Forecasting

HARVARD UNIVERSITY

Agenda

Start	End	Item
		What is Forecasting?
		Naïve Forecasting Methods
		Time Series Decomposition
		HoltWinters Forecasting
		Appendix

HARVARD UNIVERSITY

Appendix I – Regression Forecasting

HARVARD UNIVERSITY

Regression Based Forecasting

Ridership Y is a function of time (t) and noise (error = e)

$Y_i = B_0 + B_1 * t + e$

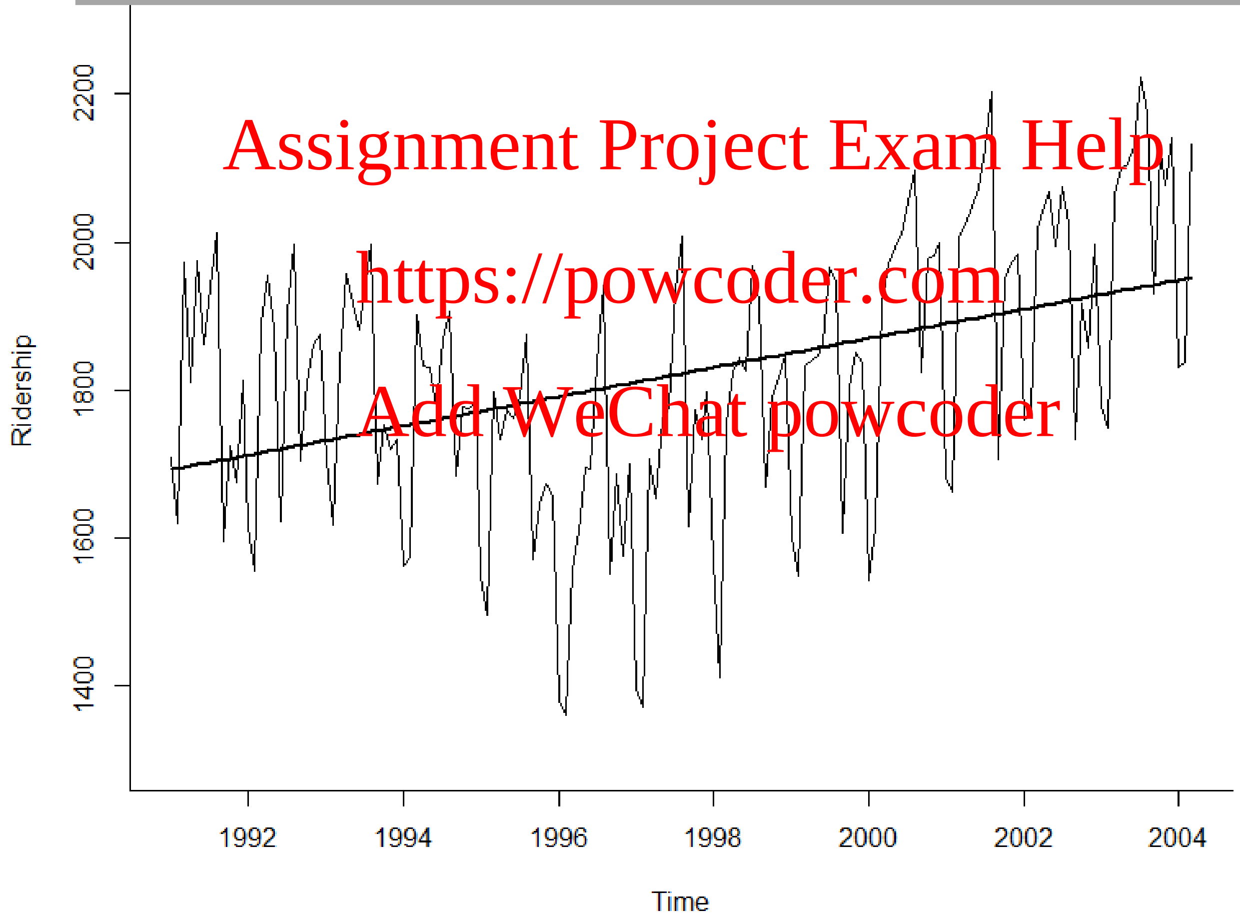
Thus we model 3 of the 4 components:

- Level (B_0)
- Trend* (B_1)
- Noise (e)

HARVARD UNIVERSITY

Time Series Linear Trend

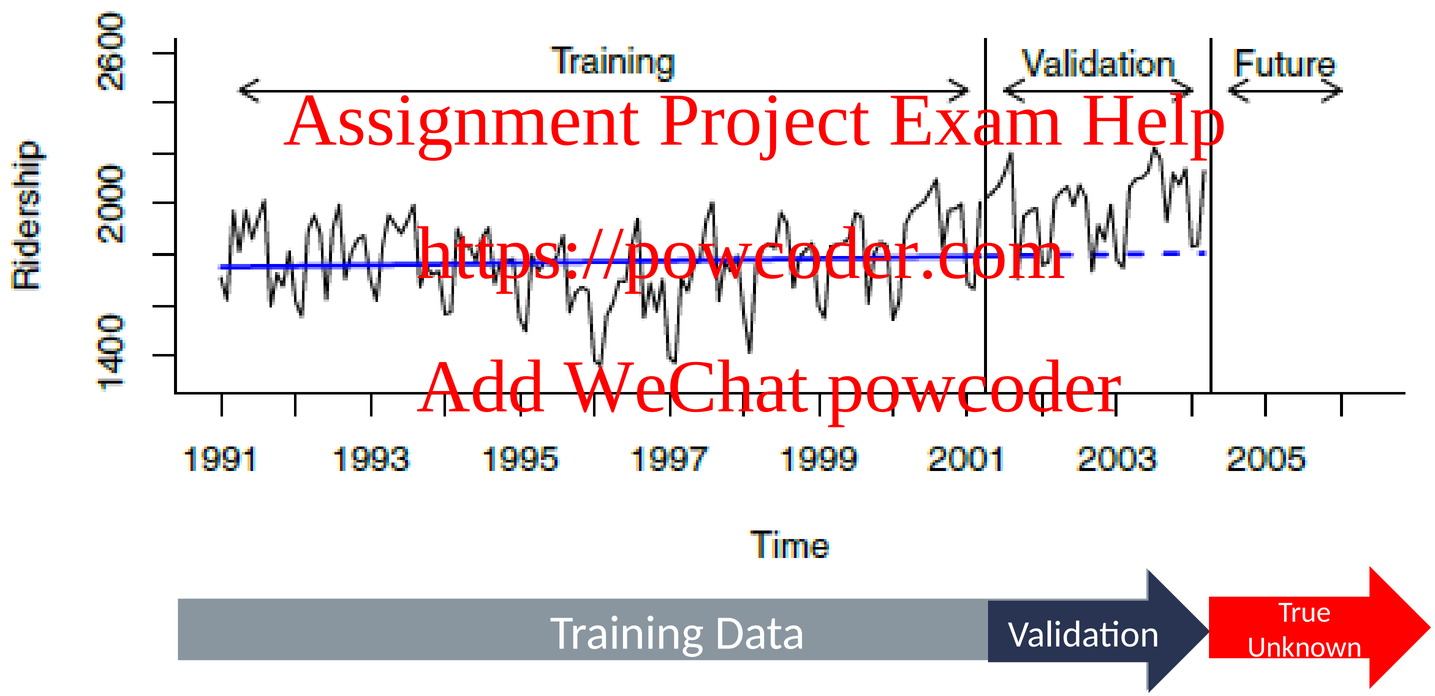
```
# produce linear trend model
ridership.lm <- tslm(ridership.ts ~
trend)
```



HARVARD UNIVERSITY

Linear Trend Predictions

```
train.lm.pred <- forecast(train.lm, h = nValid, level = 0)
```



Open 5_Ch17.R

Exponential Trend – like amazon's revenue

Appropriate model when increase/decrease in series over time is multiplicative

Replace Y with log(Y) then fit linear regression

$log(Y_i) = B_0 + B_1 t + e$

HARVARD UNIVERSITY

Natural Logs – not to hard

“e” raised what power equals the time series value

- Where “e” = ~2.718


```
log(2)
[1] 0.6931472
```


$e^{0.693}$
$2.719^{(0.6931472)}$
1.99


```
log(13)
[1] 2.564949
```

$e^{2.56949}$
$2.719^{(2.56949)}$
13.05568

Don’t worry, R handles with the log() function.

HARVARD UNIVERSITY

Exponential trend - forecast errors

Note that performance measures in standard linear regression software are not in original units

Model forecasts will be in the form log(*Y*)

Return to original units by taking exponent of model forecasts using the function exp()

Calculate standard deviation of these forecast errors to get RMSE

HARVARD UNIVERSITY

Open 5_Ch17.R (AGAIN)

Other Trends Polynomial Trend

Add additional predictors as appropriate

For example, for quadratic relationship add a t^2 predictor

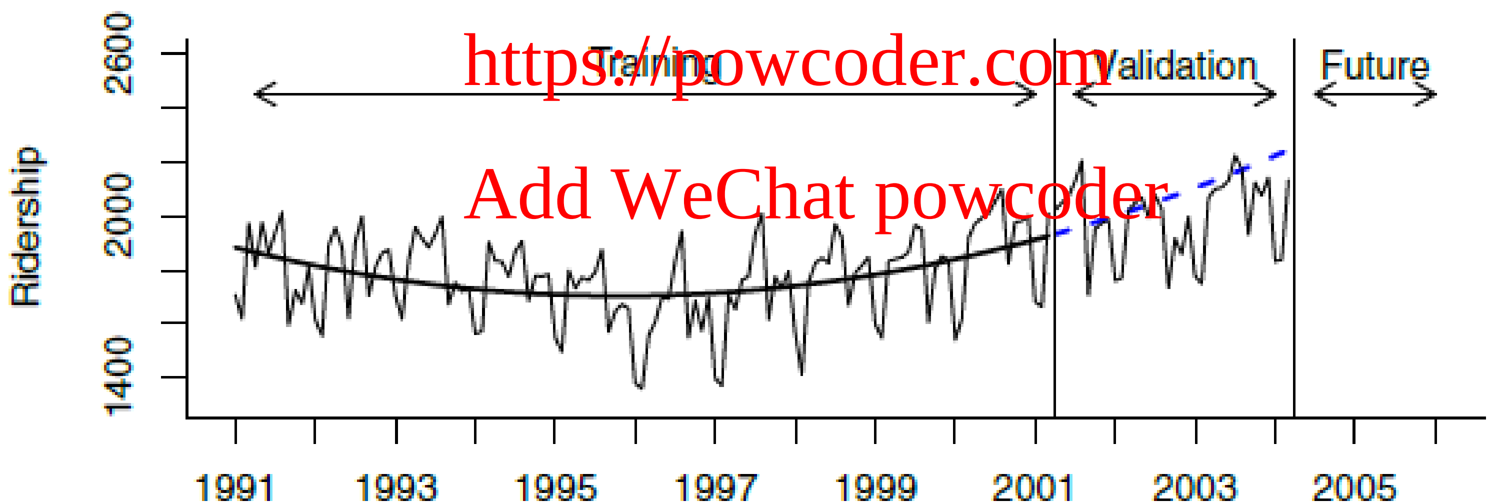
Fit linear regression using both t and t^2

HARVARD UNIVERSITY

Quadratic Trend

```
# fit quadratic trend using function I(), which treats an
# object "as is".

train.lm.poly.trend <- tslm(train.ts ~ trend + I(trend^2))
summary(train.lm.poly.trend)
train.lm.poly.trend.pred <- forecast(train.lm.poly.trend,
     h = nValid, level = 0)
```

Due to time constraints, please review pg 408.

Handling Seasonality in Regression

Just make dummy variables for seasons…but beware of multi-collinearity!

Month	Ridership	Season
Jan 1991	1709	Jan
Feb 1991	1621	Feb
Mar 1991	1973	Mar

Month	Ridership	Season	Jan	Feb	Mar	…	Nov
Jan 1991	1709	Jan	1	0	0	…	0
Feb 1991	1621	Feb	0	1	0	…	0
Mar 1991	1973	Mar	0	0	1	…	0
…	…	…	.	.	.	…	1
Nov 1991	1675	Nov	0	0	0	…	1
Dec 1991	1813	Dec	0	0	0	0	0

To avoid multi-collinearity, there is no Dec.

Final model, Amtrak data

Incorporates trend and seasonality

13 predictors

- 11 monthly dummies
- t = trends
- t^2 = quadratic trend (to get the positive and negative trend slopes)


```
train.lm.trend.season <- tslm(train.ts ~ trend +
                                   I(trend^2) + season)
```

Regression Based Forecasting is great for events

Month	Ridership	Season	Jan	Feb	Mar	…	Nov	Summer Promo	Holiday Promo
Jan 1991	1709	Jan	1	0	0	…	0	0	0
Feb 1991	1621	Feb	0	1	0	…	0	0	0
Mar 1991	1973	Mar	0	0	1	…	0	0	0
…	…	…	0	0	0	0	0	0	.0
Nov 1991	1675	Nov	0	0	0	…	1	0	1
Dec 1991	1813	Dec	0	0	0	0	0	0	1

HARVARD UNIVERSITY

Summary – Regression Based Forecasting

- Can use linear regression for exponential models (use logs) and polynomials (exponentiation)
- For seasonality, use categorical variable (make dummies)
- For Events, use more dummy variables

HARVARD UNIVERSITY

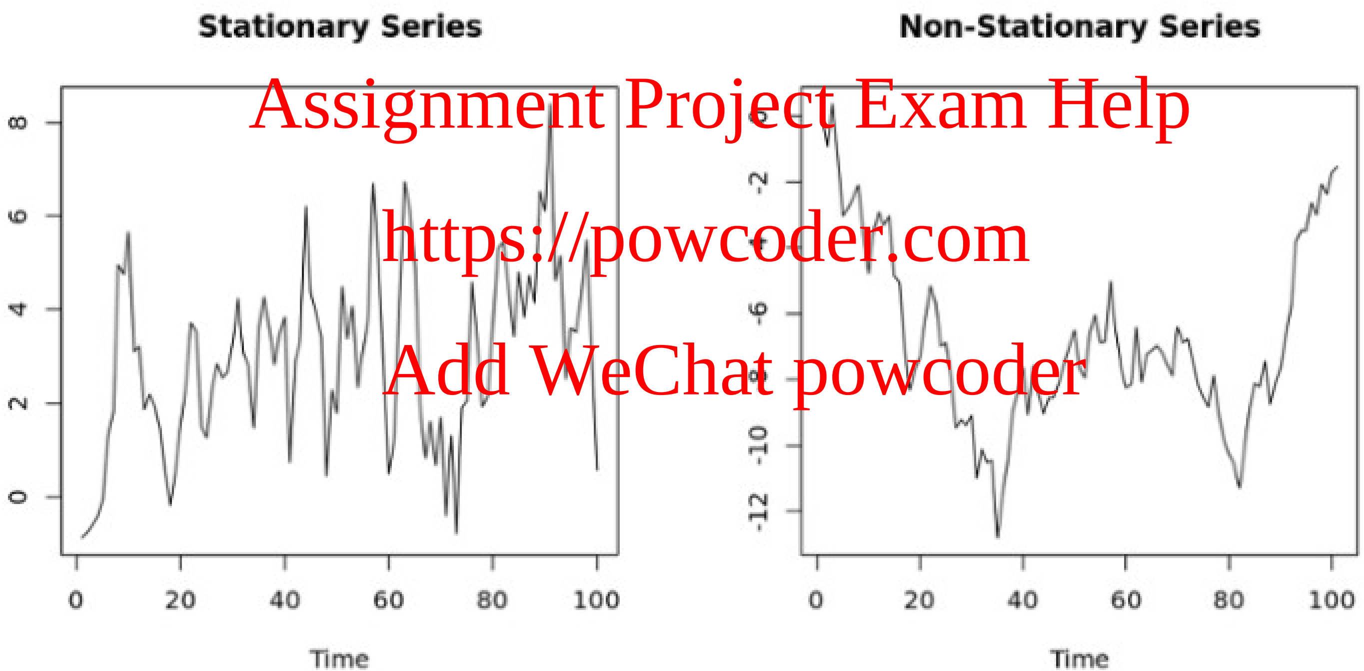
Open 6_TK_RegressionModel.R

Appendix II - ARIMA

HARVARD UNIVERSITY

ARIMA - Stationary

Fitting an ARIMA model requires the series to be **stationary**. A series is said to be stationary when its mean, variance, and autocovariance are time invariant.



More simply non-stationary means the average values change through time, levels change, etc.

HARVARD UNIVERSITY

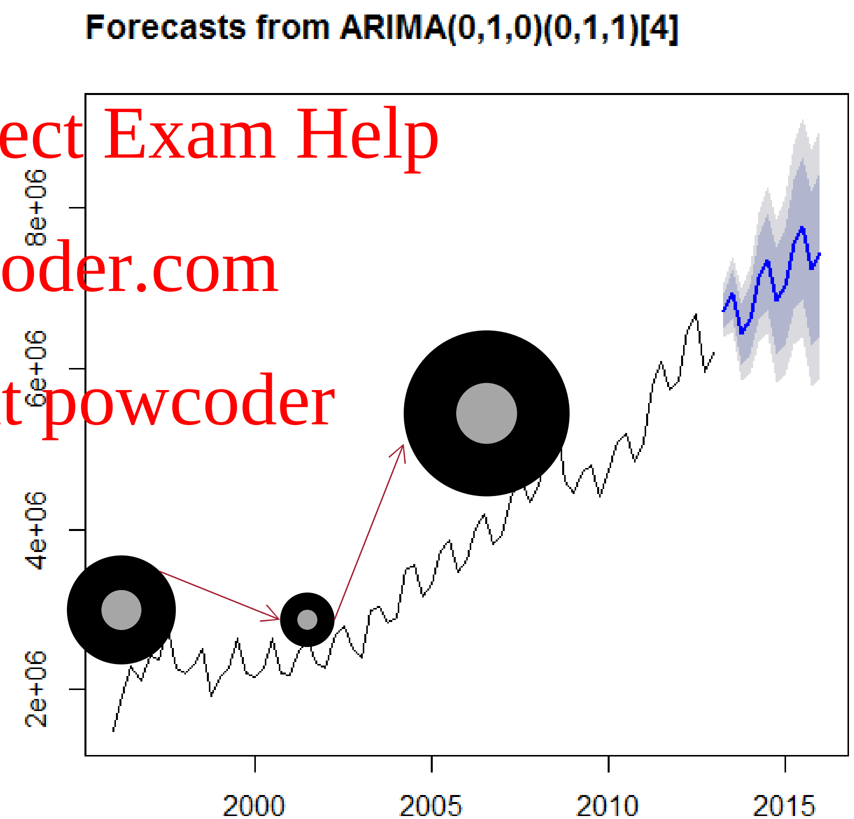
Auto Regressive Integrated Moving Averages ARIMA Analogy

Arima forecasts using a combination of p, d, q inputs

- **p** is the number of autoregressive terms,
- **d** is the number of nonseasonal differences, and
- **q** is the number of lagged forecast errors in the prediction equation

Good Reference:
http://people.duke.edu/~rnau/411arim.htm

As a tire rolls across a bumpy road, one can adjust the tread, air pressure, and diameter to get the smoothest ride. ARIMA adjust these inputs to get a close fit to the bumpy road. Think of these inputs as similar to the PDQ



Auto.arima() will adjust lags and p/d/q to extract more of the auto correlation (information shared between rows)

HARVARD UNIVERSITY

Let’s Practice

7_autoArima_AMZN.R

Before you Embark on Forecasting - Random walks

Before forecasting, consider "is the time series predictable or is it a random walk?

When we do any forecasting first try to do an AR(1) model.

- Test that slope = 1 in an AR(1) model (i.e. that the forecast for a period is the most recently-observed value)
- If the beta coefficient has a small p-value then the values are predictable and you should do a forecast (not a random walk)

HARVARD UNIVERSITY