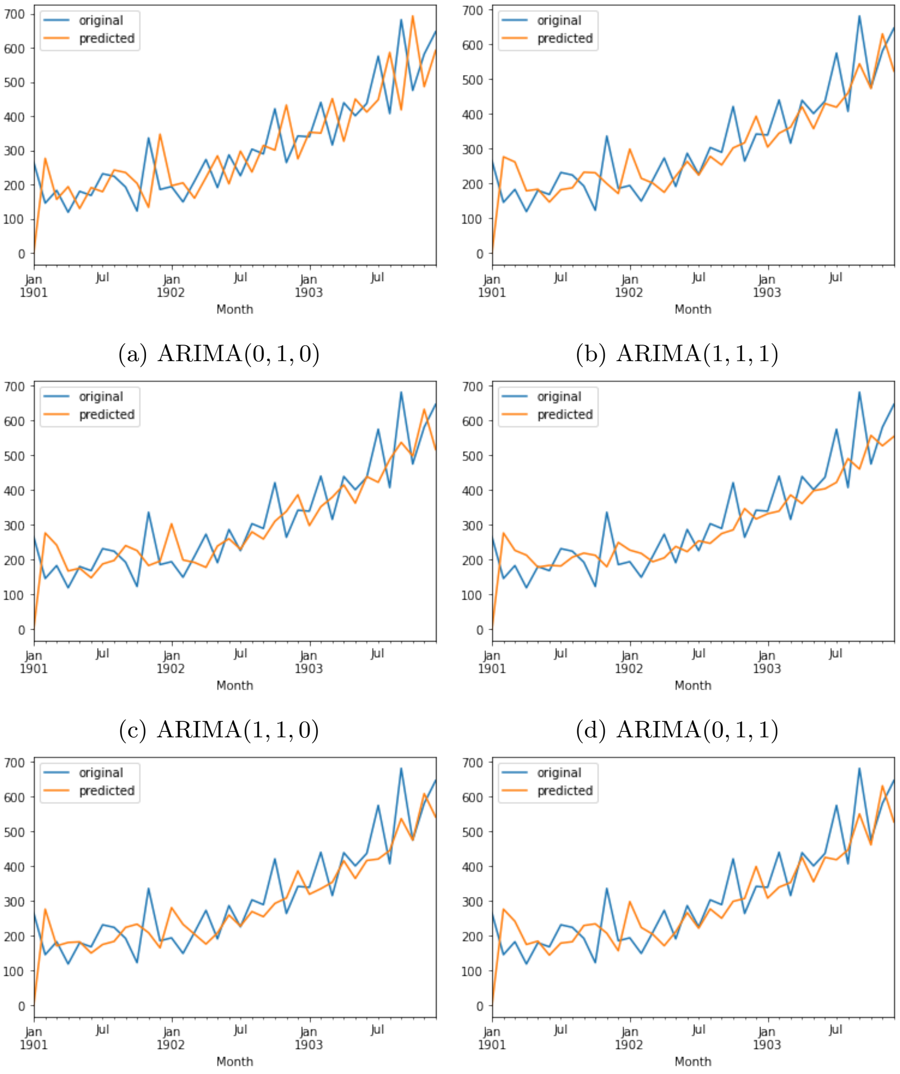
where y'_t is the differenced data. Note that the backshift notation already takes into account the order of differencing d in its equation.

An ARIMA model has 3 parameters and can be expressed as ARIMA(p, d, q).

- p is order of autoregressive component
- d is the order of differencing to be performed on the data
- q is the order of moving averages component

This generalised model is great since we don't have to difference the data ourselves and allows for the combination of AR(p) and MA(q) models.

Running through a list of parameters, we can see effects of different values for the parameters.



(a) ARIMA$(0, 1, 0)$

(b) ARIMA$(1, 1, 1)$

(c) ARIMA$(1, 1, 0)$

(d) ARIMA$(0, 1, 1)$

(e) ARIMA$(2, 1, 1)$

(f) ARIMA$(1, 1, 2)$

Figure 10: Fits for different values of p and q

3.2.5 Model Scoring

There are a few ways to score how well ARIMA models fit and predict. The most popular way is using Akaine's Information Criterion (AIC). It can be written as

$$AIC = -2log(L) + 2(p + q + k + 1),$$

where L is the likelihood of the data, $k = 1$ if $\mu \neq 0$ and $k = 0$ if $\mu = 0$. The general goal is to pick the model which minimises this value. Other values such as the Bayesian Information Criterion (BIC) and root mean squared error are also good metrics to measure and consider for your model.

Looking at the previous models, we can compare their AIC:

Model	AIC
ARIMA$(0, 1, 0)$	447.8405
ARIMA$(1, 1, 1)$	430.1539
ARIMA$(1, 1, 0)$	429.1197
ARIMA$(0, 1, 1)$	433.8894
ARIMA$(2, 1, 1)$	430.3742
ARIMA$(1, 1, 2)$	431.4727

Here, it looks like our ARIMA$(1, 1, 0)$ and ARIMA$(1, 1, 1)$ models had the best AIC scores.

3.2.6 Coefficient Estimation

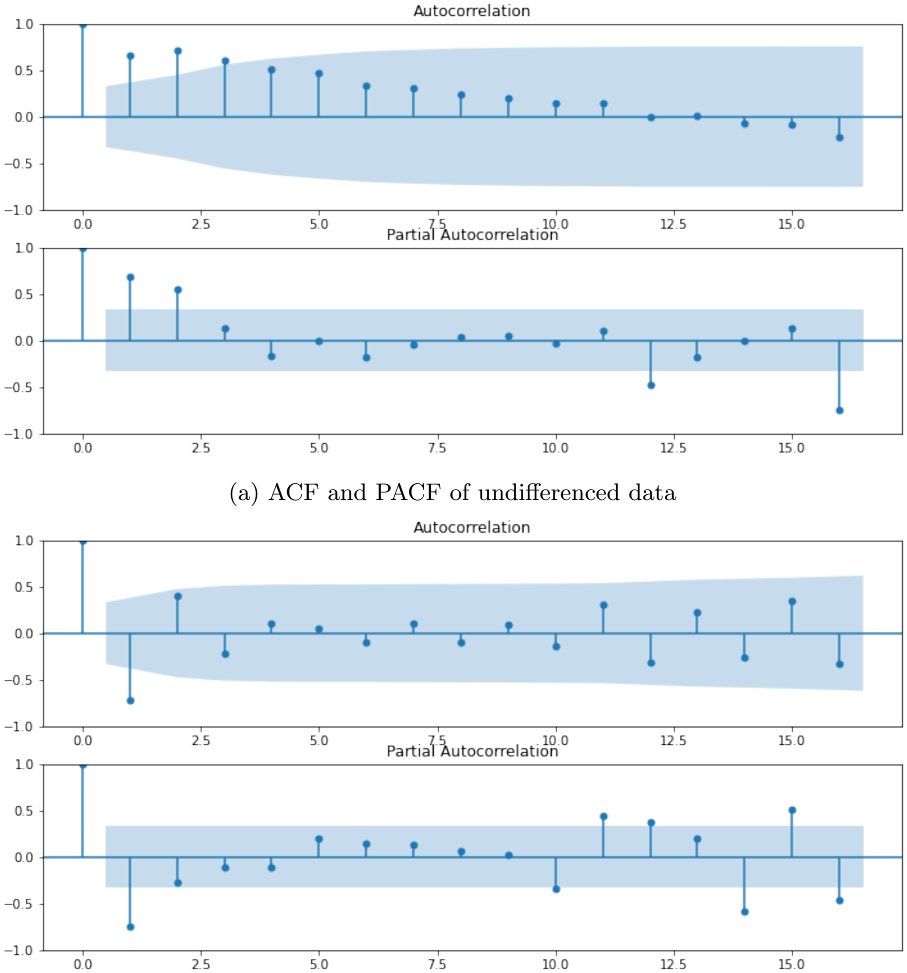
Estimating the coefficients for an ARIMA model is often a tricky task as there needs to be a balance between having a good fit for the data while not overfitting. For my implementation, both the AR and MA terms were calculated similarly using the Constrained Optimization BY Linear Approximation (COBYLA) algorithm which optimises based on the sum of squared errors. To avoid overfitting the data, I set constraints on the coefficients namely,

- $-1 \leq \phi_p \leq 1$
- $\phi_i + \phi_{i+1} \leq 1 \quad \text{for } i = 1, 2, ..., p - 1$
- $\phi_{i+1} - \phi_i \leq 1 \quad \text{for } i = 1, 2, ..., p - 1$

The same constraints are put for $\theta_1, ..., \theta_q$. In practice, ARIMA packages in Python and R use Maximum Likelihood Estimation to estimate the coefficients which often invoves much more complicated computations. There are also more complicated constraints, especially as the values of p and q increase, which these packages handle that account for the invertability and stationarity constraints.

3.2.7 Order Choosing

Up until now, we have been manually setting the parameters p, d, and q seemingly by observing which model fits the best or which one has the best AIC score. However, there are systematic methods of narrowing down the search space. This is where we look at the autocorrelation function (ACF) plots and partial autocorrelation (PACF) plots to help determine our optimal parameters. Autocorrelation is a way of measuring the similarity between values in a time series and their past. An autocorrelation of $+1$ means that there is a perfect positive correlation while -1 represents a perfect negative correlation. Partial autocorrelation is similar to autocorrelation but slightly more complicated to compute. PACF again measures similarity between y_t and $y_{t-\Delta}$ but adjusts for the presence of the other terms of short lag $(y_{t-1}, y_{t-2}, ..., y_{t-\Delta-1})$.



(a) ACF and PACF of undifferenced data

(b) ACF and PACF of order 1 differenced data

In a perfect scenario, we would want no correlation between the series and the lags of itself. Of course, this is almost impossible to achieve so the blue region highlights where we wish our spikes to lie.

Firstly, we have to choose the order of differencing. It is not sufficient to say that because a once differenced series is stationary then that should be the order. Higher order differencing may produce better results. This is where we look at the autocorrelation function (ACF) plot to help pick the value for d.

Duke University professor, Dr Robert Nau presents a few rules for interpreting these graphs and how to estimate the optimal value for d.

"Rule 1: If the series has positive autocorrelations out to a high number of lags (say, 10 or more), then it probably needs a higher order of differencing."
"Rule 2: If the lag-1 autocorrelation is zero or negative, or the autocorrelations are all small and patternless, then the series does not need a higher order of differencing. If the lag-1 autocorrelation is -0.5 or more negative, the series may be overdifferenced. BEWARE OF OVERDIFFERENCING."
Dr Robert Nau, Statistical Forecasting

Looking at the ACF plot for our undifferenced data, it mostly stays positive until around lag 11 so by rule 1, we should difference at least once. We can see that rule 1 no longer applies and rule 2 is applied. Observe that the lag-1 autocorrelation does go negative so a good choice would be $d = 1$. Note that the value is less than -0.5 which the rule warns us of overdifferencing. However, previously we looked at the ADF tests for differenced and non-differenced and it was clear that differencing was required so $d = 1$ is a good choice.

Using this method can be preferred over the trial and error method with the ADF test if we only care about differencing our data.
By looking at the ACF and PACF plots of a stationary series, we can generally identify the numbers of AR and/or MA terms that are needed. Dr Nau has identified more rules in identifying these values.

"Rule 6: If the partial autocorrelation function (PACF) of the differenced series displays a sharp cutoff and/or the lag-1 autocorrelation is positive, then consider adding one or more AR terms to the model. The lag which the PACF cuts off is the indicated number of AR terms."
"Rule 7: If the autocorrelation function (ACF) of the differenced series displays a sharp cutoff and/or the lag-1 autocorrelation is negative, then consider adding an MA term to the model. The lag which the ACF cuts off is the indicated number of MA terms."

Rule 6 helps us determine the AR terms. From the PACF plot, at lag-1 there is a steep drop into the negatives then settles into the blue region. Hence, we probably need 1 AR term. Similarly by rule 7, we should pick 1 MA term.

This gives us an ARIMA$(1, 1, 1)$ which we have seen before. This gives us an ARIMA$(1, 1, 1)$ model which we have seen before. In Python there is a package known as pmd_arima which has a nice function called auto_arima which computes the best parameters given some data. In this case, it gives back a parameter list of $(1, 1, 2)$ so our estimation was pretty close and shows that these rules are not absolute.

3.2.8 Forecasting

So far, we have shown how to fit an ARIMA model on a well-defined time series. But we have not gone into how it forecasts unseen data. It is quite simple and involves 3 steps:

1. Expand the model's ARIMA backshift equation and group by the powers of B
2. Convert to equation with only y_t's and move all terms that isn't y_t to the right side of the equation
3. Replace t with $T + 1$ is the number of future time steps and update time series with predicted value

Let's try this with an example using an ARIMA(2,1,2) model as an example. In backshift notation, the model is written as

$$(1 - \phi_1 B - \phi_2 B^2)(1 - B)y_t = \mu + (\theta_1 B + \theta_2 B^2)\varepsilon_t.$$

Using step 1, we expand and group:

$$\begin{aligned}(1 - B - \phi_1 B + \phi_1 B^2 - \phi_2 B^2 + \phi_2 B^3)y_t &= \mu + (\theta_1 B + \theta_2 B^2)\varepsilon_t \\ [1 - (1 + \phi_1)B - (\phi_2 - \phi_1)B^2 + \phi_2 B^3]y_t &= \mu + (\theta_1 B + \theta_2 B^2)\varepsilon_t.\end{aligned}$$

With second step, we convert giving

$$y_t - (1 + \phi_1)y_{t-1} - (\phi_2 - \phi_1)y_{t-2} + \phi_2 y_{t-3} = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2},$$

then moving terms

$$y_t = \mu + (1 + \phi_1)y_{t-1} + (\phi_2 - \phi_1)y_{t-2} - \phi_2 y_{t-3} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}.$$

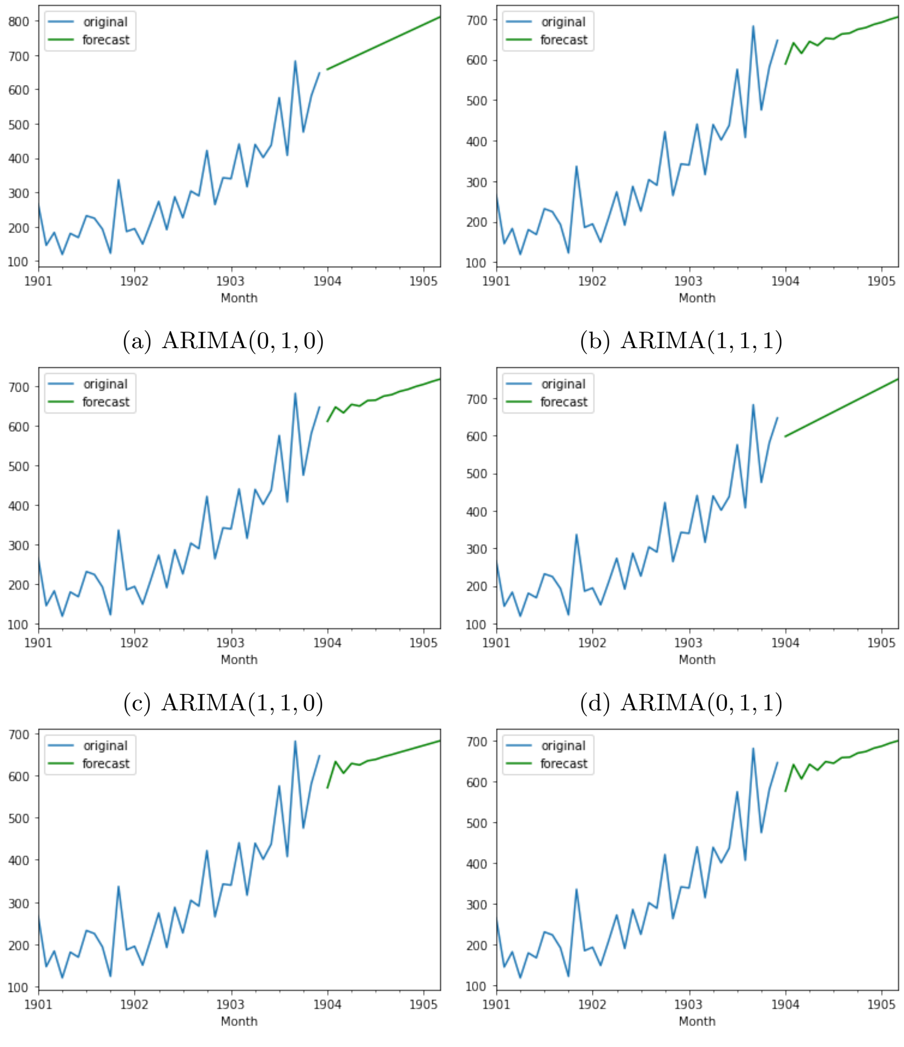
Third step is to replace t with $T + 1$:

$$y_{T+1} = \mu + (1 + \phi_1)y_T + (\phi_2 - \phi_1)y_{T-1} - \phi_2 y_{T-2} + \theta_1 \varepsilon_{T-1} + \theta_2 \varepsilon_{T-2}.$$

To predict further into the future say at time $T + 2$, then just repeat step 3 which produces

$$y_{T+2} = \mu + (1 + \phi_1)y_{T+1} + (\phi_2 - \phi_1)y_T - \phi_2 y_{T-1} + \theta_2 \varepsilon_{T-1}.$$

Observe that there should be a $\theta_1 \varepsilon_T$ term but since we set the time series value at T to be equal to the predicted value, this error term evaluates to 0. We are essentially taking our predicted value as the truth which is why all the error terms eventually become 0. Hence, if you predict $S > q$ future time steps, all MA terms disappear.



(a) ARIMA$(0, 1, 0)$

(b) ARIMA$(1, 1, 1)$

(c) ARIMA$(1, 1, 0)$

(d) ARIMA$(0, 1, 1)$

(e) ARIMA$(2, 1, 1)$

(f) ARIMA$(1, 1, 2)$

Figure 12: Forecasting for 15 future time steps from different ARIMA models. (c) and (d) are equivalent to an AR(1) and MA(1) with 1 order of differencing.

4 Seasonal models

Many time series data in the real work exhibit some form of seasonality. This includes higher network traffic during peak hours of the day or increased spending during Christmas. As such, there is a defined frequency of seasonality commonly denoted as m. For example $m = 4$ could represent quarterly data and $m = 12$ for monthly data. We will again explore the ARIMA and Exponential Smoothing algorithms and how they were extended to account for seasonality. For these examples, we will use the monthly milk dataset. This dataset is great since there is a clear seasonal pattern and the data is trending upward.

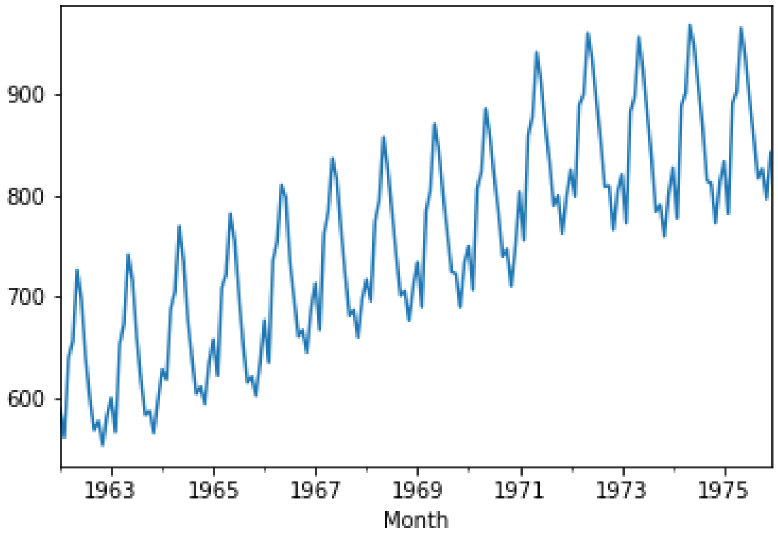


Figure 13: Milk dataset

4.1 Holt-Winters' Seasonal Method

Holt and Winters improved Holt's linear method to capture seasonality. This method includes a level equation, trend equation, seasonal equation, and forecasting equation. It introduces a new smoothing parameter γ and m initial seasonal values that have to be estimated. This method comes in two flavours, an additive and multiplicative method. Which one you pick is dependent on your data. The additive method is preferred when seasonal variations and trend change at a constant rate. Peaks and valleys in your data are roughly of the same magnitude. In contrast, multiplicative is preferred when the changes are proportional to the level of the series.

4.1.1 Additive Method

$$\begin{aligned}
\text{Level equation} \qquad & \ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend equation} \qquad & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
\text{Seasonal equation} \qquad & s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \\
\text{Forecast equation} \qquad & \hat{y}_{t+h} = \ell_t + hb_t + s_{t+h-m(k+1)},
\end{aligned}$$

where k is defined as $\lfloor \frac{h-1}{m} \rfloor$. This guarantees that that every forecast uses the seasonal component from the most recently observed season.

4.1.2 Multiplicative Method

$$\begin{aligned}
\text{Level equation} \qquad & \ell_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend equation} \qquad & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
\text{Seasonal equation} \qquad & s_t = \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m} \\
\text{Forecast equation} \qquad & \hat{y}_{t+h} = (\ell_t + hb_t)s_{t+h-m(k+1)},
\end{aligned}$$

4.1.3 Initial Values for Trend and Seasonal Components

Similar to the non seasonal method, values can be estimated for the initial trend and initial seasonal components to speed up the optimisation process.

Initial Trend

For Holt Linear, we simply used the first two data points for the initial trend. Since we have seasonal data, we can make a better estimate by taking the average trend between the first two seasons:

$$b_0 = \frac{1}{m}\left(\frac{y_{m+1} - y_1}{m} + \frac{y_{m+2} - y_2}{m} + ... + \frac{y_{m+m} - y_m}{m}\right)$$

Initial Seasonal Components

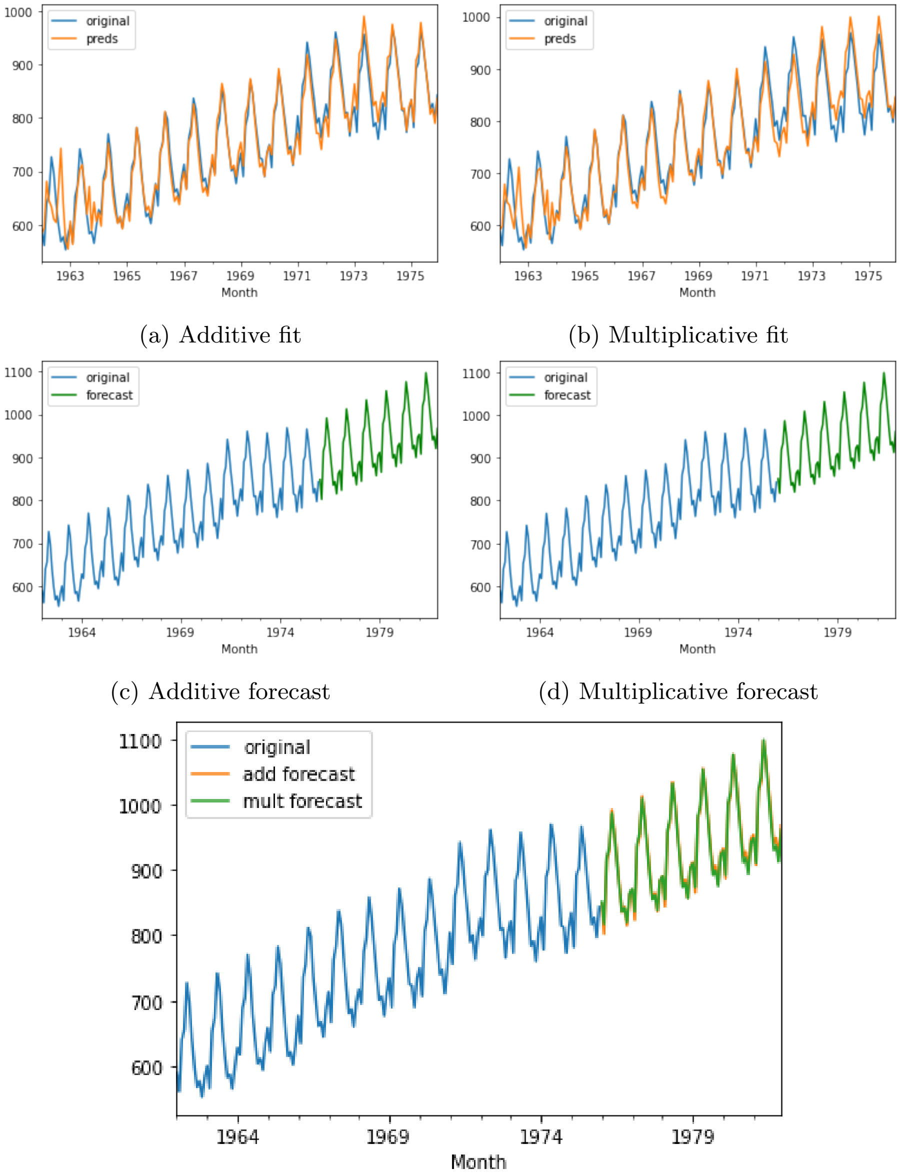
Calculating the initial seasonal components is done in a relatively similar manner. First we compute the averages A_n for each of the N years,

$$A_n = \frac{\sum_{i=1}^{m} y_i}{m}, \qquad n = 1, 2, ..., N.$$

The next step depends on whether the additive or multiplicative method is used. For additive, we sum up the differences between the time associated data points in each season with their respective yearly average and finally divide by N,

$$\begin{aligned} s_1 &= \frac{1}{N}\left[(y_1 - A_1) + (y_{m+1} - A_2) + ... + (y_{m(N-1)+1} - A_N)\right] \\ s_2 &= \frac{1}{N}\left[(y_2 - A_1) + (y_{m+2} - A_2) + ... + (y_{m(N-1)+2} - A_N)\right] \\ &\vdots \end{aligned}$$

For multiplicative, instead of taking the difference, we divide the data point by the respective average, i.e. $\frac{y_1}{A_1}$.



(a) Additive fit

(b) Multiplicative fit

(c) Additive forecast

(d) Multiplicative forecast

(e) Comparitive forecast

Figure 14: Fits and forecasts for Holt-Winters' method

Looking at the comparitive forecast, the differences between additive and multiplicative become apparent. The difference between the peaks and dips for each season become larger as time goes on whereas it stays constant in the additive case.

4.2 Seasonal ARIMA

If non-seasonal ARIMA wasn't complicated enough, there exists a seasonal extension which introduces 3 new parameters. A seasonal ARIMA model has two components, non-seasonal and seasonal, which are both structurally the same. Each have their own AR, MA and order of differencing parameters. Combined with non-seasonal differencing, taking a seasonal difference can in some instances help make the data more stationary. Seasonal differencing is simply $y'_t = y_t - y_{t-m}$. Seasonal ARIMA models can be written as

$$ARIMA(p,d,q) \times (P,D,Q)_m$$

where P, D, and Q are the seasonal AR (SAR) terms, seasonal differencing order, and seasonal MA (SMA) terms respectively with m being the period.

Using backshift notation, the entire equation can be written as:

$$(1 - \phi_1 B - ... - \phi_p B^p)(1 - \Phi_1 B - ...\Phi_P B^{mP})(1-B)^d(1-B)^D y_t = \\ \mu + (1 + \theta_1 B + ...\theta_q B^q)(1 + \Theta_1 B + ...\Theta_Q B^{mQ})\varepsilon_t.$$

Using the milk data, we pick $m = 12$ since the data is monthly. First let's figure out our order of non-seasonal and seasonal orders of differencing. We can first try 1 order of seasonal differencing and looking at the ACF and PACF graphs.

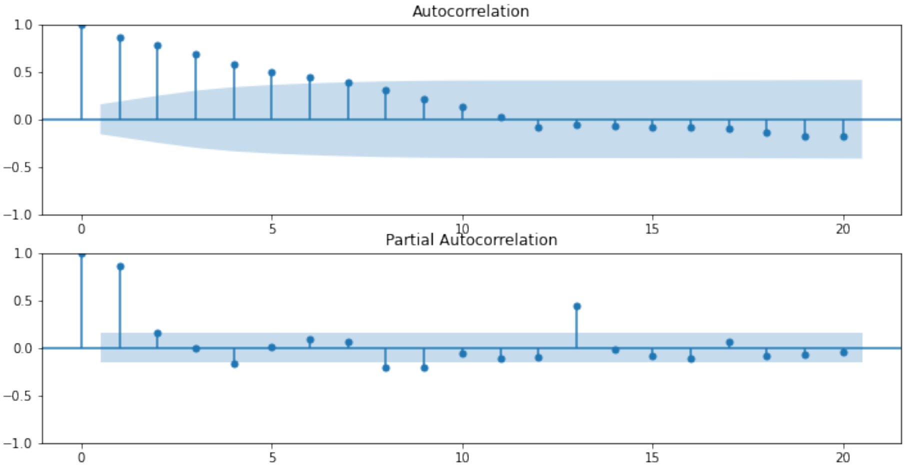


Figure 15: ACF and PACF for seasonal differenced data

The slow decrease in the ACF signifies and p-value $\approx 0.1608 >= 0.05$ indicates that we need more orders of differencing. This time we will try include a non-seasonal difference.

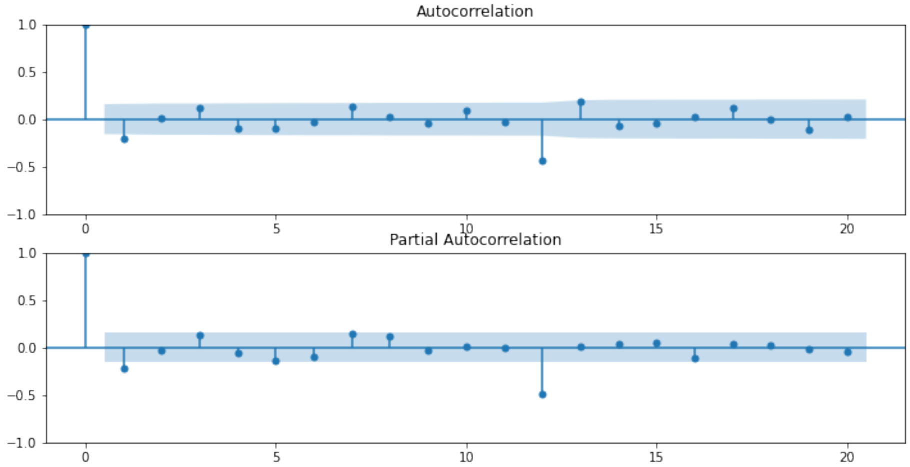


Figure 16: ACF and PACF for seasonal differenced + non seasonal differenced data

This looks a lot better. With a p-value $\approx 1.8654 \times 10^{-5} < 0.05$ we can be happy with our orders of differencing being $d = 1$ and $D = 1$. Note, by the equation, the order in which you difference the series, either seasonal or non-seasonal first, does not make a difference to the values of the overall differenced series.

Once again, Dr Nau has 2 rules in picking these parameters:

"Rule 12: If the series has a strong and consistent seasonal pattern, then you should use an order of seasonal differencing–but never use more than one order of seasonal differencing or more than 2 orders of total differencing (seasonal+nonseasonal)."

"Rule 13: If the autocorrelation at the seasonal period is positive, consider adding an SAR term to the model. If the autocorrelation at the seasonal period is negative, consider adding an SMA term to the model. Try to avoid mixing SAR and SMA terms in the same model, and avoid using more than one of either kind."

Combining these and the previous rules, it may be wise to pick a SARIMA$(1, 1, 1) \times (0, 1, 1)_{12}$ model which comes out with an AIC = 1068.064. Running this through the auto_arima function, surprisingly yields a SARIMA$(2, 0, 0) \times (0, 1, 1)_{12}$ model with an AIC = 1076.388. Both models fit the data extremely well and the forecasts are able to replicate the rise and fall pattern.

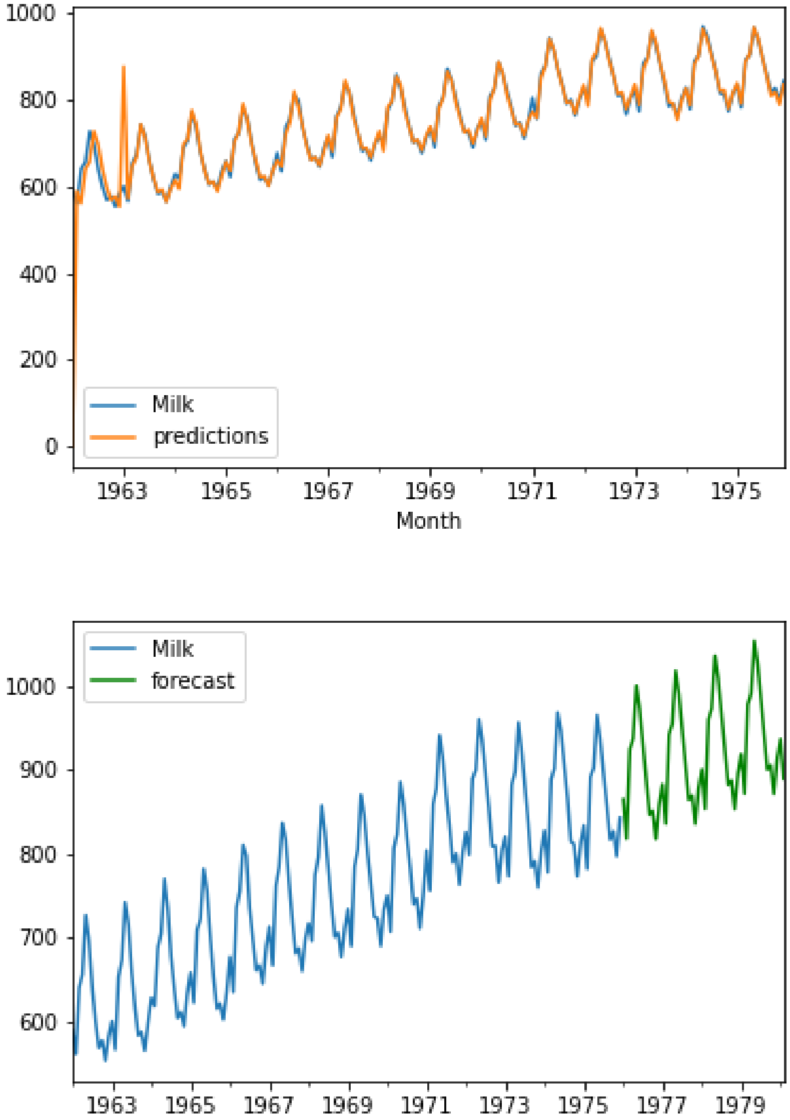
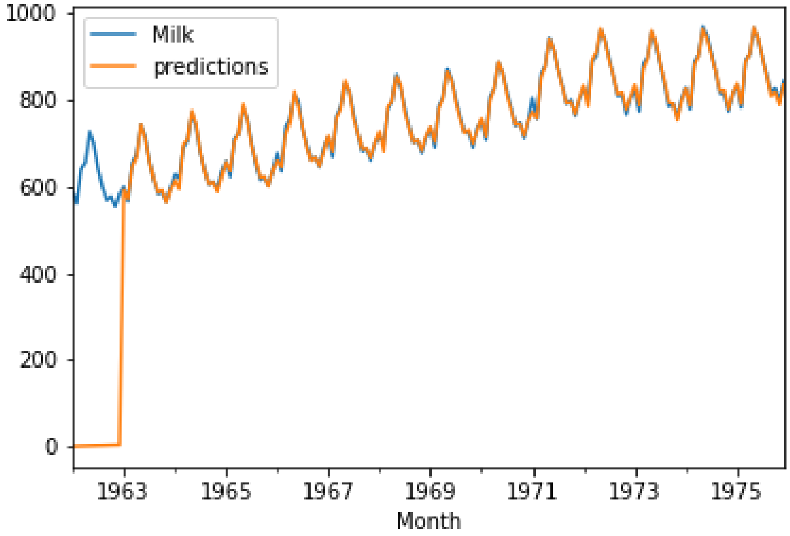


Figure 17: SARIMA$(1, 1, 1) \times (0, 1, 1)_{12}$ fit and forecasts



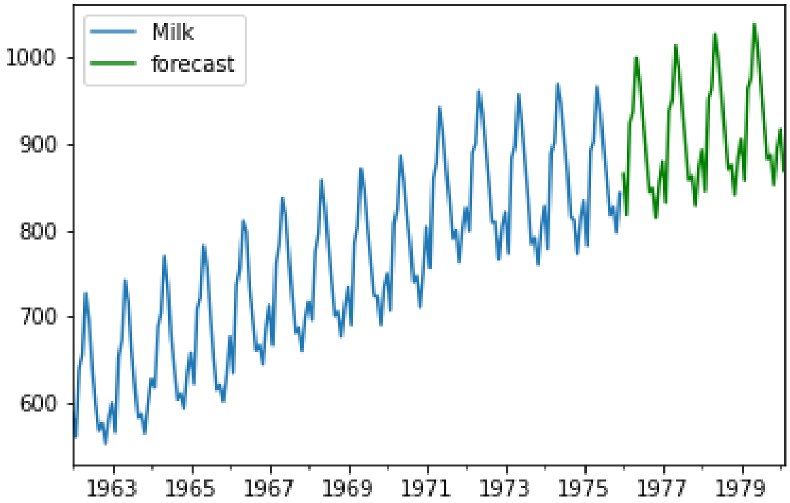


Figure 18: SARIMA$(2, 0, 0) \times (0, 1, 1)_{12}$ fit and forecasts

5 Conclusion

6 Resources

https://www.probabilitycourse.com/chapter10/10_1_4_stationary_processes.php https://www.statology.org/detrend-data/

bibliography[1]

References

[1] ABC News. "Fact check: Do more people die in Australia than Sweden due to poorly heated homes?" In: *JAMA Dermatol* (2018). DOI: https://www.abc.net.au/news/2017-08-09/fact-check-australia-sweden-winter-fatalities-heating/8780588. (accessed: 29.09.2021).