EMET 3007 Major Project

Forcasting Monthly Marriages In Australia

U6662757
Anurag Yadav
November 10, 2022

Introduction

In this report we will be using Australian Bureau of Statistics (ABS) marriage data to forecast the number of monthly marriages from 2020 December onwards and a link to the data can be found in the bibliography. The latest release only contained data up to January 2016 and I had to manually find the data for previous periods by finding older releases. It should be noted for data integrity purpose that there was a change in the method that the data was collected in 2013 January onwards but looking at the raw data no obvious outliers/anomalies in the data can be found. Furthermore, in assessing the reliability of the data source, the ABS in an independent government funded national statistic agency and has no conflict of interest in providing false or misleading data so we can conclude that the data source is reliable. The next release of the data is scheduled for the 10th November 11.30 PM.

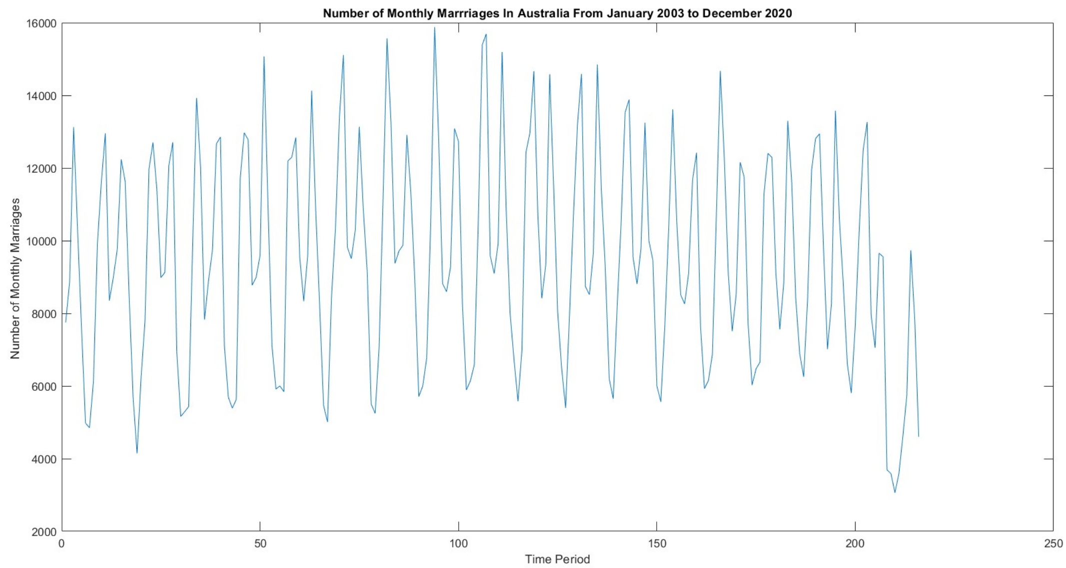


Figure 1: Raw Data

Figure 1 shows the raw data of the number of monthly marriages, in which we can see 3 clear trends. The first is that there is clear seasonality in the data in which certain months have a higher than average number of marriages and certain months consistently have a lower number of marriages than the average. The second trend is that from 2003 to 2011 (Time period 94) the upper range and general trend for the number of marriages in constantly increasing but after this time period there is a clear decrease in the number of marriages. Finally, the last major trend in the data is the clear decrease in the number of overall marriages after time period 200. This is clearly the negative effects of the coronavirus pandemic which caused people to isolate and prohibited large gatherings i.e. (weddings) thus the drastic decreases in the number of marriages.

Drawing on this I would propose two possible hypothesis which will be discussed in the final report after the data has been released. First is that due to COVID-19 there was an increase in restrictions on large gatherings and travelling therefore many couples delayed getting married as seen in the clear decrease in the overall number of marriages therefore as restrictions ease up in the future data releases we should see a higher than normal increase in the number of marriages overall. Alternatively, the second hypothesis proposes the current trend of overall decrease will continue and we should be pessimistic when creating models favouring those model who better fit the downward trend.

Additionally, when conducting initial exploratory data analysis I noticed an interesting pattern in the data, such that months which have a larger number of public holidays consistently have higher than average number of weddings. This could be used an a potential explanatory variable by future researchers.

Covariance Stationarity

A lot of the models which where explored in class has the assumption that the probabilistic structure of the data is not changing over time. In other words first that the mean is constant over time and secondly that the covariance between different elements in the sequence depends of how far apart they are not where in the sequence they are.

$$1.E(y_t) = \mu$$
$$2.Cov(y_t, y_{t-s}) = \gamma(s)$$

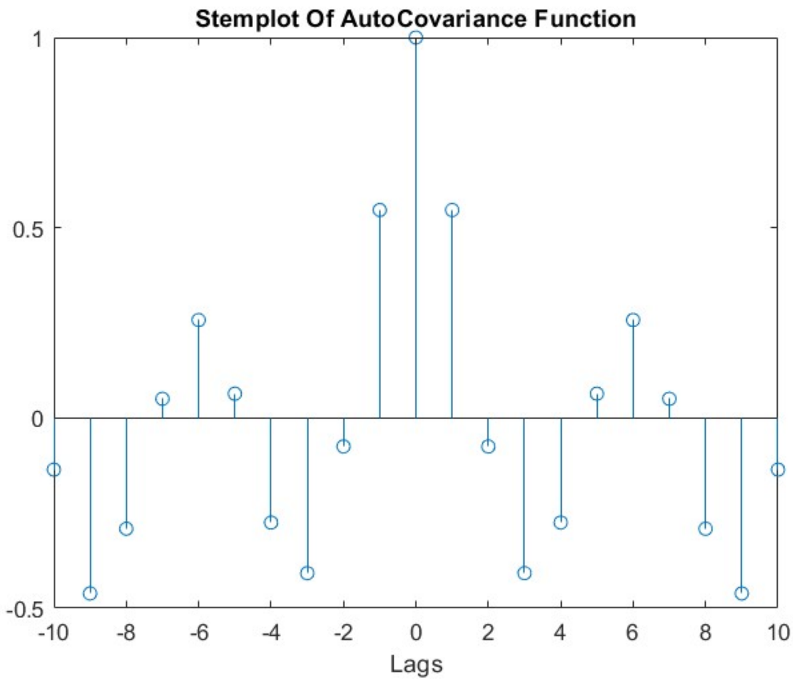


Figure 2: Sample AutoCovariance Function

To make this data covariance stationary we begin by taking the log of the dataset and then we also take the first difference as well.

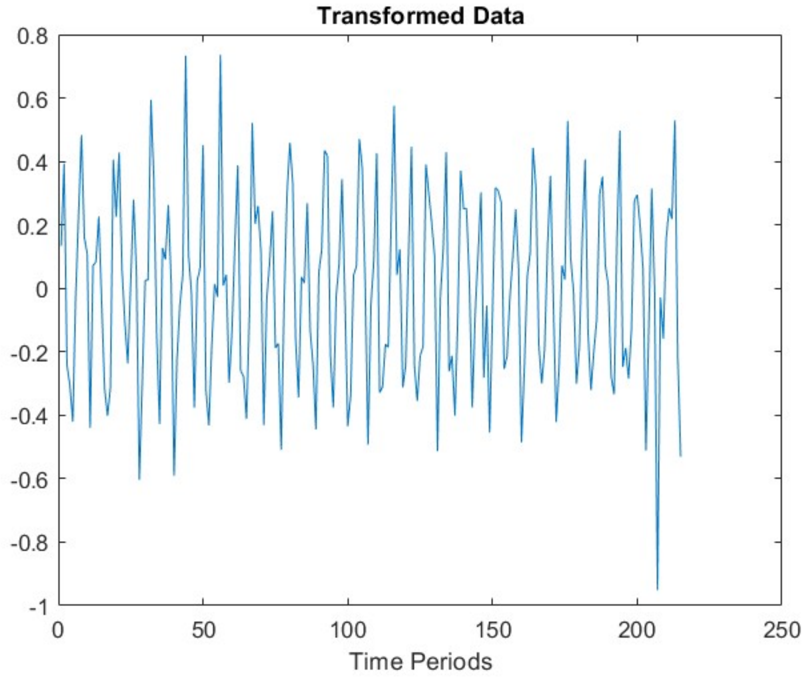


Figure 3: Transformed Data

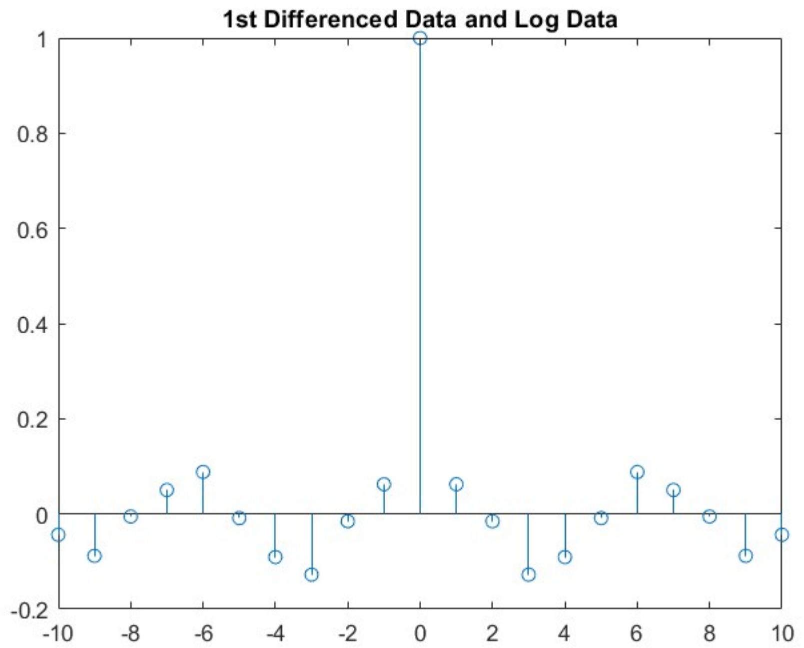


Figure 4: AutoCovariance Function of Transformed Data

Looking at figure 3,4 we can see that both criteria for covariance stationarity have now been satisfied. Also looking at Figure 4 it is difficult to ascertain weather the data is following an AR or a MA process. An MA process would have the closest lags from 0 be large values while everything elese would be close to zero, while an AR process would have desending pattern in which values decrease the further away you get from 0.

Model Selection

Now comes the question of how we decide which is the best model for forecasting monthly marriages, two ideas where explored in class. The first is that we can use In sample fitting, in which we use all of the data to find the optimal values for our parameters which minimize a loss function. The loss function most discussed in class was MSE:

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(y_t - \widehat{y_t})^2$$

The first issue with MSE is that it is prone to overfitting, such that it favours more complex models because they have a smaller MSE and these tend to be poor forecast because we pick up a lot of noise. To combat this issue it was discussed in class that we can use Akaike information criterion (AIC) and the Bayesian information criterion (BIC):

$$AIC = MSE \times T + 2k$$

$$BIC = MSE \times T + klog(T)$$

As we can see from the formula AIC/BIC will penalize model complexity.

The second issue with using in sample fitting methods such as MSE is that we are essentially finding parameters for our model which "best fit" the data, but this is not strictly the same as finding the model which "best forecasts" the data. This is where pseudo out of sample forecasting was discussed in class. In which we use all the data from t = 1 to some point t_0 ,then create parameters from this data and forecast h steps ahead, then we again repeat the process until we reach the end of the dataset and by taking the average of these forecast and comparing them to the actual values we can get the MSFE.

$$\text{MSFE} = \frac{1}{T - h - T_0 + 1}\sum_{t=T_0}^{T-h}(y_{t+h} - \hat{y}_{t+h|t})^2$$

Therefore since this report is evaluated in how accurate the forecast is for future values of the number of monthly marriages, it makes since to use MSFE in this assignment. Additionally we don't have to worry about the overfitting issue as encountered in the MSE above. Furthermore since we are interested in forcasting for the next 12 month period we will be using $h = 1$ and $h = 12$.

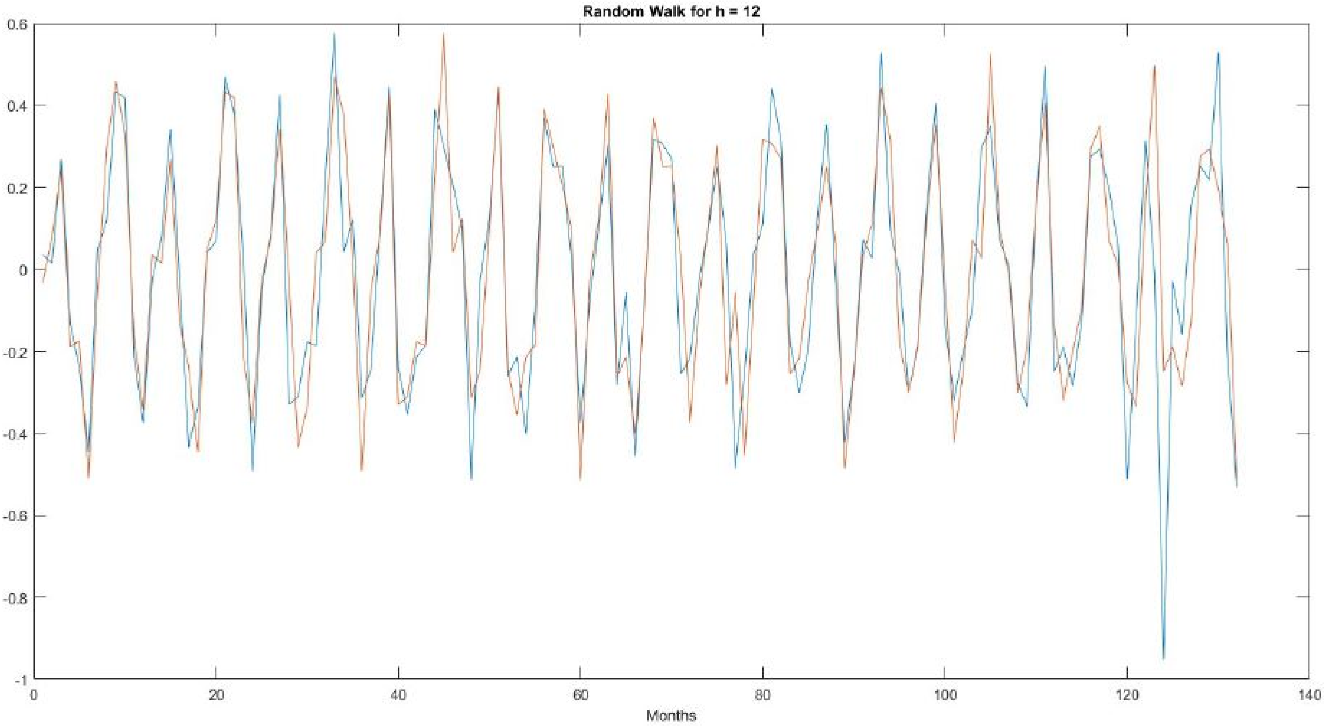
Benchmark Model

It was recommended in class that we use the random walk model as a benchmark for the MSFE and give all other model MSFE as a ratio of the random walk MSFE to make it easier for the marker to interpret the data. Therefore we will be implementing the random walk model here.
Note that in class it was recommended by the lecturer that we use 1/3 of the dataset as our "Training Period" therefore from here onwards $T_0 = 72$.

$$y_t = y_{t-1} + \varepsilon$$

	Random Walk Model	
	h = 1	h = 12
AIC	28.66419	28.66419
BIC	32.03483	32.03483
MSE	0.124019	0.124019
MSFE	0.117146	0.023858

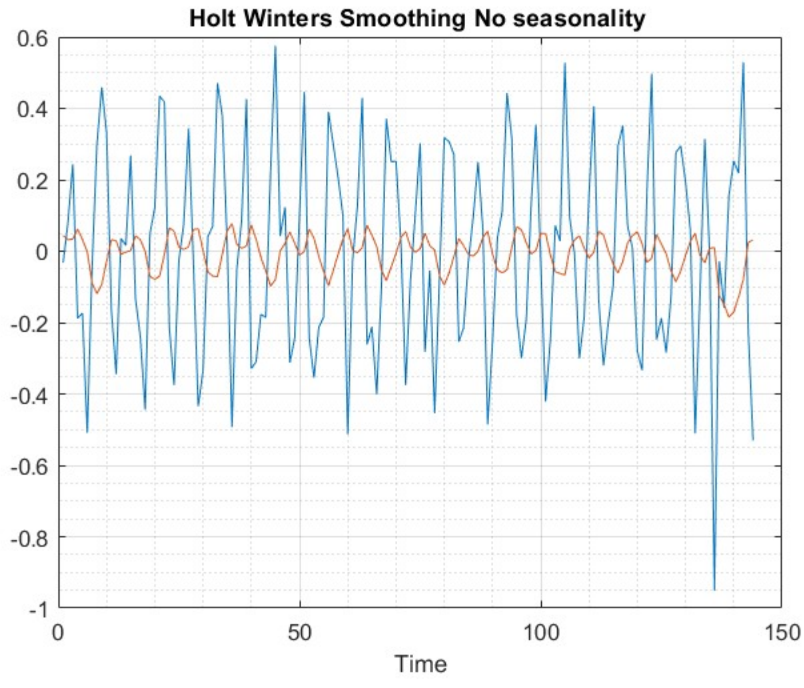
Note that it should be expected that MSE would be the same h = 1 or h = 12, since MSE uses all the data that is available to estimate model parameters. Surprisingly for h = 12 Random walk seems to be a pretty good model but for h= 1 there is room for improvement. For h = 12 the low MSFE also logically seems to make sense we observed previously that marriages where occurring in a yearly pattern where certain months had on average a high number of marriages whilst other months had on average low number of marriages. Therefore by using h = 12 we are just saying for example if there where around 10,000 marriages in the month of January 2019 then there will be exactly 10,000 marriages in the month of January 2020.



We can see from the graph that that Random walk follows the data rather well and the only areas where it fails is when there is a large deviation from the values of the previous years month such as the drop that occurs after month 120.

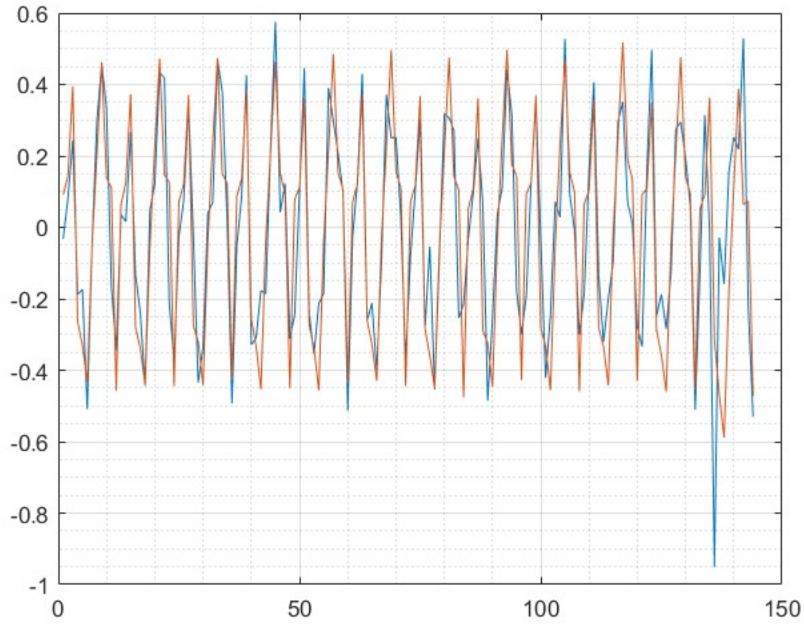
Holt Winters Smoothing Model

Lets begin by applying the Holt Winters smoothing Model with no seasonality and h = 1



Additionally parameters for alpha and beta where found using trial and error method. When alpha = 0.1 and beta = 0.45 we got the MSFE ratio as 0.994 which is not much of an improvement over the benchmark model.

Next lets apply seasonality and optimized values using a recursive algorithm for h = 12 forcast. The optimized values are alpha = 0.057971; beta = 0.927536; gamma = 0.173913; with a MSFE of 0.024 which although is an improvement from the model without seasonality is still less then the random walk.



	Holt Winters	Holt Winters + Seasonality	Random Walk
MSFE(H=1)	0.116507518	0.117070225	0.117146
MSFE(H=12)	0.026323441	0.024578564	0.023858

Note additionally that since Holt Winters is an Ad Hoc method it only offers point forcast.

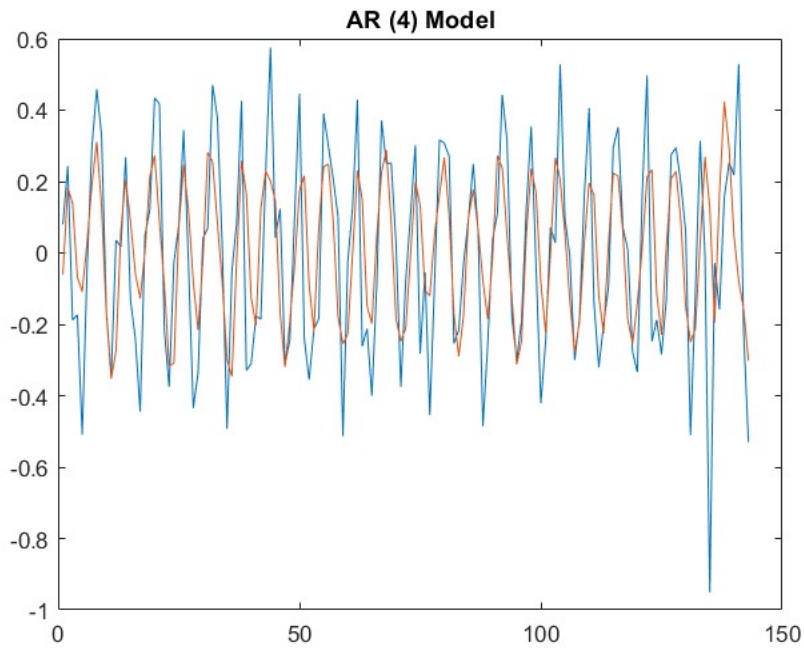
AR(p) Model

Note that for the following model we will use m= 4 lag as was done in class so we can compare the models.

	AR(1)	AR(2)	AR(3)	AR(4)	Random Walk
AIC	18.63364677	18.31992616	17.21743005	20.502505	
BIC	21.98550491	25.02364243	27.27300445	37.261795	
MSE	0.078832449	0.067866949	0.053163176	0.0497749	
MSFE H=1	0.076790787	0.064140895	0.051565553	0.0490393	0.117146
MSFE H=12	0.023151425	0.024023715	0.024233807	0.0227001	0.023858

From the above table above me can see that as complexity is increasing the MSE is decreasing and this is to be expected. Looking at AIC and BIC is not really helpfull in this scenario as AIC is giving preference to AR(2) while BIC is best for AR(1). On the otherhand for h = 1 all models outperfom the random walk while for h = 12 only AR(1) and AR(4) beat it. Overall I think AR(1) might be too simple and AR(4) is the better model.

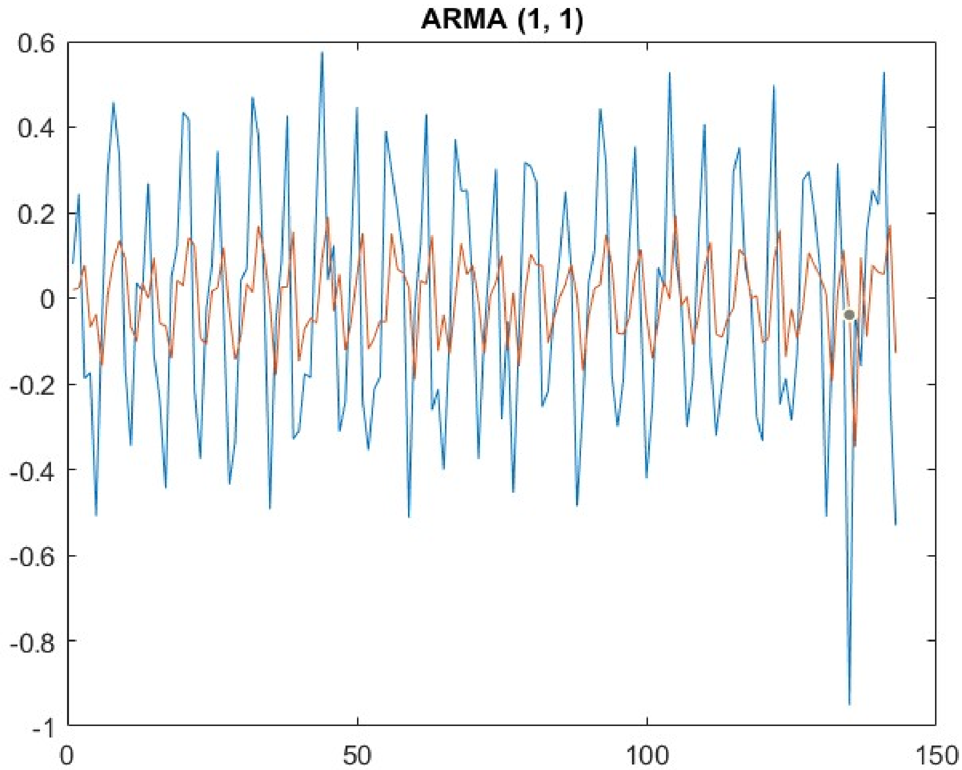
$$y_{t+1} = -0.003 + 0.0783 * y_t - 0.2365 * y_{t-1} - 0.4288 * y_{t-2} - 0.2484 * y_{t-3}$$



ARMA(p,q) Model

	ARMA(1,1)	ARMA(2,1)	ARMA(2,2)	Random Walk
MSFE H=1	0.073348518	0.064140895	0.059039299	0.117146
MSFE H=12	0.046461274	0.054023715	0.048720056	0.023858

I find the results of the ARMA model suprising, due to the nature of the ACF stemplot as discussed earlier, I assumed the data would best be modelled by a mix of both MA and AR process. But none of the ARMA models perform any better then the random walk for h = 12. Although for h = 1 ARMA(2,2) model had the lowest MSFE.



State Space Model

The state space models where covered in the final week of this course and for this assignment we will be implementing the Unobserved Component model which is a simple state space model.

$$y_t = \tau_t + \epsilon_t$$

$$\epsilon \sim N(0, \sigma^2)$$

y_t here represents what we see ie the number of marriages in a month, while τ_t is the actual true underlying value plus some ϵ_t which is a noisy error term. This should intuitively make sense because when we collect data there is always some noise for example measurement error plus the true underlying data.

The issue with this model is that it has too many parameters, for example we have τ_1 to τ_T and we also have σ^2 as parameters, thus you can see we have more parameters then data!. To combat this we assume some of the values (Vh and omega) to be known.

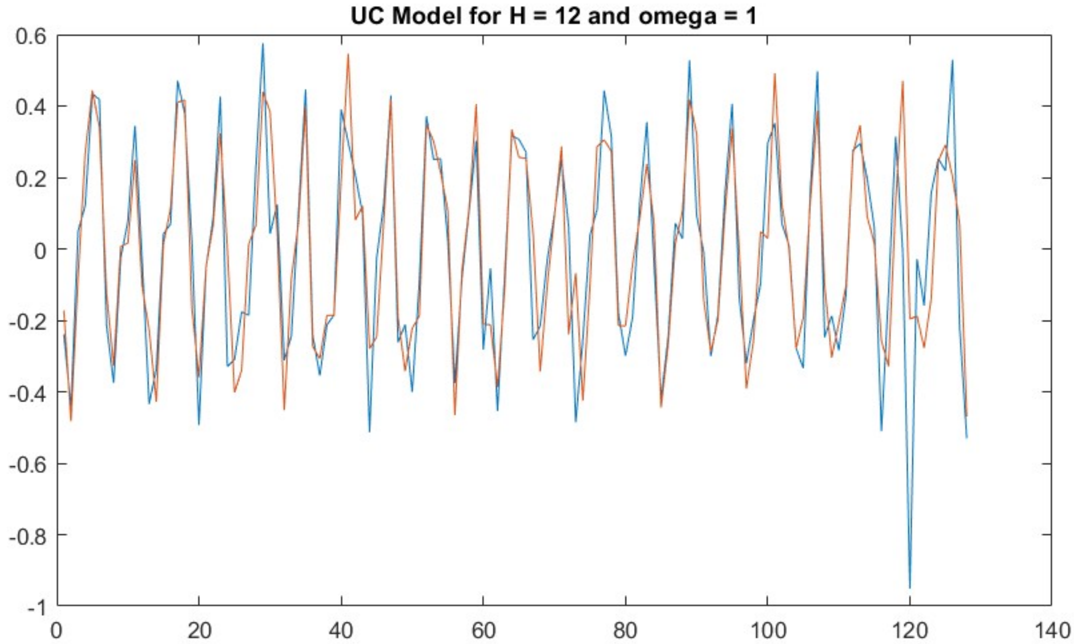
For this model we will assume a random walk specification as was done in the lectures and this should produce a relatively good forcast considering when looking at raw data the number of marriages by month is related such that in January you normally have x number of marriages plus some variation higher/lower than x for the next year.

$$\textit{measurement equation}: y_t = \tau_t + \epsilon_t$$

$$\textit{transition equation}: \tau_t = \tau_{t-1} + u_t$$

To choose the values for Vh since it is only used in the first instant τ_1 so given a large enough training period it has no real influence on the final model. In regards to picking omega it governs how quickly/slowly τ changes therefore we chose a range of values between 0.25 and 1.

	omega = 0.25	omega = 0.5	omega = 0.75	omega = 1
MSFE(H=1)	0.116507518	0.117070225	0.117415201	0.1176548
MSFE(H=12)	0.026323441	0.024578564	0.024235251	0.0241522



Overall visually it seems to me that the UC model best fits the data compared to all previous models. Although the MSFE is suprising since for h = 1 only omega = 0.25 and omega = 0.5 out perform the benchmark model and for h = 12 the MSFE of none of the models outperforms the benchmark.

Conclusion

To conclude the best model in terms of MSFE was the AR(4) model

As y_{T+1} is normal, the 95% confidence interval is $\mathbf{x}_{T+1}\hat{\boldsymbol{\beta}} \pm 1.96\hat{\sigma}$

In regards to density forcasting the values for y_{t+h} should lie between [-0.1692919, -0.36058797] where $h = 1$. Next looking at the 1 year ahead forcast the next 12 months where calculated in MATLAB and are shown below.

Variables - yhatA

yhatAR4_T

12x1 double

	1
1	-0.2649
2	0.0677
3	0.3460
4	0.2514
5	-0.0326
6	-0.2310
7	-0.2055
8	-0.0107
9	0.1527
10	0.1556
11	0.0267
12	-0.1016
13	

The data has been differenced and logged and these things will need to be undone when doing the final next part of this assingment.

Bibliography

1. Forcasting - Diebold: Book
2. ABS Website Data: https://www.abs.gov.au/statistics/people/people-and-communities/marriages-and-divorces-australia/2020)
3. Jackson Micallef - Student

Code

```
y = xlsread("data.xlsx");

plot(y)
%%
[cov_y, lags_y] = xcov(y, 10, 'coeff');
stem(lags_y, cov_y); title('Data');
%%

y = xlsread("data.xlsx");
y = log(y)
T = length(y);
L = spdiags(ones(T-1, 1), -1, T, T);
I = speye(T);
%%

D = I - L;
y = D*y;
[cov_ylog, lags_ylog] = xcov(y, 10, 'coeff');
stem(lags_ylog, cov_ylog); title('1st Differenced Data and Log Data')
y = y(2:end);
% plot(y(2:end))
save('fdata.mat',"y");
z = load('fdata.mat');

%%
y = z.y;
T0 = 72;
T = length(y);
h = 12;
ytph = y(T0+h:end);
syhat = y(T0: T-h);
MSE_rw = mean((y(2:end) - y(1:T-1)).^2);
AIC_rw = MSE_rw*T + 1*2;
BIC_rw = MSE_rw*T + 1*log(T);
MSFE_rw = mean((ytph - syhat).^2);
plot(ytph)
hold on
plot(syhat); xlabel('Months'); ylabel('# of Marriages and logged and
differenced)')
hold off

%% Holt winters no seasonality
y = z.y;
T = length(y);
h = 12;
T0 = 72-h;
alpha = 0.1; beta = 0.45;
ytph = y(T0+h:end);
syhat = zeros(T-h-T0+1, 1);
Lt = y(1); bt = y(2) - y(1);
plot(ytph)
hold on
for t = 2:T-h
 newLt = alpha*y(t) + (1-alpha)*(Lt + bt);
 newbt = beta*(newLt - Lt) + (1 - beta)*bt;
 yhat = newLt + h*newbt;
 Lt = newLt;
 bt = newbt;
 if t >= T0
```

```
 syhat(t-T0+1, :) = yhat;
 end
end
periods = [T0+h:T]';
plot(syhat);
MSFE_HW1 = mean((ytph - syhat).^2);
hold off
grid on
grid minor

%% holt winters with seasonality = 12

y = z.y;
T = length(y);
h = 1;
T0 = 72-h;
iterable = linspace(0,1,70);
St = zeros(T-h,1);
ytph = y(T0+h:end);
s = 12;
% initialize
Lt = mean(y(1:s)); bt = 0; St(1:12) = y(1:s) - Lt;
MSFE1 = [100]
syhat = zeros(T-h-T0+1, 1);
for alphai = iterable
 for betai = iterable
 for gammai = iterable
 for t = s+1:T-h
 newLt = alphai*(y(t) - St(t-s)) + (1-alphai)*(Lt+bt);
 newbt = betai*(newLt-Lt) + (1-betai)*bt;
 St(t) = gammai*(y(t)-newLt) + (1-gammai)*St(t-s);
 yhat = newLt + h*newbt + St(t+h-s);
 Lt = newLt; bt = newbt; % update Lt and bt
 if t>= T0
 syhat(t-T0+1,:) = yhat;
MSFE2 = mean((ytph-syhat).^2);
if MSFE2 < MSFE1
 MSFE1 = MSFE2;
 txt = sprintf('alpha = %f' ,alphai)
 txt = sprintf('beta = %f' ,betai)
 txt = sprintf('gamma = %f' ,gammai)
 MSFE1
 end
 end
 end
 end
 end
end


y = z.y
T = length(y);
h = 1;
T0 = 72-h;
s = 12;
syhat_3 = zeros(T-h-T0+1,1);
ytph_3 = y(T0+h:end); % observed y {t+h}
alpha = 0.057971; beta = 0.927536; gamma = 0; % smoothing parameters
St = zeros(T-h,1);
```

```
Lt = mean(y(1:s)); bt = 0; St(1:12) = y(1:s) - Lt;
for t = s+1:T-h
 newLt = alpha*(y(t) - St(t-s)) + (1-alpha)*(Lt+bt);
 newbt = beta*(newLt-Lt) + (1-beta)*bt;
 St(t) = gamma*(y(t)-newLt) + (1-gamma)*St(t-s);
 yhat = newLt + h*newbt + St(t+h-s);
 Lt = newLt; bt = newbt; % update Lt and bt
 if t>= T0
 syhat_3(t-T0+1,:) = yhat;
 end
end
MSFE_HW2 = mean((ytph_3-syhat_3).^2);
plot(ytph_3)
hold on
plot(syhat_3)
hold off
grid on
grid minor

%% AR1

%% Running an AR(1) model
z = load('fdata.mat');
m = 4; h = 12;
y = z.y;
T0 = 72 - 4;
y0 = y(1:m); y = y(m+1:end); T = length(y);
ytph = y(T0+h: end);
yhatAR = zeros(T-h-T0+1, 1);
for t = T0:T-h
 yt = y(h:t);
 xt = [ones(t-h+1, 1) [y0(m); y(1:t-h)]];
 betahat = (xt'*xt)\(xt'*yt);
 yhatAR(t-T0+1,:) = [1 y(t) ]*betahat;
end
MSFE_AR1 = mean((ytph - yhatAR).^2);
plot(ytph); title('AR (1) Model');
hold on
plot(yhatAR)
hold off
Y = y(2:end);
X = [ones(T-1, 1) y(1:end-1)];
betahatAR1 = (X'*X)\(X'*Y);
yhat = X*betahatAR1;
MSE_AR1 = mean((yhat - Y).^2);
k = 1;
AIC_AR1 = MSE_AR1*T + k*2;
BIC_AR1 = MSE_AR1*T + k*log(T);
sig1 = ((Y - yhat)'*(Y - yhat))/T;
yhatAR1_T=zeros(12,1);
yhatAR1_T(1) = [1 y(end)]*betahatAR1;
for i =2:12
yhatAR1_T(i) = [1 yhatAR1_T(i-1)]*betahatAR1;
end

%% Running an AR(2) model
z = load('fdata.mat');
m = 4; h = 12;
y = z.y;
```

```
T0 = 72 - 4;
y0 = y(1:m); y = y(m+1:end); T = length(y);
ytph = y(T0+h: end);
yhatAR2 = zeros(T-h-T0+1, 1);
for t = T0:T-h
 yt = y(h:t);
 xt = [ones(t-h+1, 1) [y0(m); y(1:t-h)] [y0(m-1:end); y(1:t-h-1)] ];
 betahat2 = (xt'*xt)\(xt'*yt);
 yhatAR2(t-T0+1,:) = [1 y(t) y(t-1) ]*betahat2;
end
MSFE_AR2 = mean((ytph - yhatAR2).^2);
plot(ytph); title('AR (2) Model');
hold on
plot(yhatAR2)
hold off
Y = y(3:end);
X = [ones(T-2, 1) y(2:T-1) y(1:T-2)];
betahatAR2 = (X'*X)\(X'*Y);
yhat = X*betahatAR2;
MSE_AR2 = mean((yhat - Y).^2);
k = 2;
AIC_AR2= MSE_AR2*T + k*2;
BIC_AR2 = MSE_AR2*T + k*log(T);
yhatAR2_T=zeros(12,1);
yhatAR2_T(1) = [1 y(end) y(end-1)]*betahatAR2;
yhatAR2_T(2) = [1 yhatAR2_T(1) y(end)]*betahatAR2;
for i =3:12
yhatAR2_T(i) = [1 yhatAR2_T(i-1) yhatAR2_T(i-2)]*betahatAR2;
end
sig2 = ((Y - yhat)'*(Y - yhat))/T;
%% Running AR(3) model
z = load('fdata.mat');
m = 4; h = 12;
y = z.y;
T0 = 72 - m;
y0 = y(1:m); y = y(m+1:end); T = length(y);
ytph = y(T0+h: end);
yhatAR3 = zeros(T-h-T0+1, 1);
for t = T0:T-h
 yt = y(h:t);
 xt = [ones(t-h+1, 1) [y0(m); y(1:t-h)] [y0(m-1:end); y(1:t-h-1)] [y0(m-2:end);
y(1:t-h-2)] ];
 betahat3 = (xt'*xt)\(xt'*yt);
 yhatAR3(t-T0+1,:) = [1 y(t) y(t-1) y(t-2) ]*betahat3;
end
MSFE_AR3 = mean((ytph - yhatAR3).^2);
plot(ytph); title('AR (3) Model');
hold on
plot(yhatAR3)
hold off
Y = y(4:end);
X = [ones(T-3, 1) y(3:T-1) y(2:T-2) y(1:T-3)];
betahatAR3 = (X'*X)\(X'*Y);
yhat = X*betahatAR3;
MSE_AR3 = mean((yhat - Y).^2);
k = 3;
AIC_AR3= MSE_AR3*T + k*2;
BIC_AR3 = MSE_AR3*T + k*log(T);
sig3 = ((Y - yhat)'*(Y - yhat))/T;
```

```matlab
%% Running AR(4) model
z = load('fdata.mat');
m = 4; h = 1;
y = z.y;
T0 = 72 - m;
y0 = y(1:m); y = y(m+1:end); T = length(y);
ytph = y(T0+h: end);
yhatAR4 = zeros(T-h-T0+1, 1);
for t = T0:T-h
 yt = y(h:t);
 xt = [ones(t-h+1, 1) [y0(m); y(1:t-h)] [y0(m-1:end); y(1:t-h-1)] [y0(m-2:end);
y(1:t-h-2)] [y0(m-3:end); y(1:t-h-3)] ];
 betahat4 = (xt'*xt)\(xt'*yt);
 yhatAR4(t-T0+1,:) = [1 y(t) y(t-1) y(t-2) y(t-3) ]*betahat4;
end
MSFE_AR4 = mean((ytph - yhatAR4).^2);
plot(ytph); title('AR (4) Model');
hold on
plot(yhatAR4)
hold off
Y = y(5:end);
X = [ones(T-4, 1) y(4:T-1) y(3:T-2) y(2:T-3) y(1:T-4) ];
betahatAR4 = (X'*X)\(X'*Y);
yhat = X*betahatAR4;
MSE_AR4 = mean((yhat - Y).^2);
k = 5;
AIC_AR4= MSE_AR4*T + k*2;
BIC_AR4 = MSE_AR4*T + k*log(T);
yhatAR4_T=zeros(12,1);
yhatAR4_T(1) = [1 y(end) y(end-1) y(end-2) y(end-3)]*betahatAR4;
yhatAR4_T(2) = [1 yhatAR4_T(1) y(end) y(end-1) y(end-2)]*betahatAR4;
yhatAR4_T(3) = [1 yhatAR4_T(2) yhatAR4_T(1) y(end) y(end-1)]*betahatAR4;
yhatAR4_T(4) = [1 yhatAR4_T(3) yhatAR4_T(2) yhatAR4_T(1) y(end)]*betahatAR4;
for i = 5:12
yhatAR4_T(i) = [1 yhatAR4_T(i-1) yhatAR4_T(i-2) yhatAR4_T(i-3) yhatAR4_T(i-
4)]*betahatAR4;
end
sig4 = ((Y - yhat)'*(Y - yhat))/T;

%% Running an ARMA(p, q) Model
z = load('fdata.mat');
m = 4; h = 12;
y = z.y;
T0 = 72 - m;
y0 = y(1:m); y = y(m+1:end); T = length(y);
yhatARMA11 = zeros(T-h-T0+1, 1);
 % theta = [phi1 mu psi];
f = @(theta) loglike_ARMA11(theta, y0, y(1:T0));
thetahat = fminsearch(f, [0.5, 0, 0.5]);
for t = T0:T-h
 yt = y(1:t);
 % find the MLE
 f = @(theta) loglike_ARMA11(theta, y0, yt);
 thetahat = fminsearch(f, thetahat);
 % make the uhats
 Gam = speye(t) + spdiags(ones(t-1,1),[-1],t,t)*thetahat(3);
 X = [[y0(m); y(1:t-1)] ones(t, 1)];
 uhat = Gam\(yt-X*thetahat(1:2)');
 % store the forecasts
```

```
 yhatARMA11(t-T0+1, :) = thetahat(2) + thetahat(1)*y(t) + thetahat(3)*uhat(end);
end
ytph = y(T0+h:end);
MSFE_ARMA11 = mean((ytph - yhatARMA11).^2);

% Error Estimates for ARMA(1,1)
m = 4; h = 1;
y = z.y;
T0 = 72 - m;
y0 = y(1:m); y = y(m+1:end); T = length(y);
Y = y(2:end);
X = [ones(T-1, 1) y(1:end-1)];
betahatARMA11 = (X'*X)\(X'*Y);
yhat = X*betahatARMA11;
MSE_ARMA11 = mean((yhat - Y).^2);
k = 3;
AIC_ARMA11 = MSE_ARMA11*T + k*2;
BIC_ARMA11 = MSE_ARMA11*T + k*log(T);
plot(ytph)
hold on
plot(yhatARMA11); title('ARMA (1, 1)');
hold off

z = load('fdata.mat');
y = z.y;
m = 4; %% the first m points as lags
y0 = y(1:m); y = y(m+1:end);
T = length(y);
Vtau = 10; %% initial condition
omega = 1; %% fix omega

T0 = 72;
h = 12; % h-step-ahead forecast
yhatUC = zeros(T-h-T0+1,1);
ytph = y(T0+h:end); % observed y {t+h}
sighat = 1;

for t = T0:T-h
yt = y(1:t);
f = @(sig) cologlike_UC(sig,omega,yt,Vtau);
sighat = fminsearch(f,sighat);
H = speye(t) - spdiags(ones(t-1,1),-1,t,t);
invOmega =sparse(1:t,1:t,[1/Vtau 1/omega*ones(1,t-1)],t,t);
HinvOmegaH = H'*invOmega*H;
K = speye(t)/sighat + HinvOmegaH;
tauhat = K\(yt/sighat);
yhatUC(t-T0+1) = tauhat(end); %store the forecasts
end

MSFE_UC = mean((ytph - yhatUC).^2);

plot(ytph); title('UC Model for H = 12 and omega = 1');
hold on
plot(yhatUC)
hold off
```

```
%% functions
function ell = loglike_ARMA11(x, y0, y)
phi = zeros(2, 1);
phi(1) = x(1); phi(2) = x(2); psi = x(3); % phi(2) is mu
N = length(y); m = length(y0);
A = speye(N); B = sparse(2:N, 1:N-1, ones(1, N-1), N, N);
Gam = A + B*psi; Gam2 = Gam*Gam';
X = [[y0(m); y(1:N-1)] ones(N, 1)];
 % transformed ell removing constant term
ell = -(y-X*phi)'*(Gam2\(y-X*phi));
ell = -ell;%% Negative of the log-likelihood for ARMA(1,1)
end

function ell = loglike_ARMA21(x, y0, y)
phi = zeros(3, 1);
phi(1) = x(1); phi(2) = x(2); phi(3) = x(3); psi = x(4); % phi(3) is mu
N = length(y); m = length(y0);
A = speye(N); B = sparse(2:N, 1:N-1, ones(1, N-1), N, N);
Gam = A + B*psi; Gam2 = Gam*Gam';
X = [[y0(m); y(1:N-1)] [y0(m-1:end); y(1:N-2)] ones(N, 1)];
 % transformed ell removing constant term
ell = -(y-X*phi)'*(Gam2\(y-X*phi));
ell = -ell;
end


function ell = cologlike_UC(sig,omega,y,Vtau)
T = length(y);
H = speye(T) - sparse(2:T,1:(T-1),ones(1,T-1),T,T);
invOmega = sparse(1:T,1:T,[1/Vtau 1/omega*ones(1,T-1)]);
HinvOmegaH = H'*invOmega*H;
tauhat = 1/sig*(1/sig*speye(T) + sig*HinvOmegaH)\y;
err = (y-tauhat)'*(y-tauhat)/sig + tauhat'*HinvOmegaH*tauhat;
ell = -T/2*log(sig) - (T-1)/2*log(omega) - .5*err;
ell = -ell;
end
```