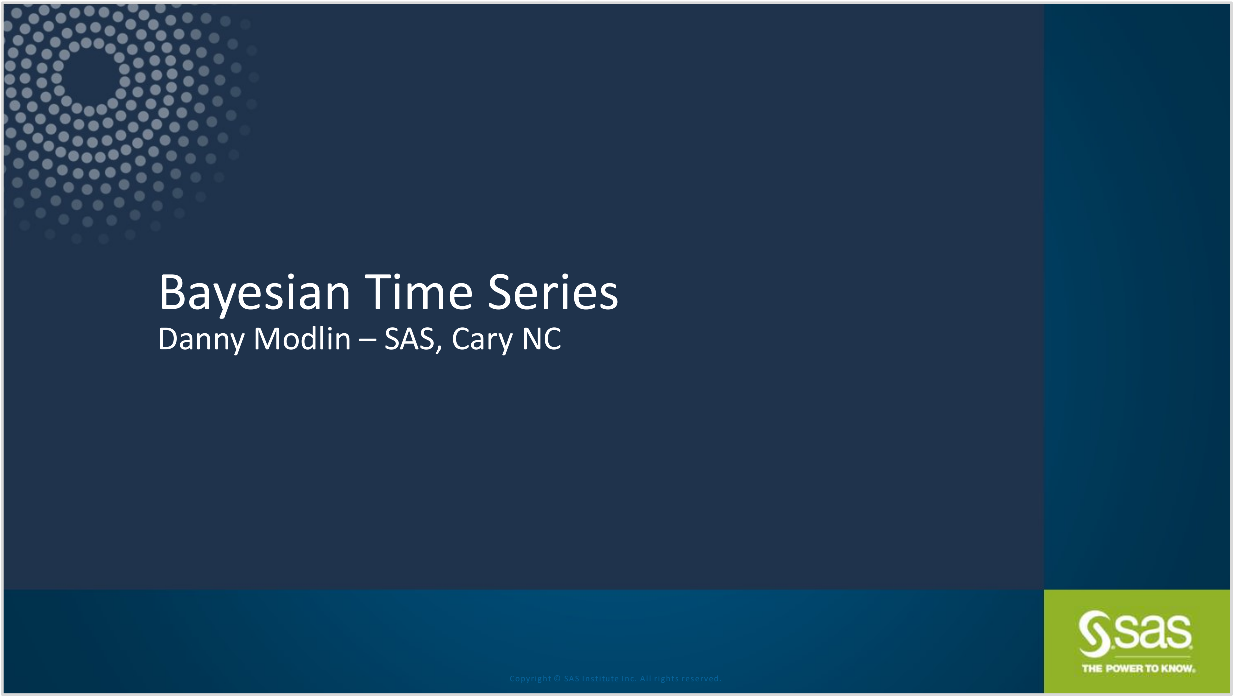
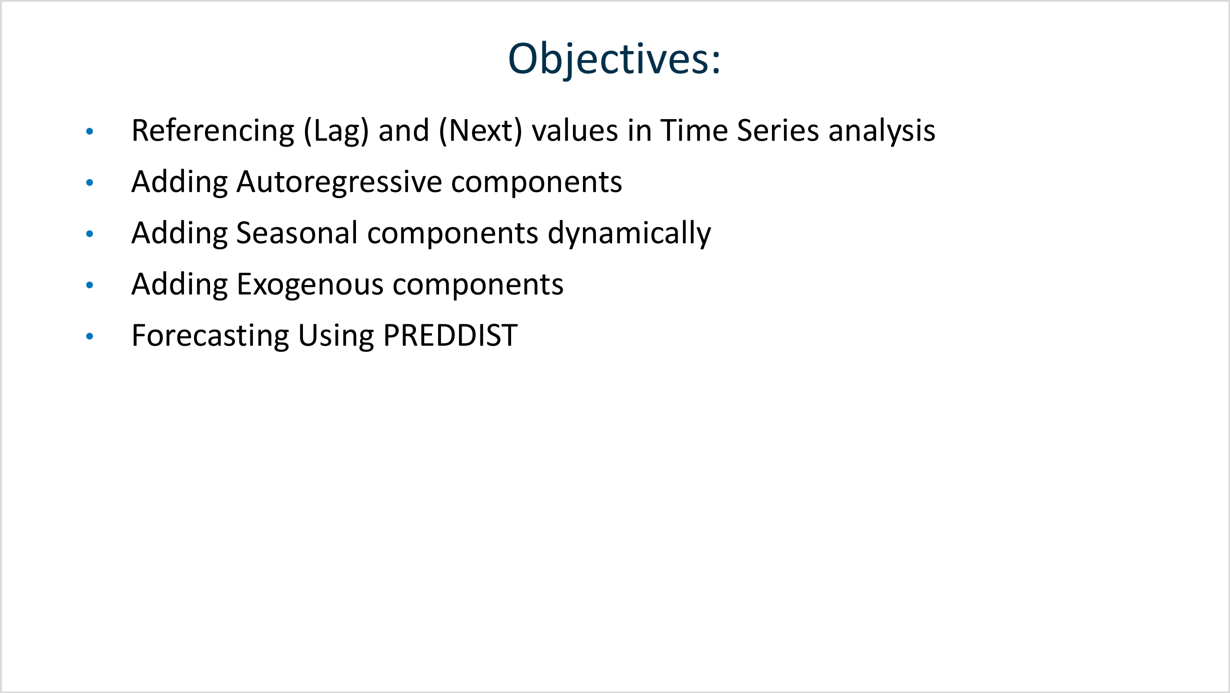


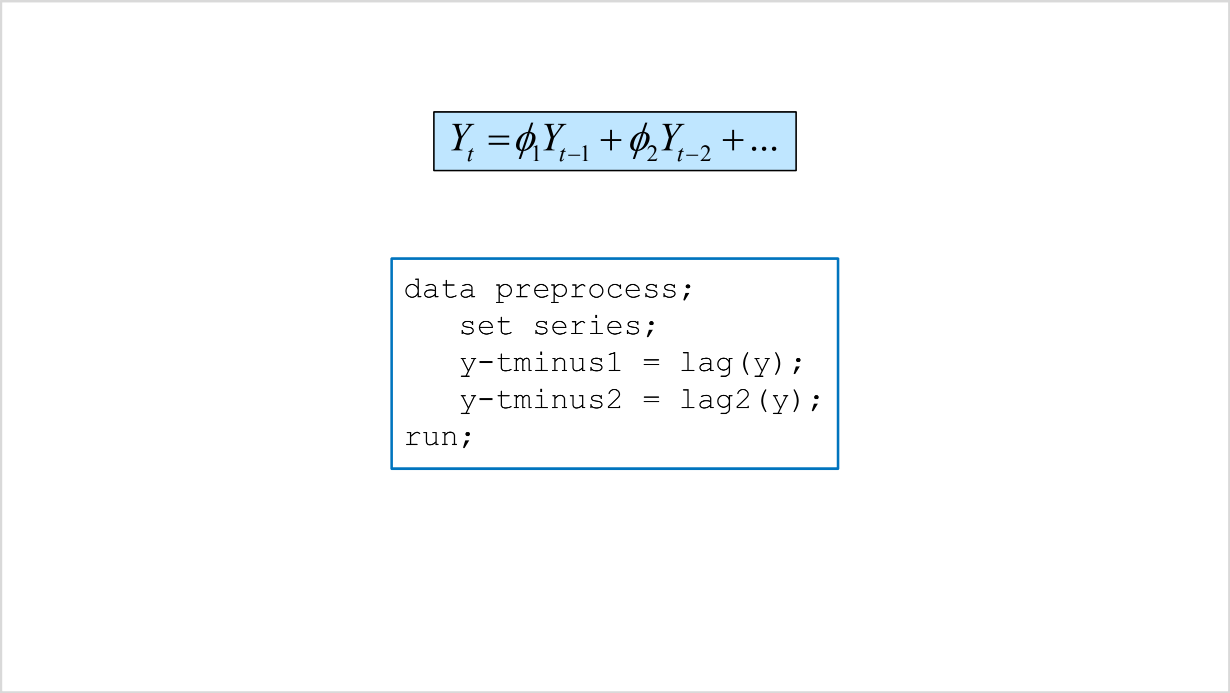
Bayesian Time Series

Danny Modlin – SAS, Cary NC



Objectives:

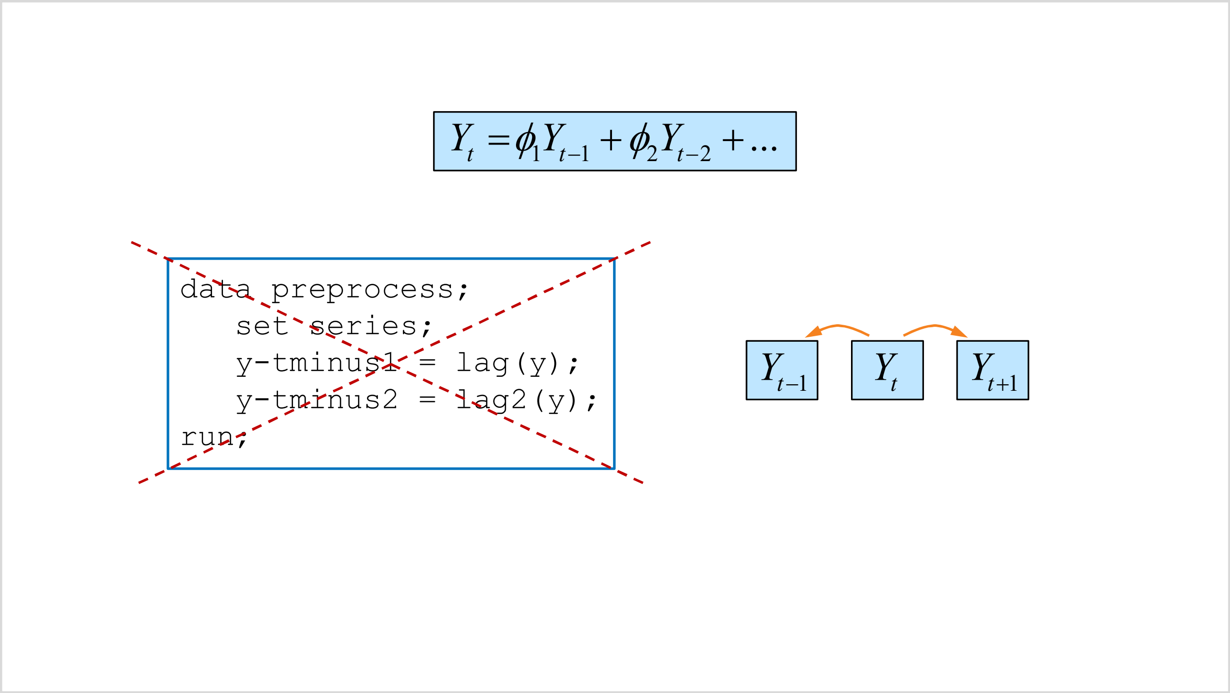
- Referencing (Lag) and (Next) values in Time Series analysis
- Adding Autoregressive components
- Adding Seasonal components dynamically
- Adding Exogenous components
- Forecasting Using PREDDIST



$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots$$

```
data preprocess;
   set series;
   y-tminus1 = lag(y);
   y-tminus2 = lag2(y);
run;
```

Let’s begin our discussion of Bayesian time series structure with autoregressive elements. Prior to SAS/STAT 14.1, coding these elements was more time consuming than it is now. To fit autoregressive time series models in the past, you had to preprocess the data. This put the work on you to create variables within the data set that contained the lagged values of the response series.

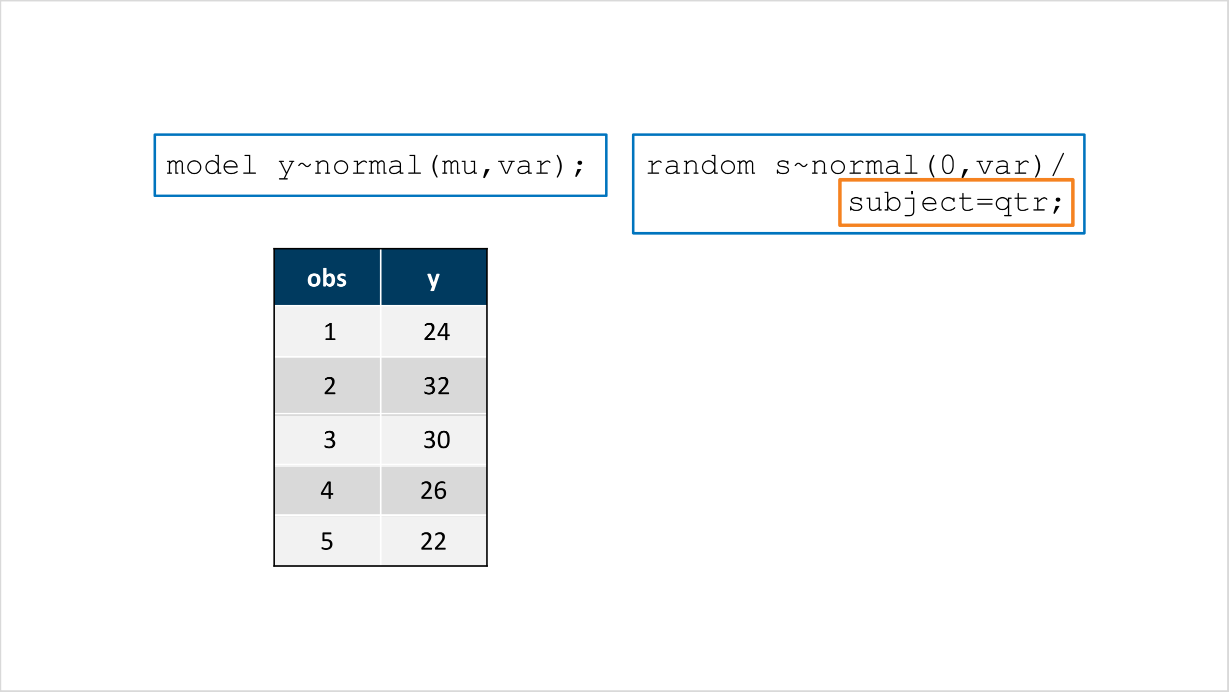


$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots$$

```
data preprocess;
   set series;
   y-tminus1 = lag(y);
   y-tminus2 = lag2(y);
run;
```

Y_{t-1} Y_t Y_{t+1}

Now, we have access to lead and lagged values for random variables that are indexed. What do I mean by *indexed*? Two types of random variables are indexed in the MCMC procedure.

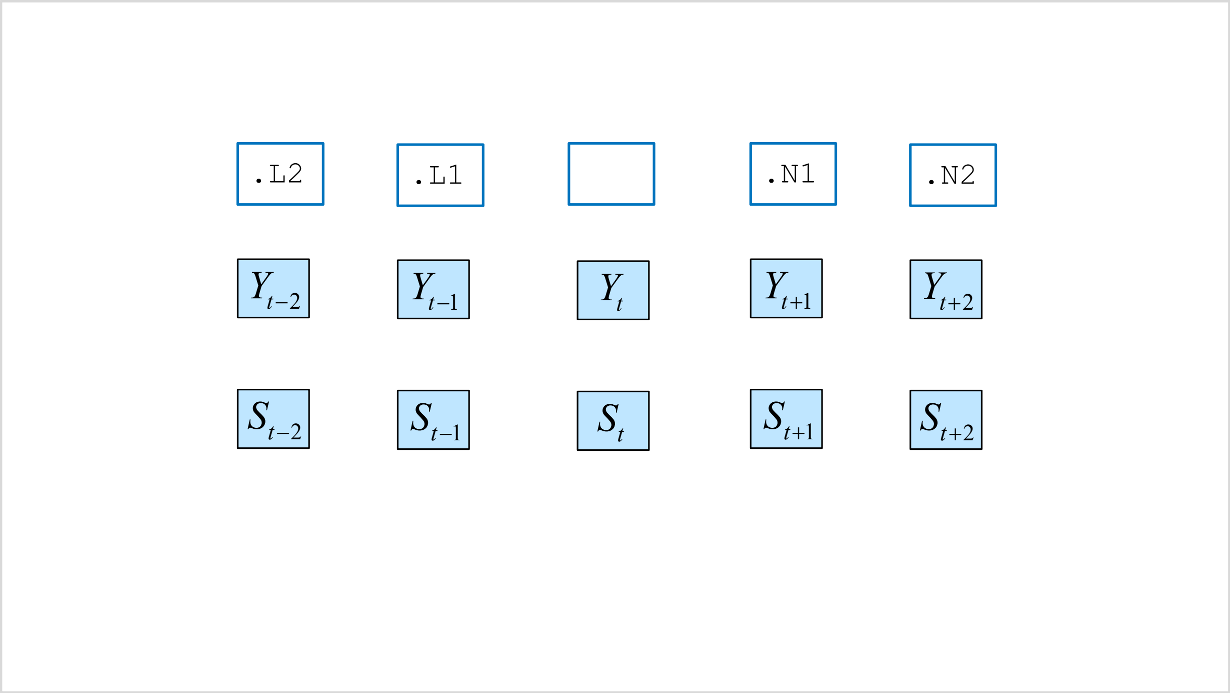


```
model y~normal(mu,var);
```

```
random s~normal(0,var)/
                 subject=qtr;
```

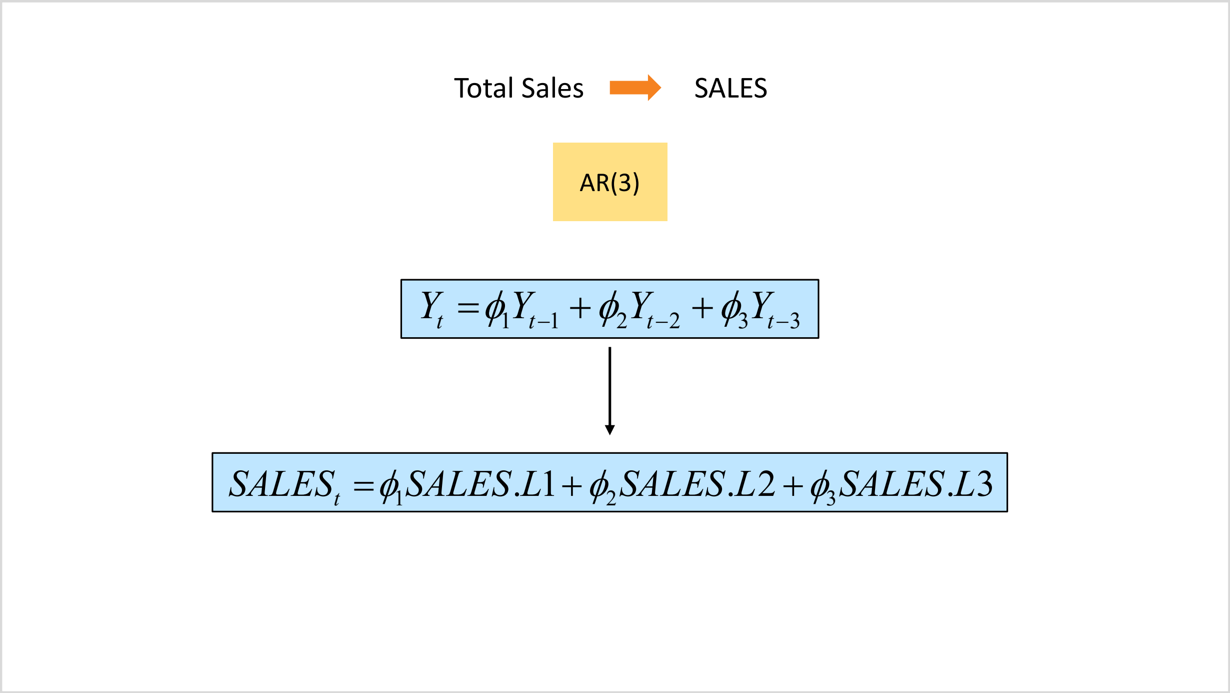
obs	y
1	24
2	32
3	30
4	26
5	22

The first is the response variable, our time series. In the MODEL statement, this variable is indexed by observations. The second is a random variable placed in the RANDOM statement. This variable is indexed by the SUBJECT= option.



.L2	.L1		.N1	.N2
Y_{t-2}	Y_{t-1}	Y_t	Y_{t+1}	Y_{t+2}
S_{t-2}	S_{t-1}	S_t	S_{t+1}	S_{t+2}

To access both the lead and lagged values of these indexed variables, we state the variable name followed by either .L or .N to access lagged or next values respectively. Let’s look at an example.



Total Sales → SALES

AR(3)

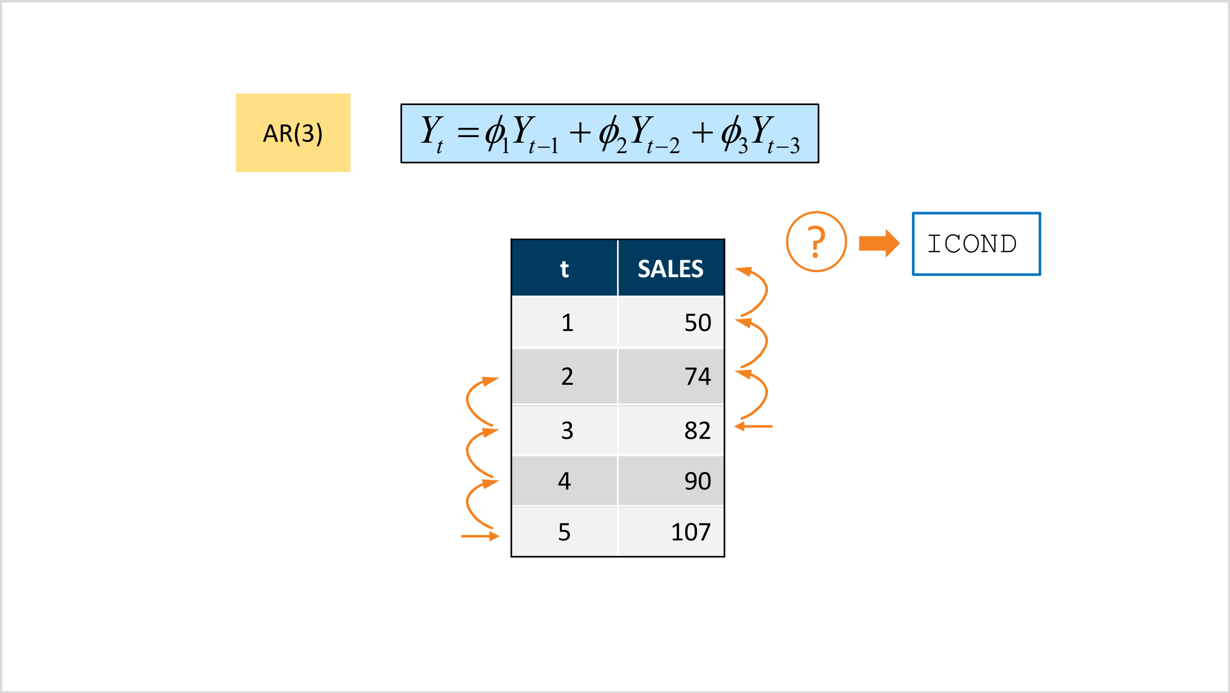
$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3}$$

$$SALES_t = \phi_1 SALES.L1 + \phi_2 SALES.L2 + \phi_3 SALES.L3$$

Suppose you have a time series of total sales data accumulated monthly. During your exploration, you determine that a third-order autoregressive model (AR(3)) would be appropriate for your series. So, three lagged values of the response series would be included in your time series model.

When adding these elements to our model, we would code **SALES.L1** for the first lagged value, **SALES.L2** for the second, and **SALES.L3** for the third. We do not use it here, but if we wanted to look forward in time, we would use **SALES.N1** to access the value one time unit ahead in the series.

Creating these lagged values using a data set was not too complex, but with .L and .N, it is much easier. However, there was no built-in way to account for the initial states of these lagged variables. What do I mean by *initial states*?



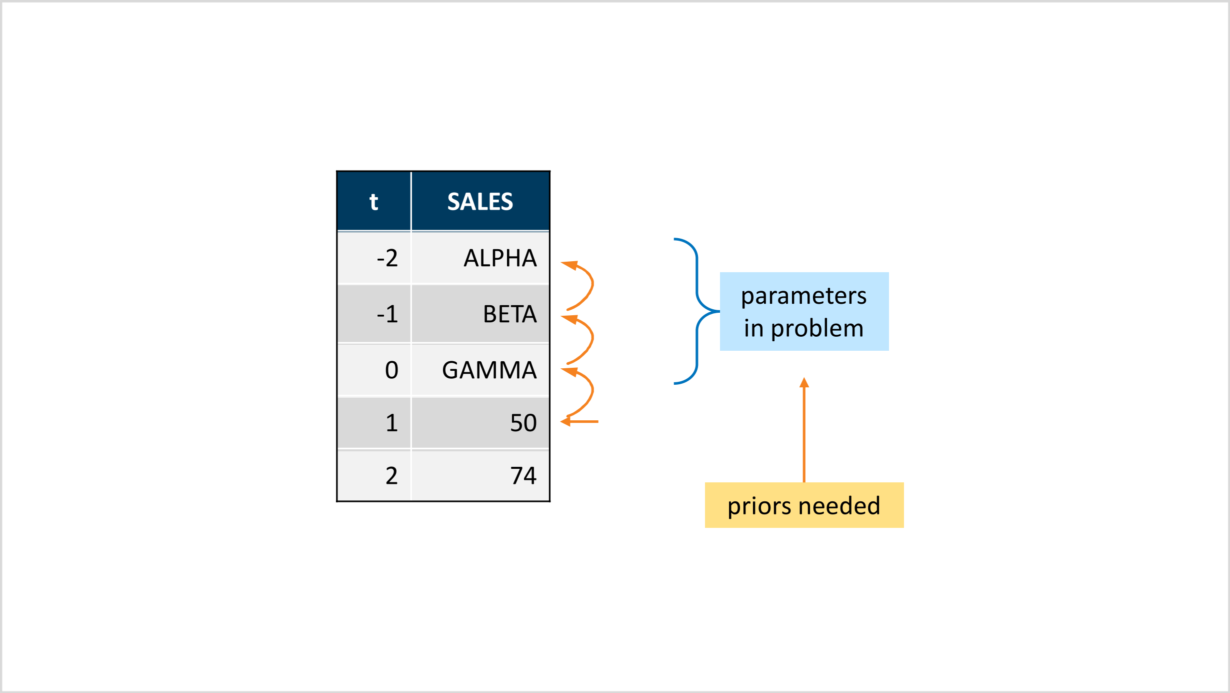
AR(3)

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3}$$

t	SALES
1	50
2	74
3	82
4	90
5	107

To forecast the total sales at time position 5 in the series, we would include the values from time positions 4, 3, and 2. This is not a problem because those values are found within our response series. What happens if we wanted to forecast position 3 in the series? We have the total sales of time positions 2 and 1. You might now see the problem that we have.

As we approach the start of the time series, we run out of information for our lagged time values. This was a problem before the ICOND= option. In the MODEL statement or the RANDOM statement, we can now account for these initial states (or initial conditions).

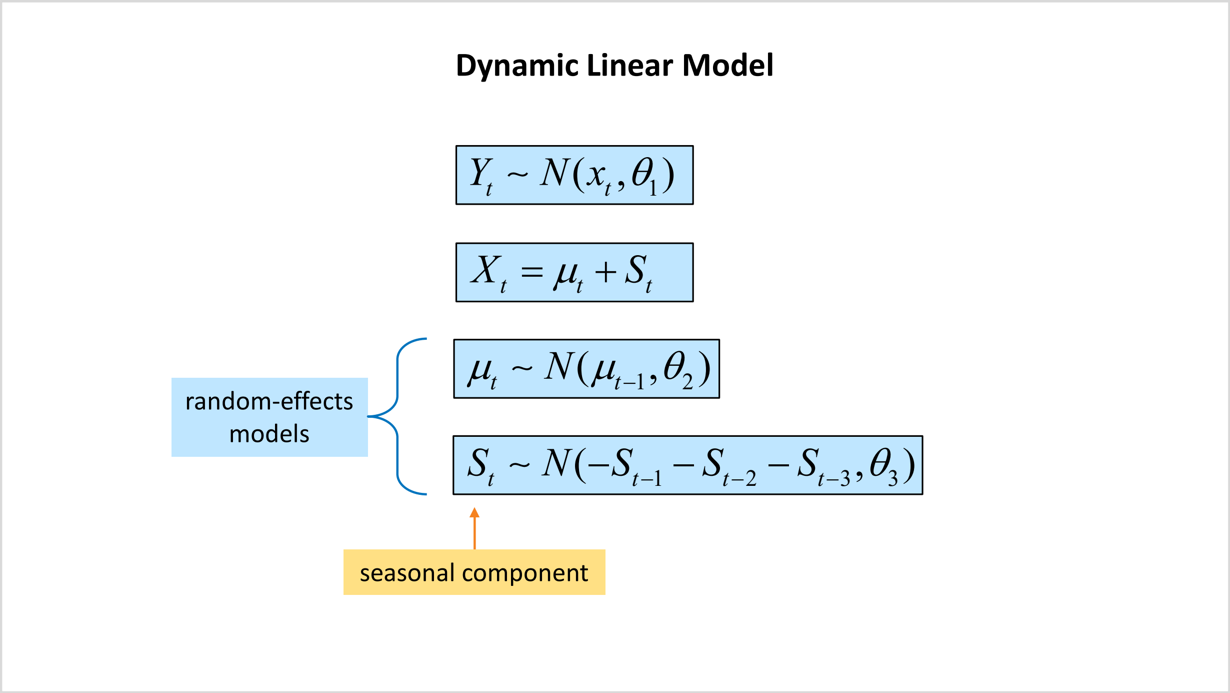


t	SALES
-2	ALPHA
-1	BETA
0	GAMMA
1	50
2	74

parameters in problem

priors needed

In our example, we can include ICOND= (alpha beta gamma) in the MODEL statement. These initial states are treated as parameters in the problem, and we place priors against them just like any other parameter in our model. Three items are listed due to the maximum number of initial states needed being three when we are at the very beginning of the series. Using this technique, we do not lose data at the front from missing values.



Dynamic Linear Model

$$Y_t \sim N(x_t, \theta_1)$$

$$X_t = \mu_t + S_t$$

$$\mu_t \sim N(\mu_{t-1}, \theta_2)$$

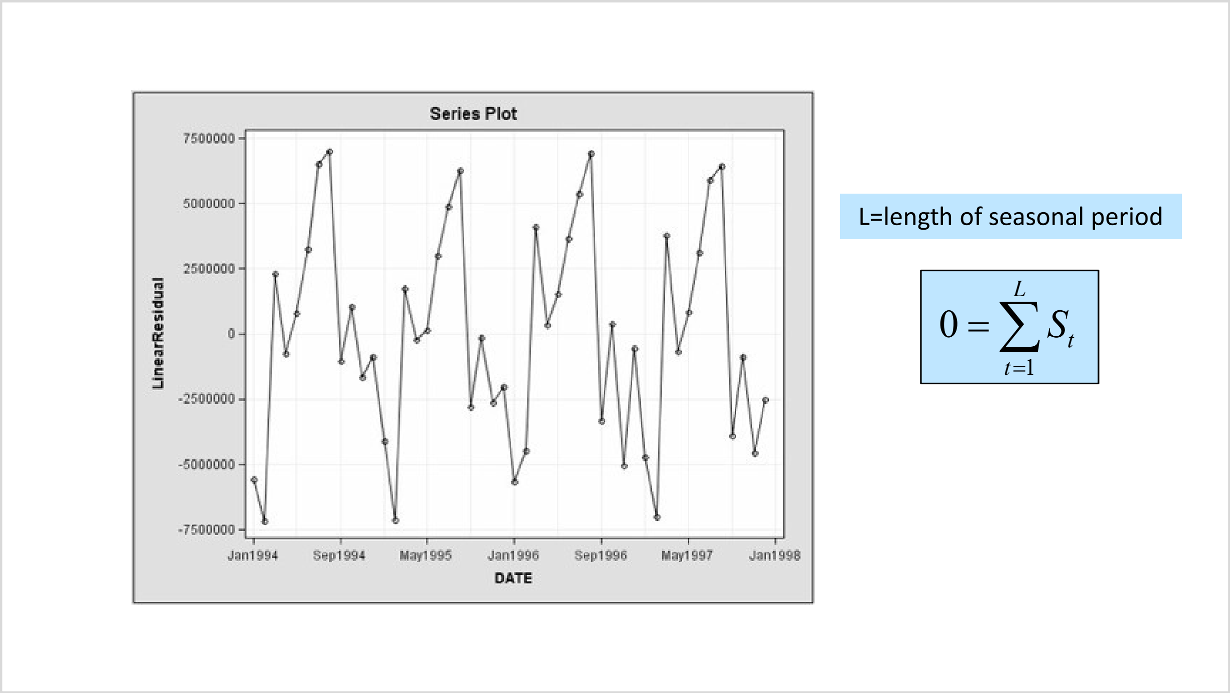
$$S_t \sim N(-S_{t-1} - S_{t-2} - S_{t-3}, \theta_3)$$

random-effects models

seasonal component

Performing a Bayesian time series analysis also enables you to use a dynamic linear model setup. This setup is a very general type of nonstationary time series model. With this, you can create models with time-varying coefficients where you can explore stochastic shifts in regression parameters.

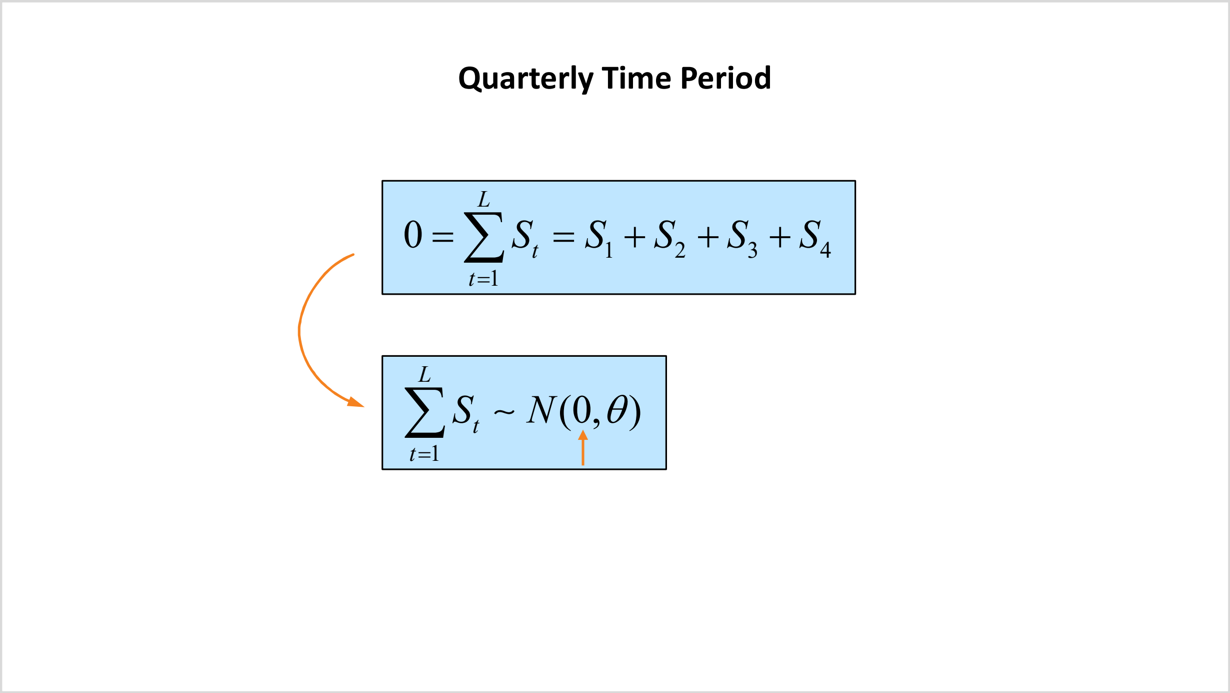
To do this, we use random-effects models that specify time dependence between successive parameter values in the form of smoothness priors. The best application of this structure is for seasonality components.



L=length of seasonal period

$$0 = \sum_{t=1}^{L} S_t$$

As you recall, seasonality components are deviations from the trend. These seasonality values sum to zero across the length of the seasonal period. For example, let’s look at sales data that have been accumulated to quarterly averages. Upon inspection, we determine that there is a seasonal pattern existing across the quarterly values.

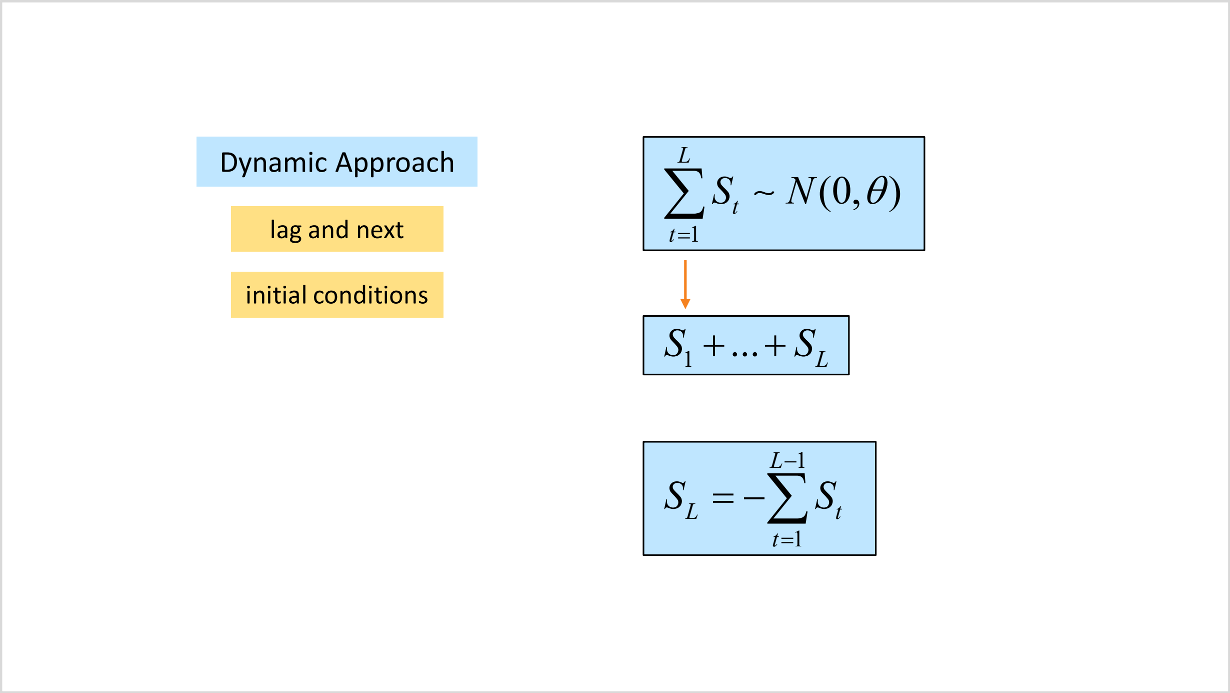


Quarterly Time Period

$$\boxed{0 = \sum_{t=1}^{L} S_t = S_1 + S_2 + S_3 + S_4}$$

$$\boxed{\sum_{t=1}^{L} S_t \sim N(0, \underset{\uparrow}{\theta})}$$

From a deterministic approach, these quarterly seasonal component values will sum to zero across four consecutive time points. This is due to the period of quarterly data being 4 in length. Taking a more dynamic approach, this sum is zero in the mean of the distribution with an additional variability.



Dynamic Approach

lag and next

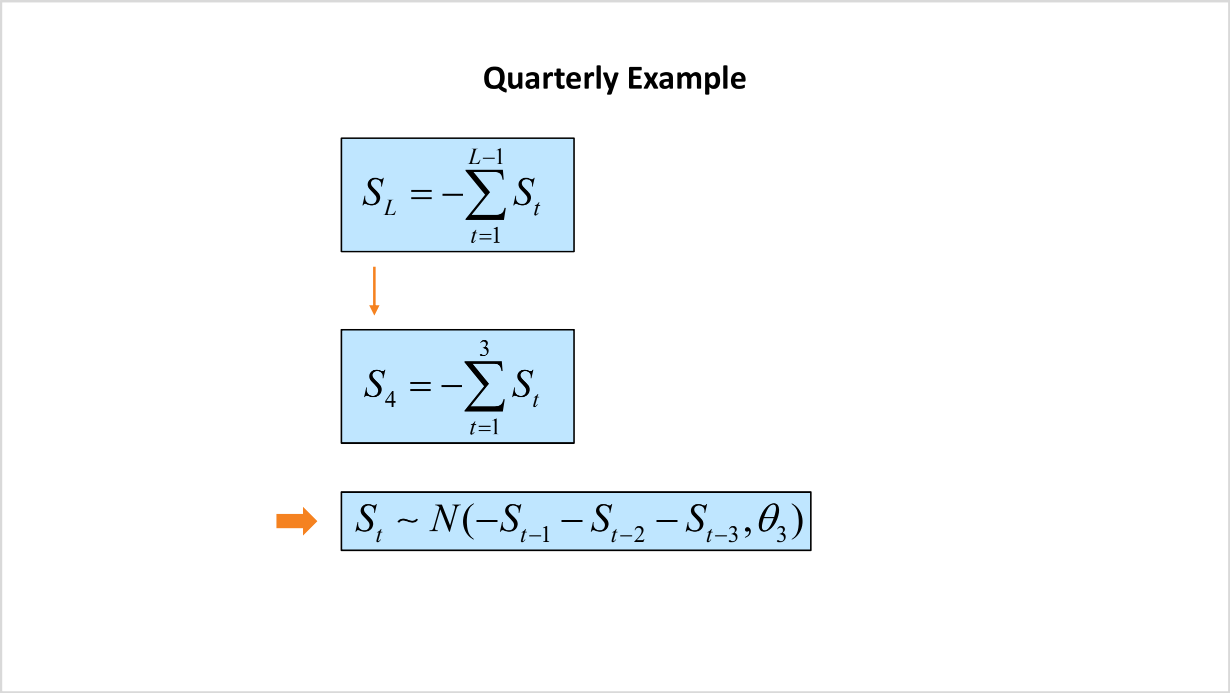
initial conditions

$$\sum_{t=1}^{L} S_t \sim N(0,\theta)$$

$$S_1 + \ldots + S_L$$

$$S_L = -\sum_{t=1}^{L-1} S_t$$

The additional benefit of using the dynamic approach to this seasonal component is that we can now use the lag and next elements as well as initial conditions during the modeling process. Because we know that the sum of all the seasonal components should add to zero in the mean, we can model the seasonal component at the current time point as the sum of the negative previous seasonal components.



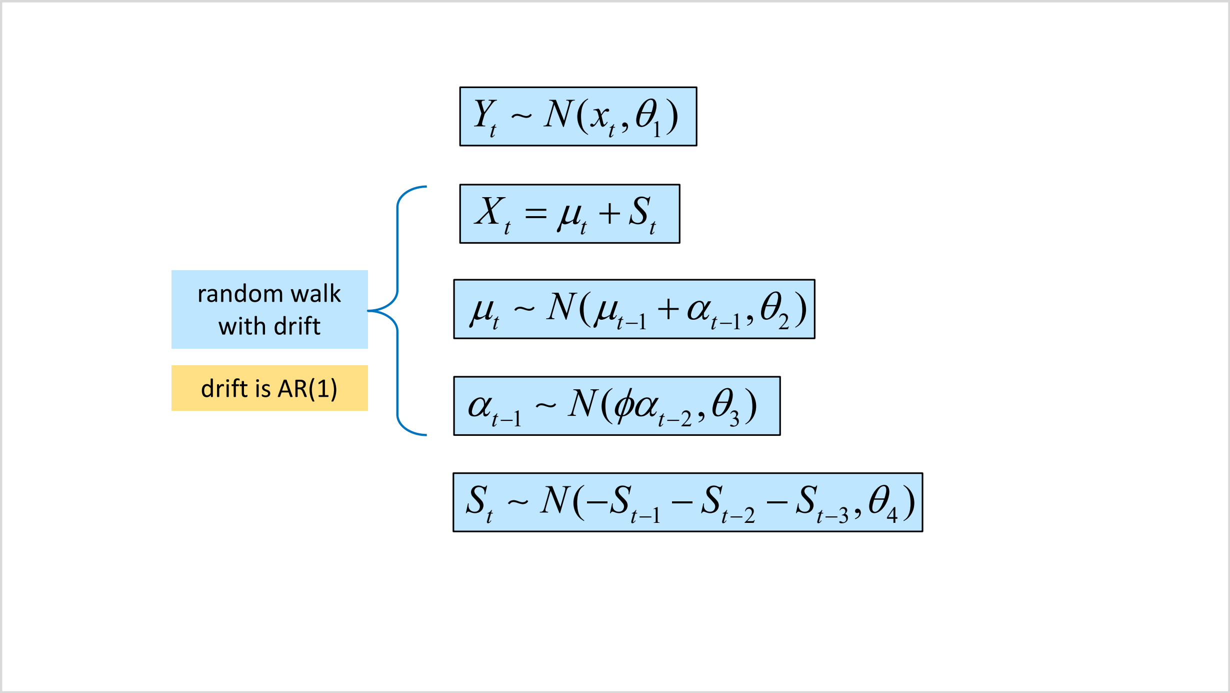
Quarterly Example

$$S_L = -\sum_{t=1}^{L-1} S_t$$

$$S_4 = -\sum_{t=1}^{3} S_t$$

$$S_t \sim N(-S_{t-1} - S_{t-2} - S_{t-3}, \theta_3)$$

In our quarterly example, this would make the value of our current time point equal to the negative of the sum of the previous three seasonal components. We place this calculation in the mean of our distribution with a smoothness variance component.



$$Y_t \sim N(x_t, \theta_1)$$

$$X_t = \mu_t + S_t$$

$$\mu_t \sim N(\mu_{t-1} + \alpha_{t-1}, \theta_2)$$

$$\alpha_{t-1} \sim N(\phi \alpha_{t-2}, \theta_3)$$

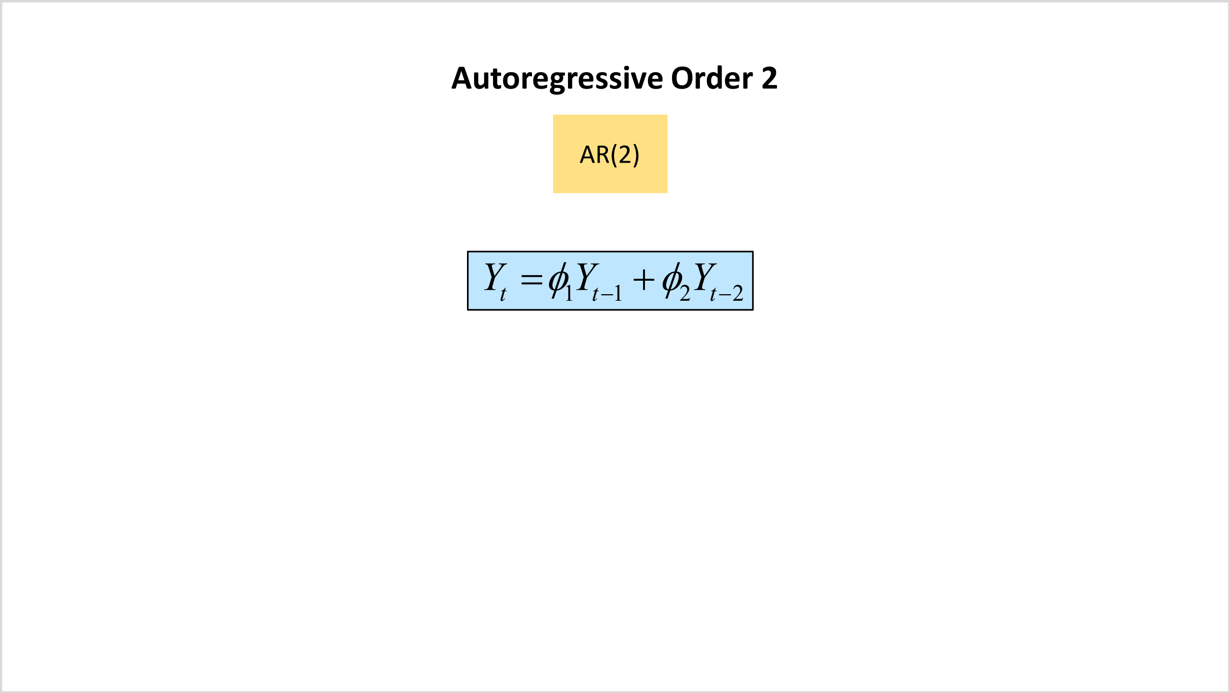
$$S_t \sim N(-S_{t-1} - S_{t-2} - S_{t-3}, \theta_4)$$

random walk with drift

drift is AR(1)

In addition to seasonality components, we could entertain trends that follow a random walk with drift. This drift could follow a first-order autoregressive process.

The application of dynamic linear model setups within our time series models greatly expands the ability of our Bayesian approach to modeling.

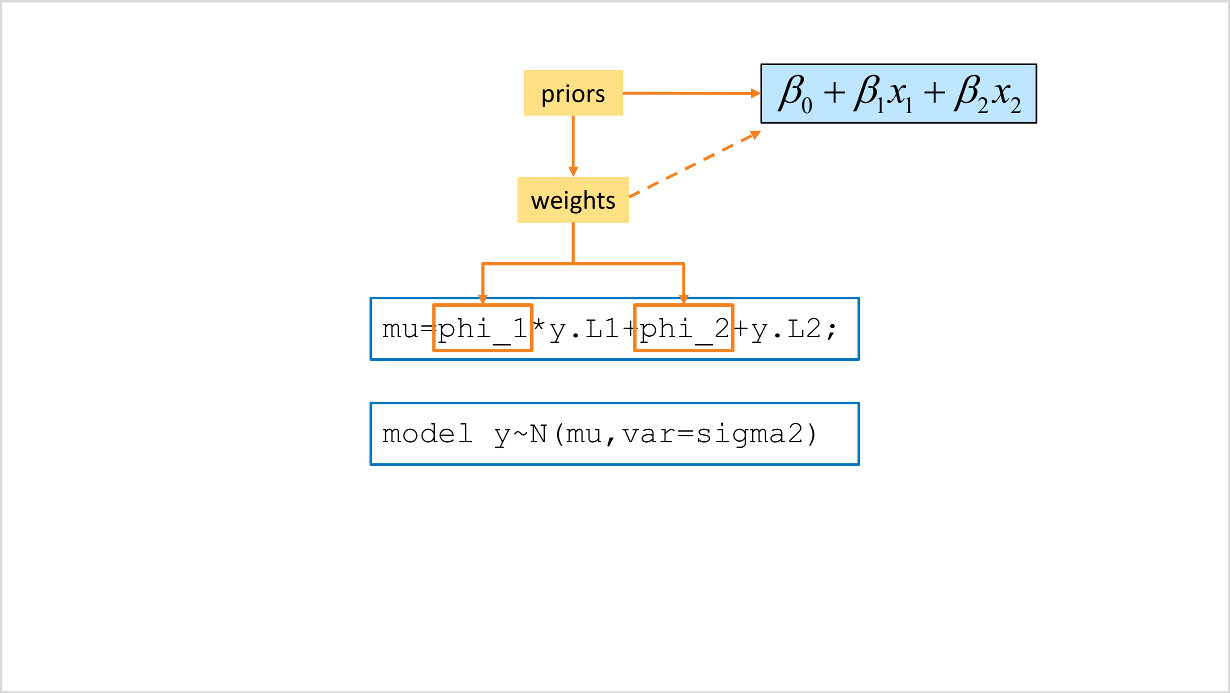


Autoregressive Order 2

AR(2)

$$\boxed{Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2}}$$

Previously, we discussed and practiced adding an autoregressive component to our model. The example was an AR(2).



$$\beta_0 + \beta_1 x_1 + \beta_2 x_2$$

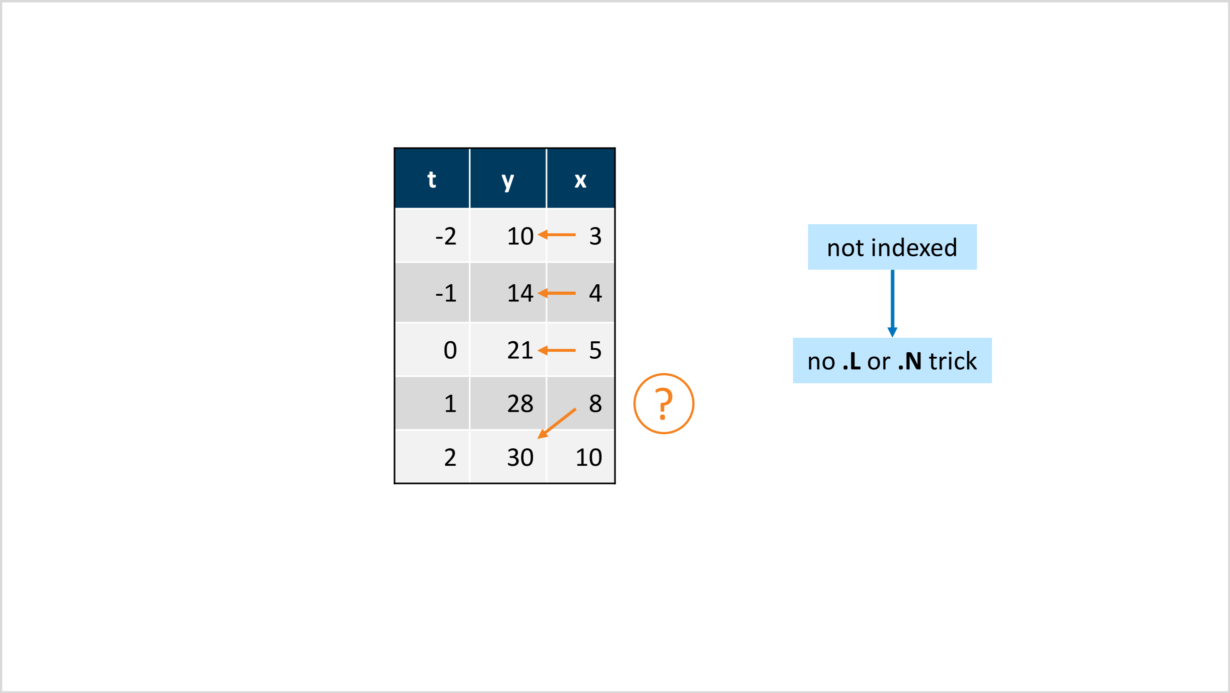
```
mu=phi_1*y.L1+phi_2+y.L2;
```

```
model  y~N(mu,var=sigma2)
```

As you recall, prior to the model line in our code, we composed the mean from a weighted combination of the previous two values of the time series. The phi coefficients in the model were the weights.

Think back to traditional regression equations from your past. You might see a resemblance of this autoregressive setup to that of a regression equation. In the regression version, our weights/coefficients are our beta estimates.

Much like in the autoregressive viewpoint, we place prior distributions against the coefficients, as they are the parameters of the problem.



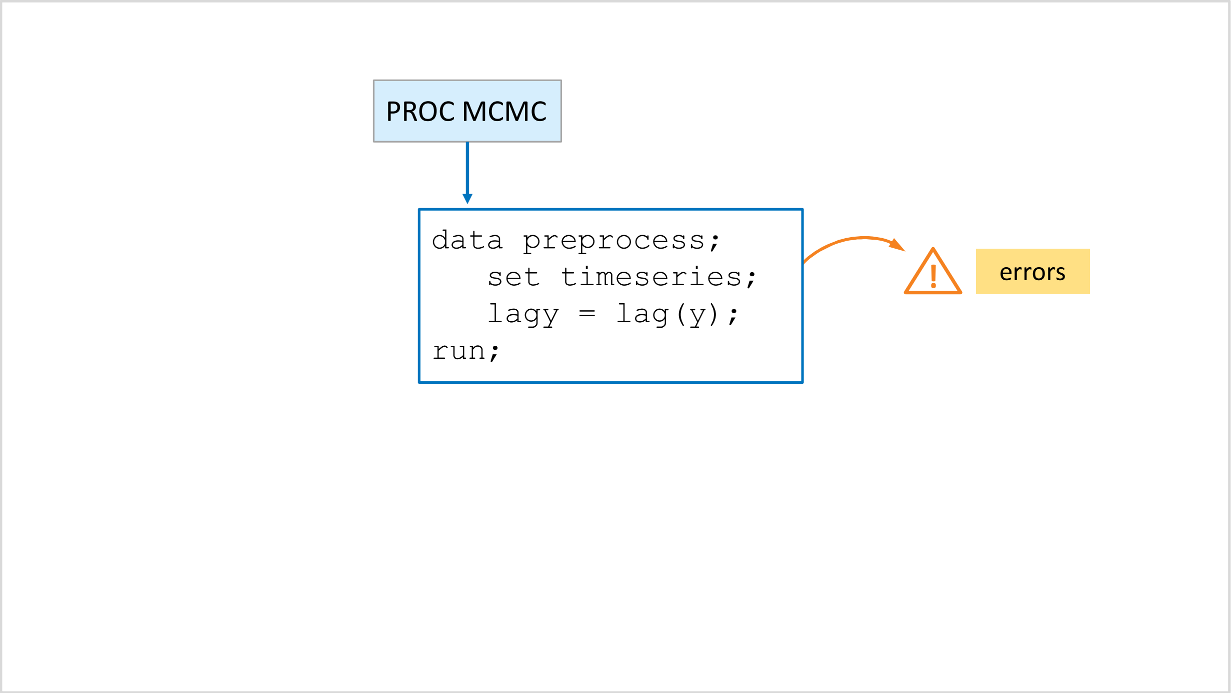
t	y	x
-2	10	3
-1	14	4
0	21	5
1	28	8
2	30	10

not indexed

no **.L** or **.N** trick

Adding a contemporaneous exogenous effect, where x at time t affects y at time t, is rather simple. However, what happens if there is a lag of the exogenous variable that influences the value of the series at time t?

Unlike the series y and our seasonal components from before, our exogenous variable does not appear on a code line that makes it indexed. Therefore, we do not have the luxury of the .L and .N elements. So how do we bring in these lagged values on an exogenous variable?



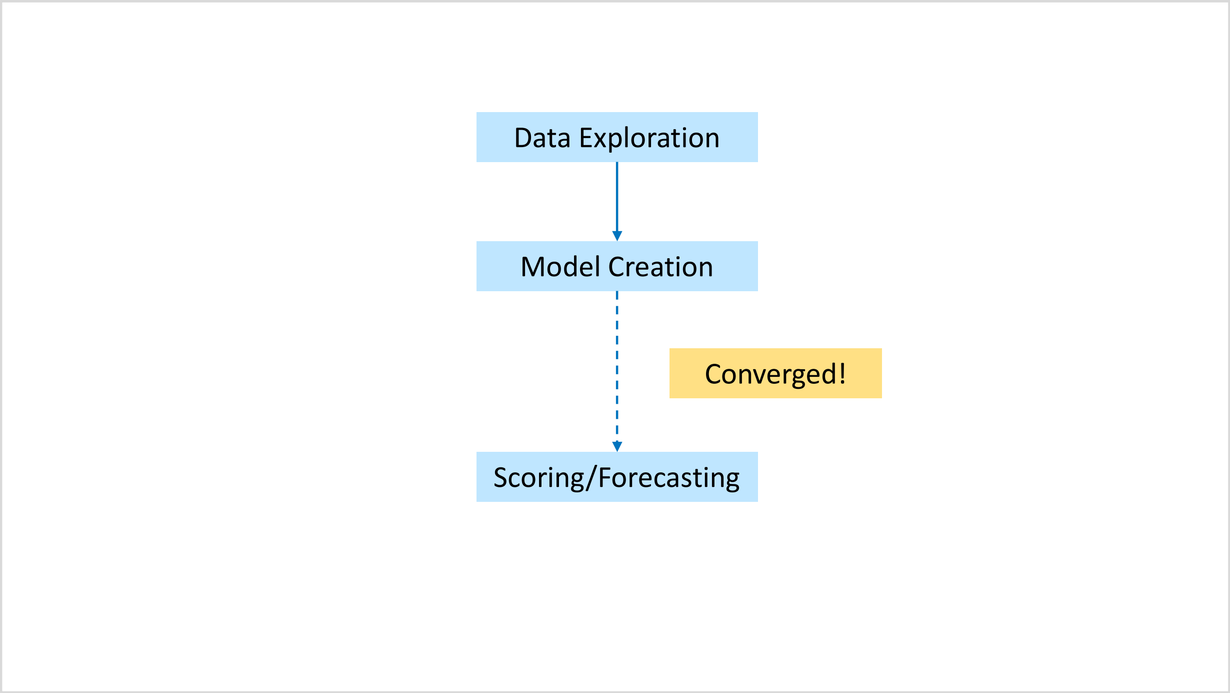
PROC MCMC

```
data preprocess;
   set timeseries;
   lagy = lag(y);
run;
```

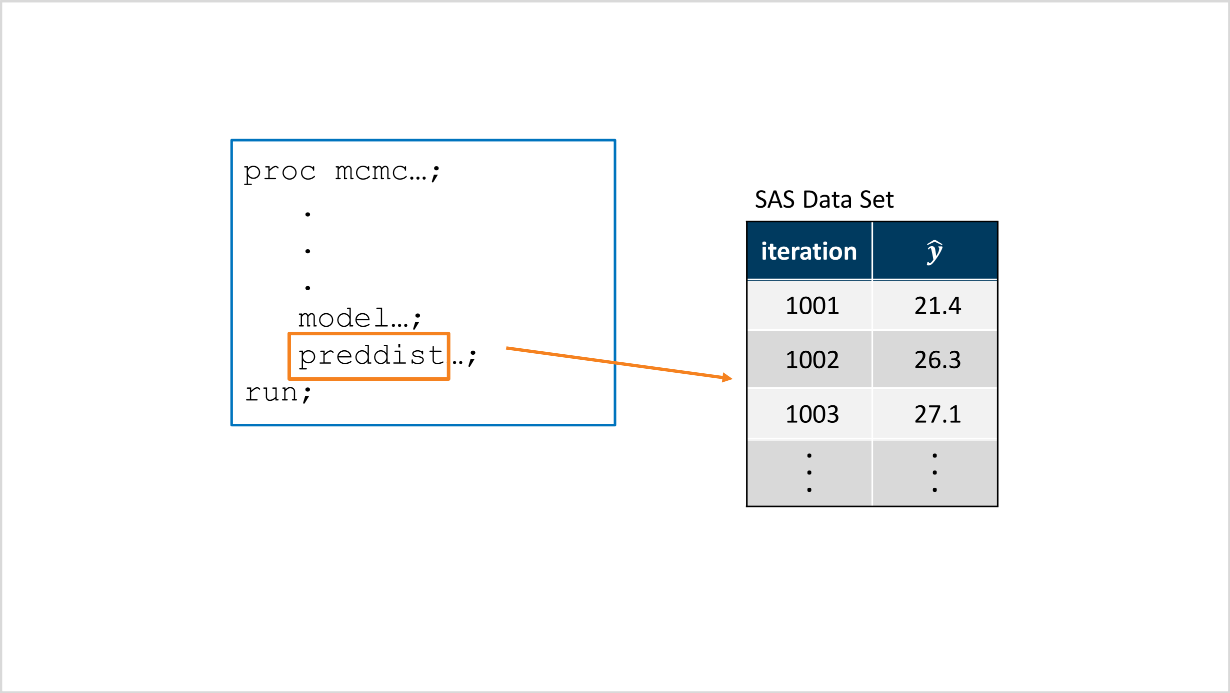
errors

Prior to the execution of the Bayesian code, we will use a DATA step to produce the needed lag terms for the model. PROC MCMC does allow for the use of DATA step procedures within open code to construct the mean element of the model. However, using the lag DATA step function will generate missing values at the start of the series, and these will cause errors in the execution.

There are several options of working with lagged values of exogenous variables. Using the external DATA step approach is the most direct. We will talk about this in the demonstration.



Once you have confirmed that you have a converged solution to your Bayesian analysis, you can then proceed with using this model to create forecasts (scoring). Trying to create forecasts before the solution has converged will waste your time.

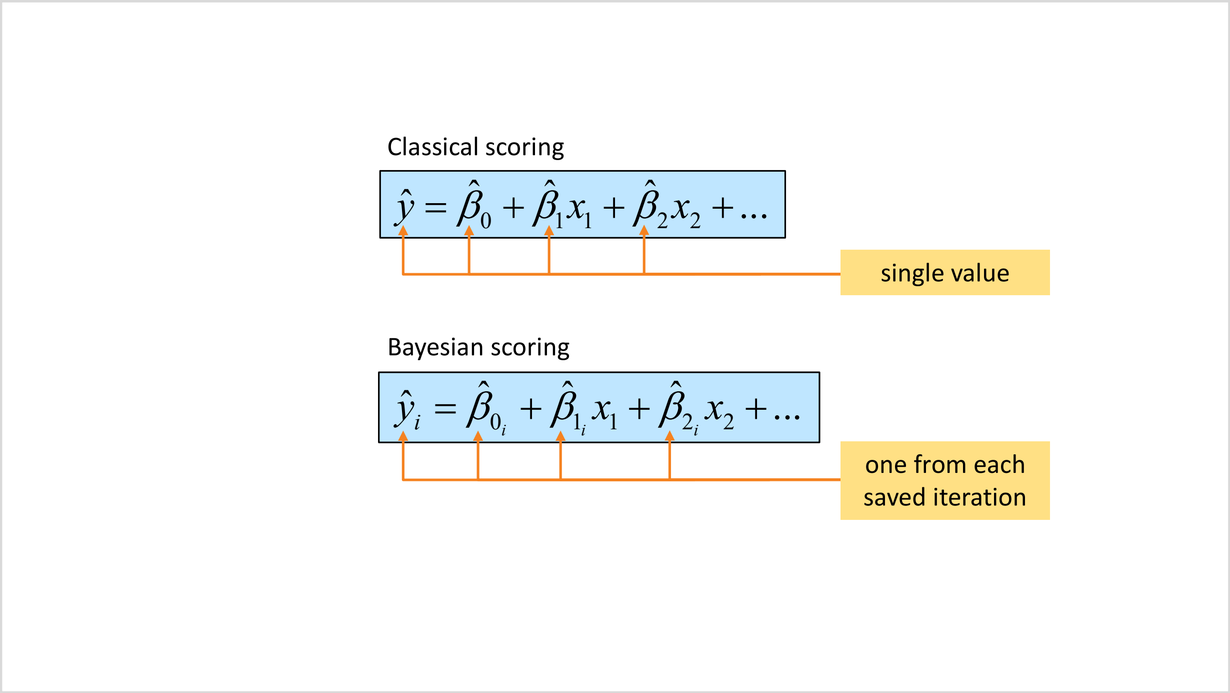


```
proc mcmc…;
     .
     .
     .
    model…;
    preddist…;
run;
```

SAS Data Set

iteration	$\hat{y}$
1001	21.4
1002	26.3
1003	27.1
⋮	⋮

The PREDDIST statement in PROC MCMC is the tool for creating a new SAS data set that contains random samples from the posterior predictive distribution of the response variable, our time series.



Classical scoring

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots$$

single value

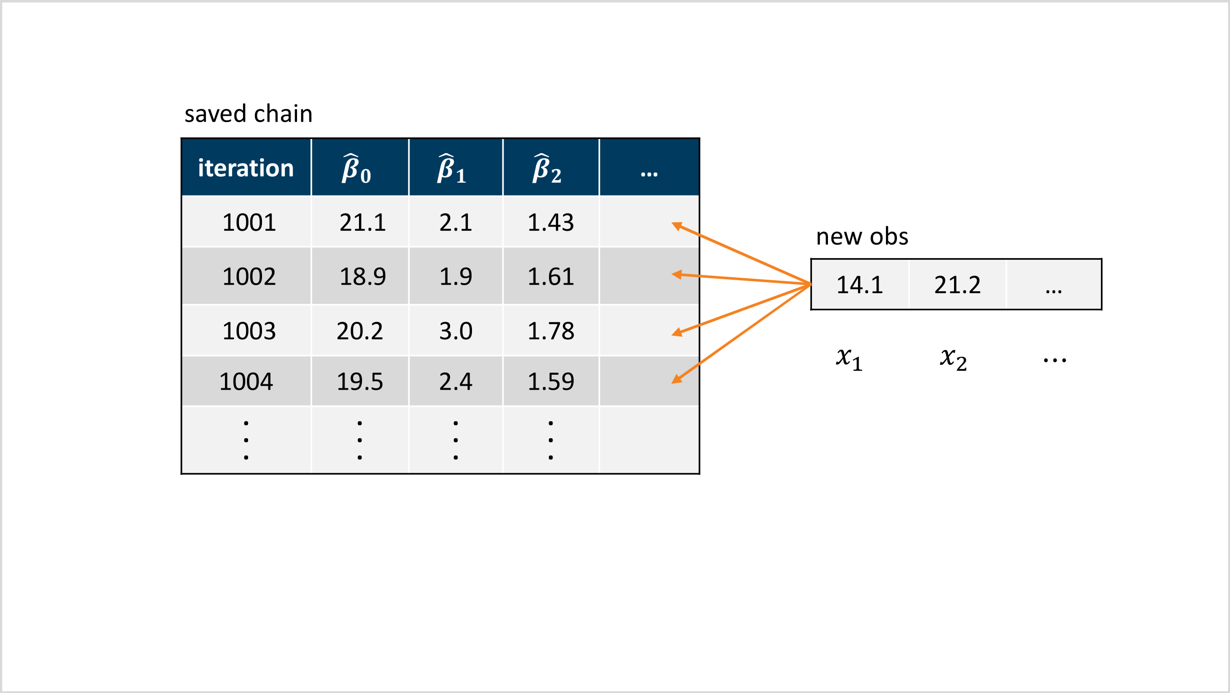
Bayesian scoring

$$\hat{y}_i = \hat{\beta}_{0_i} + \hat{\beta}_{1_i} x_1 + \hat{\beta}_{2_i} x_2 + \ldots$$

one from each saved iteration

There is a big difference between a classical approach to scoring and a Bayesian approach to scoring. In the classical approach, a single value is estimated for each of the parameters in the model. With the provided observation information and these parameter estimates, we arrive at a single prediction value for the current time point.

In the Bayesian approach, this differs because there is no longer a single estimate of the parameter. Recall that we are treating the parameters as random variables and ultimately ending on a posterior distribution for the parameters given the data. It is for this reason that scoring changes in the Bayesian approach.



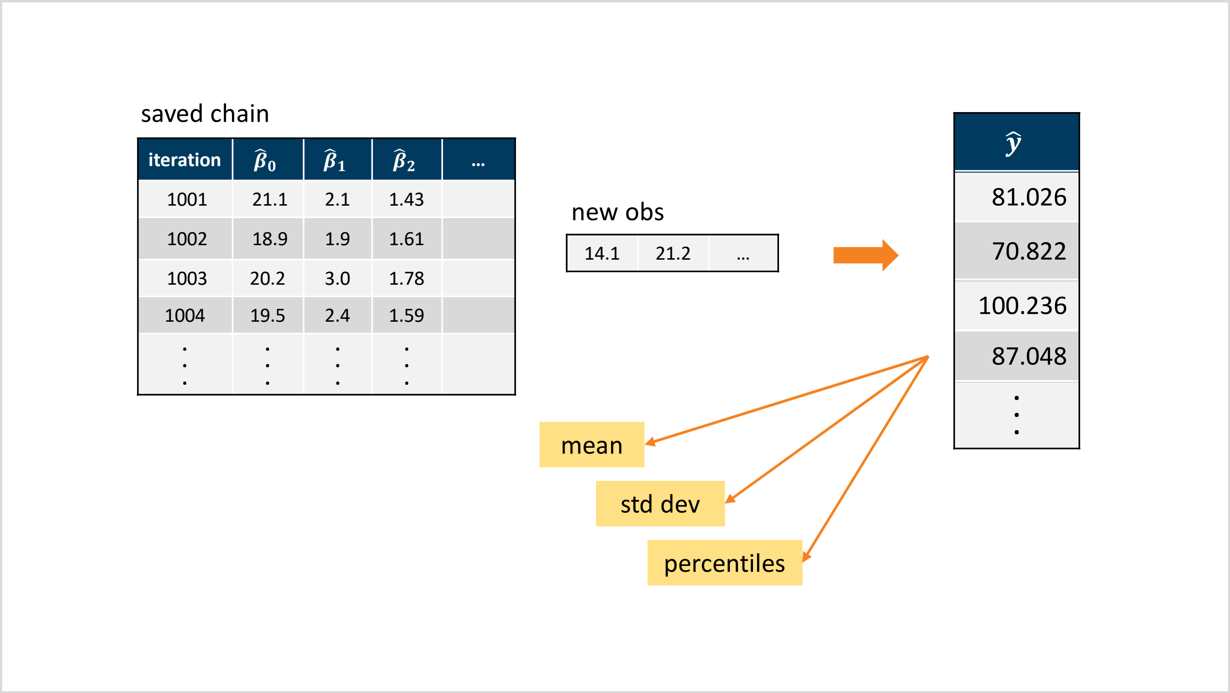
saved chain

iteration	$\widehat{\beta}_0$	$\widehat{\beta}_1$	$\widehat{\beta}_2$	…
1001	21.1	2.1	1.43	
1002	18.9	1.9	1.61	
1003	20.2	3.0	1.78	
1004	19.5	2.4	1.59	
⋮	⋮	⋮	⋮	

new obs

x_1	x_2	$\cdots$
14.1	21.2	…

PROC MCMC uses an iterative approach to sample from the posterior distribution of the parameters. At each individual iteration, the current value of the parameter is what, at this moment, we perceive the value of that parameter to be. In the end, we have a chain of iterations saved to represent the sample from the posterior distribution of the parameter. When PREDDIST is active, the values of the observations are combined with each iteration value of the parameters, yielding a predicted response value for each iteration. This chain of predictions creates the sample of the posterior distribution of the prediction at a time point of the series.



saved chain

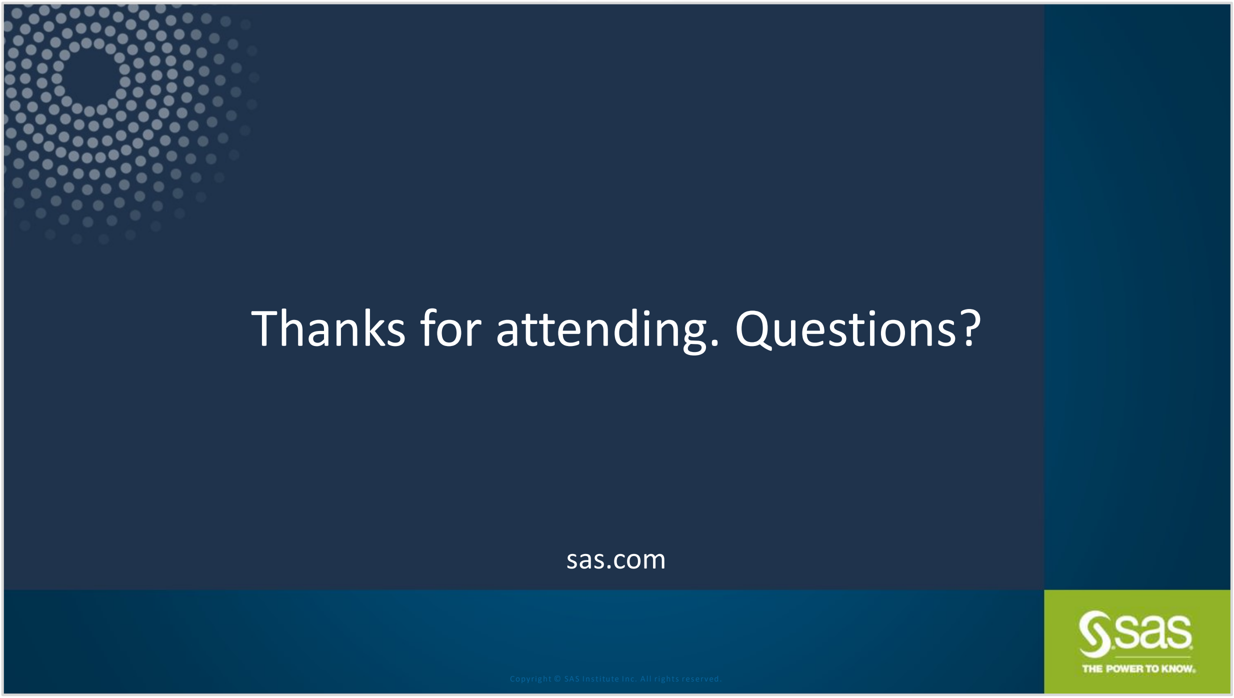
iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	...
1001	21.1	2.1	1.43	
1002	18.9	1.9	1.61	
1003	20.2	3.0	1.78	
1004	19.5	2.4	1.59	
⋮	⋮	⋮	⋮	

new obs

14.1	21.2	...

$\hat{y}$
81.026
70.822
100.236
87.048
⋮

From this posterior sample, we can calculate the mean, standard deviation, and percentiles of that distribution to aid in discussion and presentation.



Thanks for attending. Questions?

sas.com

THE POWER TO KNOW.