Personal project

Data description

There are 217 observations recorded as of the making of this report (October 2020) and the latest observation pertains to July 2020. The next two observations will be released on the 9th of October (pertains to August data) and the 10th of November (pertains to September data) respectively. Since the requirement of this report asks for forecast of the observation released in November 2020, the forecast is going to be a two-step ahead forecast. The forecast pertains to the September 2020 observation.

Problems with the time-series

Upward trend: The original series shows an upward trend. To remove the trend and improve stationarity, I took the first difference (figure 2). Throughout the report, I will be using the original series as well as the first difference as the basis for forecast.

Seasonality: Autocovariance plot of the original series (figure 3) show seasonality for data points 12 months apart. In the simple regression model, I will be using “month” and “quarter” as dummy variables.

Throughout the report, I will be using MSFE as the sole measure of forecast accuracy.

To improve the accuracy of the data, I also consider using building approval (A422064L) as exogenous explanatory variables. In addition to the simple regression model mentioned before, I will also be adding building approval as explanatory variables. I will also use ARMAX model where building approval is an exogenous explanatory variable.

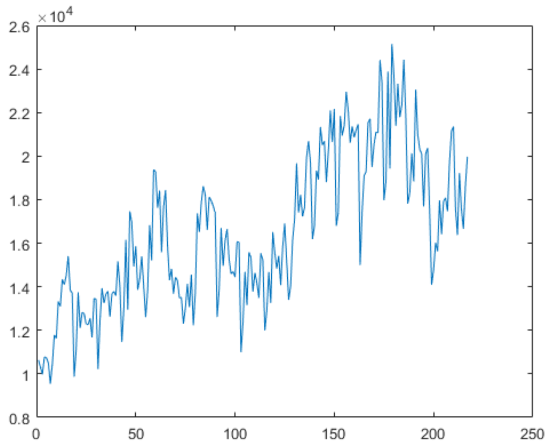


Figure 1: Original time series

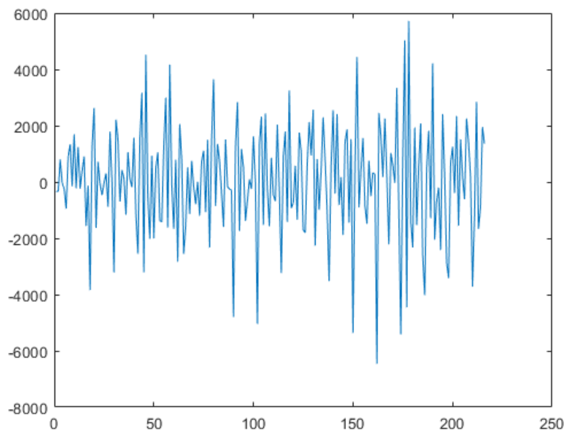


Figure 2: First difference

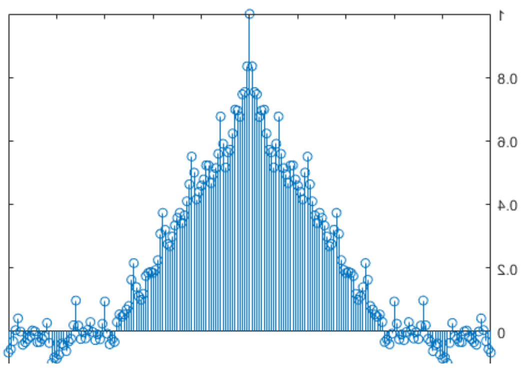


Figure 3: Autocovariance plot or the original time series

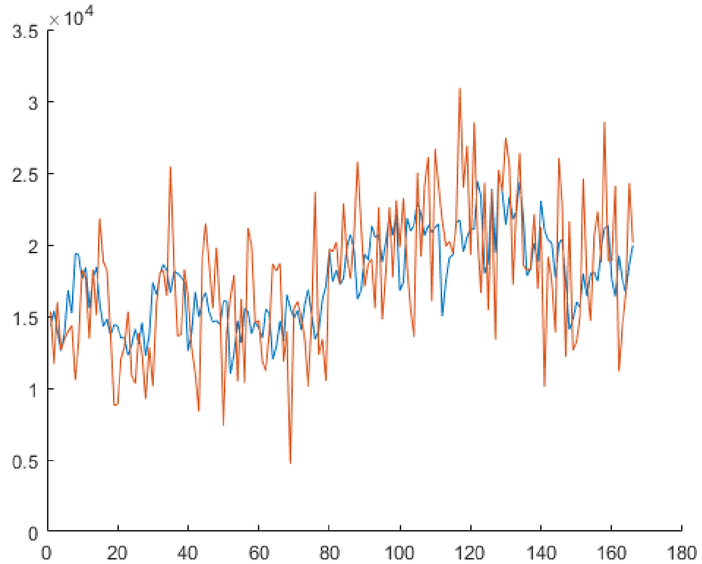
Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Model used

I first conduct a random walk forecast on the data. This will serve as a benchmark for all the later forecasts.

1. The first batch of models I use are regression models with monthly and quarterly dummy variables only (no quantitative variable). S
2. The second batch of models I use are regression models similar to 1, but with building approval as a quantitative explanatory variable.
3. The third model I use is Holt-Winters smoothing without adjustment to increase in the seasonality parameter
4. The fourth model is HW smoothing model with adjustment to increase in the seasonality parameter (Same as we did in the week 6 lab)
5. The fifth model is IAR on the first difference, with up to 3 lags
6. The sixth model is IMA model on first difference, with up to 3 lags
7. The seventh model is ARMA model on the original series with up to 3 lags
8. The eighth model is AMAR model on the original series with up to 3 lags
9. The ninth model is ARMAX model with up to 3 lags

Random walk



	Random walk
MSFE	2.5099e+07

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Month and quarter as explanatory dummy variables

$$S1: y_t = a_0 + a_1 t + a_4 D_{4t} + u_t;$$

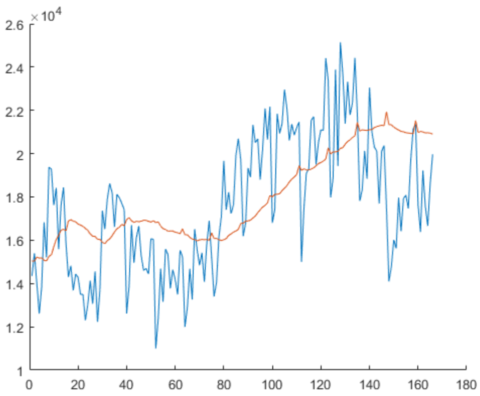
$$S2: y_t = a_1 t + \sum_{i=1}^{4} a_i D_{it} + u_t;$$

$$S3: y_t = a_0 + a_1 t + a_{12} M_{12,t} + u_t;$$

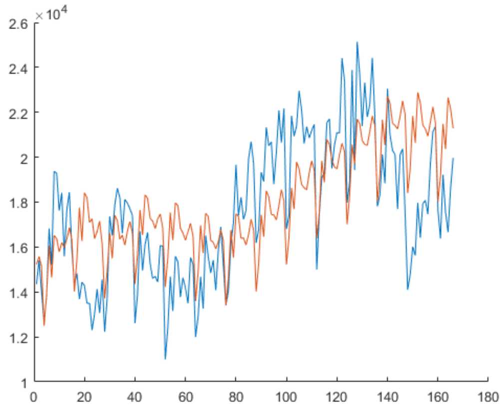
$$S4: y_t = a_1 t + \sum_{i=1}^{12} a_i M_{it} + u_t;$$

The model that uses all four quarters as explanatory dummies (S2) has the highest forecast accuracy, with MSFE of 5.7068e+06.

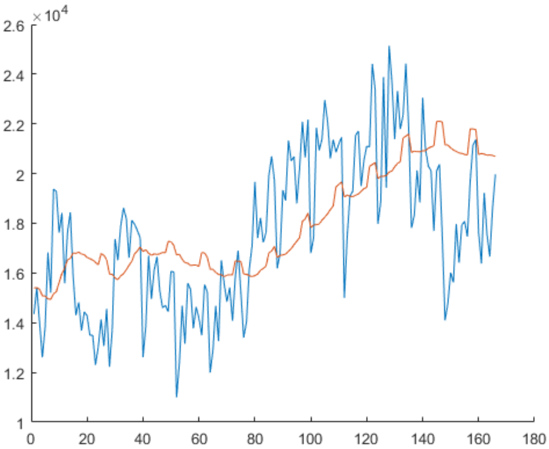
	S1	S2	S3	S4	RW
MSFE	7.0694e+06	5.7068e+06	6.8486e+06	6.3107e+06	2.5099e+07



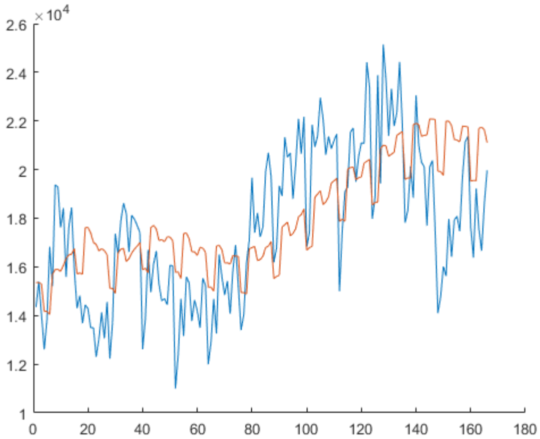
S1



S2



S3



S4

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Using month and quarter as explanatory dummy variables and building approval as explanatory variable

$$S1': y_t = a_0 + a_1 t + a_4 D_{4t} + \beta * Approval_{t-h} + u_t;$$

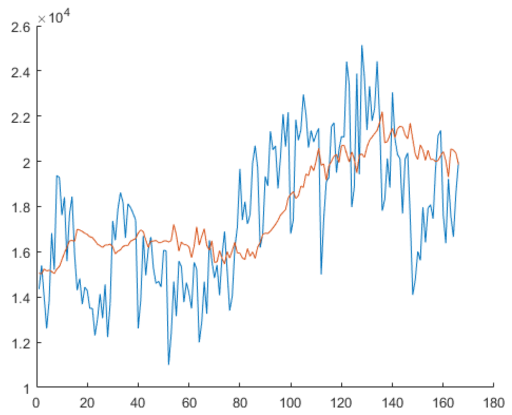
$$S2': y_t = a_1 t + \sum_{i=1}^{4} a_i D_{it} + \beta * Approval_{t-h} + u_t;$$

$$S3': y_t = a_0 + a_1 t + a_{12} M_{12,t} + \beta * Approval_{t-h} + u_t;$$

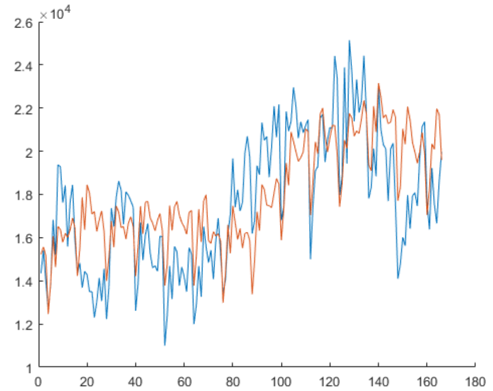
$$S4': y_t = a_1 t + \sum_{i=1}^{12} a_i M_{it} + \beta * Approval_{t-h} + u_t;$$

Adding building approval as explanatory somewhat improves forecast accuracy, with S2 still being the best model. Adding the additional variable has reduced S2's MSFE from 5.7068e+06 to 4.9057e+06.

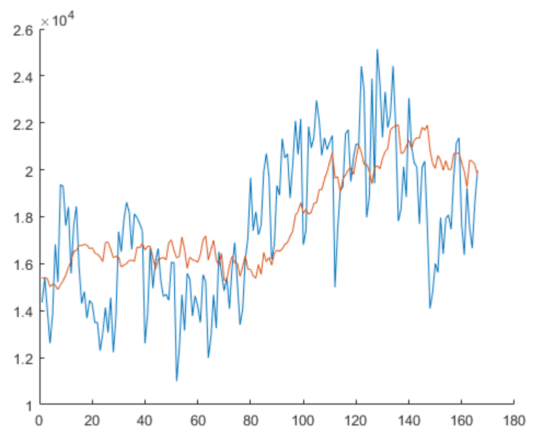
	S1'	S2'	S3'	S4'	RW
MSFE	6.4737e+06	4.9057e+06	6.3762e+06	6.0116e+06	2.5099e+07



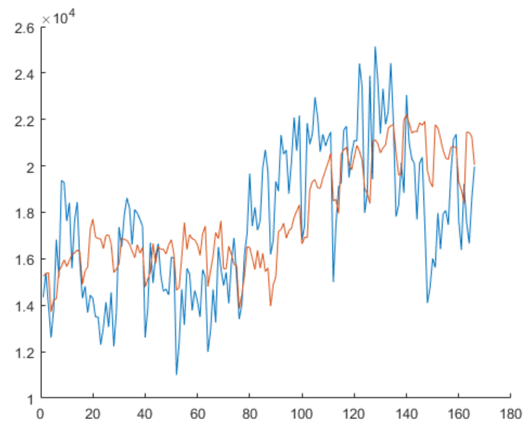
S1'



S2'



S3'



S4'

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Holt winters smoothing

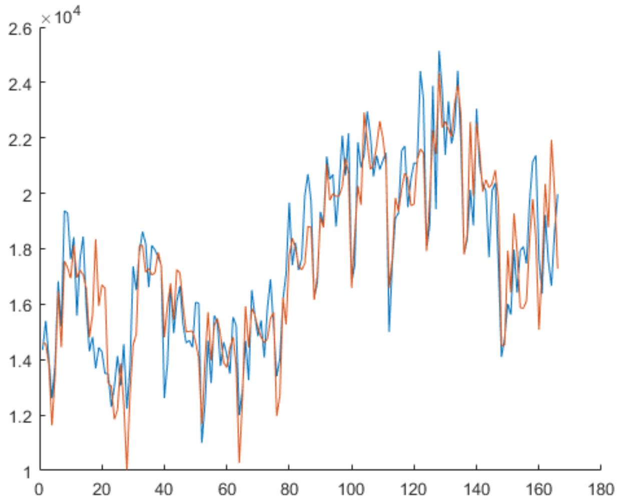
I use Holt winters forecasting method on both the original data series as well as the first difference. I used the version of Holt Winters method that considers seasonality for both data series.

When forecasting using the original data series, I considered adding an incremental term to the seasonality parameter, like what we did in tutorial. Adding the additional term assumes that the seasonal variation has a trend in December. This trend could either be upward or downward depending on the sign of term “a”.

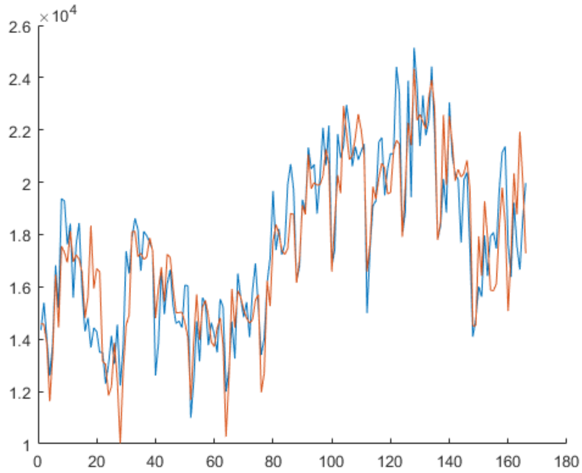
$$St = \begin{cases} \gamma(y_t - L_t) + (1-\gamma)S_{t-s} + a;\ for\ t\ in\ December \\ \gamma(y_t - L_t) + (1-\gamma)S_{t-s}; otherwise \end{cases}$$

After optimizing the coefficients, I found no improvement in forecasting accuracy after adding the extra term “a” to the seasonality parameter.

	HW without seasonality parameter adjustment	HW with seasonality parameter adjustment	RW
MSFE	1.8370e+06	1.8370e+06	2.5099e+07



Holt winters without seasonal adjustment



Holt winters with seasonal adjustment

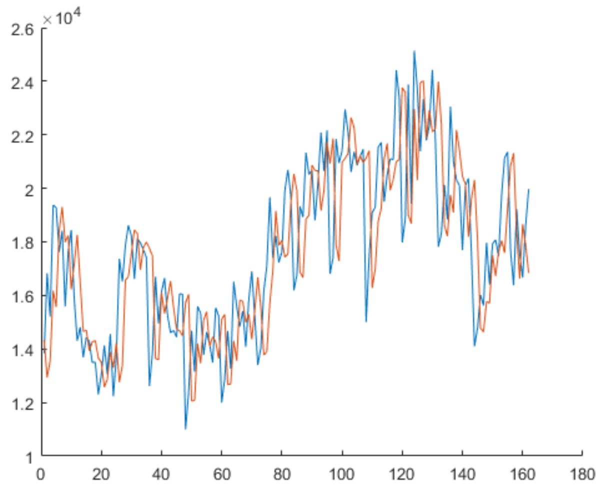
Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

IAR model

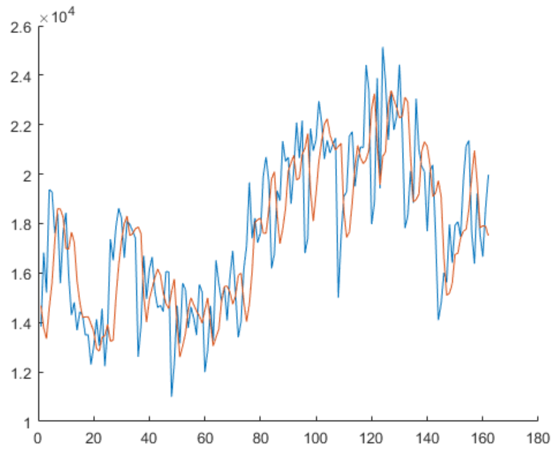
$$IAR(p)\colon \Delta y_t = \rho_1 \Delta y_{t-1} + \cdots + \rho_3 \Delta y_{t-3} + u_t;$$

The IAR models are systematically better than almost all regression models. This is with the exception of S2' (uses four quarterly dummies and building approval as explanatory variable), which has MSFE of 4.9057e+06

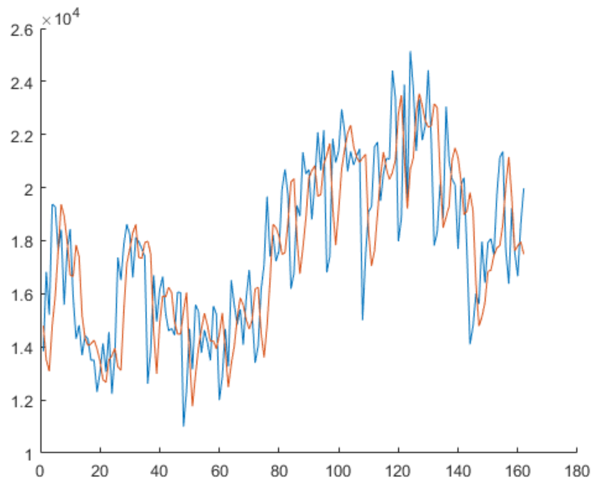
	IAR(1,1)	IAR(1,2)	IAR(1,3)	RW
MSFE	5.6598e+06	5.2361e+06	5.4737e+06	2.5099e+07



IAR(1,1)



IAR(1,2)



IAR(1,3)

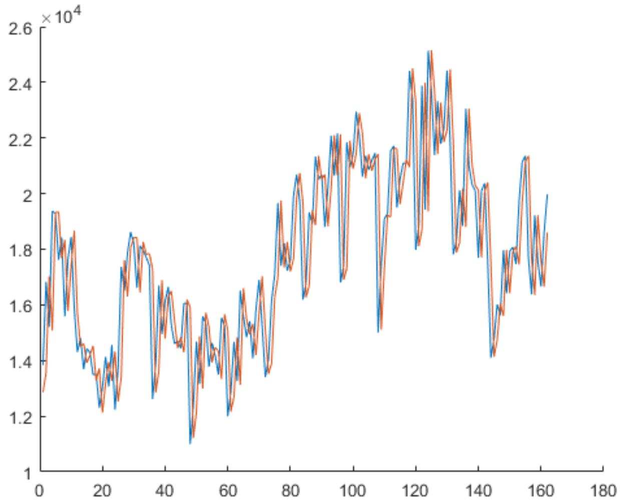
Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

IMA(1, p) model

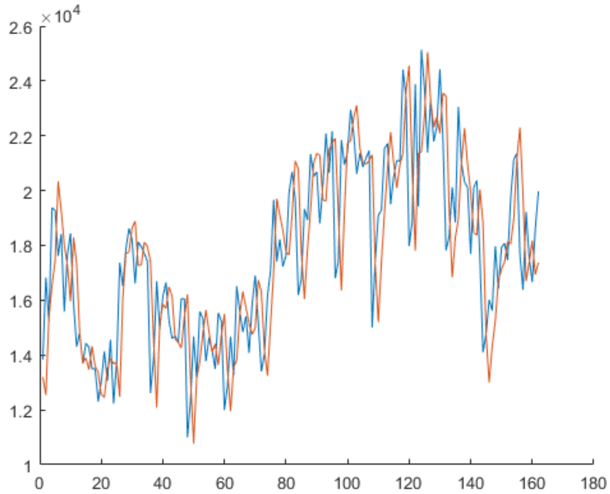
$$IMA(p, 1): \ \Delta y_t = u_t + \varphi_1 u_{t-1} + \dots + \varphi_3 u_{t-3};$$

IMA models are systematically better than IAR models, with IMA(1,2) performing the best at MSFE of 4.6633e+06.

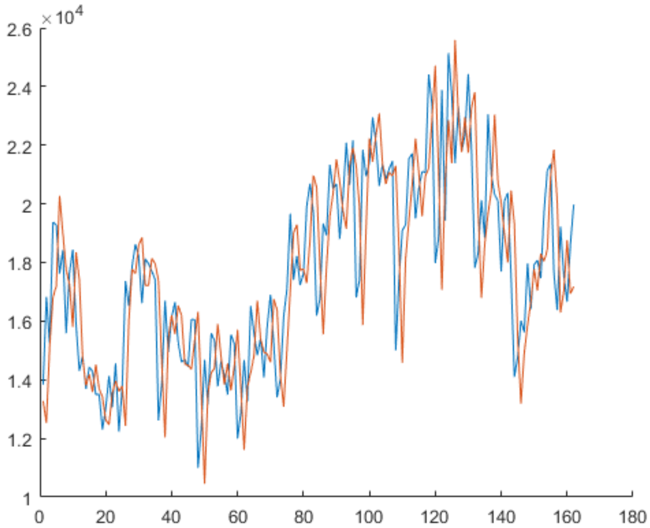
	IMA(1,1)	IMA(1,2)	IMA(1,3)	RW
MSFE	4.7950e+06	4.6633e+06	4.8344e+06	2.5099e+07



IMA(1,1)



IMA(1,2)



IMA(1,3)

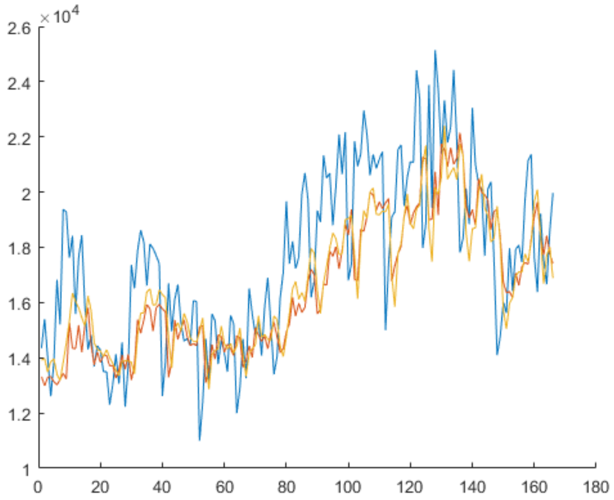
Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

ARMA(p, 2) model

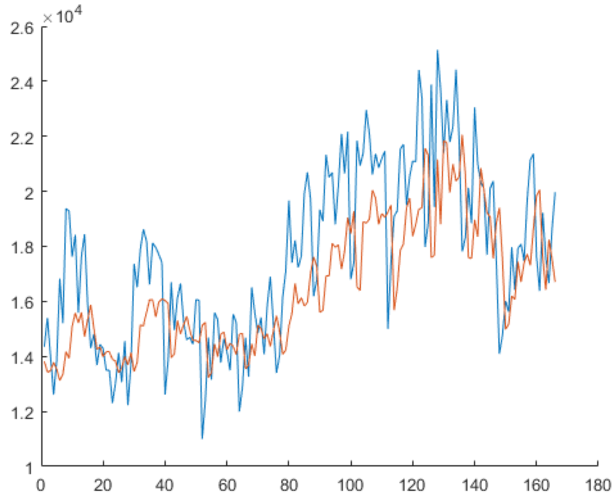
$$ARMA(p,q): y_t = \mu + \rho_1 y_{t-1} + \cdots + \rho_p y_{t-p} + u_t + \varphi_1 u_{t-1} + \ldots + \varphi_q u_{t-q};$$

Interestingly, despite added complexity, ARMA models only represent marginal improvement over simple regression models. All ARMA models are not on par, in terms of forecasting accuracy, with AR counterparts.

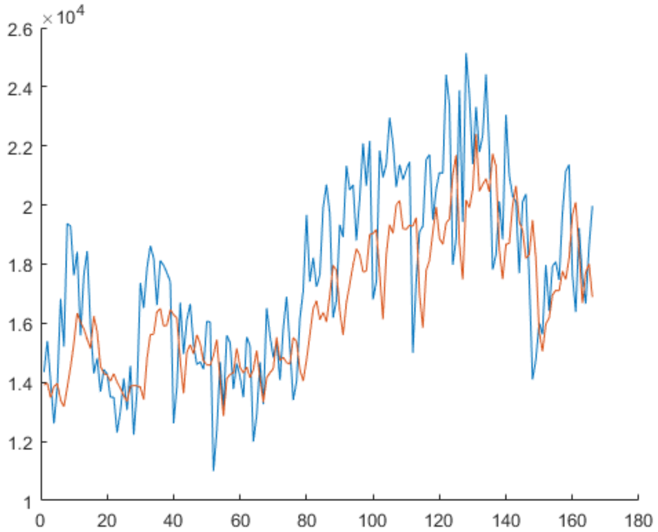
	ARMA(1,2)	ARMA(2,2)	ARMA(3, 2)	RW
MSFE	6.1079e+06	6.1489e+06	5.6699e+06	2.5099e+07



ARMA(1,2)



ARMA(2,2)



ARMA(3,2)

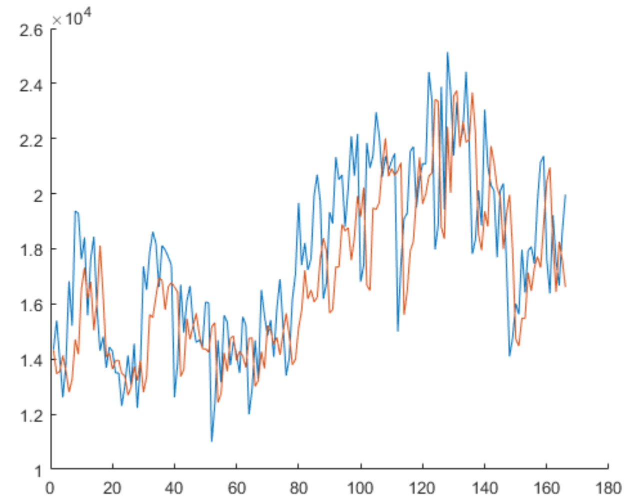
Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

ARAR(p, q) model

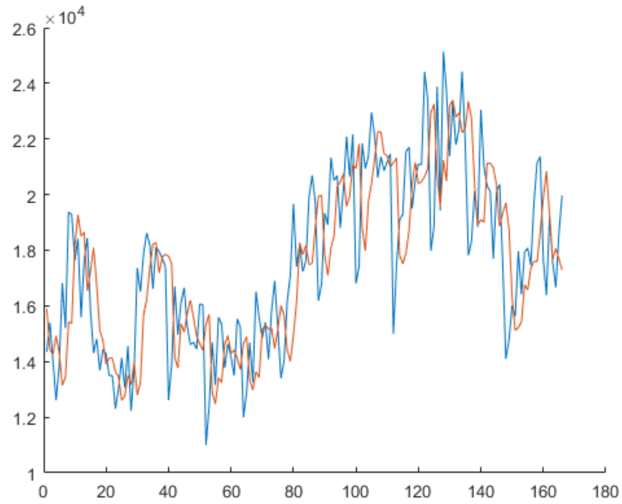
$$ARAR(p,q): y_t = \mu + \rho_1 y_{t-1} + \cdots + \rho_p y_{t-p} + e_t;\ e_t = \theta_1 e_{t-1} + \cdots + \theta_{1_p} e_{t-p} + u_t$$

ARAR models are only comparable to AR models, despite being more complex.

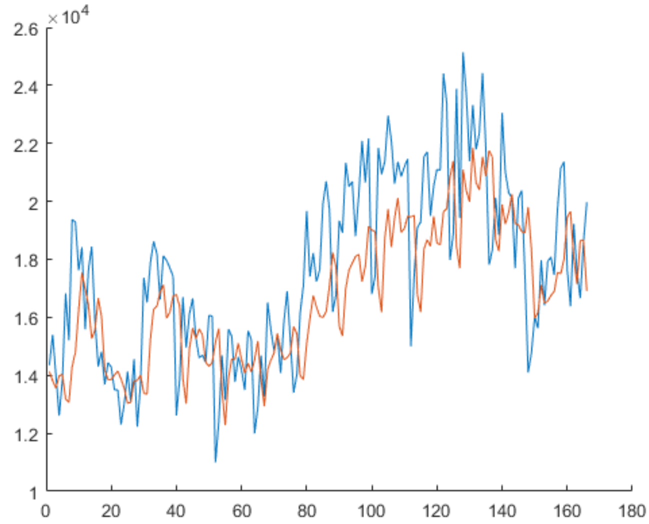
	ARAR(1,2)	ARAR(2,2)	ARRA(3,2)	RW
MSFE	6.0064e+06	5.2742e+06	5.5629e+06	2.5099e+07



ARAR (1,2)



ARAR (2,2)



ARAR (3,2)

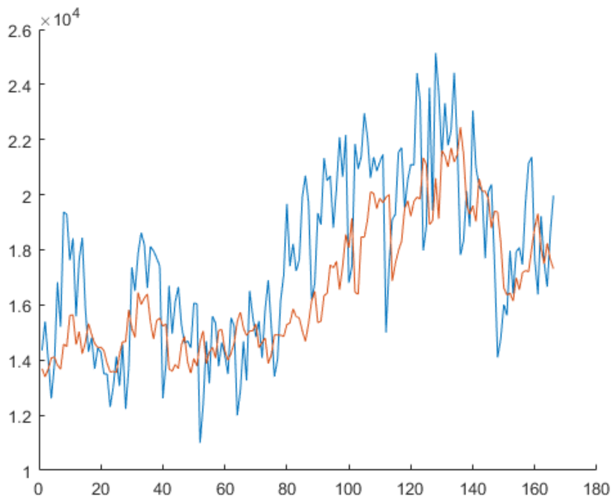
Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

ARMAX(p, 2) model

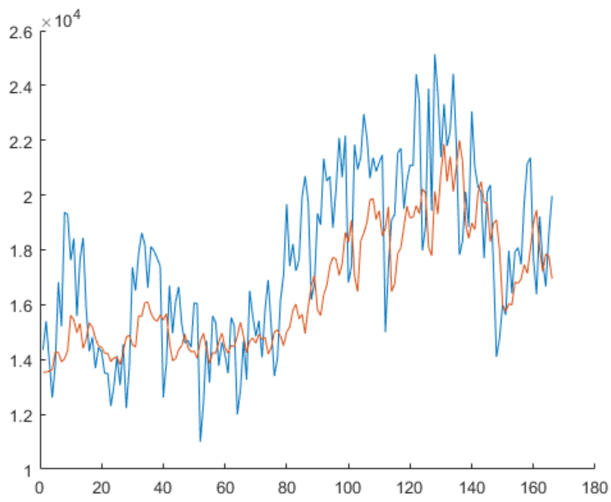
$$ARMA(p,q): y_t = \mu + \rho_1 y_{t-1} + \cdots + \rho_p y_{t-p} + u_t + \varphi_1 u_{t-1} + \ldots + \varphi_q u_{t-q} + \beta * Approval$$

ARMAX models are only comparable to AR models, despite being more complex.

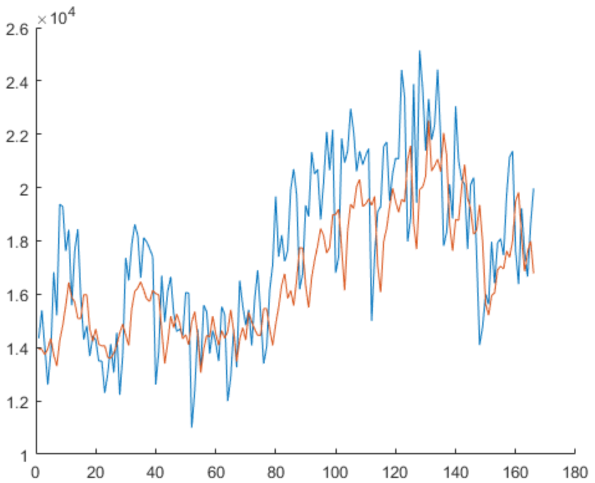
	ARMAX(1,2)	ARMAX(2,2)	ARMAX(3,2)	RW
MSFE	6.0797e+06	5.9442e+06	5.6699e+06	2.5099e+07



ARMAX (1,2)



ARMAX (2,2)



ARMAX (3,2)

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Result summary

The following table summarizes the forecast accuracy of all the models I used.

	S1	S2	S3	S4	RW
MSFE	7.0694e+06	5.7068e+06	6.8486e+06	6.3107e+06	2.5099e+07
	S1’	S2’	S3’	S4’	
MSFE	6.4737e+06	4.9057e+06	6.3762e+06	6.0116e+06	

	HW without seasonality parameter adjustment	HW with seasonality parameter adjustment
MSFE	1.8370e+06	1.8370e+06

	IAR(1,1)	IAR(1,2)	IAR(1,3)
MSFE	5.6598e+06	5.2361e+06	5.4737e+06
	IMA(1,1)	IMA(1,2)	IMA(1,3)
MSFE	4.7950e+06	4.6633e+06	4.8344e+06
	ARMA(1,2)	ARMA(2,2)	ARMA(3, 2)
MSFE	6.1079e+06	6.1489e+06	5.6699e+06
	ARAR(1,2)	ARAR(2,2)	ARRA(3,2)
MSFE	6.0064e+06	5.2742e+06	5.5629e+06
	ARMAX(1,2)	ARMAX(2,2)	ARMAX(3,2)
MSFE	6.0797e+06	5.9442e+06	5.6699e+06

IMA(1, 2) model seems to have the highest forecast accuracy of all the models, at MSFE of 4.6633e+06.
I will be using IMA(1, 2) model for density forecast of the September total housing loan in Australia.

Density forecast

IMA model has the following specification

$$IMA(1,1): \ \Delta y_t = u_t + \varphi_1 u_{t-1}$$

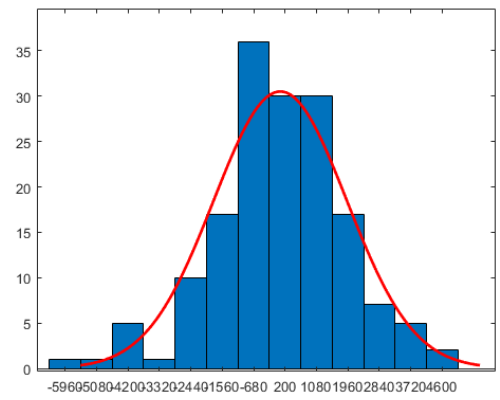
Since the IMA model calculates changes in the original series, I use the following equations to calculate point forecast of the original series.

$$\widehat{y_{t+1}} = y_t + \widehat{\Delta y}_{t+1}; where\ \widehat{\Delta y}_{t+1} = \widehat{u_{t+1}} + \varphi_1 \widehat{u_t} = \varphi_1 \widehat{u_t}$$
$$\widehat{y_{t+2}} = \widehat{y_{t+1}} + \widehat{\Delta y}_{t+2}; where\ \widehat{\Delta y}_{t+2} = \widehat{u_{t+2}} + \varphi_1 \widehat{u_{t+1}} = 0$$
$$\therefore \widehat{y_{t+2}} = \widehat{y_{t+1}} = y_t + \varphi_1 \widehat{u_t}$$

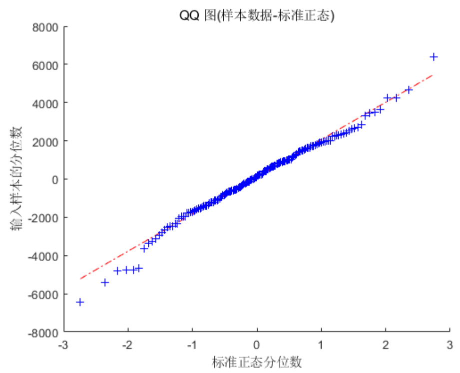
The point forecast calculated by MATLAB is 1.8628e+04

To make density forecast, I first estimate the distribution of $\Delta y_{t+1} - \widehat{\Delta y}_{t+1}$. The following is the histogram plot of the distribution of $\Delta y_{t+1} - \widehat{\Delta y}_{t+1}$ with fitted normal distribution curve

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.



The diagram below shows the QQ plot of $\Delta y_{t+1} - \widehat{\Delta y}_{t+1}$. The QQ plot shows there are some outliers where $\Delta y_{t+1} - \widehat{\Delta y}_{t+1}$ is large in absolute value. However, judging from the fact that most of the points lie on the line representing normal distribution quantiles, I feel safe assuming $\Delta y_{t+1} - \widehat{\Delta y}_{t+1}$ follows a normal distribution.



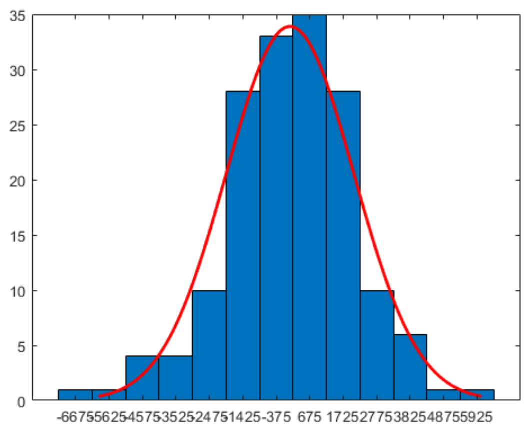
Using MATLAB function “fitdist”, I found the estimated parameters of normal distribution $\Delta y_{t+1} - \widehat{\Delta y}_{t+1}$ follows

$$\Delta y_{t+1} - \widehat{\Delta y}_{t+1} \sim N(64.8156, 1863.74^2), \textit{with } 95\% \textit{ CI of mean being } [-224.354, 353.985], \textit{and } 95\% \textit{ CI of std being } [1680.5, 2092.18]$$

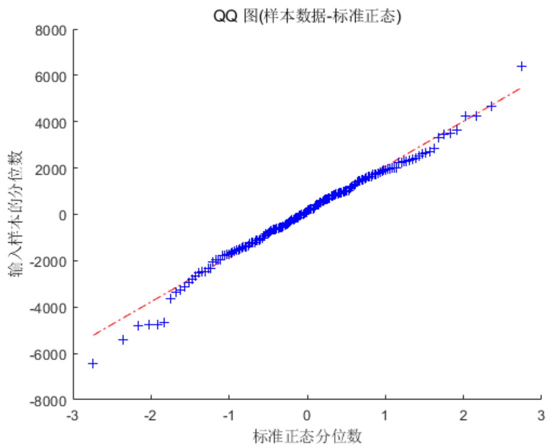
Or equivalently

$$\Delta y_{t+1} \sim N\left(64.8156 + \widehat{\Delta y_{t+1}}, 1863.74^2\right)$$

Secondly, I estimate the distribution of $\Delta y_{t+2} - \widehat{\Delta y}_{t+2}$. The following is the histogram plot and QQ plot of $\Delta y_{t+2} - \widehat{\Delta y}_{t+2}$, with fitted normal distribution curve overlying on the histogram.



Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.



Again, I feel confident that $\Delta y_{t+2} - \widehat{\Delta y}_{t+2}$ follows a normal distribution.

$$\Delta y_{t+2} - \widehat{\Delta y}_{t+2} \sim N(54.7949, 2001.85^2), with\ 95\%\ CI\ of\ mean\ being\ [-255.804, 365.393], and\ 95\%\ CI\ of\ std\ being\ [1805.03, 2247.22]$$

Or equivalently

$$\Delta y_{t+2} \sim N\left(54.7949 + \widehat{\Delta y_{t+2}}, 2001.85^2\right)$$

Lastly, I found $cov(\Delta y_{t+1}, \Delta y_{t+2})$ to be -1.0632e+06.

Therefore, using simple statistics, I conclude

$$\begin{aligned} y_{t+2} &= y_t + \Delta y_{t+1} \\ &\quad + \Delta y_{t+2} \sim N\left(y_t + \widehat{\Delta y_{t+1}} + \widehat{\Delta y_{t+2}} + 64.8156 + 54.7949, 2001.85^2 + 1863.74^2 - 2 * 1.0632\text{e} + 06\right) \\ &\quad = N(18748, 2314^2) \end{aligned}$$

95% CI for y_{t+2} is therefore [14212.56, 23283.44]. In other words, I am 95% confident that the new loan commitments for total housing in September 2020 will be between 14212.56 million and 23283.44 million.

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Appendix (Code used)

Random walk

```
load project_data.csv
y = project_data(:, 1);
T = length(y); T0 = 50; h = 2;
ytph = y(T0+h:end);
syhat = zeros(T-T0-h+1, 1);
for t = T0: T-h
    yt = y(1:t);
    sigma2 = var(yt);
    yhat = yt(end) + sqrt(h * sigma2) * randn;
    syhat(t-T0+1) = yhat;
end
MSFE_RW = mean((ytph - syhat).^2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Month and quarter as dummy variables

```
load project_data.csv
y = project_data(:, 1); M = project_data(:, 2); D = project_data(:, 3);
T = length(y); h = 2; T0 = 50;
```

S1

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

M12 = (M==12);

for t = T0: T - h
    Xt = [ones(t, 1), (1: t)', M12(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [1, t+h, M12(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE1 = mean((ytph - syhat).^2);
```

S2

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

M1 = (M==1); M2 = (M==2);M3 = (M==3);M4 = (M==4);M5 = (M==5);M6 = (M==6);M7 = (M==7);M8 = (M==8);M9 = (M==9);M10 =
(M==10);M11 = (M==11);M12 = (M==12);

for t = T0: T - h
    Xt = [(1: t)', M1(1:t), M2(1:t),  M3(1:t), M4(1:t), M5(1:t), M6(1:t), M7(1:t), M8(1:t), M9(1:t), M10(1:t),
M11(1:t), M12(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [t+h, M1(t+h), M2(t+h), M3(t+h), M4(t+h), M5(t+h), M6(t+h), M7(t+h), M8(t+h), M9(t+h), M10(t+h),
M11(t+h), M12(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE2 = mean((ytph - syhat).^2);
```

S3

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

D12 = (D==4);

for t = T0: T - h
    Xt = [ones(t, 1), (1: t)', D12(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [1, t+h, D12(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE3 = mean((ytph - syhat).^2);
```

S4

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

D1 = (D==1); D2 = (D==2);D3 = (D==3); D4 = (D==4);

for t = T0: T - h
    Xt = [(1: t)', D1(1:t), D2(1:t), D3(1:t), D4(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [t+h, D1(t+h), D2(t+h), D3(t+h), D4(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE4 = mean((ytph - syhat).^2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Month and quarter as dummy variables, with exogenous building approval variable

```
load project_data.csv; load project_data_2.csv;
y = project_data(:, 1); x = project_data_2(:, 1);
M = project_data(:, 2); D = project_data(:, 3);
T = length(y); h = 2; T0 = 50;
```

S1'

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

M12 = (M==12);

for t = T0: T - h
    Xt = [ones(t, 1), (1: t)', M12(1:t), x(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [1, t+h, M12(t+h), x(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE1 = mean((ytph - syhat).^2);
```

S2'

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

M1 = (M==1); M2 = (M==2);M3 = (M==3);M4 = (M==4);M5 = (M==5);M6 = (M==6);M7 = (M==7);M8 = (M==8);M9 = (M==9);M10 =
(M==10);M11 = (M==11);M12 = (M==12);

for t = T0: T - h
    Xt = [(1: t)', M1(1:t), M2(1:t),  M3(1:t), M4(1:t), M5(1:t), M6(1:t), M7(1:t), M8(1:t), M9(1:t), M10(1:t),
M11(1:t), M12(1:t), x(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [t+h, M1(t+h), M2(t+h), M3(t+h), M4(t+h), M5(t+h), M6(t+h), M7(t+h), M8(t+h), M9(t+h), M10(t+h),
M11(t+h), M12(t+h), x(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE2 = mean((ytph - syhat).^2);
```

S3'

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

D12 = (D==4);

for t = T0: T - h
    Xt = [ones(t, 1), (1: t)', D12(1:t), x(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [1, t+h, D12(t+h), x(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE3 = mean((ytph - syhat).^2);
```

S4’

```
ytph = y(T0+h: end);

syhat = zeros(T - T0 - h + 1, 1);

D1 = (D==1); D2 = (D==2);D3 = (D==3); D4 = (D==4);

for t = T0: T - h
    Xt = [(1: t)', D1(1:t), D2(1:t), D3(1:t), D4(1:t), x(1:t)];
    yt = y(1:t);
    betahat = (Xt'* Xt) \ Xt' * yt;
    yhat = [t+h, D1(t+h), D2(t+h), D3(t+h), D4(t+h), x(t+h)] * betahat;
    syhat(t-T0+1) = yhat;
end
MSFE4 = mean((ytph - syhat).^2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

Holt winters smoothing

```
opt_params = fminsearch(@HW_MSFE, [0.5, 0.5, 0.5, 0.1]);

load project_data.csv
alpha = opt_params(1); beta = opt_params(2);
gamma = opt_params(3); a = 0;
y = project_data(:, 1); M = project_data(:, 2); D = project_data(:, 2);
T = length(y); T0 = 50; h = 2; s = 12;
syhat = zeros(T-h-T0+1, 1);
ytph = y(T0+h:end);
St = zeros(T-h, 1);
Lt = mean(y(1:s)); bt = 0; St(1:s) = 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;
    if M(t) == 12
        St(t) = gamma * (y(t) - newLt) + (1-gamma) * St(t-s) + a;
    else
        St(t) = gamma * (y(t) - newLt) + (1-gamma) * St(t-s);
    end
    yhat = newLt + h * newbt + St(t+h-s);
    Lt = newLt; bt = newbt;
    if t >= T0
        syhat(t-T0+1, :) = yhat;
    end
end
MSFE = mean((ytph - syhat).^2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
opt_params = fminsearch(@HW_MSFE, [0.5, 0.5, 0.5, 0.1]);

load project_data.csv
alpha = opt_params(1); beta = opt_params(2);
gamma = opt_params(3); a = opt_params(4);
y = project_data(:, 1); M = project_data(:, 2); D = project_data(:, 2);
T = length(y); T0 = 50; h = 2; s = 12;
syhat = zeros(T-h-T0+1, 1);
ytph = y(T0+h:end);
St = zeros(T-h, 1);
Lt = mean(y(1:s)); bt = 0; St(1:s) = 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;
    if M(t) == 12
        St(t) = gamma * (y(t) - newLt) + (1-gamma) * St(t-s) + a;
    else
        St(t) = gamma * (y(t) - newLt) + (1-gamma) * St(t-s);
    end
    yhat = newLt + h * newbt + St(t+h-s);
    Lt = newLt; bt = newbt;
    if t >= T0
        syhat(t-T0+1, :) = yhat;
    end
end
MSFE = mean((ytph - syhat).^2);
```

Function for finding optimal HW coefficients

```
function MSFE = HW_MSFE(params)
alpha = params(1); beta = params(2); gamma = params(3); a = params(4);
load project_data.csv
y = project_data(:, 1); M = project_data(:, 2); D = project_data(:, 2);
T = length(y); T0 = 50; h = 2; s = 12;
syhat = zeros(T-h-T0+1, 1);
ytph = y(T0+h:end);
St = zeros(T-h, 1);
Lt = mean(y(1:s)); bt = 0; St(1:s) = 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;
    if M(t) == 12
        St(t) = gamma * (y(t) - newLt) + (1-gamma) * St(t-s) + a;
    else
        St(t) = gamma * (y(t) - newLt) + (1-gamma) * St(t-s);
    end
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
    yhat = newLt + h * newbt + St(t+h-s);
    Lt = newLt; bt = newbt;
    if t >= T0
        syhat(t-T0+1, :) = yhat;
    end
end
MSFE = mean((ytph - syhat).^2);
end
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

IAR(1,1)

```
load project_data.csv
y = project_data(:, 1);
m = 4;
y0 = y(1:m); y = y(m+1:end);
T = length(y);
dely0 = y(1) - y0(end);
dely = y(2:end) - y(1:end-1);
T0 = 50; h = 2;
syhat = zeros(T - T0 - h + 1, 1);
ytph = y(T0+h: end);
for t = T0: T-h
    delyt = dely(1: t-1);
    X = [dely0;delyt(1:end-1)];
    betahat = (X' * X) \ (X' * delyt);
    delytp1hat = [delyt(end)] * betahat; ytp1hat = y(t) + delytp1hat;
    delytp2hat = [delytp1hat] * betahat; ytp2hat = ytp1hat + delytp2hat;
    syhat(t-T0+1) = ytp2hat;
end
MSFE_IAR11 = mean((ytph - syhat).^2);
```

IAR(1,2)

```
load project_data.csv
y = project_data(:, 1);
m = 4;
y0 = y(1:m); y = y(m+1:end);
T = length(y);
dely0 = y(1) - y0(end);
dely = y(2:end) - y(1:end-1);
T0 = 50; h = 2;
syhat = zeros(T - T0 - h + 1, 1);
ytph = y(T0+h: end);
for t = T0: T-h
    delyt = dely(1: t-1);
    X = [[dely0;delyt(1:end-1)], [y0(end)-y0(end-1); dely0; delyt(1:end-2)]];
    betahat = (X' * X) \ (X' * delyt);
    delytp1hat = [delyt(end), delyt(end-1)] * betahat; ytp1hat = y(t) + delytp1hat;
    delytp2hat = [delytp1hat, delyt(end)] * betahat; ytp2hat = ytp1hat + delytp2hat;
    syhat(t-T0+1) = ytp2hat;
end
MSFE_IAR12 = mean((ytph - syhat).^2);
```

IAR(1, 3)

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
load project_data.csv
y = project_data(:, 1);
m = 4;
y0 = y(1:m); y = y(m+1:end);
T = length(y);
dely0 = y(1) - y0(end);
dely = y(2:end) - y(1:end-1);
T0 = 50; h = 2;
syhat = zeros(T - T0 - h + 1, 1);
ytph = y(T0+h: end);
for t = T0: T-h
    delyt = dely(1: t-1);
    X = [[dely0;delyt(1:end-1)], [y0(end)-y0(end-1); dely0; delyt(1:end-2)], [y0(end-1) - y0(end-2); y0(end) -
y0(end-1); dely0; delyt(1:end-3)]];
    betahat = (X' * X) \ (X' * delyt);
    delytp1hat = [delyt(end), delyt(end-1), delyt(end-2)] * betahat; ytp1hat = y(t) + delytp1hat;
    delytp2hat = [delytp1hat, delyt(end), delyt(end-1)] * betahat; ytp2hat = ytp1hat + delytp2hat;
    syhat(t-T0+1) = ytp2hat;
end
MSFE_IAR13 = mean((ytph - syhat).^2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

IMA(1, 1)

```
load project_data.csv; y = project_data(:, 1);
m=4; h=2; y0 = y(1: m); y = y(m+1: end); T = length(y);
dely = y(2:end) - y(1:end-1);
T0 = 50;
syhat = zeros(T-T0-h+1, 1);
ytph = y(T0+h:end);
f = @(psi)loglike_IMA11(psi, dely(1:T0-1));
params_hat = fminsearch(f, [0.5]);
for t = T0: T-h
    delyt = dely(1: t-1);
    f = @(psi) loglike_IMA11(psi, delyt(1:T0-1));
    params_hat = fminsearch(f, params_hat);
    H = speye(t-1) + params_hat(1) * spdiags(ones(t-1-1, 1), -1, t-1, t-1);
    uhat = H \ delyt;
    delytp1hat = params_hat(1) * uhat(end); ytp1hat = y(t) + delytp1hat;
    delytp2hat = 0; ytp2hat = ytp1hat + delytp2hat;
    syhat(t-T0+1, :) = ytp2hat;
end
MSFE_IMA11 = mean((ytph - syhat).^2);
```

IMA(1, 2)

```
load project_data.csv; y = project_data(:, 1);
m=4; h=2; y0 = y(1: m); y = y(m+1: end); T = length(y);
dely = y(2:end) - y(1:end-1);
T0 = 50;
syhat = zeros(T-T0-h+1, 1); sdelytp2hat = zeros(T-T0-h, 1); sdelytp1hat = zeros(T-T0-h, 1);
ytph = y(T0+h:end); delytp2 = dely(T0+1:end); delytp1 = dely(T0: end-1);
f = @(psi)loglike_IMA12(psi, dely(1:T0-1));
params_hat = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    delyt = dely(1: t-1);
    f = @(psi) loglike_IMA12(psi, delyt(1:T0-1));
    params_hat = fminsearch(f, params_hat);
    H = speye(t-1) + params_hat(1) * spdiags(ones(t-1-1, 1), -1, t-1, t-1) + params_hat(2) * spdiags(ones(t-1-2, 1),
-2, t-1, t-1);
    uhat = H \ delyt;
    delytp1hat = params_hat(1) * uhat(end) + params_hat(2) * uhat(end-1); ytp1hat = y(t) + delytp1hat;
    delytp2hat = params_hat(2) * uhat(end); ytp2hat = ytp1hat + delytp2hat;
    syhat(t-T0+1, :) = ytp2hat;
    sdelytp2hat(t-T0+1, :) = delytp2hat;
    sdelytp1hat(t-T0+1, :) = delytp1hat;
end
histfit(delytp1 - sdelytp1hat);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
fitdist(delytp1 - sdelytp1hat, 'Normal')
histfit(delytp2 - sdelytp2hat);
fitdist(delytp2 - sdelytp2hat, 'Normal')
MSFE_IMA12 = mean((ytph - syhat).^2);

f = @(psi) loglike_IMA12(psi, dely);
params_hat = fminsearch(f, params_hat);
H = speye(T-1) + params_hat(1) * spdiags(ones(T-1-1, 1), -1, T-1, T-1) + params_hat(2) * spdiags(ones(T-1-2, 1), -2, T-1, T-1);
uhat = H \ dely;
delytp1hat = params_hat(1) * uhat(end) + params_hat(2) * uhat(end-1); ytp1hat = y(end) + delytp1hat;
delytp2hat = params_hat(2) * uhat(end); ytp2hat = ytp1hat + delytp2hat;
```

IMA(1, 3)

```
load project_data.csv; y = project_data(:, 1);
m=4; h=2; y0 = y(1: m); y = y(m+1: end); T = length(y);
dely = y(2:end) - y(1:end-1);
T0 = 50;
syhat = zeros(T-T0-h+1, 1);
ytph = y(T0+h:end);
f = @(psi)loglike_IMA13(psi, dely(1:T0-1));
params_hat = fminsearch(f, [0.5, 0.5, 0.5]);
for t = T0: T-h
    delyt = dely(1: t-1);
    f = @(psi) loglike_IMA13(psi, delyt(1:T0-1));
    params_hat = fminsearch(f, params_hat);
    H = speye(t-1) + params_hat(1) * spdiags(ones(t-1-1, 1), -1, t-1, t-1) + params_hat(2) * spdiags(ones(t-1-2, 1), -2, t-1, t-1) + params_hat(3) * spdiags(ones(t-1-3, 1), -3, t-1, t-1);
    uhat = H \ delyt;
    delytp1hat = params_hat(1) * uhat(end) + params_hat(2) * uhat(end-1) + params_hat(3) * uhat(end-2); ytp1hat = y(t) + delytp1hat;
    delytp2hat = params_hat(2) * uhat(end) + params_hat(3) * uhat(end-1); ytp2hat = ytp1hat + delytp2hat;
    syhat(t-T0+1, :) = ytp2hat;
end
MSFE_IMA13 = mean((ytph - syhat).^2);
```

Functions for numerically finding IMA MLEs.

```
function ell = loglike_IMA11(params, y)
psi1 = params(1);
T = length(y);
I = speye(T);
A = spdiags(ones(T-1, 1), [-1], T, T);
gam = I + A * psi1;
gam2 = gam * gam';
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
sigma2 = y' * (gam2\y)/T;
ell = -T/2 * log(2 * pi * sigma2) - 1/(2 * sigma2) * y' * (gam2 \ y);
ell = -ell;
end
```

```
function ell = loglike_IMA12(params, y)
psi1 = params(1);  psi2 = params(2);
T = length(y);
I = speye(T);
A = spdiags(ones(T-1, 1), [-1], T, T); B = spdiags(ones(T-2, 1), -2, T, T);
gam = I + A * psi1 + B * psi2;
gam2 = gam * gam';
sigma2 = y' * (gam2\y)/T;
ell = -T/2 * log(2 * pi * sigma2) - 1/(2 * sigma2) * y' * (gam2 \ y);
ell = -ell;
end
```

```
function ell = loglike_IMA13(params, y)
psi1 = params(1);  psi2 = params(2); psi3 = params(3);
T = length(y);
I = speye(T);
A = spdiags(ones(T-1, 1), [-1], T, T); B = spdiags(ones(T-2, 1), -2, T, T);
C = spdiags(ones(T-3, 1), -3, T, T);
gam = I + A * psi1 + B * psi2 + C * psi3;
gam2 = gam * gam';
sigma2 = y' * (gam2\y)/T;
ell = -T/2 * log(2 * pi * sigma2) - 1/(2 * sigma2) * y' * (gam2 \ y);
ell = -ell;
end
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

ARMA(1, 2)

```
load project_data.csv
y = project_data(:,1);
T = length(y);
T0 = 50; h = 2;
% options = optimset('MaxFunEvals', 2000, 'MaxIter', 2000, 'Display', 'Iter');
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARMA12(params, y(1:T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    yt = y(1:t);
    f = @(params) loglike_ARMA12(params, yt);
    phi_MLE = fminsearch(f, phi_MLE);
    X = [ones(t, 1), [0; y(1: t-1)]];
    % H
    L1 = spdiags(ones(t-1, 1), -1, t, t);
    L2 = L1 ^ 2;
    H = speye(t) + L1 * phi_MLE(1) + L2 * phi_MLE(2);
    betahat = (X' / (H * H') * X) \ (X' / (H * H') * yt);
    uhat = H \ (yt - X * betahat);
    % 1 step ahead forecast
    % yt+1 hat
    ytp1hat = [1, yt(end)] * betahat + phi_MLE(1) * uhat(end) + phi_MLE(2) * uhat(end-1);
    % yt+2 hat
    ytp2hat = [1, ytp1hat] * betahat + phi_MLE(2) * uhat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARMA12 = mean((ytph - syhat).^2);
```

ARMA(2, 2)

```
load project_data.csv
y = project_data(:, 1);
T = length(y);
T0 = 50; h = 2;
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARMA22(params, y(1: T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    yt = y(1:t);
    f = @(params) loglike_ARMA22(params, yt);
    phi_MLE = fminsearch(f, phi_MLE);
    X = [ones(t, 1), [0;y(1:t-1)], [0;0;y(1:t-2)]];
    L1 = spdiags(ones(t-1, 1), -1, t, t);
    L2 = L1 ^2;
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
    H = speye(t) + L1 * phi_MLE(1) + L2 * phi_MLE(2);
    H2 = H * H';
    betahat = (X' / H2 * X) \ (X' / H2 * yt);
    uhat = H \ (yt - X * betahat);
    ytp1hat = [1, y(t), y(t-1)] * betahat + phi_MLE(1) * uhat(end) + phi_MLE(2) * uhat(end-1);
    ytp2hat = [1, ytp1hat, y(t)] * betahat + phi_MLE(2) * uhat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARMA22 = mean((ytph - syhat).^2);
```

ARMA(3, 2)

```
load project_data.csv
y = project_data(:, 1);
T = length(y);
T0 = 50; h = 2;
ytph = y(T0 + h: T);
syhat = zeros(T - T0 - h + 1, 1);
f = @(params) loglike_ARMA32(params, y(1:T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T - h
    yt = y(1:t);
    f = @(params) loglike_ARMA32(params, yt);
    phi_MLE = fminsearch(f, phi_MLE);
    L1 = spdiags(ones(t-1, 1), -1, t, t); L2 = L1 ^ 2;
    H = speye(t) + phi_MLE(1) * L1 + phi_MLE(2) * L2; H2 = H * H';
    X = [ones(t, 1), [0; y(1: t-1)], [0;0;y(1:t-2)], [0;0;0;y(1:t-3)]];
    betahat = (X' / H2 * X) \ (X' / H2 * yt);
    uhat = H \ (yt - X * betahat);
    ytp1hat = [1, y(t), y(t-1), y(t-2)] * betahat + phi_MLE(1) * uhat(end) + phi_MLE(2) * uhat(end-1);
    ytp2hat = [1, ytp1hat, y(t), y(t-1)] * betahat + phi_MLE(2) * uhat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARMA32 = mean((ytph - syhat).^2);
```

Functions for numerically finding ARMA MLEs.

```
function ell = loglike_ARMA12(params, y)
% overhead
T = length(y);
phi1 = params(1); phi2 = params(2);
% X
X = [ones(T, 1), [0; y(1:end-1)]];
% H
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1 ^ 2;
H = speye(T) + L1 * phi1 + L2 * phi2;
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
H2 = H * H';
betahat = (X' / H2 * X) \ (X' / H2 * y);
sigma2 = 1/T * (y-X * betahat)' / H2 * (y- X*betahat);
ell = -T/2 * log(2 * pi * sigma2)-1/(2*sigma2) * (y-X * betahat)' / H2 * (y - X*betahat);
ell = -ell;
end
```

```
function ell = loglike_ARMA22(params, y)
phi1 = params(1); phi2 = params(2);
T = length(y);
X = [ones(T, 1), [0; y(1:T-1)], [0; 0; y(1:T-2)]];
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1 ^ 2;
H = speye(T) + L1 * phi1 + L2 * phi2;
H2 = H * H';
betahat = (X' / H2 * X) \ (X' / H2 * y);
sigma2 = 1/T * (y-X*betahat)' / H2 * (y-X*betahat);
ell = -T/2 * log(2*pi*sigma2) - 1/(2*sigma2) * (y-X*betahat)' * H' * H * (y-X*betahat);
ell = -ell;
end
```

```
function ell = loglike_ARMA32(params, y)
phi1 = params(1); phi2 = params(2);
T = length(y);
X = [ones(T, 1), [0;y(1:T-1)], [0;0;y(1:T-2)], [0;0;0;y(1:T-3)]];
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1^2;
H = speye(T) + L1 * phi1 + L2 * phi2;
H2 = H * H';
betahat = (X' / H2 * X) \ (X' / H2 * y);
sigma2 = 1/T*(y-X*betahat)' / H2 * (y-X*betahat);
ell  = -T/2 * log(2 * pi * sigma2) - 1 / (2 * sigma2) * (y - X * betahat)' / H2 * (y-X*betahat);
ell = -ell;
end
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

ARAR(1, 2)

```
load project_data.csv
y = project_data(:, 1);
T = length(y);
T0 = 50; h = 2;
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARAR12(params, y(1: T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    yt = y(1:t);
    f = @(params) loglike_ARAR12(params, yt);
    phi_MLE = fminsearch(f, phi_MLE);
    X = [ones(t, 1), [0; y(1: t-1)]];
    L1 = spdiags(ones(t-1, 1), -1, t, t);
    L2 = L1 ^ 2;
    H = speye(t) - L1 * phi_MLE(1) - L2 * phi_MLE(2);
    betahat = (X' * (H' * H) * X) \ X' * (H' * H) * yt;
    ehat = yt - X * betahat;
    ytp1hat = [1, yt(end)] * betahat + phi_MLE(1) * ehat(end) + phi_MLE(2) * ehat(end-1);
    ytp2hat = [1, ytp1hat] * betahat + phi_MLE(1).^2 * ehat(end) + phi_MLE(1) * phi_MLE(2) * ehat(end-1) + 
phi_MLE(2) * ehat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARAR12 = mean((ytph - syhat).^2);
```

ARAR(2, 2);

```
load project_data.csv
y = project_data(:, 1); T = length(y); T0 = 50; h = 2;
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARAR22(params, y(1: T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    yt = y(1:t);
    f = @(params) loglike_ARAR22(params, yt);
    phi_MLE = fminsearch(f, phi_MLE);
    L1 = spdiags(ones(t-1, 1), -1, t, t); L2 = L1^2;
    H = speye(t) - L1 * phi_MLE(1) - L2 * phi_MLE(2);
    X = [ones(t, 1), [0;y(1:t-1)], [0;0;y(1:t-2)]];
    betahat = (X' * (H' * H) * X) \ X' * (H' * H) * yt;
    ehat = yt - X * betahat;
    ytp1hat = [1, y(t), y(t-1)] * betahat + phi_MLE(1) * ehat(end) + phi_MLE(2) * ehat(end-1);
    ytp2hat = [1, ytp1hat, y(t)] * betahat + phi_MLE(1).^2 * ehat(end) + phi_MLE(1) * phi_MLE(2) * ehat(end-1) + 
phi_MLE(2) * ehat(end);
    syhat(t-T0+1) = ytp2hat;
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
end
MSFE_ARAR22 = mean((ytph - syhat).^2);
```

ARAR(3, 2);

```
load project_data.csv
y = project_data(:, 1); T = length(y); T0 = 50; h = 2;
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARAR32(params, y(1: T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    yt = y(1:t);
    f = @(params) loglike_ARAR32(params, yt);
    phi_MLE = fminsearch(f, phi_MLE);
    L1 = spdiags(ones(t-1, 1), -1, t, t); L2 = L1^2;
    H = speye(t) - L1 * phi_MLE(1) - L2 * phi_MLE(2);
    X = [ones(t, 1), [0;y(1:t-1)], [0;0;y(1:t-2)], [0;0;0;y(1:t-3)]];
    betahat = (X' * (H' * H) * X) \ X' * (H' * H) * yt;
    ehat = yt - X * betahat;
    ytp1hat = [1, y(t), y(t-1), y(t-2)] * betahat + phi_MLE(1) * ehat(end) + phi_MLE(2) * ehat(end-1);
    ytp2hat = [1, ytp1hat, y(t), y(t-1)] * betahat + phi_MLE(1).^2 * ehat(end) + phi_MLE(1) * phi_MLE(2) * ehat(end-
1) + phi_MLE(2) * ehat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARAR32 = mean((ytph - syhat).^2);
```

Functions for finding ARAR MLEs.

```
function ell = loglike_ARAR12(params, y)
T = length(y);
phi1 = params(1); phi2 = params(2);
X = [ones(T, 1), [0; y(1:end-1)]];
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1 ^ 2;
H = speye(T) - L1 * phi1 - L2 * phi2;
betahat = (X' * (H' * H) * X) \ X' * (H' * H) * y;
sigma2 = 1/T * (y-X*betahat)' * (H' * H) * (y-X*betahat);
ell = -T/2 * log(2 * pi * sigma2) - 1/(2*sigma2) * (y-X*betahat)' * (H' * H) * (y-X*betahat);
ell = -ell;
end
```

```
function ell = loglike_ARAR22(params, y)
T = length(y); phi1 = params(1); phi2 = params(2);
X = [ones(T, 1), [0; y(1:T-1)], [0;0;y(1:T-2)]];
L1 = spdiags(ones(T-1, 1), -1, T, T); L2 = L1 ^ 2;
H = speye(T) - phi1 * L1 - phi2 * L2;
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
betahat = (X' * (H' * H) * X) \ X' * (H' * H )* y;
sigma2 = 1/T * (y - X * betahat)' * (H' * H) * (y - X * betahat);
ell = -T/2 * log(2 * pi * sigma2) - 1/(2*sigma2) * (y-X*betahat)' *(H' * H)*(y-X*betahat);
ell = -ell;
end
```

```
function ell = loglike_ARAR32(params, y)
T = length(y); phi1 = params(1); phi2 = params(2);
X = [ones(T, 1), [0; y(1:T-1)], [0;0;y(1:T-2)], [0; 0; 0; y(1:T-3)]];
L1 = spdiags(ones(T-1, 1), -1, T, T); L2 = L1 ^ 2;
H = speye(T) - phi1 * L1 - phi2 * L2;
betahat = (X' * (H' * H) * X) \ X' * (H' * H )* y;
sigma2 = 1/T * (y - X * betahat)' * (H' * H) * (y - X * betahat);
ell = -T/2 * log(2 * pi * sigma2) - 1/(2*sigma2) * (y-X*betahat)' *(H' * H)*(y-X*betahat);
ell = -ell;
end
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

ARMAX (1, 2)

```
warning('off');
load project_data.csv
load project_data_2.csv
y = project_data(:, 1); x = project_data_2(:, 1);
T = length(y); T0 = 50; h = 2;
ytph = y(T0 + h: T); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARMAX12(params, y(1:T0), x(1:T0));
phi_MLE = fminsearch(f, [0.5, 0.5]);
for t = T0: T-h
    yt = y(1:t); xt = x(1:t);
    f = @(params) loglike_ARMAX12(params, yt, xt);
    phi_MLE = fminsearch(f, [0.5, 0.5]);
    L1 = spdiags(ones(t-1, 1), -1, t, t); L2 = L1 ^ 2;
    H = speye(t) + L1 * phi_MLE(1) + L2 * phi_MLE(2);
    X = [ones(t, 1), [0; y(1:t-1)], x(1:t)];
    betahat = (X' / (H * H') * X) \ (X' / (H * H') * yt);
    uhat = H \ (yt - X * betahat);
    ytp1hat = [1, y(t), x(t+1)] * betahat + phi_MLE(1) * uhat(end) + phi_MLE(2) * uhat(end-1);
    ytp2hat = [1, ytp1hat, x(t+2)] * betahat + phi_MLE(2) * uhat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARMAX12 = mean((ytph - syhat).^2);
```

ARMAX(2, 2)

```
load project_data.csv
load project_data_2.csv
y = project_data(:, 1); x = project_data_2(:, 1);
T = length(y);
T0 = 50; h = 2;
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARMAX22(params, y(1: T0), x(1:T0));
phi_MLE = fminsearch(f, [0.1, 0.1]);
for t = T0: T-h
    yt = y(1:t); xt = x(1:t);
    f = @(params) loglike_ARMAX22(params, yt, xt);
    phi_MLE = fminsearch(f, [0.1, 0.1]);
    X = [ones(t, 1), [0;y(1:t-1)], [0;0;y(1:t-2)], x(1:t)];
    L1 = spdiags(ones(t-1, 1), -1, t, t);
    L2 = L1 ^2;
    H = speye(t) + L1 * phi_MLE(1) + L2 * phi_MLE(2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
    H2 = H * H';
    betahat = (X' / H2 * X) \ (X' / H2 * yt);
    uhat = H \ (yt - X * betahat);
    ytp1hat = [1, y(t), y(t-1), x(t+1)] * betahat + phi_MLE(1) * uhat(end) + phi_MLE(2) * uhat(end-1);
    ytp2hat = [1, ytp1hat, y(t), x(t+2)] * betahat + phi_MLE(2) * uhat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARMAX22 = mean((ytph - syhat).^2);
```

ARMAX(3, 2)

```
load project_data.csv
load project_data_2.csv
y = project_data(:, 1); x = project_data_2(:, 1);
T = length(y);
T0 = 50; h = 2;
ytph = y(T0+h:end); syhat = zeros(T-T0-h+1, 1);
f = @(params) loglike_ARMAX32(params, y(1: T0), x(1:T0));
phi_MLE = fminsearch(f, [0.0, 0.0]);
for t = T0: T-h
    yt = y(1:t); xt = x(1:t);
    f = @(params) loglike_ARMAX32(params, yt, xt);
    phi_MLE = fminsearch(f, phi_MLE);
    X = [ones(t, 1), [0;y(1:t-1)], [0;0;y(1:t-2)], [0;0;0;y(1:t-3)],x(1:t)];
    L1 = spdiags(ones(t-1, 1), -1, t, t);
    L2 = L1 ^2;
    H = speye(t) + L1 * phi_MLE(1) + L2 * phi_MLE(2);
    H2 = H * H';
    betahat = (X' / H2 * X) \ (X' / H2 * yt);
    uhat = H \ (yt - X * betahat);
    ytp1hat = [1, y(t), y(t-1), y(t-2), x(t+1)] * betahat + phi_MLE(1) * uhat(end) + phi_MLE(2) * uhat(end-1);
    ytp2hat = [1, ytp1hat, y(t), y(t-1), x(t+2)] * betahat + phi_MLE(2) * uhat(end);
    syhat(t-T0+1) = ytp2hat;
end
MSFE_ARMAX32 = mean((ytph - syhat).^2);
hold on
plot(ytph);
plot(syhat);
```

Functions used for finding ARMAX MLEs

```
function ell = loglike_ARMAX12(params, y, x)
% overhead
T = length(y);
phi1 = params(1); phi2 = params(2);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
% X
X = [ones(T, 1), [0; y(1:end-1)], x(1:end)];
% H
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1 ^ 2;
H = speye(T) + L1 * phi1 + L2 * phi2;
H2 = H * H';
betahat = (X' / H2 * X) \ (X' / H2 * y);
sigma2 = 1/T * (y-X * betahat)' / H2 * (y- X*betahat);
ell = -T/2 * log(2 * pi * sigma2)-1/(2*sigma2) * (y-X * betahat)' / H2 * (y - X*betahat);
ell = -ell;
end
```

```
function ell = loglike_ARMAX22(params, y, x)
% overhead
T = length(y);
phi1 = params(1); phi2 = params(2);
% X
X = [ones(T, 1), [0; y(1:end-1)], [0;0;y(1:end-2)], x(1:end)];
% H
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1 ^ 2;
H = speye(T) + L1 * phi1 + L2 * phi2;
H2 = H * H';
betahat = (X' / H2 * X) \ (X' / H2 * y);
sigma2 = 1/T * (y-X * betahat)' / H2 * (y-X*betahat);
ell = -T/2 * log(2 * pi * sigma2)-1/(2*sigma2) * (y-X * betahat)' / H2 * (y - X*betahat);
ell = -ell;
end
```

```
function ell = loglike_ARMAX32(params, y, x)
% overhead
T = length(y);
phi1 = params(1); phi2 = params(2);
% X
X = [ones(T, 1), [0; y(1:end-1)], [0;0;y(1:end-2)], [0;0;0;y(1:end-3)], x(1:end)];
% H
L1 = spdiags(ones(T-1, 1), -1, T, T);
L2 = L1 ^ 2;
H = speye(T) + L1 * phi1 + L2 * phi2;
H2 = H * H';
betahat = (X' / H2 * X) \ (X' / H2 * y);
sigma2 = 1/T * (y-X * betahat)' / H2 * (y-X*betahat);
ell = -T/2 * log(2 * pi * sigma2)-1/(2*sigma2) * (y-X * betahat)' / H2 * (y - X*betahat);
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.

```
ell = -ell;
end
```

Throughout the report, the blue line in the plot represents original data and the orange line represents forecasts.