We have shown how to forecast (predict) future values $Y_{t+\ell}$, but it is also important to assess the precision of our predictions.

We can do this by obtaining prediction limits (i.e., a prediction interval) for $Y_{t+\ell}$.

To obtain these intervals, we will have to make an assumption about the distribution of the stochastic component (white noise terms) in our model.

The formulas we will use will assume the white noise terms follow a normal distribution.

If this assumption does not hold for the original data, we can transform the data (possibly using evidence from a Box-Cox analysis).
With a deterministic trend model, $Y_t = \mu_t + X_t$, where $\mu_t$ is some deterministic trend and the stochastic component $X_t$ has mean zero, the forecast is

$$\hat{Y}_t(\ell) = \mu_{t+\ell}.$$ 

If $X_t$ is normally distributed, then the forecast error $e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = X_t$ is also normally distributed.

And $var[e_t(\ell)] = \gamma_0$, which is the noise variance.

This implies that

$${Y}_{t+\ell} - \hat{Y}_t(\ell) \over \sqrt{var[e_t(\ell)]}$$

follows a standard normal distribution.
So with probability $1 - \alpha$, the future observation ($\ell$ time units ahead), $Y_{t+\ell}$, falls within the interval

$$\hat{Y}_t(\ell) \pm z_{\alpha/2} \sqrt{\text{var}[e_t(\ell)]}$$

Note that this is technically a prediction interval rather than a confidence interval, since the quantity that we hope the interval contains is a random quantity.

Consider the Dubuque temperature data, for which we used a harmonic regression model for the trend.

The forecast of the June 1976 average temperature was 68.3, and the estimate of the noise standard deviation (see R code) was 3.7.

So a 95% prediction interval for the June 1976 average temperature is $68.3 \pm (1.96)(3.7)$ or $(61.05, 75.55)$. 
The Prediction Limits are only Approximate

- The above prediction interval method would be correct if the parameters of the trend model were known exactly.
- In practice, however, we *estimate* these parameters from our sample data.
- When our prediction is based on estimated parameters, the forecast error variance is not really $\gamma_0$, but rather $\gamma_0[1 + 1/n + c(n, \ell)]$, where $c(n, \ell)$ is some function of the sample size and the lead time.
- But for the trend models we typically consider (harmonic, linear, or quadratic trends), both $1/n$ and $c(n, \ell)$ are typically quite small when the sample size is large.
- For a harmonic model with period 12, $c(n, \ell) = 2/n$.
- And for a linear trend model, $c(n, \ell) \approx 3/n$ for moderate lead time $\ell$ and large $n$.
- Therefore, using $\gamma_0$ as the forecast error variance produces an approximately correct interval when $n$ is large.
Now consider models in the ARIMA class (including AR, MA, and ARMA models).

If the white noise terms are normally distributed, then the forecast error $e_t(\ell)$ is again normally distributed.

But for ARIMA models, the forecast error variance is a function of both the noise variance and the $\psi$-weights:

$$\text{var}[e_t(\ell)] = \sigma_e^2 \sum_{j=0}^{\ell-1} \psi_j^2.$$ 

In reality, the $\psi$-weights are functions of the $\phi$’s and $\theta$’s, which must be estimated, and the $\sigma_e^2$ must be estimated as well.

But plugging in these estimates has little effect on the validity of the prediction limits, for large sample sizes.
Prediction Intervals with an AR(1) Model

- With an AR(1) model, the forecast error variance formula is fairly simple:

\[
\text{var}[e_t(\ell)] = \sigma_e^2 \frac{1 - \phi^{2\ell}}{1 - \phi^2}
\]

- Consider the AR(1) model for the color property series. Using ML, we obtained the estimates \( \hat{\phi} = 0.5705 \), \( \hat{\mu} = 74.3293 \), and \( \hat{\sigma}_e^2 = 24.8 \).

- Our forecast one time unit ahead (\( \ell = 1 \)) was 70.14793.

- The 95\% prediction interval for this forecast is

\[
70.14793 \pm (1.96) \sqrt{(24.8) \frac{1 - 0.5705^2}{1 - 0.5705^2}} = 70.14793 \pm (1.96) \sqrt{24.8},
\]

or (60.39, 79.91).
More Prediction Intervals with an $AR(1)$ Model

- Our forecast two time units ahead ($\ell = 2$) was 71.94342.
- The 95% prediction interval for this forecast is
  
  $$71.94342 \pm (1.96)\sqrt{(24.8) \frac{1 - 0.5705^{2(2)}}{1 - 0.5705^2}},$$

  or (60.71, 83.18).

- Our forecast ten time units ahead ($\ell = 10$) was 74.30249 (very near $\hat{\mu}$, recall).
- The 95% prediction interval for this forecast is
  
  $$74.30249 \pm (1.96)\sqrt{(24.8) \frac{1 - 0.5705^{2(10)}}{1 - 0.5705^2}},$$

  or (62.41, 86.20).

- As $\ell$ gets larger, for this $AR(1)$ model, both the forecast and the prediction limits converge to some fixed long-lead values.
These formulas can be used to calculate the forecast and prediction limits for one forecast at a time, but often it is more useful to plot forecasts and prediction limits for several future values.

The \texttt{arima} function in R can generate an object from which we can plot the observed time series, plus the forecasts and 95\% prediction limits at any desired number of future time points.

See R example with the harmonic regression on the Dubuque temperature data.

In this example, we append 2 years of missing values to the \texttt{tempdub} data in order to forecast the temperature for two years into the future.
▶ See R example with the $AR(1)$ model on the color property data.
▶ Note that the forecasts and the 95% prediction limits converge toward their long-lead values, getting near them just a few time units into the future.
▶ The long-lead forecast for this model is simply the estimated process mean (see plot).
More Plots of Forecasts and Prediction Limits: $AR(p)$ Models

- See R example with the $AR(3)$ model on the (square-root-transformed) hare data.
- Note that the forecasts and the 95% prediction limits take longer to converge toward their long-lead values.
- The long-lead forecast plot for this $AR(3)$ model still shows the cyclical pattern even going 25 years into the future (see plot).
- What if we go even further into the future (say, 100 years)?
- See another R example with the `sarima.for` function in the `astsa` package, with the $AR(2)$ model on the recruitment data.
We have seen that for an $MA(1)$ model, the best forecast is 
\[ \hat{Y}_t(1) = \mu - \theta e_t \] for $\ell = 1$ and 
\[ \hat{Y}_t(\ell) = \mu \] for $\ell > 1$.

The forecast error variance $\text{var}[e_t(\ell)]$ for the $MA(1)$ model is 
\[ \sigma^2_e \] for $\ell = 1$ and 
\[ \sigma^2_e(1 + \theta^2) \] for $\ell > 1$.

By plugging the estimates into the formula

\[ \hat{Y}_t(\ell) \pm z_{\alpha/2} \sqrt{\text{var}[e_t(\ell)]} \]

we obtain a $(1 - \alpha)100\%$ prediction interval in the usual way.

In practice, we can easily obtain the forecasts and prediction limits for MA models (or any ARIMA models) using the sarima.for function in R.
Recall from our previous example with the *random walk with drift* model (an ARIMA(0, 1, 0) model), the presence or absence of a constant term $\theta_0$ in the model made a big difference in the forecasts.

In that example, we saw that, as a function of the lead time $\ell$, the forecasts increased (or decreased) linearly, with slope $\theta_0$ (the $\theta_0$ represented the “drift”).

In general, with ARIMA models that include differencing (having $d > 0$), the presence or absence of a constant term changes the forecasts substantially.
However, the \texttt{arima} function in the TSA package does not allow you to include a mean $\mu$ or constant term $\theta_0$ in the model unless $d = 0$.

With a nonstationary ARIMA model for differenced data, it is recommended instead to use the \texttt{sarima} function in R.

By default, \texttt{sarima} includes an intercept term, which we could estimate and check whether it was significantly different from zero.

If the intercept is not significantly different from 0, it is fine then to fit the model without it, but if the intercept is needed, we should use a model that includes it (see example with logged GNP data in R).
Updating ARIMA Forecasts

- Suppose we have yearly time series data, with the last observed year being 2018.
- We can use the data to forecast the values for 2019, 2020, 2021, etc.
- Once time passes and we actually observe the true value for 2019, we can use this additional information to update our previous forecasts for 2020, 2021, etc.
- We could simply redo the whole forecast from scratch, based on years . . . , 2017, 2018, 2019, but there is a shortcut way to update our previously obtained forecasts.
- There is a straightforward updating equation for ARIMA models in terms of the $\psi$-weights:

$$\hat{Y}_{t+1}(\ell) = \hat{Y}_t(\ell + 1) + \psi_\ell[Y_{t+1} - \hat{Y}_t(1)]$$

- The part in brackets, $Y_{t+1} - \hat{Y}_t(1)$, is the actual forecast error at time $t + 1$, which is known once $Y_{t+1}$ has been observed.
Recall the color property series in which we used the 35 observed values and an AR(1) model to forecast future values for times 36, 37, ... 

Note: For the AR(1) model, $\psi_1 = \phi_1$.

Our forecast 1 time unit into the future yielded $\hat{Y}_{35}(1) = 70.14793$, and our forecast 2 time units into the future was $\hat{Y}_{35}(2) = 71.94342$.

Suppose the actual value at time 36 becomes available, and it is 65.

Our updated forecast for the value at time 37 is then

\[
\hat{Y}_{36}(1) = \hat{Y}_{35}(2) + \psi_1[Y_{36} - \hat{Y}_{35}(1)]
\]

\[
= 71.94342 + 0.5705(65 - 70.14793) = 69.00673.
\]
Forecast Weights and EWMAs

- For ARIMA models without moving average terms, it is clear how forecasts are obtained from the observed series $Y_1, Y_2, \ldots, Y_t$.

- For models with MA terms, the noise terms appear in the forecasts.

- Recall that for any invertible ARIMA process, we can write it in terms of an infinite sum of AR terms:
  \[ Y_t = \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + \cdots + e_t. \]

- Changing $t$ to $t+1$, we have:
  \[ Y_{t+1} = \pi_1 Y_t + \pi_2 Y_{t-1} + \cdots + e_{t+1}, \]
  and taking conditional expectations of both sides (given $Y_1, Y_2, \ldots, Y_t$), we have:
  \[ \hat{Y}_t(1) = \pi_1 Y_t + \pi_2 Y_{t-1} + \cdots \]
EWMA in the $IMA(1, 1)$ Model

- In the $IMA(1, 1)$ model where $Y_t = Y_{t-1} + e_t - \theta e_{t-1}$, the $\pi$-weights are
  
  $$\pi_j = (1 - \theta) \theta^{j-1} \text{ for } j \geq 1.$$ 

- Thus the one-step-ahead forecast, called an exponentially weighted moving average (EWMA), is
  
  $$\hat{Y}_{t}(1) = (1 - \theta) Y_t + (1 - \theta) \theta Y_{t-1} + (1 - \theta) \theta^2 Y_{t-2} + \cdots$$

- These weights decrease exponentially, and by summing a geometric series, we can see that they sum to 1.

- We can write this in a recursive updating formula as
  
  $$\hat{Y}_{t}(1) = (1 - \theta) Y_t + \theta \hat{Y}_{t-1}(1).$$
Example of Forecasting with the $IMA(1, 1)$ Model

In practice, if our model specification shows that an $IMA(1, 1)$ model is appropriate for our data, we can estimate $\theta$ (and the smoothing constant, $1 - \theta$) in the usual way and compute an EWMA forecast using this formula.

See the R example of forecasting the logged oil price data with an $IMA(1, 1)$ model and the `sarima.fore` function.
If our model involves taking first differences to achieve stationarity, we could forecast future values by either

1. forecasting the original nonstationary series (as we did in the IMA(1, 1) example with the logged oil price data), or
2. forecasting the stationary differenced series \( W_t = Y_t - Y_{t-1} \) and reversing the differencing by summing the results to get the forecasts in the original terms.

Both methods lead to *exactly the same* forecasts, since differencing is a linear operation.

This fact also applies to differences of *any order*. 
Often we choose to model the natural logarithms of the original data.

Let \( \{Y_t\} \) denote the original series and let \( Z_t = \log(Y_t) \).

Then the (back-transformed) minimum mean square error forecast of \( Z_{t+\ell} \) is NOT the minimum mean square error forecast of \( Y_{t+\ell} \), since

\[
E[Y_{t+1}|Y_t, Y_{t-1}, \ldots, Y_1] \geq \exp[E(Z_{t+1}|Z_t, Z_{t-1}, \ldots, Z_1)].
\]

However, consider that if \( Z_t \) is normally distributed, then \( Y_t \) must have had a skewed distribution (specifically, a log-normal distribution).
For this skewed-right distribution, the \textit{mean absolute error} is a better criterion, and the \textit{median} of the conditional distribution (given the observed data) may be considered optimal.

And since $Z_t$ is normal, this median of its conditional distribution equals the mean of its conditional distribution.

And

$$E[Z_t] = \text{median}[Z_t] = \text{median}[\log(Y_t)] = \log[\text{median}(Y_t)].$$

So getting the forecast $\hat{Z}_t(\ell)$ in the usual way and then using $e^{\hat{Z}_t(\ell)}$ as the forecast for $Y_{t+\ell}$ is justified as minimizing the mean absolute error with respect to the distribution of $Y_t$. 

Hitchcock

STAT 520: Forecasting and Time Series
Recall that when our original observed time series is nonstationary, two important approaches to “achieve stationarity” are *detrending* or *differencing*.

In some cases, we could use either approach to forecast future values (say, at time $t + \ell$) of a nonstationary series.

We could (1) estimate a trend model and obtain the detrended (residual) series based on that; (2) fit a stationary ARMA model to the detrended data (if the detrended series is not simply white noise); (3) forecast the value of the detrended series at time $t + \ell$ using our usual ARMA forecasting technique; and (4) add that to the prediction of the trend model at time $t + \ell$. 
The other approach would just be to use a ARIMA model with differencing on the original series and forecast based on that (including a constant term in the ARIMA model if needed).

This latter approach with the ARIMA model is simpler and usually works better, unless there is some clear trend in the series that differencing cannot handle.

See the chicken price example in R for an example of both approaches.
Note the forecast of $Y_{t+\ell}$ is an expected value of that future observation (given $Y_1, \ldots, Y_t$).

Sometimes we may be interested in using our chosen model to simulate random realizations of the process (random variables, NOT an expected value) for one or more future time points.

The `simulate` function in the `forecast` package in R can randomly simulate such future observations of the process, based on the chosen model.

Note that you can think of the forecast $\hat{Y}_t(\ell)$ as approximately the average of many, many such simulated future values of the series at time $t + \ell$ (see plots in R).