FORECASTING PRODUCT DEMAND IN R

Loading data into xts object

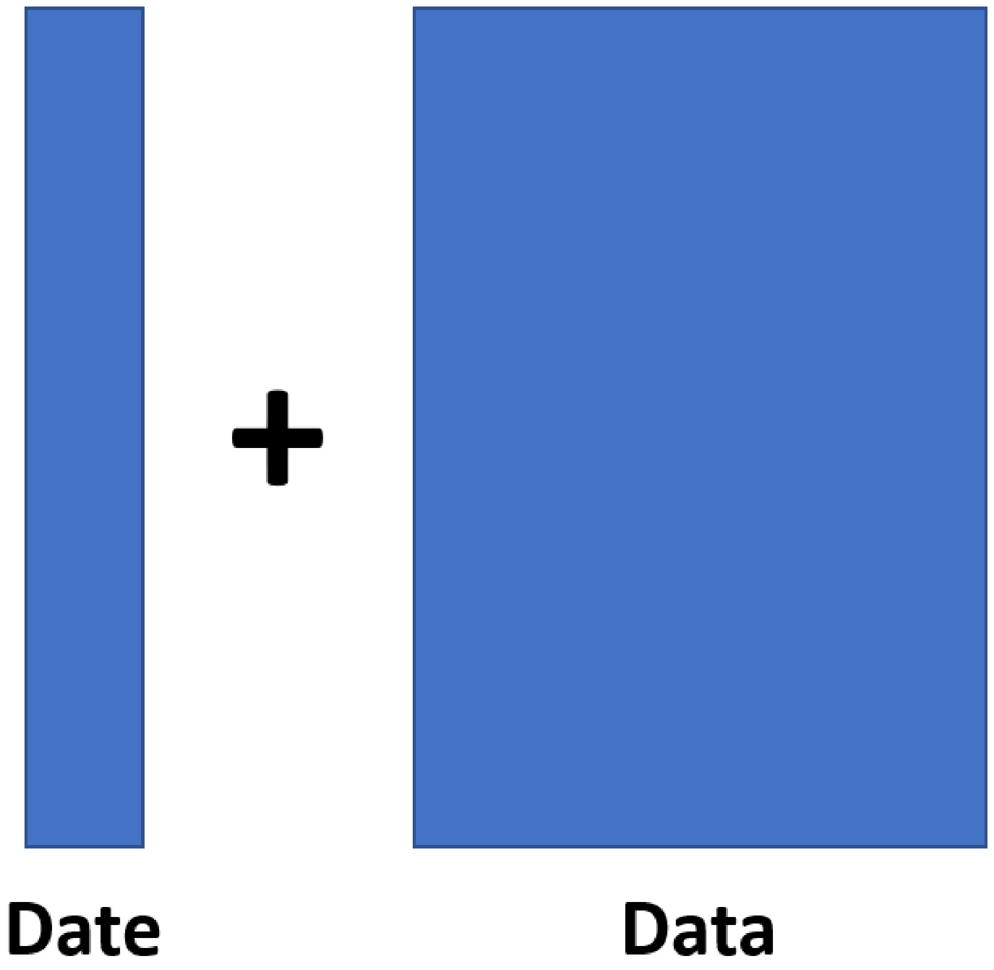
Aric LaBarr, Ph.D.
Senior Data Scientist, Elder Research

xts objects

- eXtensible Time Series object
- Builds upon zoo objects

Loading Data Into xts Object

- Attach a date index on to a data matrix
- Very easy to manipulate!



xts objects

- Manipulating Time Series Data in R with xts & zoo

- Manipulating Time Series Data in R: Case Studies

Loading Data Example

```
dates <- seq(as.Date("2014-01-19"), length = 176, by = "weeks")

bev_xts <- xts(bev, order.by = dates)

head(bev_xts[,"M.hi"])

           M.hi
2014-01-19  458
2014-01-26  477
2014-02-02  539
2014-02-09  687
2014-02-16  389
2014-02-23  399
```

FORECASTING PRODUCT DEMAND IN R

Let's practice!

FORECASTING PRODUCT DEMAND IN R

ARIMA Time Series 101

Aric LaBarr, Ph.D.
Senior Data Scientist, Elder Research

Other courses on time series

- Time Series with R skill track

What is an ARIMA Model?

- **A**uto**R**egressive Models
- **I**ntegrated
- **M**oving **A**verage

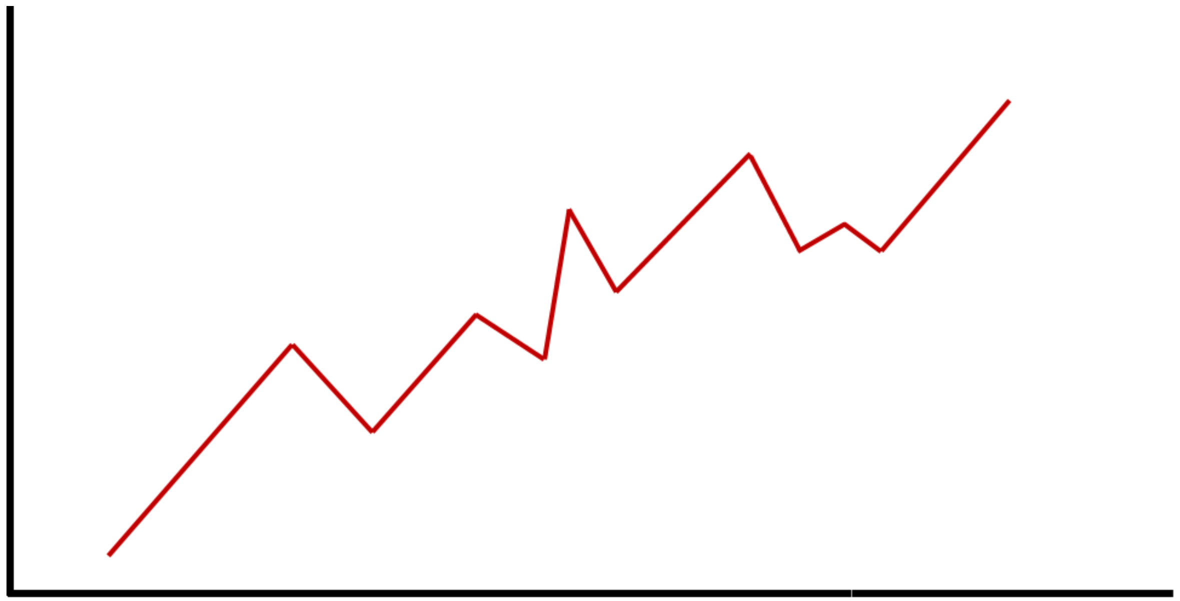
Integrated - Stationarity

- Does your data have a dependence across time?
- How long does this dependence last?
- **Stationarity**
  - Effect of an observation dissipates as time goes on
  - Best long term prediction is the mean of the series
  - Commonly achieved through differencing

Differencing

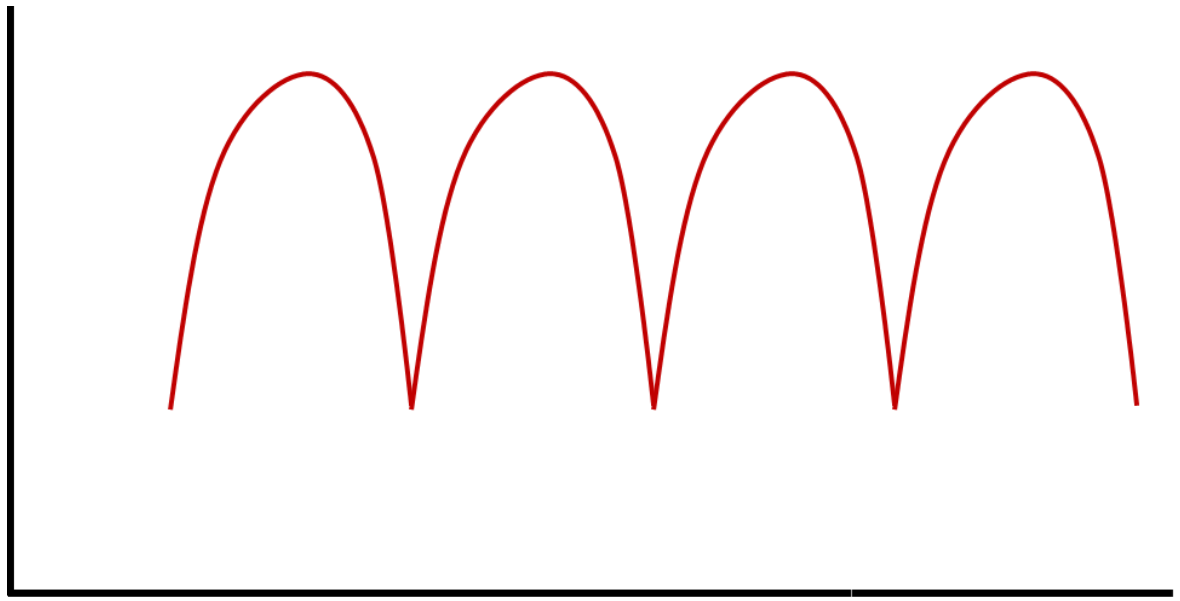
Typically used to remove trend...

$$Y_t - Y_{t-1}$$



... or seasonality

$$Y_t - Y_{t-12}$$



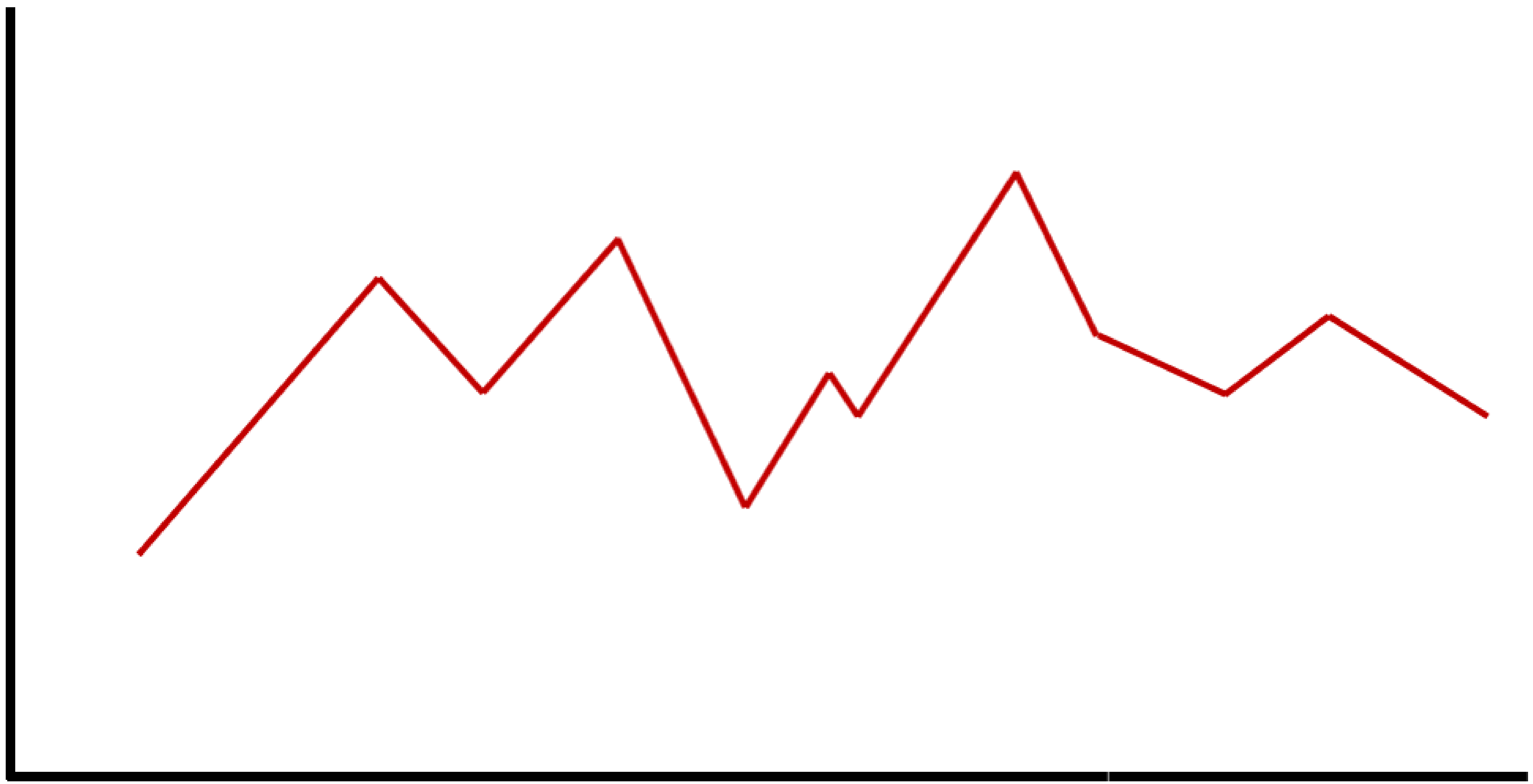
Autoregressive (AR) Piece

- **A**uto**R**egressive Models
  - Depend only on previous values - called **lags**.
  - $Y_t = \omega_0 + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + ... + \varepsilon_t$
  - Long-memory models - effect slowly dissipates

Moving Average (MA) Piece

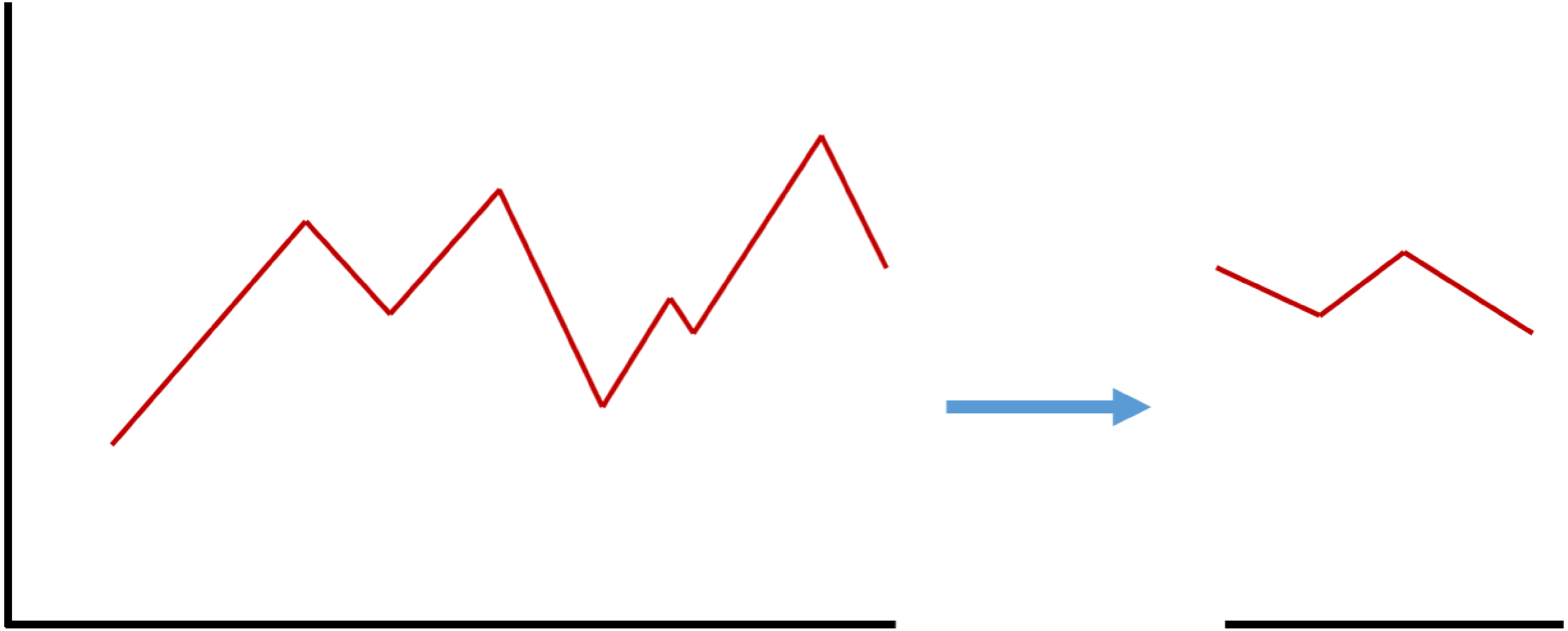
- **M**oving **A**verage Models
  - Depend only on previous "shocks" or errors
  - $Y_t = \omega_0 + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \ldots$
  - Short-memory models - effects quickly disappear completely

Training vs. Validation



```
M_t <- bev_xts[,"M.hi"] + bev_xts[,"M.lo"]
```

Training vs. Validation



```
M_t_train <- M_t[index(M_t) < "2017-01-01"]
M_t_valid <- M_t[index(M_t) >= "2017-01-01"]
```

How to Build ARIMA Models?

```
auto.arima(M_t_train)

Series: M_t_train
ARIMA(4,0,1) with non-zero mean

Coefficients:
         ar1      ar2     ar3     ar4      ma1       mean
      1.3158  -0.5841  0.1546  0.0290  -0.6285  2037.5977
s.e.  0.3199   0.2562  0.1534  0.1165   0.3089    87.5028

sigma^2 estimated as 67471:  log likelihood=-1072.02
AIC=2158.05   AICc=2158.81   BIC=2179.31
```

FORECASTING PRODUCT DEMAND IN R

Let's practice!

FORECASTING PRODUCT DEMAND IN R

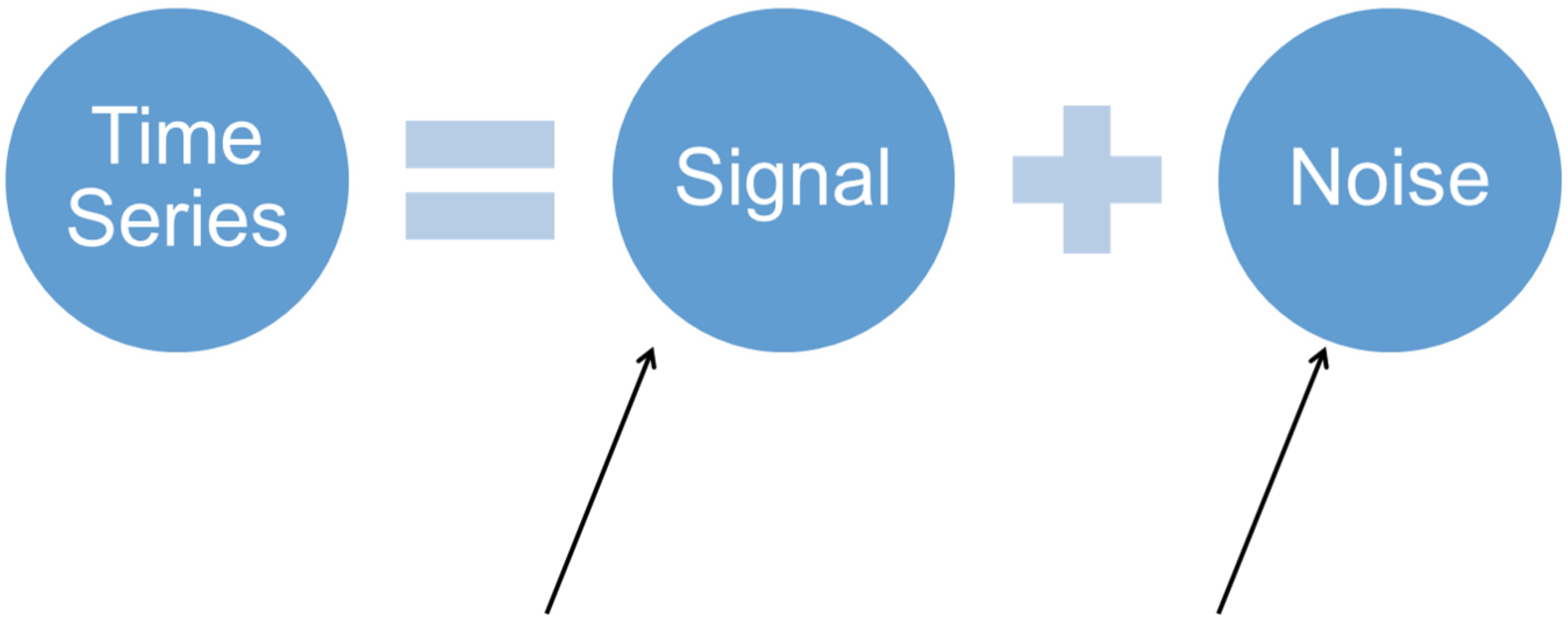
Forecasting with time series

Aric LaBarr, Ph.D.
Senior Data Scientist, Elder Research

Forecasting

- Goal of most time series models!
- Models use past values or "shocks" to predict the future
- Pattern recognition followed by pattern repetition

Forecasting



Forecasts extrapolate signal portion of model.

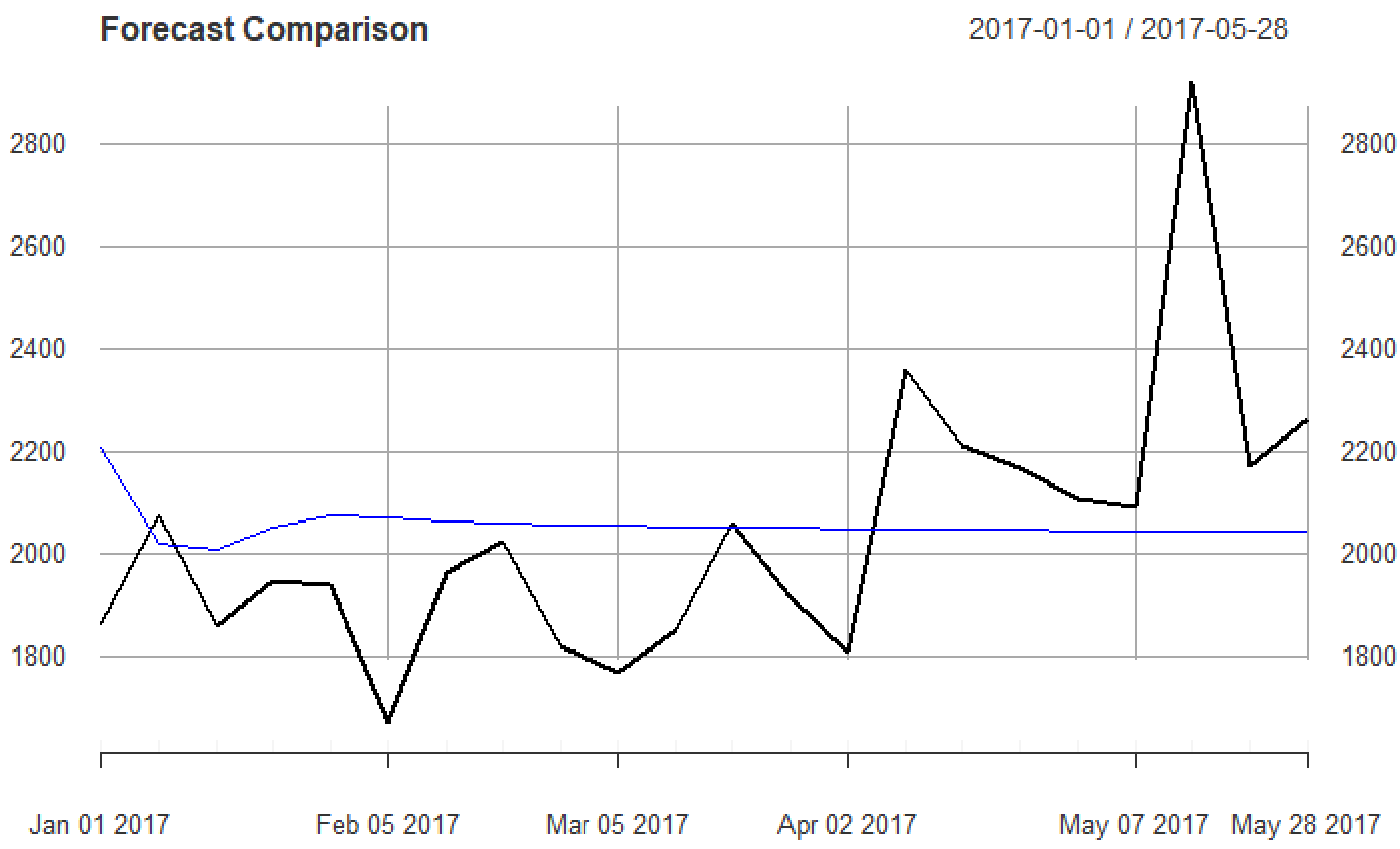
Confidence intervals account for uncertainty.

Forecasting Example

```
forecast_M_t <- forecast(M_t_model, h = 22)

for_dates <- seq(as.Date("2017-01-01"), length = 22, by = "weeks")
for_M_t_xts <- xts(forecast_M_t$mean, order.by = for_dates)

plot(M_t_valid, main = 'Forecast Comparison')
lines(for_M_t_xts, col = "blue")
```



Forecast Comparison
2017-01-01 / 2017-05-28
2800
2600
2400
2200
2000
1800
Jan 01 2017
Feb 05 2017
Mar 05 2017
Apr 02 2017
May 07 2017
May 28 2017

How to Evaluate Forecasts?

- 2 Common Measures of Accuracy:

1. **M**ean **A**bsolute **E**rror (MAE)

$$\frac{1}{n}\sum_{i=1}^{n}|Y_t - \hat{Y}_t|$$

2. **M**ean **A**bsolute **P**ercentage **E**rror (MAPE)

$$\frac{1}{n}\sum_{i=1}^{n}|\frac{Y_t - \hat{Y}_t}{Y_t}| \times 100$$

MAE and MAPE Example

```
for_M_t <- as.numeric(forecast_M_t$mean)
v_M_t <- as.numeric(M_t_valid)

MAE <- mean(abs(for_M_t - v_M_t))
MAPE <- 100*mean(abs((for_M_t - v_M_t)/v_M_t))

> print(MAE)
[1] 198.7976

> print(MAPE)
[1] 9.576247
```

FORECASTING PRODUCT DEMAND IN R

Let's practice!