Public

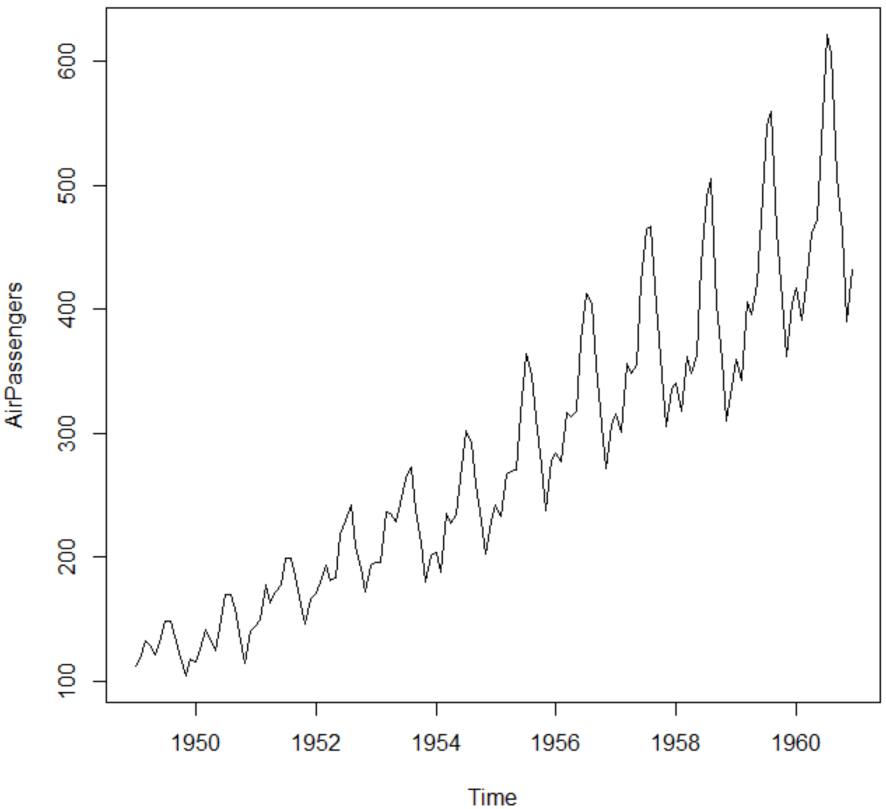
Simplicity in forecasting: Sometimes it’s better to be simple than correct

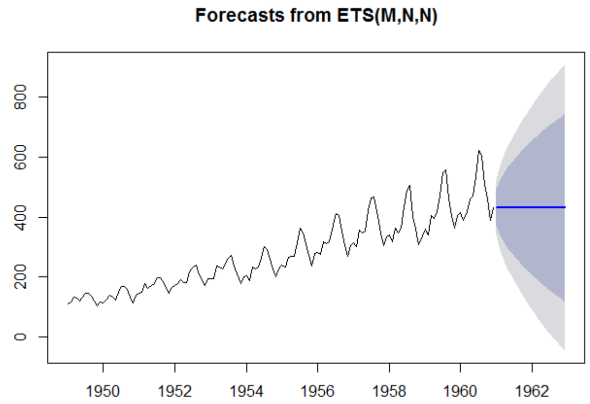
Stephan Kolassa, SAP

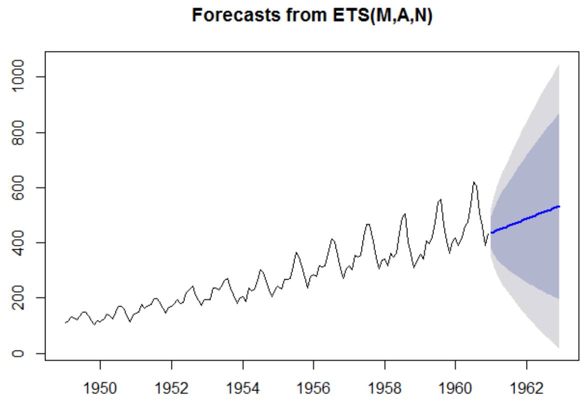
International Symposium on Forecasting, Practitioner Track – June 21, 2016

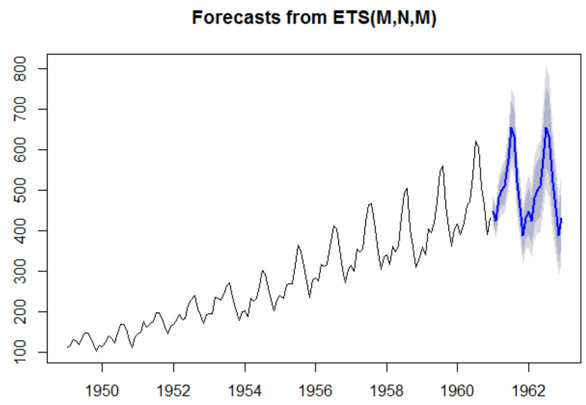
We all know we should include important drivers in our forecasts

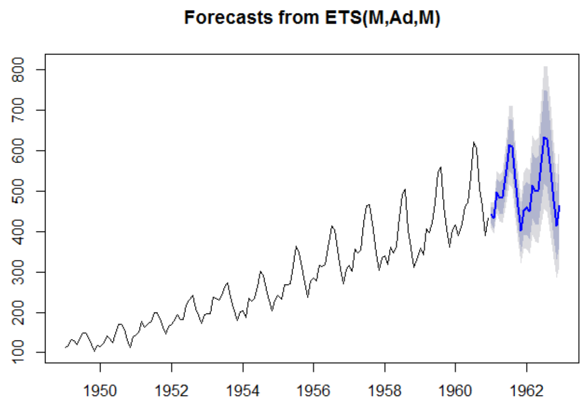
The classic airline passengers dataset & forecasts











We all know we should include important drivers in our forecasts

Today's question:

What about *unimportant* drivers?

(Or: less obviously important ones?)

Overview

- Misspecified models can yield forecasts that are badly wrong (nothing new here)
- Even correct models can yield bad forecasts (also not new)
- However, a correct model can yield systematically worse forecasts than a simpler incorrect model!
- We illustrate & investigate the problem using a simple simulation (could be run in Microsoft Excel)
- Takeaways:
  – Don’t insist on modeling weak signals
  – Use hierarchical models or regularization
- This is adapted from and expands on an article in *Foresight* (Winter 2016, #40: 20-26)

Simulated example

Data

- 10,000 “monthly” time series
- Each with 12 historical and 12 holdout observations
- Simulation:

$$y_t = b_0 + s_t + \epsilon$$

- Intercept:

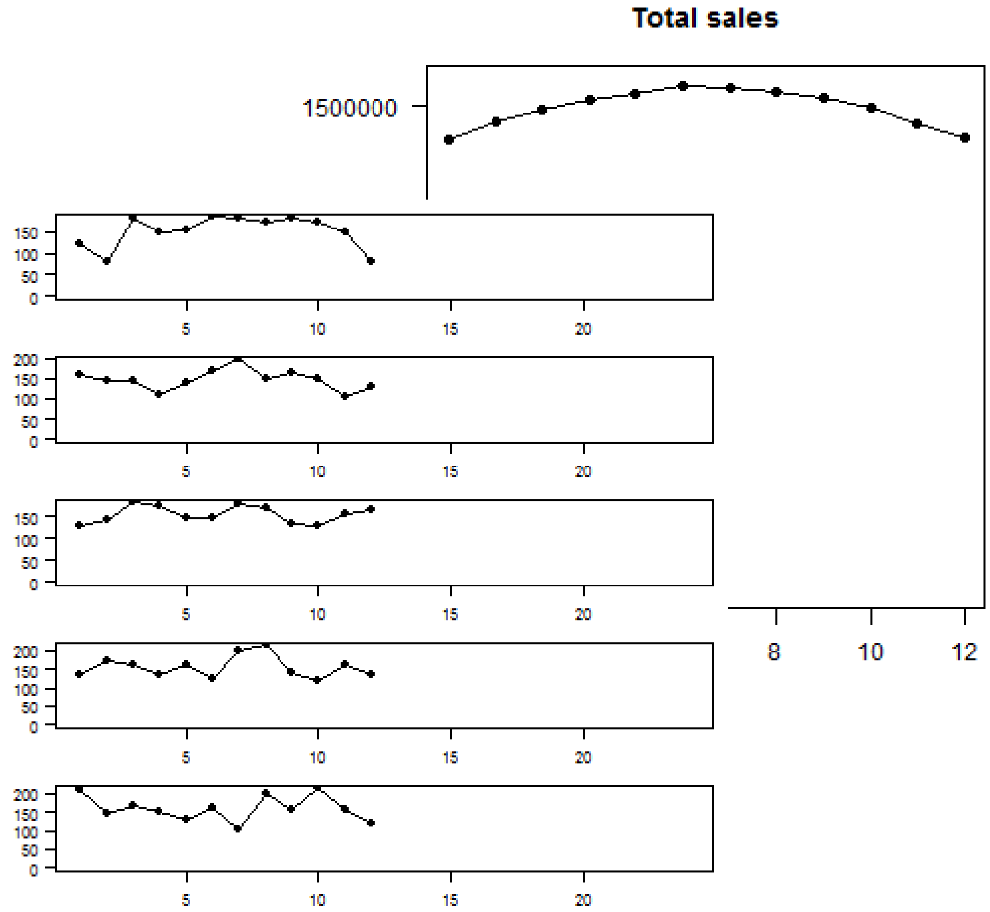
$$b_0 = 140$$

- Seasonal effects:

$$s_{\text{Jan}} = 0, \quad s_{\text{Feb}} = 5,$$
$$s_{\text{Mar}} = 9, \quad s_{\text{Apr}} = 12,$$
$$s_{\text{May}} = 14, \quad s_{\text{Jun}} = 16,$$
$$s_{\text{Jul}} = 16, \quad s_{\text{Aug}} = 14,$$
$$s_{\text{Sep}} = 12, \quad s_{\text{Oct}} = 9,$$
$$s_{\text{Nov}} = 5, \quad s_{\text{Dec}} = 0$$

- Noise term:

$$\epsilon \sim N(0{,}30^2)$$



Simulated example

The correct model

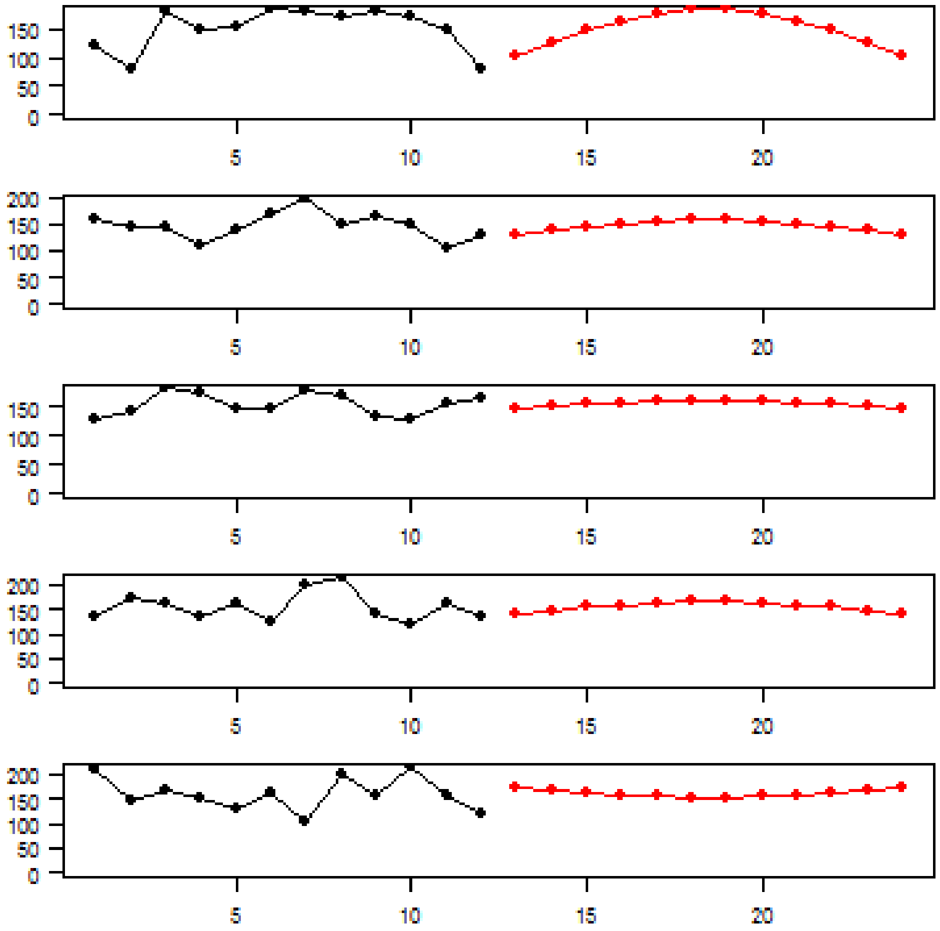
- Assume that the seasonal shape s_t is known, but not the intercept or the seasonal contribution

- Model:

$$y_t = \beta_0 + \beta_1 s_t + \epsilon_t$$

- This is the *correct* model!

- Run a separate linear regression for each time series to estimate β_0 and β_1 – and forecast



Simulated example

Models (C) and (S)

- Assume that the seasonal shape s_t is known, but not the intercept or the seasonal contribution

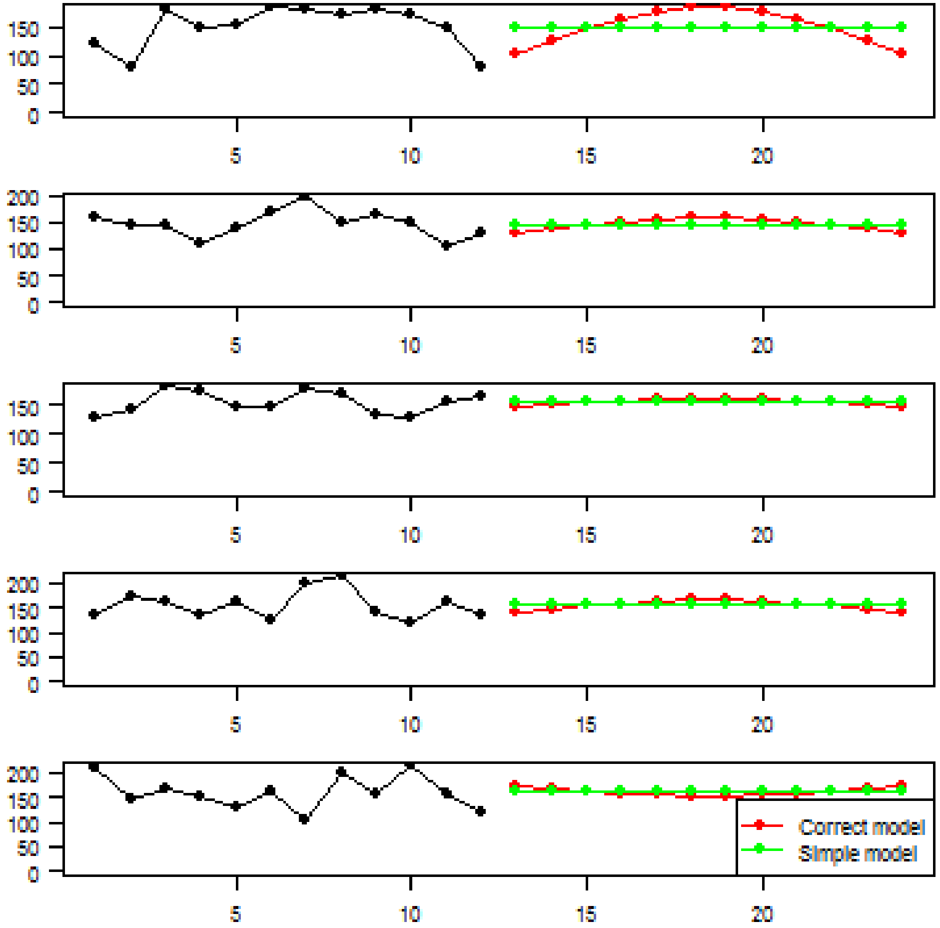
- Models:

$$y_t = \beta_0 + \beta_1 s_t + \epsilon_t \qquad \text{(C)}$$

$$y_t = \beta_0 + \epsilon_t \qquad \text{(S)}$$

- (C) for “correct” (or “complex”) and (S) for “simple”

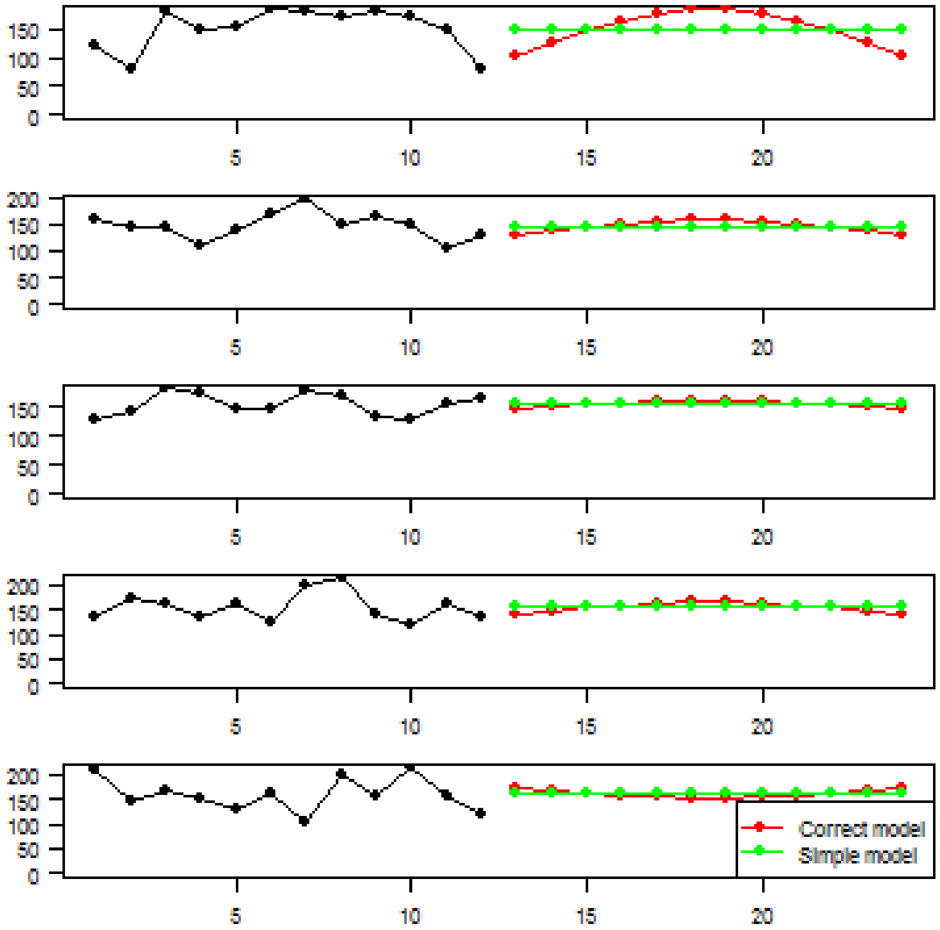
- Again, run linear regressions for each series



Simulated example

Models (C) and (S) – and their forecasts

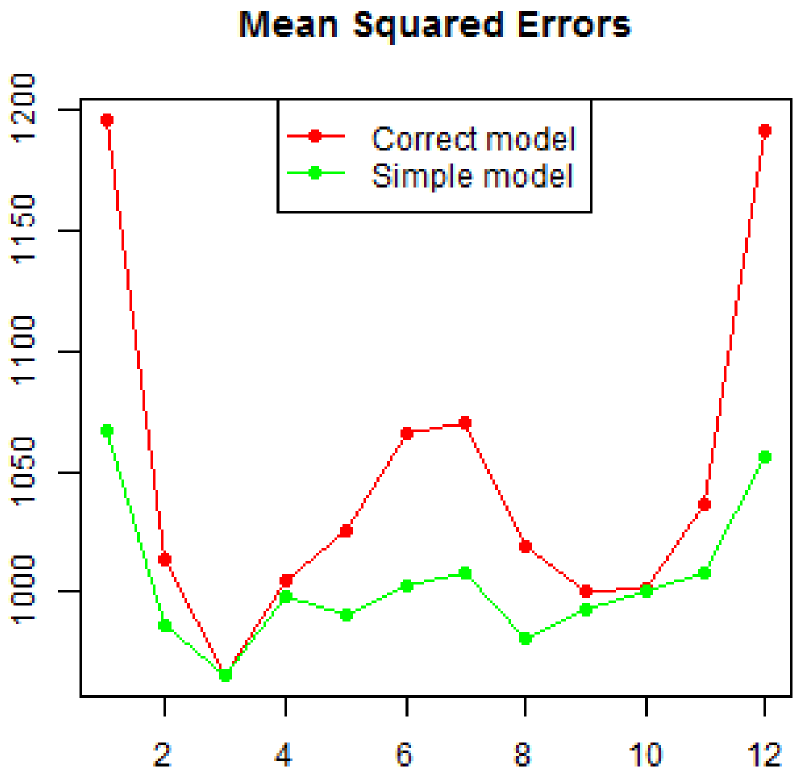
- How do forecasts perform?
- (Future data are simulated using the *same* data-generating process!)



Simulated example

Mean Squared Error in the holdout period

- The correct model (C) *never* has lower errors than the simple model (S)!
- In particular, model (C) is worse in January, June, July, December
- What's going on?



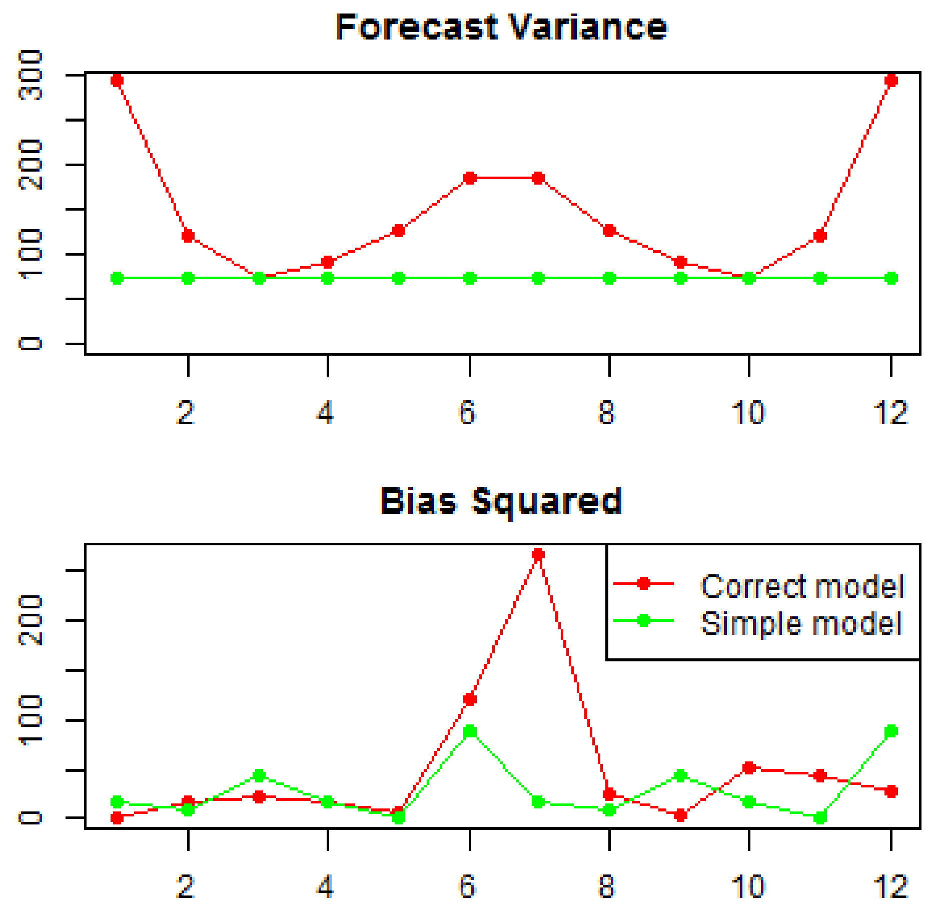
Simulated example

Expected Squared Error (ESE) decomposition

- Under certain technical conditions (which are met here),

$$ESE = \mathrm{Bias}^2 + \mathrm{Var(Fcst)} + \mathrm{Var(Actuals)}$$

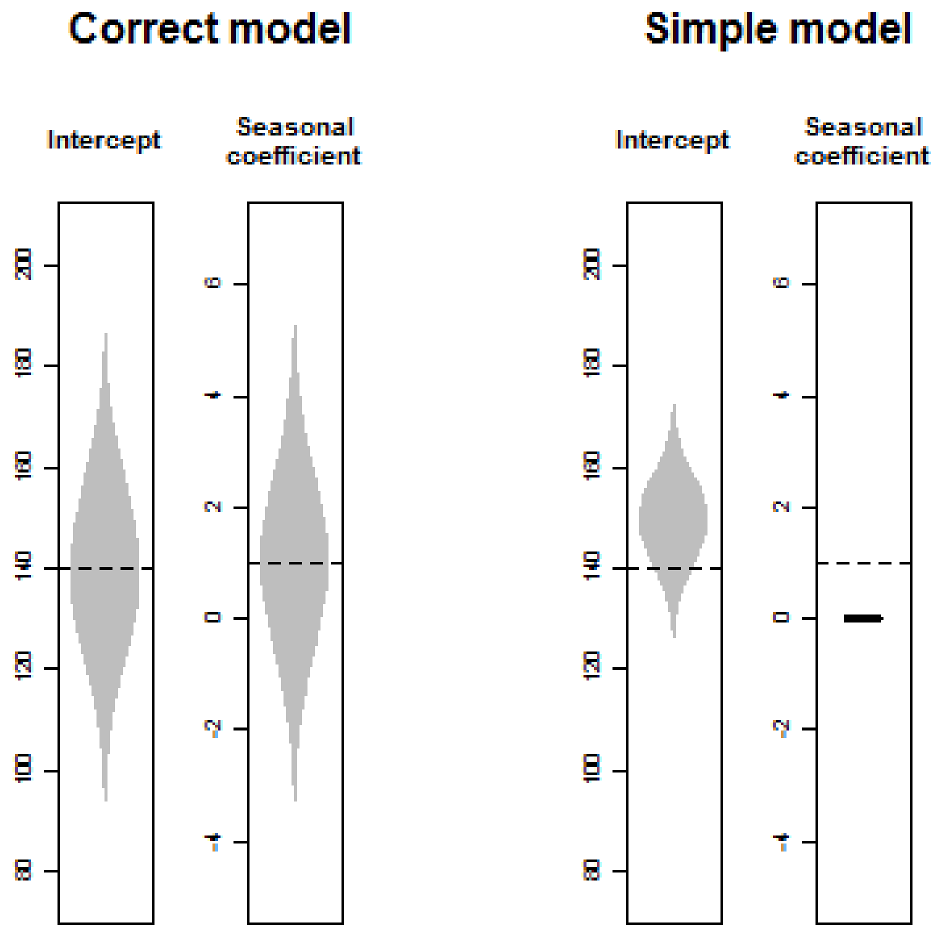
- Let's disregard the variance of the actuals
- The variance of the forecast shows the pattern we have seen in the MSE – this obviously drives the error



Simulated example

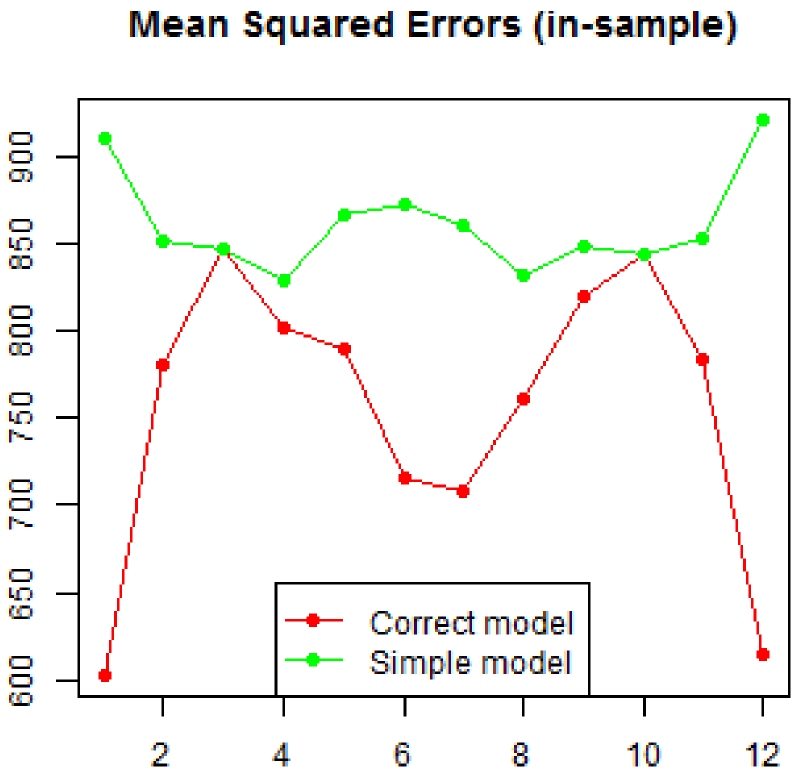
Where does the variance in the forecasts come from?

- Let's look at the distributions of parameter estimates across our 10,000 series
- Parameter estimates in the correct model are correct on average: they are *unbiased* (this is not surprising)
- Parameter estimates in the simple model are badly biased (this is not surprising, either)
- Key observation: estimates for both parameters in the correct model are more variable!

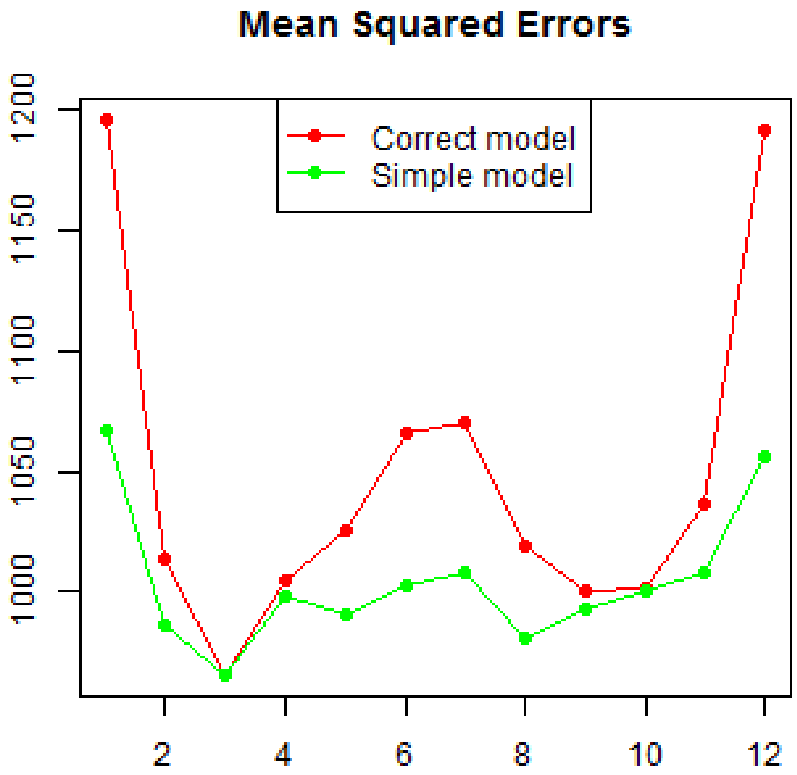


Simulated example

Mean Squared Error in-sample (left) and out-of-sample (right)



- The correct model is better in-sample, but worse out-of-sample
- In-sample fit is not a reliable guide to out-of-sample forecasting accuracy!



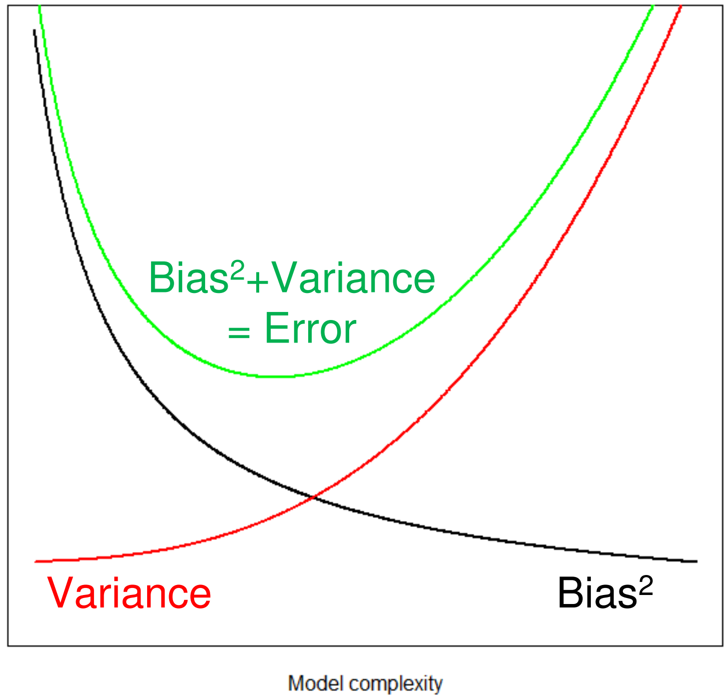
Let's stop and look back at what we have learned

- If signals are present but weak, fitting them may yield worse forecasts than not fitting them. Examples:
  - Day-of-week patterns for slow movers
  - Lifecycle patterns or seasonality for slow movers
  - Cannibalization on SKU × store level
  - Weather impact on SKU × store level

- Why? Complex models exhibit less bias but more variance!

- Good in-sample fit does not imply good forecasting accuracy!

The Bias-Variance Trade-off (Conceptual)



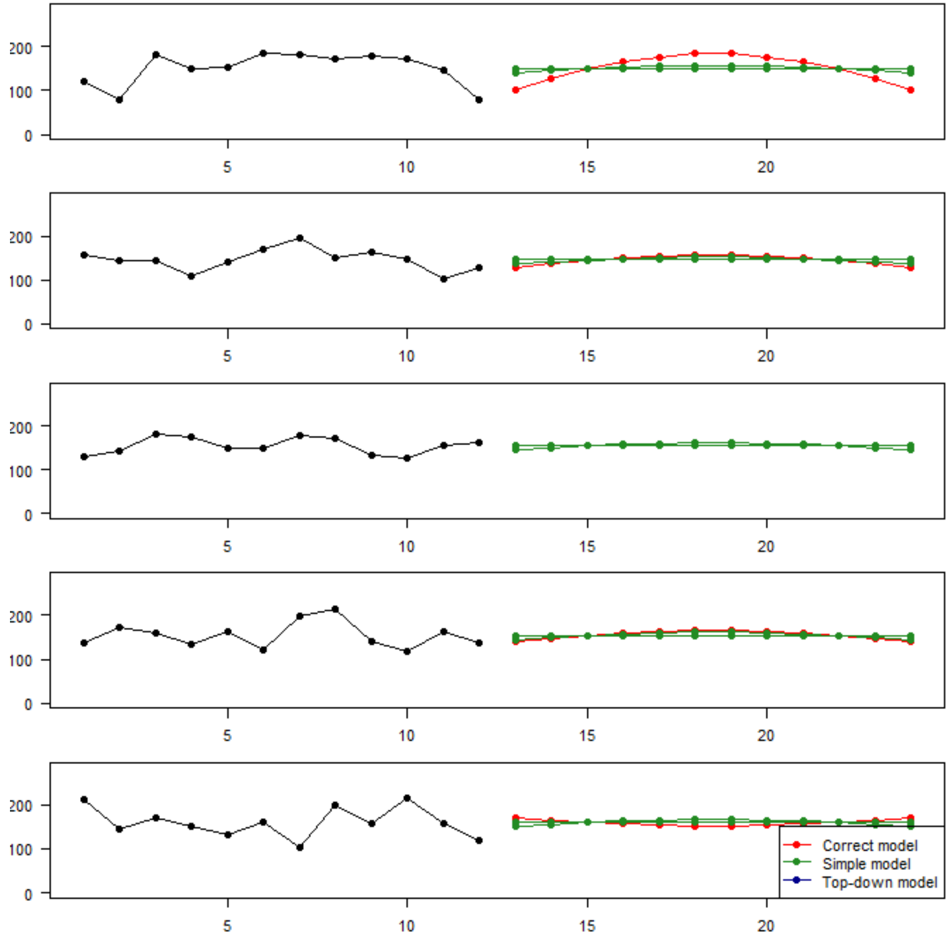
What to do?

- We *know* that there is seasonality in our data – we simulated it, and it's visible on aggregate level
- How to include seasonality in our models without variance killing us?
- *Hierarchical* models can "transfer" information between different levels
- *Regularization* increases bias & reduces variance and may yield more accurate forecasts
  - Bayesian statistics
  - Lasso
  - Ridge regression
  - Elastic net

Hierarchical forecasts

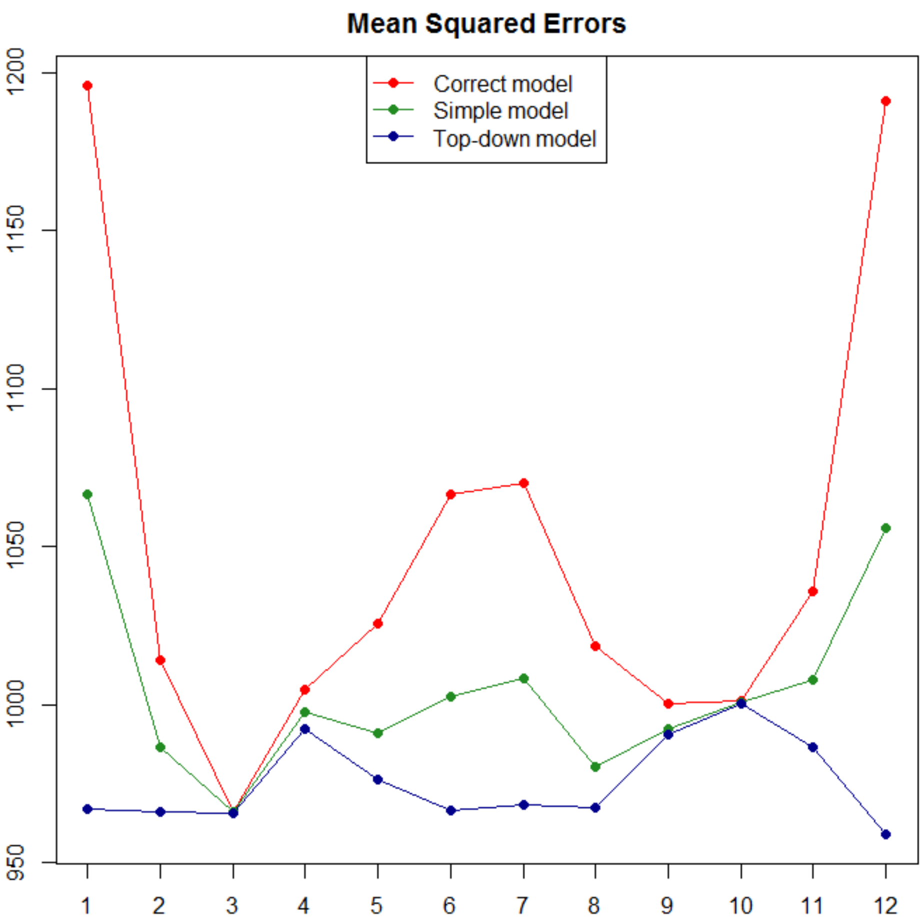
Top-down forecasting

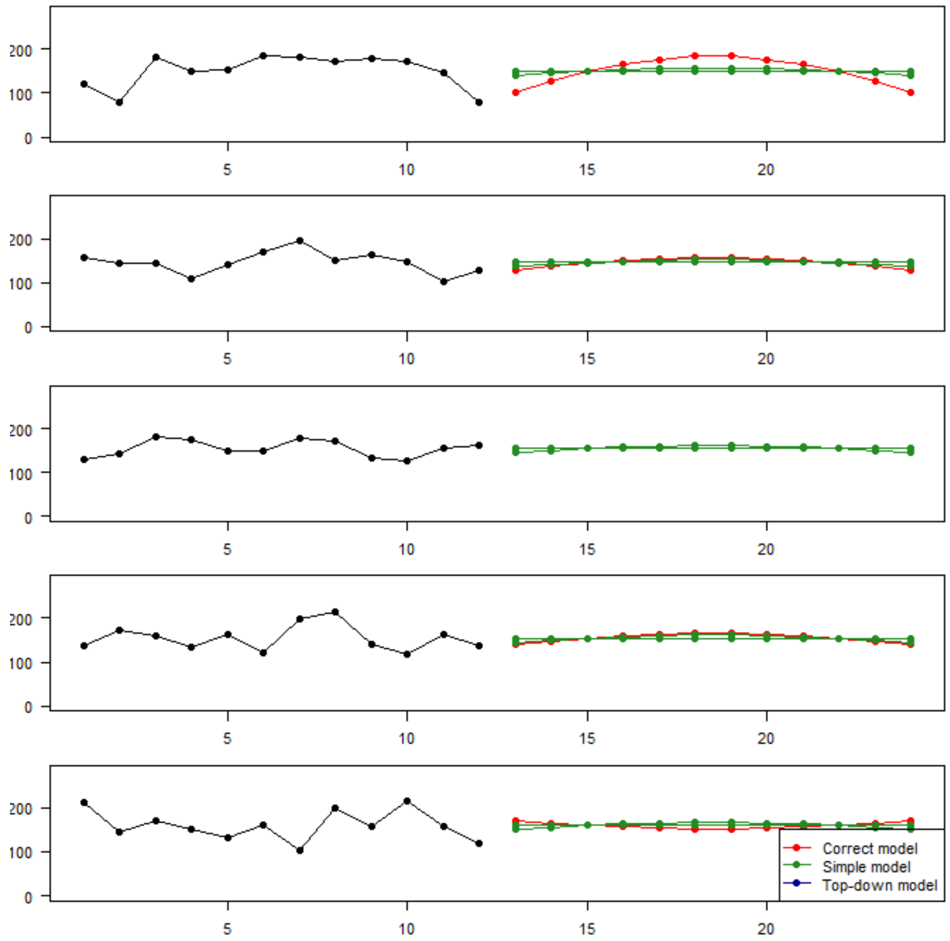
- Top-down forecasting:
  - Forecast on the top level
  - Distribute this total forecast down to individual series
    - By historical proportions (see right)
    - By forecasted proportions



Hierarchical forecasts

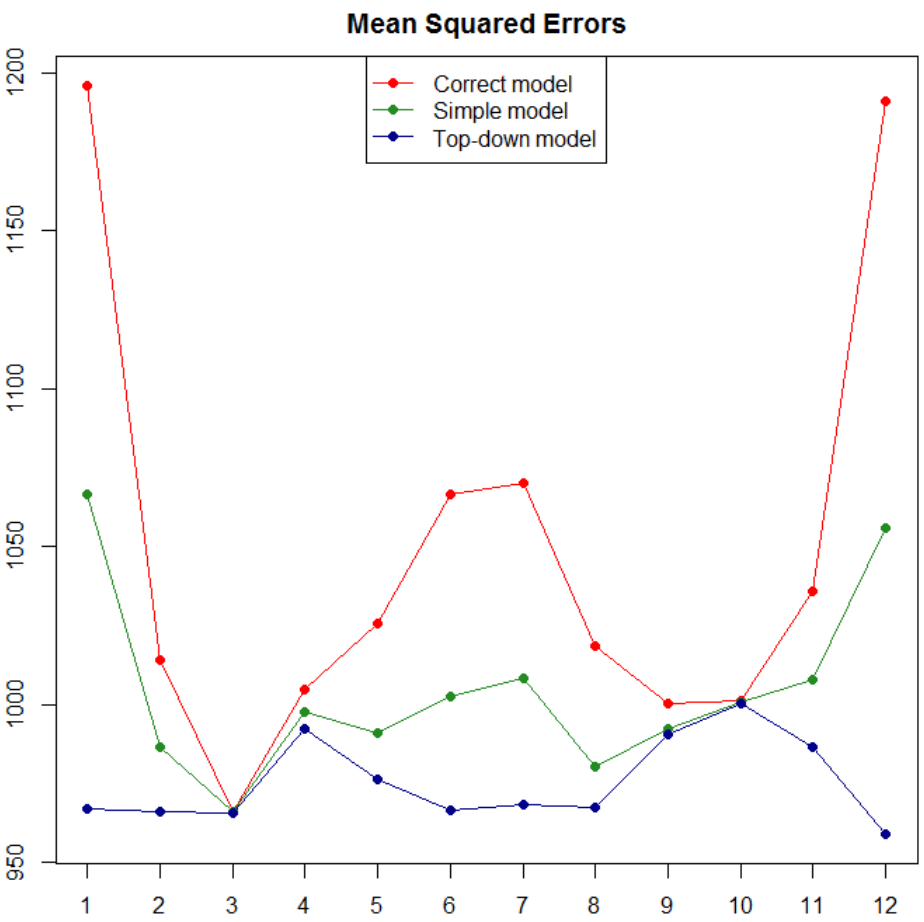
Top-down forecasting





Hierarchical forecasts

Top-down forecasting



- Advantages:
  - Models aggregate dynamics very well
  - Simple to understand and explain
- Disadvantages:
  - If the group does not *share* dynamics, top-down will be off
  - Can't account for known dynamics on granular level

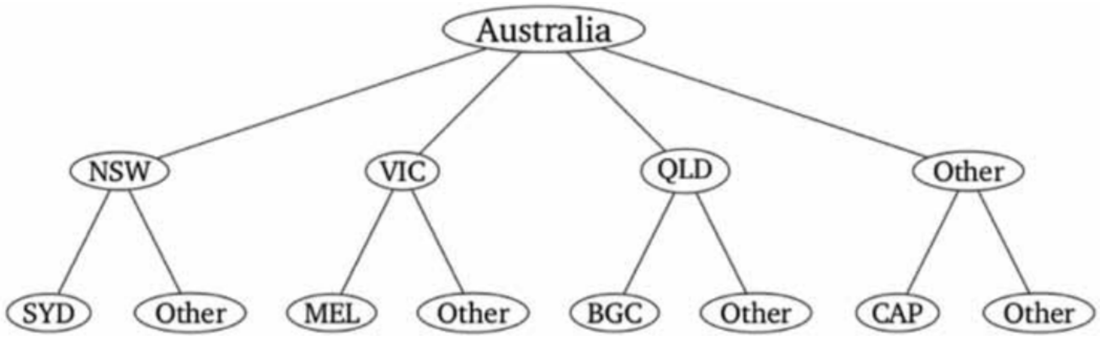
Hierarchical forecasts

Optimal combination forecasting

- Optimal combination forecasting:
  – Forecast on *all* levels separately
  – The result will not be sum-consistent…
  – … so we modify *all* forecasts "slightly"
  – The final forecasts are consistent *and* often better on all levels!

- Why?
  – All modifications are based on information from other hierarchy levels
  – → information is propagated through the hierarchy!

- See Hyndman & Athanasopoulos (*Foresight*, 2014, 35:42-48)

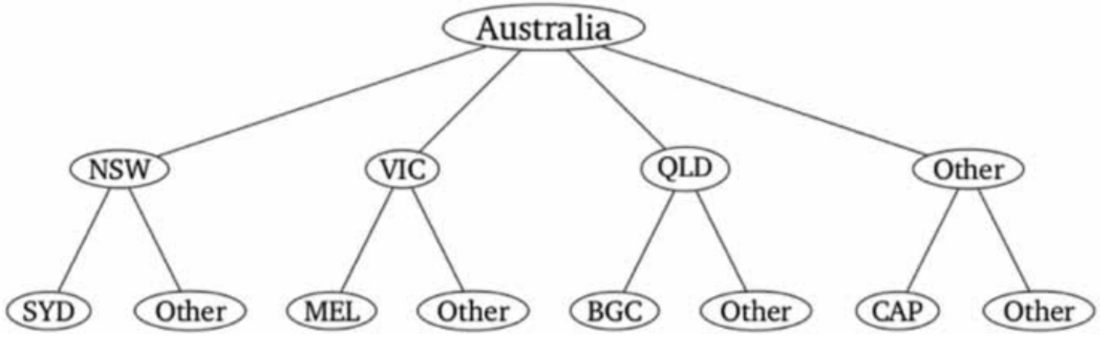


Hierarchical forecasts

Optimal combination forecasting

- Optimal combination forecasting:
  - Forecast on *all* levels separately
  - The result will not be sum-consistent…
  - … so we modify *all* forecasts "slightly"
  - The final forecasts are consistent *and* often better on all levels!
- Why?
  - All modifications are based on information from other hierarchy levels
  - → information is propagated through the hierarchy!
- See Hyndman & Athanasopoulos (*Foresight*, 2014, 35:42-48)

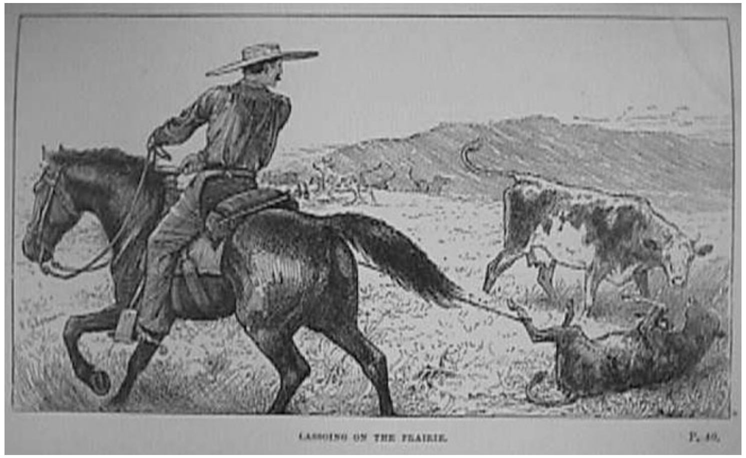
- Advantages:
  - Models dynamics on *all* levels
  - Typically improves forecasts on *all* levels
- Disadvantages:
  - Hard to explain
  - Computationally difficult (has difficulties on more realistic hierarchies, e.g., crossing location × product hierarchies)



Regularization

- Recall that the original problem was that parameter estimates in model C had low bias but high variance

- *Regularization* increases bias & reduces variance
  – Bayesian statistics
  – Lasso
  – Ridge regression
  – Elastic net



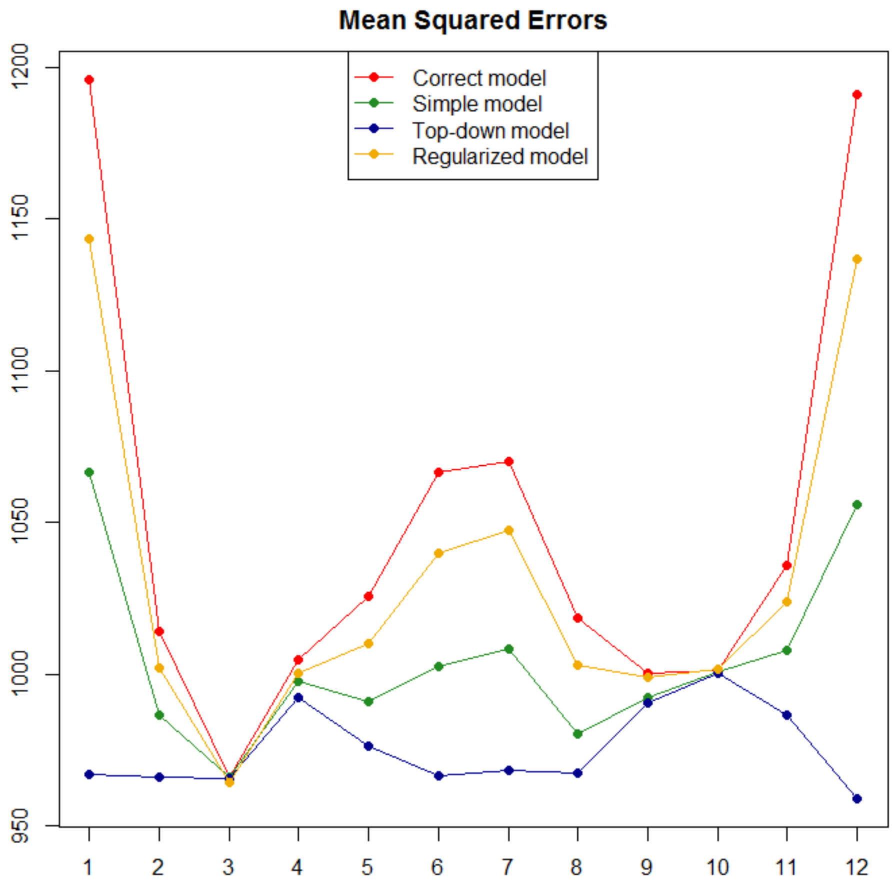
- Idea in each case:
  – Constrain the model coefficients (in different ways)
  – If something is constrained by a lasso or an elastic net, it can't vary as much!

- This will reduce in-sample fit, but often improve forecasting accuracy

Results

- Advantages of regularization:
  - Does not require maintaining a hierarchy
  - Is quite a general concept

- Disadvantages of regularization:
  - Not very common in time series analysis , more in regression modeling and machine learning
  - Hard to explain, though less so than optimal combination forecasting
  - Computationally intensive (cross-validation), though less so than optimal combination forecasting

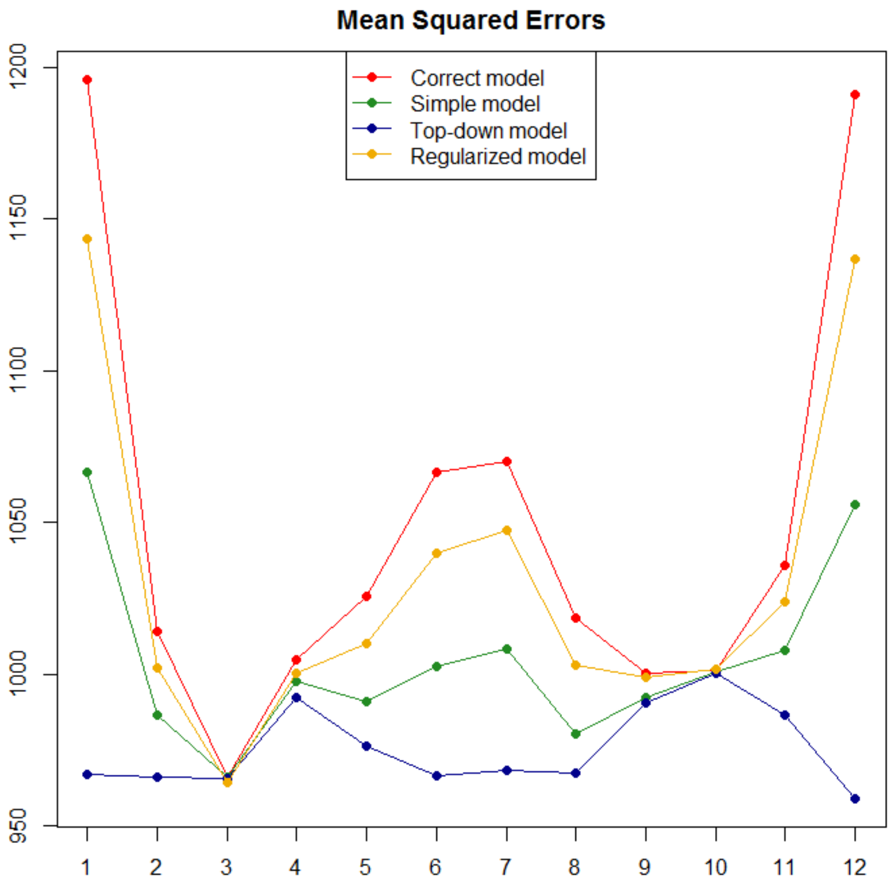


Takeaways

- There are multiple sources of forecasting errors
  - The bias of the model
  - The variance of the model
  - The residual variance

- Making our model more complex will reduce bias (and usually residual variance), but increase model variance
  - Keep this trade-off in mind!
  - Don’t automatically expect better forecasts from more complex models…
  - …. especially if we model weak signals!

- Ways forward:
  - Simple models
  - Hierarchical forecasting
  - Regularization



Thank you!

Contact information:

Stephan Kolassa
Stephan.Kolassa@sap.com

SAP (Switzerland) AG
High-Tech-Center 2
Bahnstrasse 1
8274 Tägerwilen
Switzerland