Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Business Analytics using Data Mining & Forecasting

BU7143 & BU7144

Dr. Nicholas P. Danks
Business Analytics
Email address

Overview of Today's Session

Part 1: Time Series Data

1. Core tasks/goals of Data Mining
2. The process of Data Mining

1. Regression Forecasting
    1. Variable types
    2. Outliers, missing data, normal data
    3. An example

Part I
Time Series Data

Main ideas

- Forecast future values of a time series
- Distinction between forecasting (main focus) and describing/explaining
- Four components of time series:
  - Level
  - Trend
  - Seasonality
  - noise

Explain vs. Predict

Explanation is the goal of "time series **analysis**"

- Models are based on causal argument
- Models are not "black-box"

Forecasting (our focus) seeks to **predict** future values

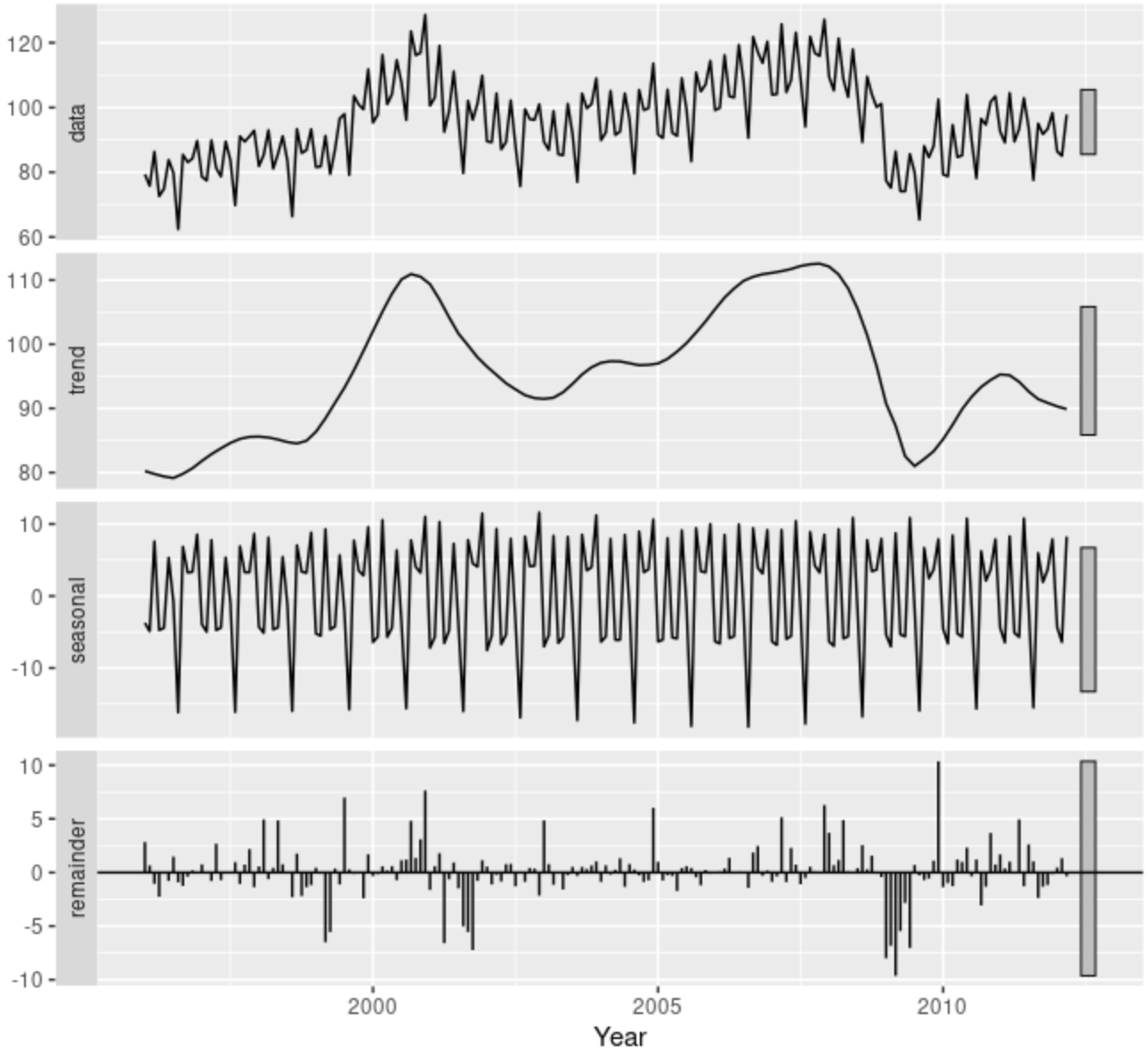
Time Series Components

(Level)

Trend

Seasonality

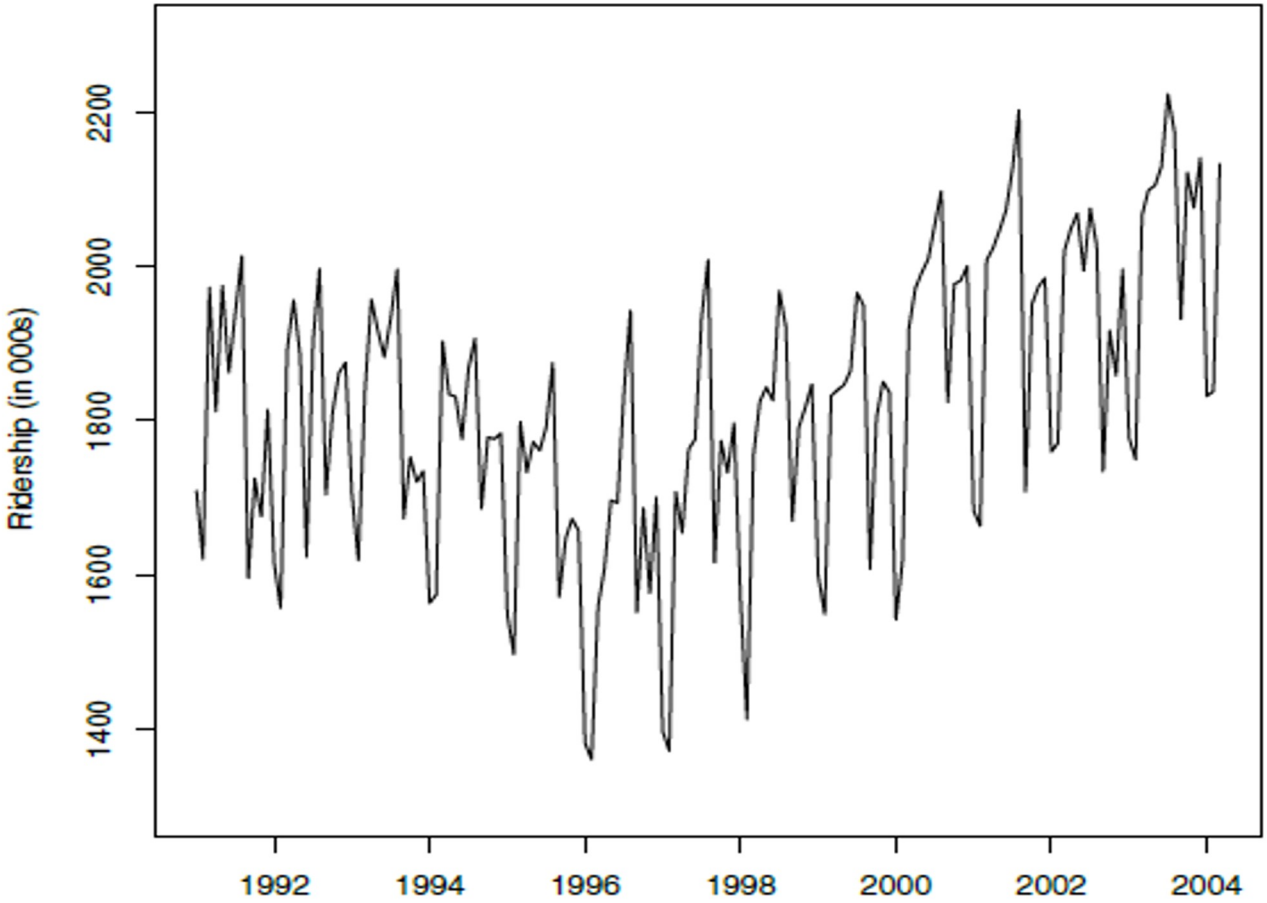
Noise



Amtrak Ridership (monthly)

Level - about 1,800,000 passengers per month

Appears to have U-shaped trend



R Code for Preceding Plot

```
library(forecast)
Amtrak.data <- read.csv("Amtrak.csv")

# create time series object using ts()
# ts() takes three arguments: start, end, and freq.
# with monthly data, the frequency of periods per cycle is 12 (per year).
# arguments start and end are (cycle [=year], seasonal period [=month] number)
# pairs.
# here start is Jan 1991: start = c(1991, 1); end is Mar 2004: end = c(2004,
# 3).

ridership.ts <- ts(Amtrak.data$Ridership,
start = c(1991, 1), end = c(2004, 3), freq = 12)

# plot the series
plot(ridership.ts, xlab = "Time", ylab = "Ridership (in 000s)", ylim = c(1300,
2300))
```

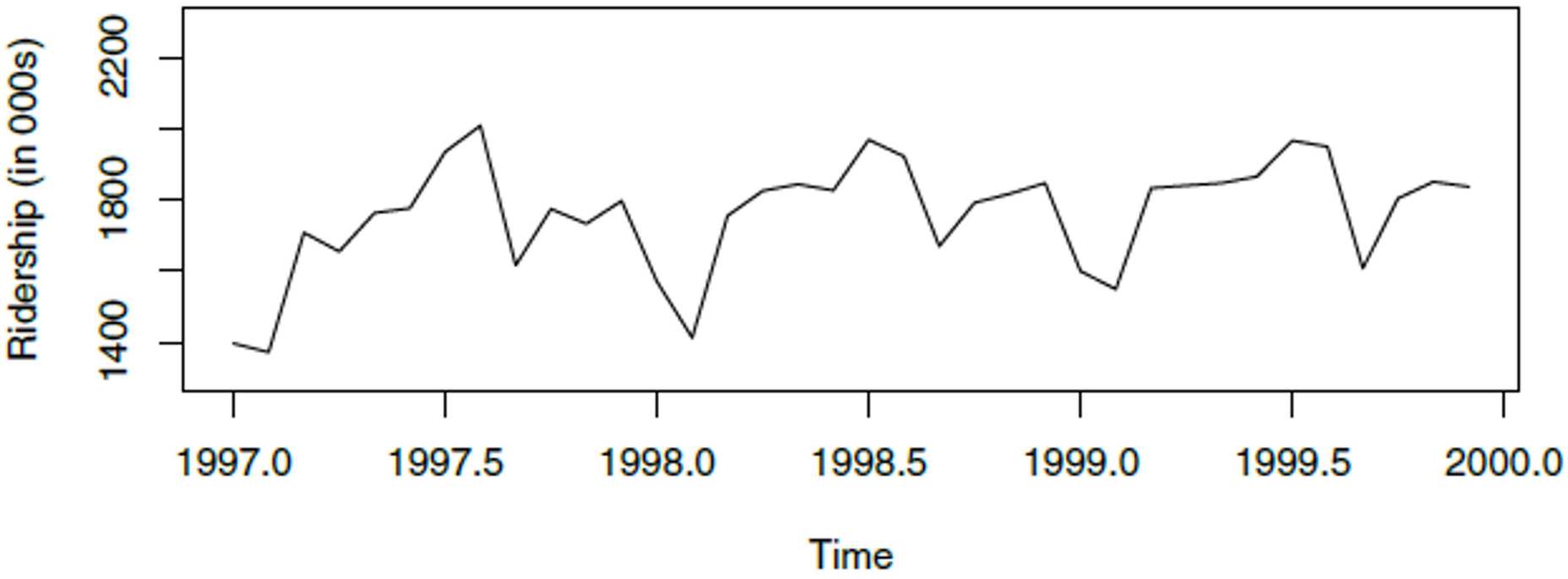
Zoom to 3 years (1997-1999)

Seasonality* appears:

Each year traffic peaks in summer

Noise:

Departure from the general level that is neither trend nor seasonality



Don’t confuse the time series term “season,” which is the period over which a cyclical pattern repeats (e.g. a year), with the standard English seasons of the year (fall, winter, etc.)

Partitioning

Divide data into training portion and validation portion

Test model on the validation portion

Random partitioning would leave holes in the data, which causes problems

Forecasting methods assume regular sequential data

Instead of random selection, divide data into two parts

Train on early data

Validate on later data

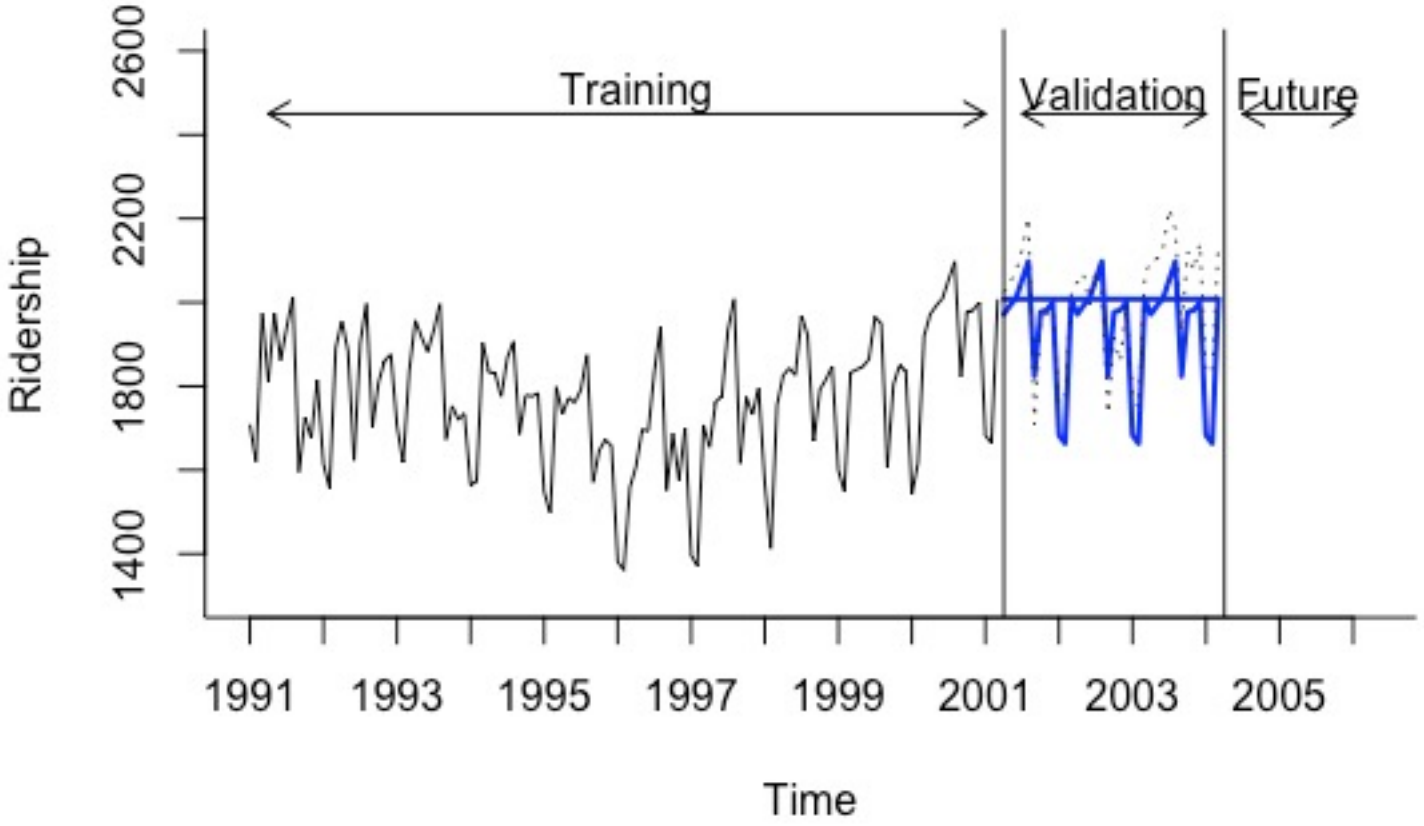
Performance can be assessed against the “naïve benchmark” – *naïve forecast* is simply the most recent value in the time series

Timeseries partitioning is not random!!

Benchmarks

Naïve benchmark is the trend, or average

Seasonal Naïve is the same value for prior season period (m,d,y)

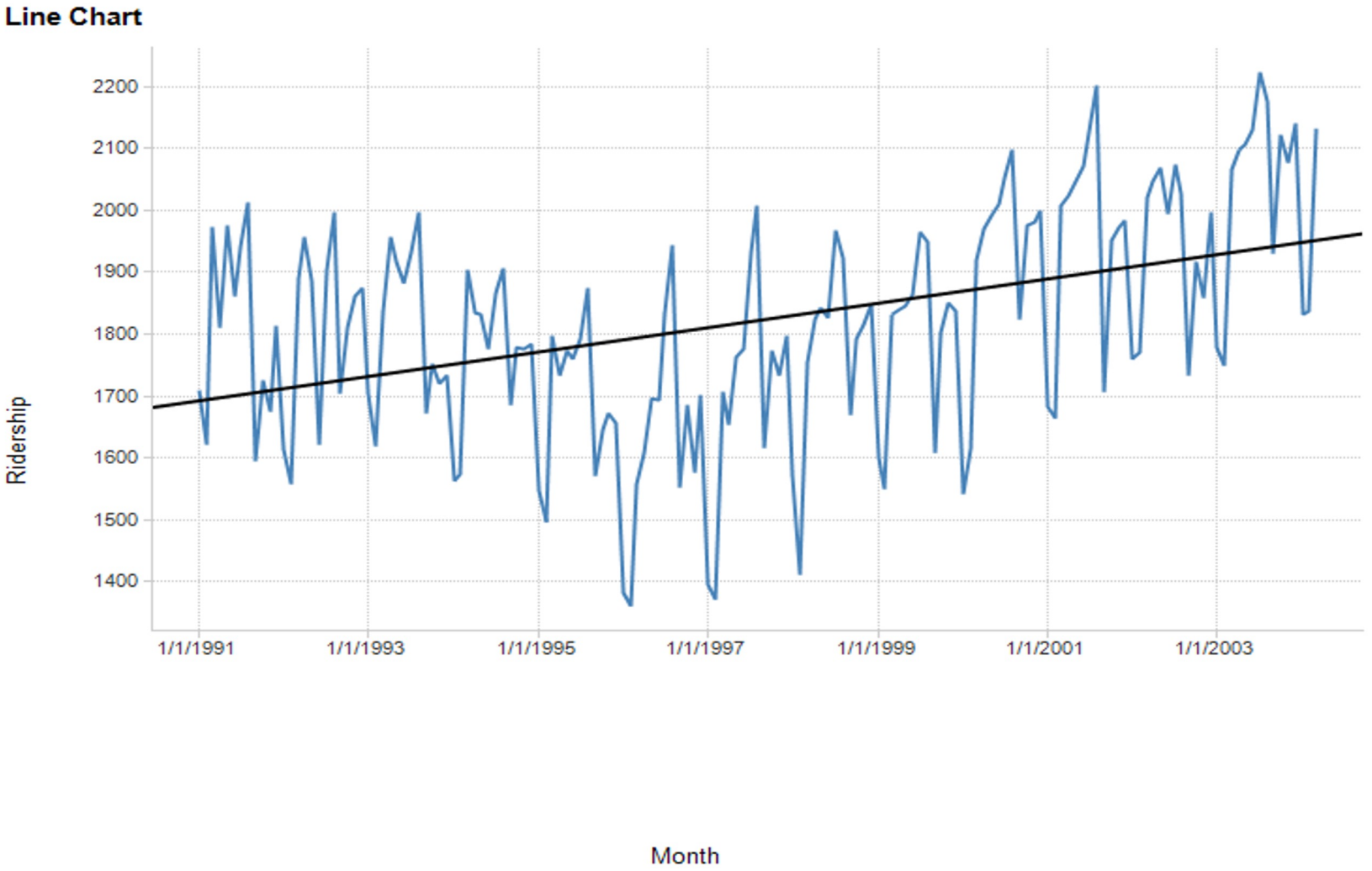


Part II
Regression Forecasting

Main ideas

- Fit linear trend, time as predictor
- Modify & use also for non-linear trends
  - Exponential
  - Polynomial
- Can also capture seasonality

Linear fit to Amtrak ridership data



(Doesn’t fit too well – more later)

The regression model

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)

Our trend model is linear, which we can see from the graph is not a good fit (more later)

function `ts` converts Amtrak .csv data into time series object

Although the original data have the time points, `ts` uses only the ridership data, and recreates the time points itself with `start`, `end`, and `freq`

```
library(forecast)
Amtrak.data <- read.csv("Amtrak.csv")

# create time series
ridership.ts <- ts(Amtrak.data$Ridership, start = c(1991,1),
     end = c(2004,3), freq = 12)

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

# plot the series
plot(ridership.ts, xlab = "Time", ylab = "Ridership", ylim =
     c(1300,2300), bty = "l")
lines(ridership.lm$fitted, lwd = 2)
```

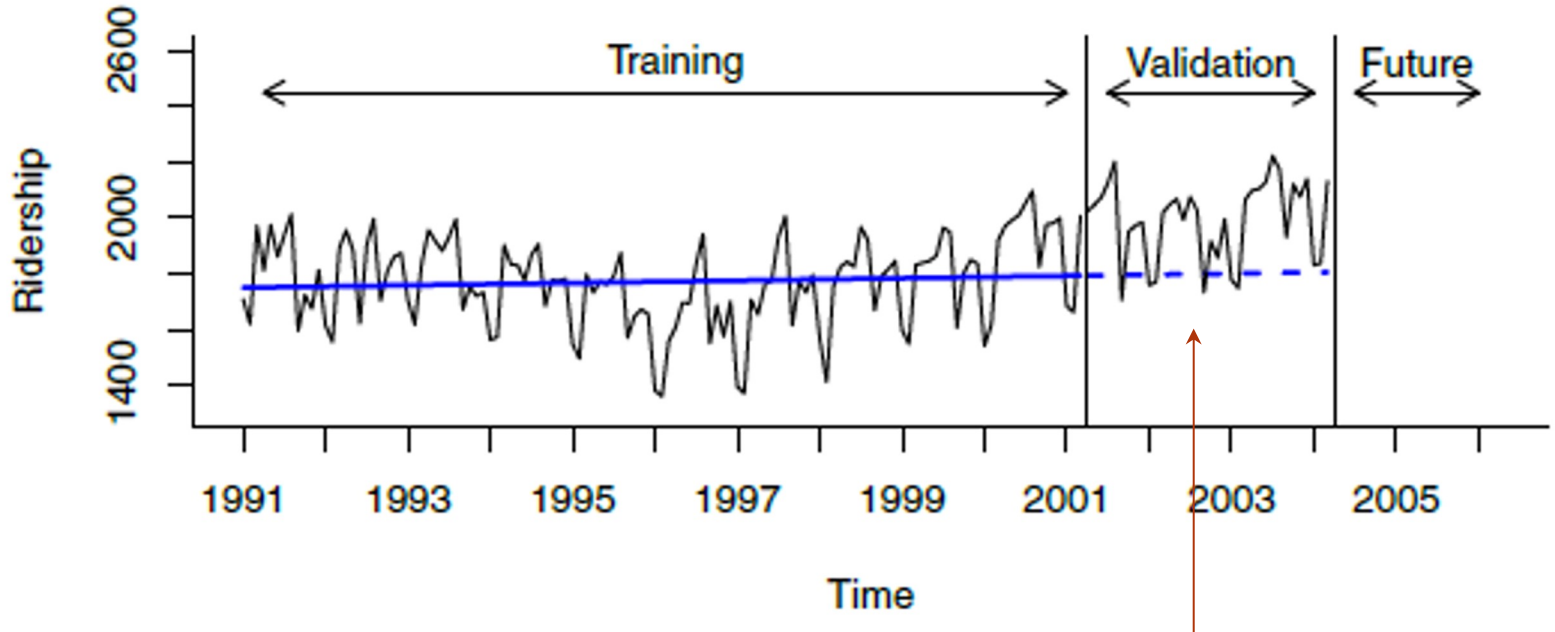
`trend` is predictor variable created on the fly from the time series data

`tslm` fits linear model to data with time series components

Partitioning the data

```
nValid <- 36
nTrain <- length(ridership.ts) - nValid

# partition the data
train.ts <- window(ridership.ts, start = c(1991, 1),
     end  = c(1991, nTrain))
valid.ts <- window(ridership.ts, start = c(1991, nTrain + 1),
     end = c(1991, nTrain + nValid))
```

```
# fit linear trend model to training set and create
forecasts
train.lm <- tslm(train.ts ~ trend)
train.lm.pred <- forecast(train.lm, h = nValid, level = 0)
```



Trend based on training data
underestimates validation period

Plotting linear fit to ridership data (previous slide), and forecast errors (not shown)

```
fit linear trend model to training set and create forecasts
train.lm <- tslm(train.ts ~ trend)
train.lm.pred <- forecast(train.lm, h = nValid, level = 0)

par(mfrow = c(2, 1))
plot(train.lm.pred, ylim = c(1300, 2600), ylab = "Ridership", xlab =
   "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "", flty = 2)
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(train.lm.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts)
plot(train.lm.pred$residuals, ylim = c(-420, 500), ylab = "Forecast Errors",
     xlab = "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "")
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(valid.ts - train.lm.pred$mean, lwd = 1)
```

Exponential Trend

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

e.g. t_1 is x% more than t_0, t_2 is x% more than t_1…

Replace Y with log(Y) then fit linear regression

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

Exponential trend - forecast errors

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

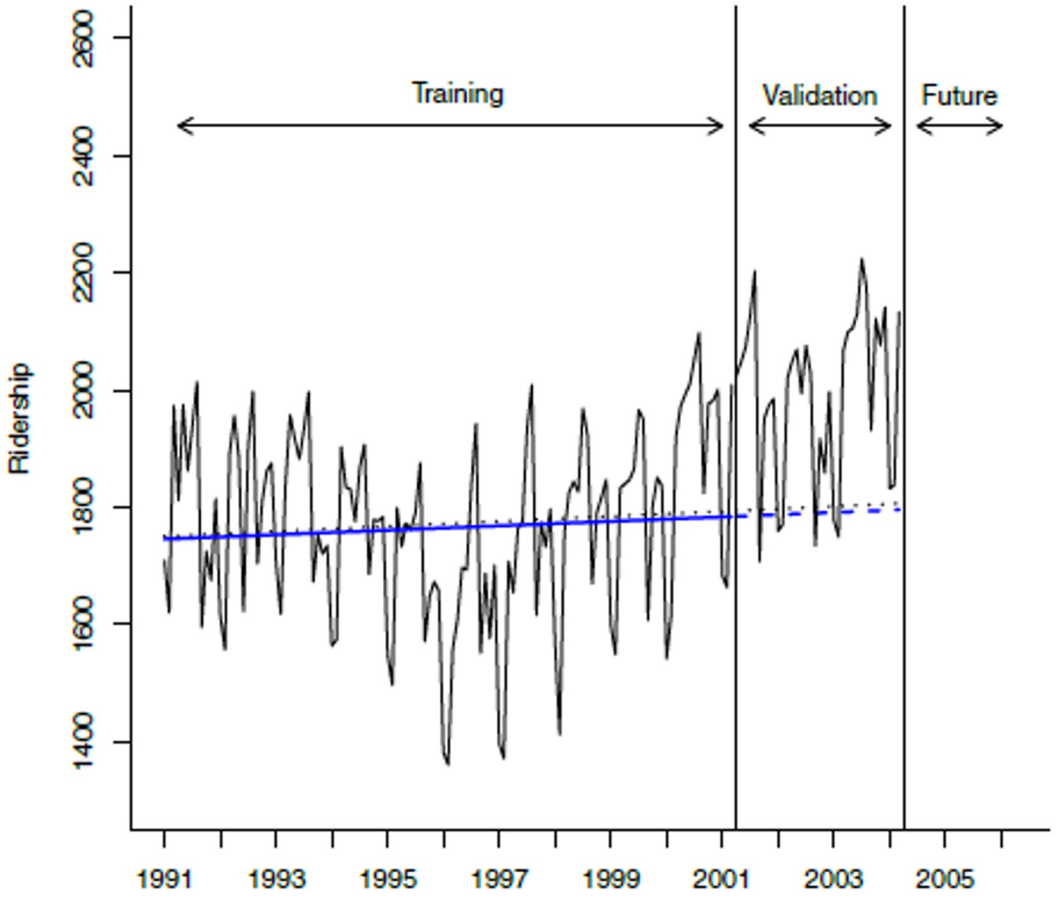
Model forecasts will be in the form log(*Y*)

Return to original units by taking exponent of model forecasts

Calculate standard deviation of these forecast errors to get RMSE

```
# fit exponential trend using tslm() with argument
# lambda = 0

train.lm.expo.trend <- tslm(train.ts ~ trend, lambda = 0)
train.lm.expo.trend.pred <- forecast(train.lm.expo.trend,
     h = nValid, level = 0)
```

Exponential trend (dotted line) very similar to linear trend (solid line)



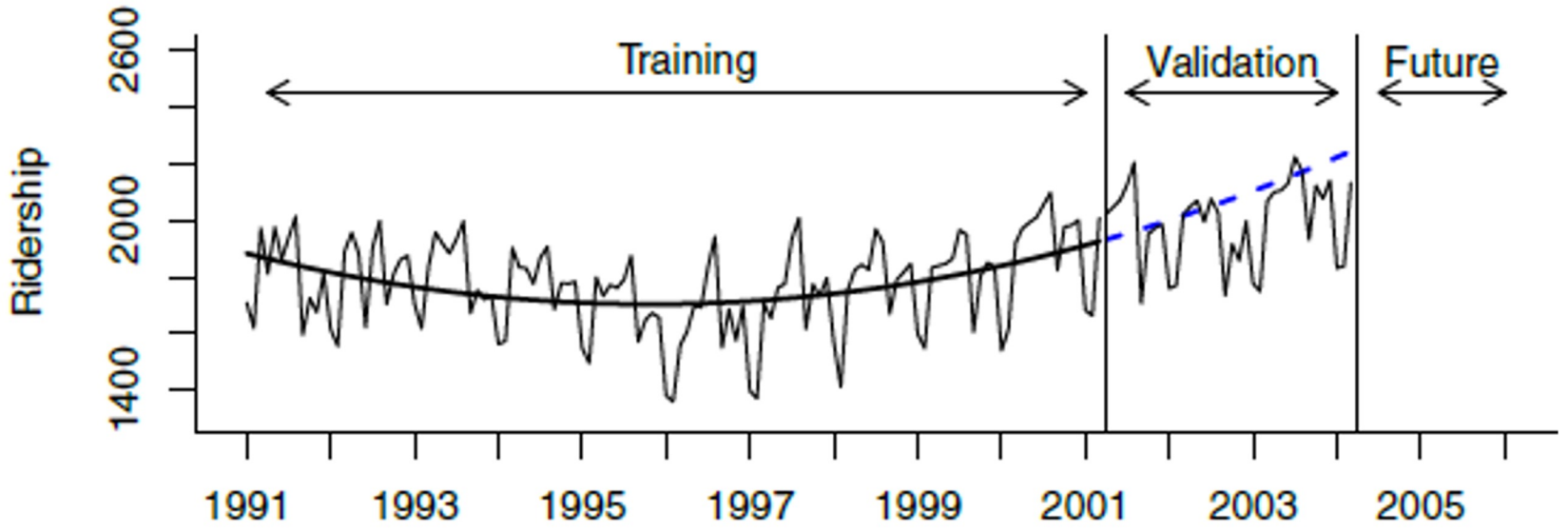
Polynomial Trend

Add additional predictors as appropriate

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

Fit linear regression using both t and t^2

```
# 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)
```



Better job capturing the trend, though it over forecasts in validation period.
Next: we'll try capturing seasonality.

Handling Seasonality

- Seasonality is any recurring cyclical pattern of consistently higher or lower values (daily, weekly, monthly, quarterly, etc.)
- Handle in regression by adding categorical variable for season, e.g.

Month	Ridership	Season
Jan-91	1709	Jan
Feb-91	1621	Feb
Mar-91	1973	March
Apr-91	1812	April

11, not 12, to avoid multicollinearity

```
# include season as a predictor in tslm(). Here it creates 11
# dummies, one for each month except for first season, January
train.lm.season <- tslm(train.ts ~ season)
summary(train.lm.season)
```

Final model, Amtrak data

Incorporates trend and seasonality

13 predictors

- 11 monthly dummies
- t
- t^2

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

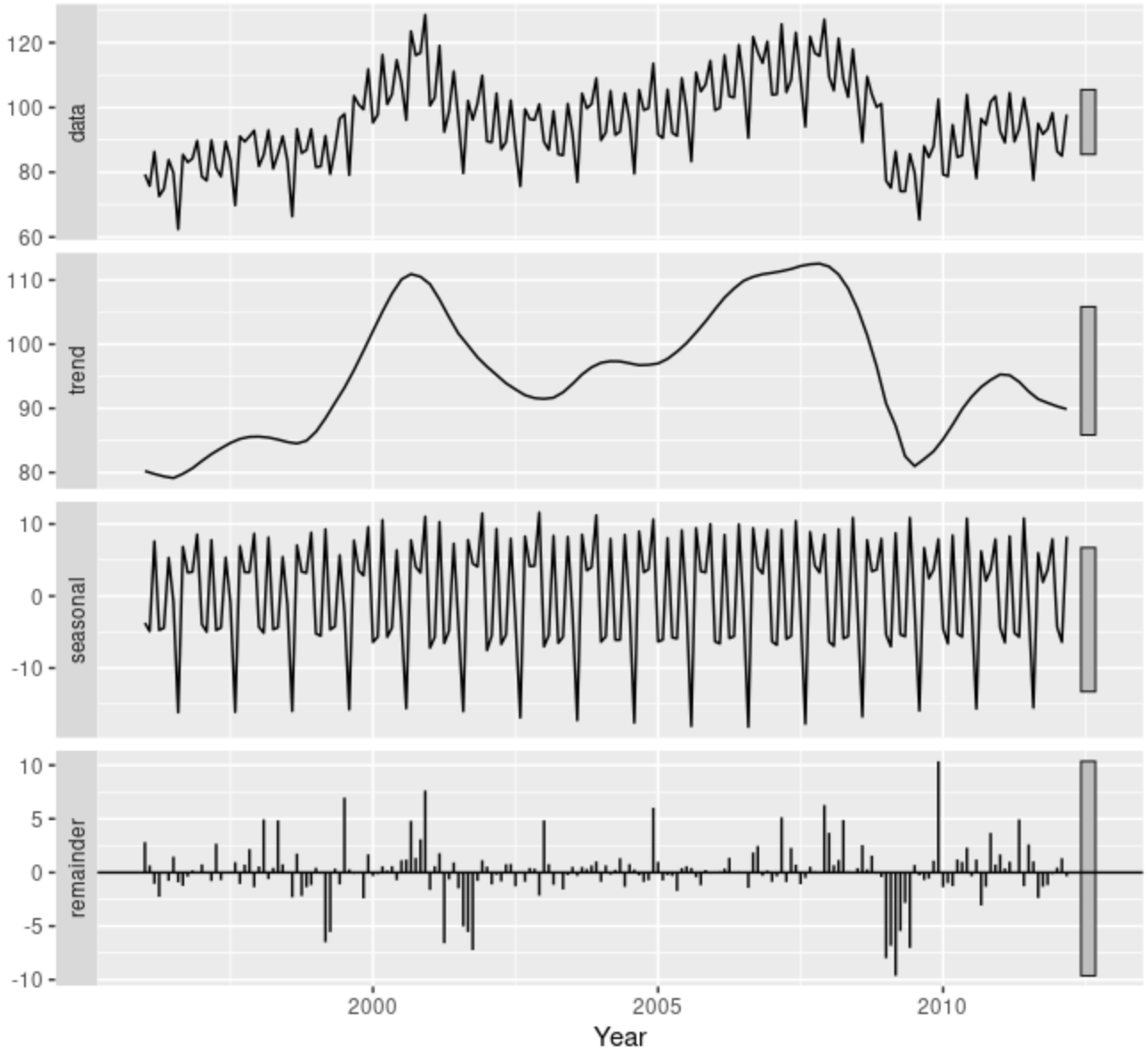
Time Series Components

(Level)

Trend

Seasonality

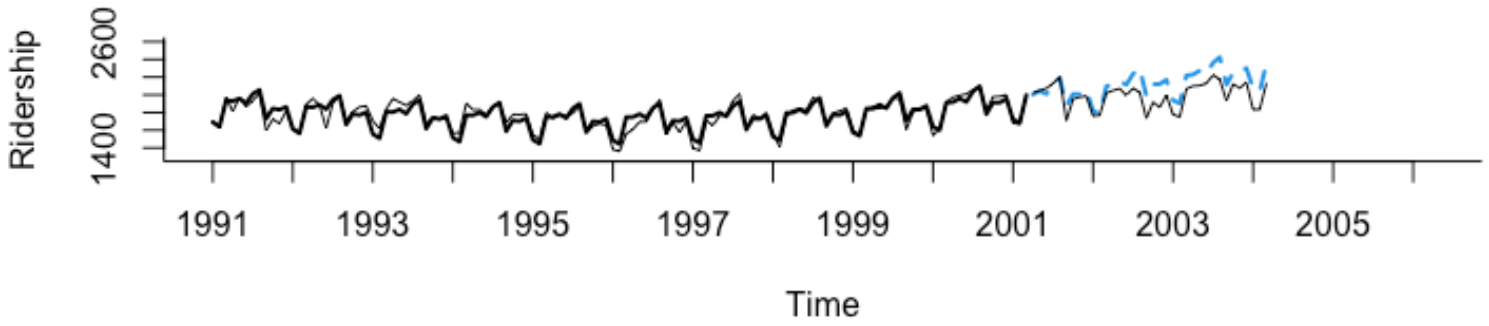
Noise

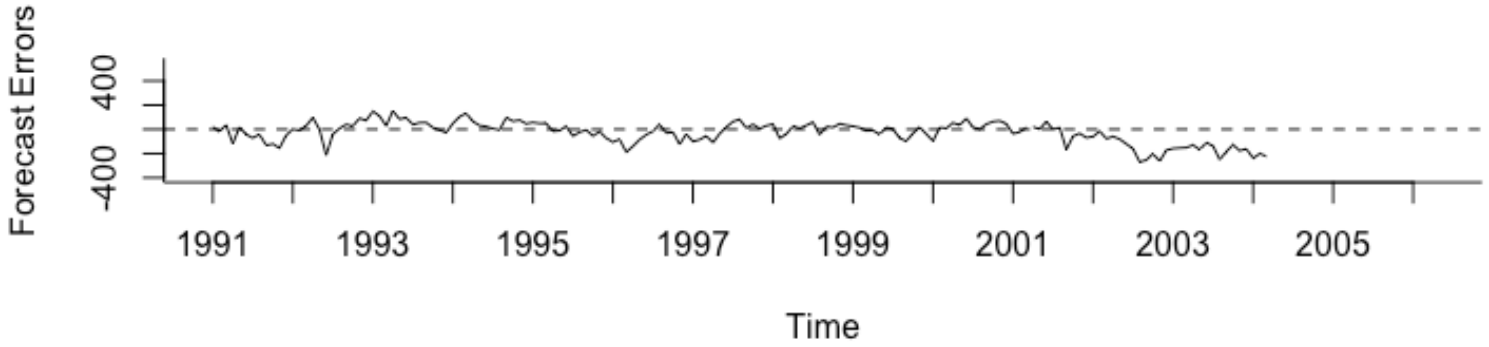


Output of full model

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

Call:
tslm(formula = train.ts ~ trend + I(trend^2) + season)

Residuals:
     Min       1Q   Median       3Q      Max
-213.775  -39.363    9.711   42.422  152.187

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.697e+03  2.768e+01  61.318  < 2e-16 ***
trend       -7.156e+00  7.293e-01  -9.812  < 2e-16 ***
I(trend^2)   6.074e-02  5.698e-03  10.660  < 2e-16 ***
season2     -4.325e+01  3.024e+01  -1.430  0.15556
season3      2.600e+02  3.024e+01   8.598 6.60e-14 ***
season4      2.606e+02  3.102e+01   8.401 1.83e-13 ***
season5      2.938e+02  3.102e+01   9.471 6.89e-16 ***
season6      2.490e+02  3.102e+01   8.026 1.26e-12 ***
season7      3.606e+02  3.102e+01  11.626  < 2e-16 ***
season8      4.117e+02  3.102e+01  13.270  < 2e-16 ***
season9      9.032e+01  3.102e+01   2.911  0.00437 **
season10     2.146e+02  3.102e+01   6.917 3.29e-10 ***
season11     2.057e+02  3.103e+01   6.629 1.34e-09 ***
season12     2.429e+02  3.103e+01   7.829 3.44e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 70.92 on 109 degrees of freedom
Multiple R-squared:  0.8246,         Adjusted R-squared:  0.8037
F-statistic: 39.42 on 13 and 109 DF,  p-value: < 2.2e-16
```

Full model Performance






```
> train.lm.trend.season.pred <- forecast(train.lm.trend.season, h = nValid, level = 0)
> accuracy(train.lm.trend.season.pred, valid.ts)
                      ME      RMSE       MAE        MPE     MAPE      MASE      ACF1 Theil's U
Training set  3.693205e-15  66.76143  51.95091 -0.1525653 3.015509 0.6297693 0.6040588        NA
Test set     -1.261654e+02 153.25066 131.72503 -6.4314945 6.698700 1.5968226 0.7069291 0.8960679
>
> train.lm.season.pred <- forecast(train.lm.season, h = nValid, level = 0)
> accuracy(train.lm.season.pred, valid.ts)
                      ME      RMSE       MAE        MPE      MAPE      MASE      ACF1 Theil's U
Training set 3.685399e-15  96.34038  75.13099 -0.3099246  4.326881 0.9107674 0.7856939        NA
Test set     2.179267e+02 229.65092 217.92668 10.8646179 10.864618 2.6417928 0.6346963  1.330938
```

Autocorrelation

Unlike cross-sectional data, time-series values are typically correlated with nearby values (“autocorrelation”)

Ordinary regression does not account for this

Computing autocorrelation

Create “lagged” series

Copy of the original series, offset by one or more timer periods

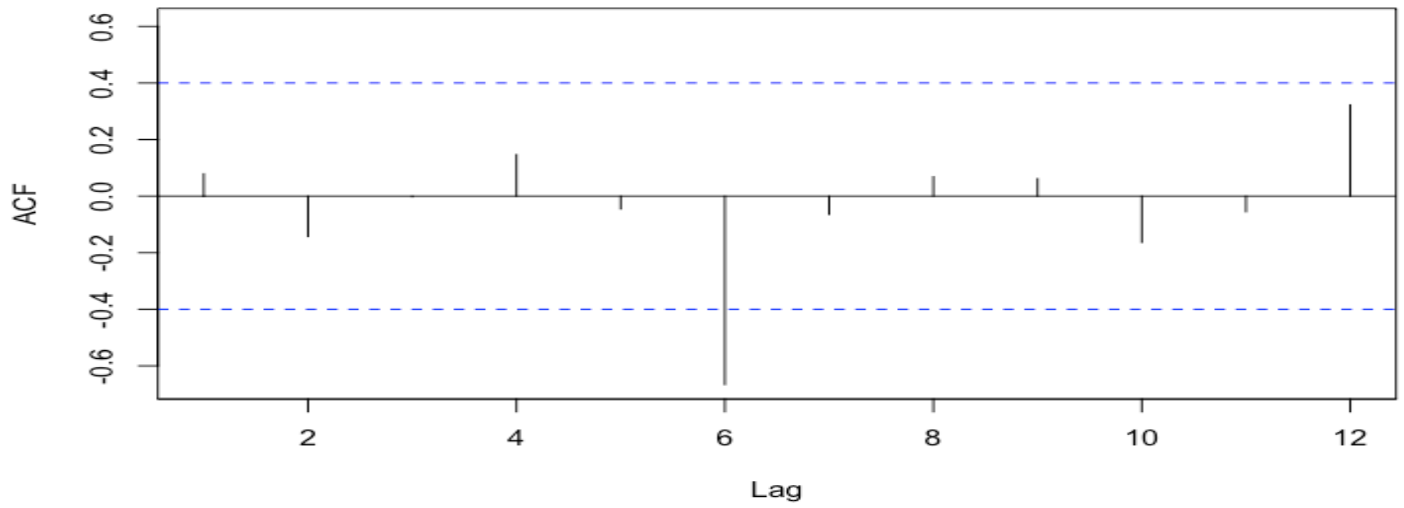
Compute correlation between original series and lagged series

- Lag-1, lag-2, etc.

Amtrak – original series and Lag-1, Lag-2

TABLE 16.1 FIRST 24 MONTHS OF AMTRAK RIDERSHIP SERIES

Month	Ridership	Lag-1 Series	Lag-2 Series
Jan-91	1709		
Feb-91	1621	1709	
Mar-91	1973	1621	1709
Apr-91	1812	1973	1621
May-91	1975	1812	1973
Jun-91	1862	1975	1812
Jul-91	1940	1862	1975
Aug-91	2013	1940	1862
Sep-91	1596	2013	1940
Oct-91	1725	1596	2013
Nov-91	1676	1725	1596
Dec-91	1814	1676	1725
Jan-92	1615	1814	1676

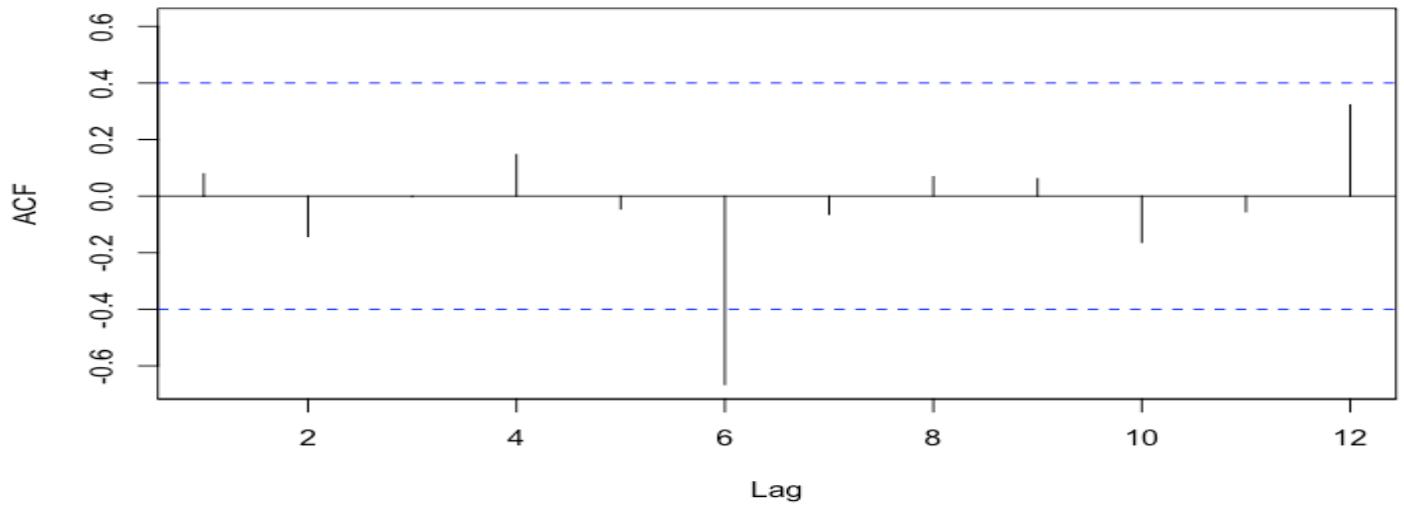


Autocorrelation

Positive autocorrelation at lag-1 = stickiness

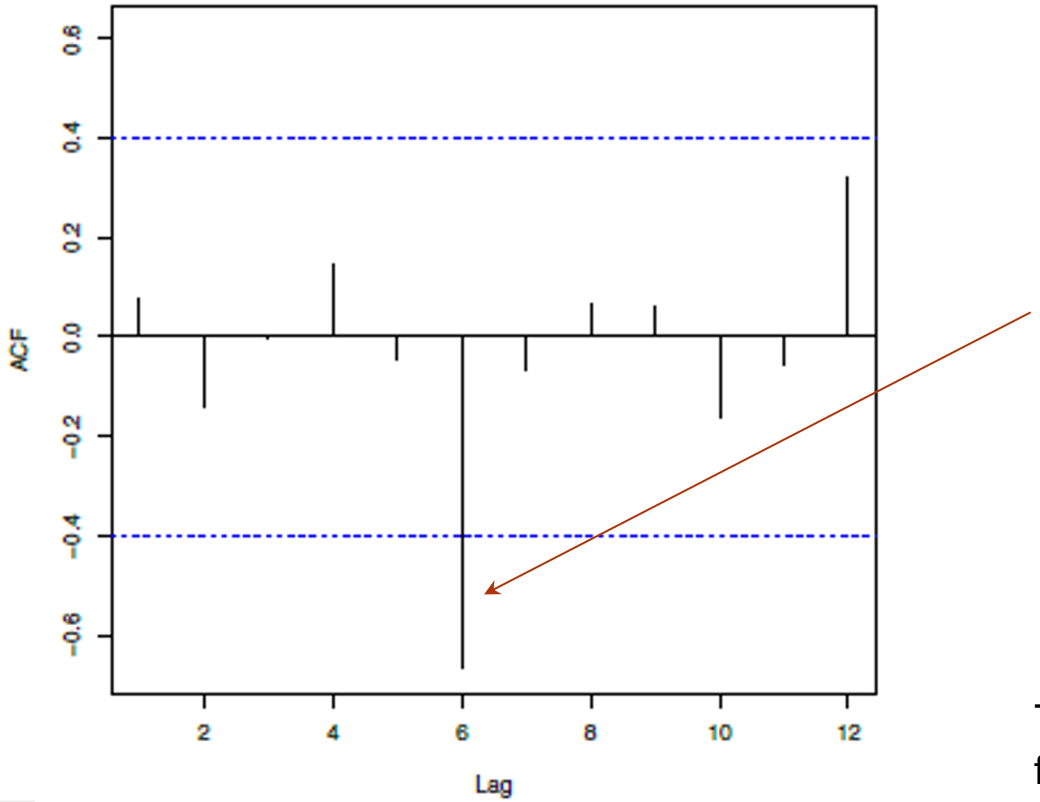
Strong autocorrelation (positive or negative) at a lag > 1 indicates seasonal (cyclical) pattern

Autocorrelation in residuals indicates the model has not fully captured the seasonality in the data



Compute & display autocorrelation for different lags, over 24 months:

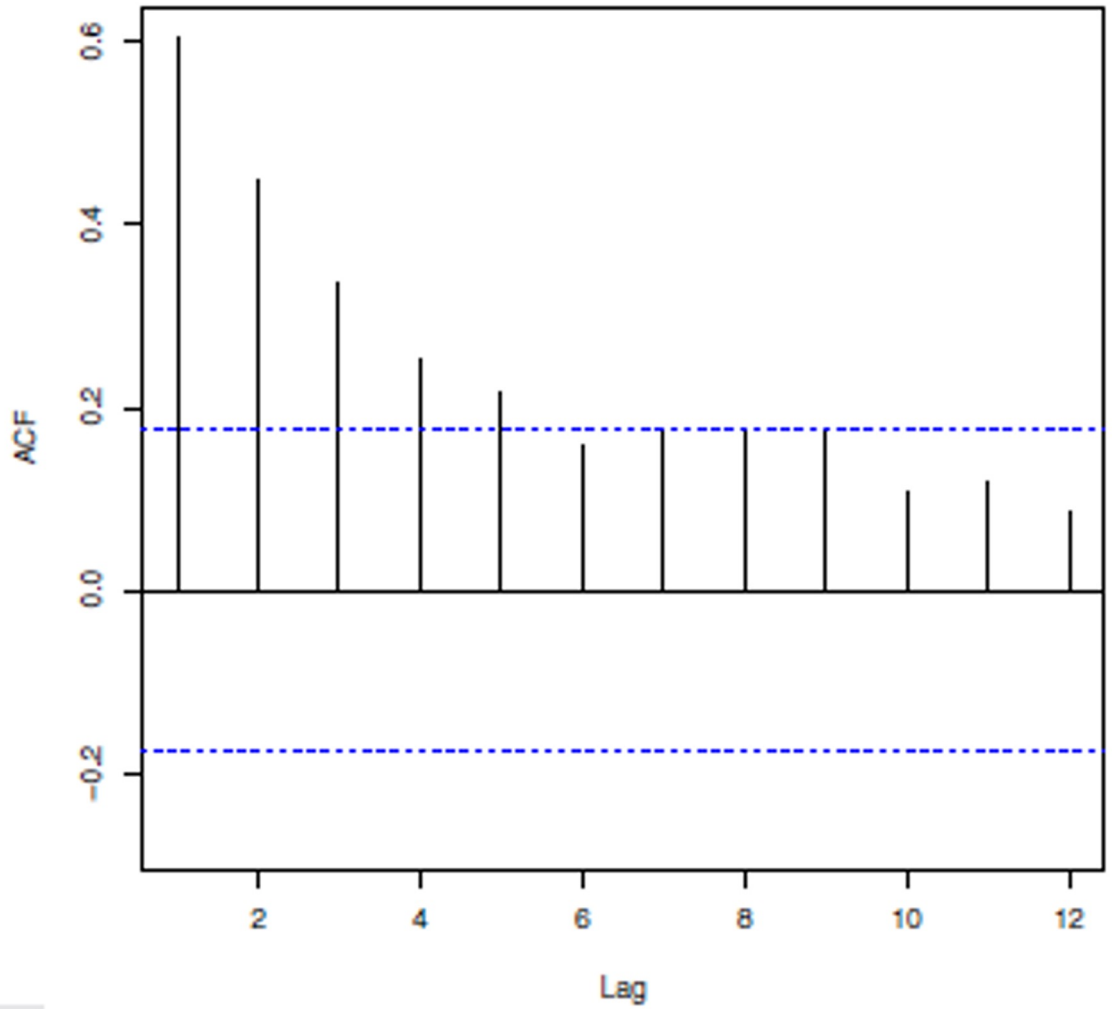
```
ridership.24.ts <- window(train.ts, start = c(1991, 1),
     end = c(1991, 24))
Acf(ridership.24.ts, lag.max = 12, main = "")
```



Strong negative correlation at 6 months shows seasonal pattern (high summer traffic, low winter)

The dotted lines are confidence bounds for judging statistical significance

It is useful to examine autocorrelation for the residuals:



Strong autocorrelation from lag 1 on, but lag 6 no longer dominates.

Note: If you have correlation at lag 1, it will naturally propagate to lag 2, 3, etc., tapering off

Incorporating autocorrelation into models

Use a forecasting method to forecast *k*-steps ahead

Fit AR (autoregressive) model to residuals

Incorporate residual forecasts

Improved $F_{t+k} = F_{t+k} + E_{t+k}$

Choose order of the AR model

If autocorrelation exists at Lag-1, a Lag-1 model should be sufficient to capture lags at other periods as well

$E_t = B_0 + B_1 E_{t-1} + e$

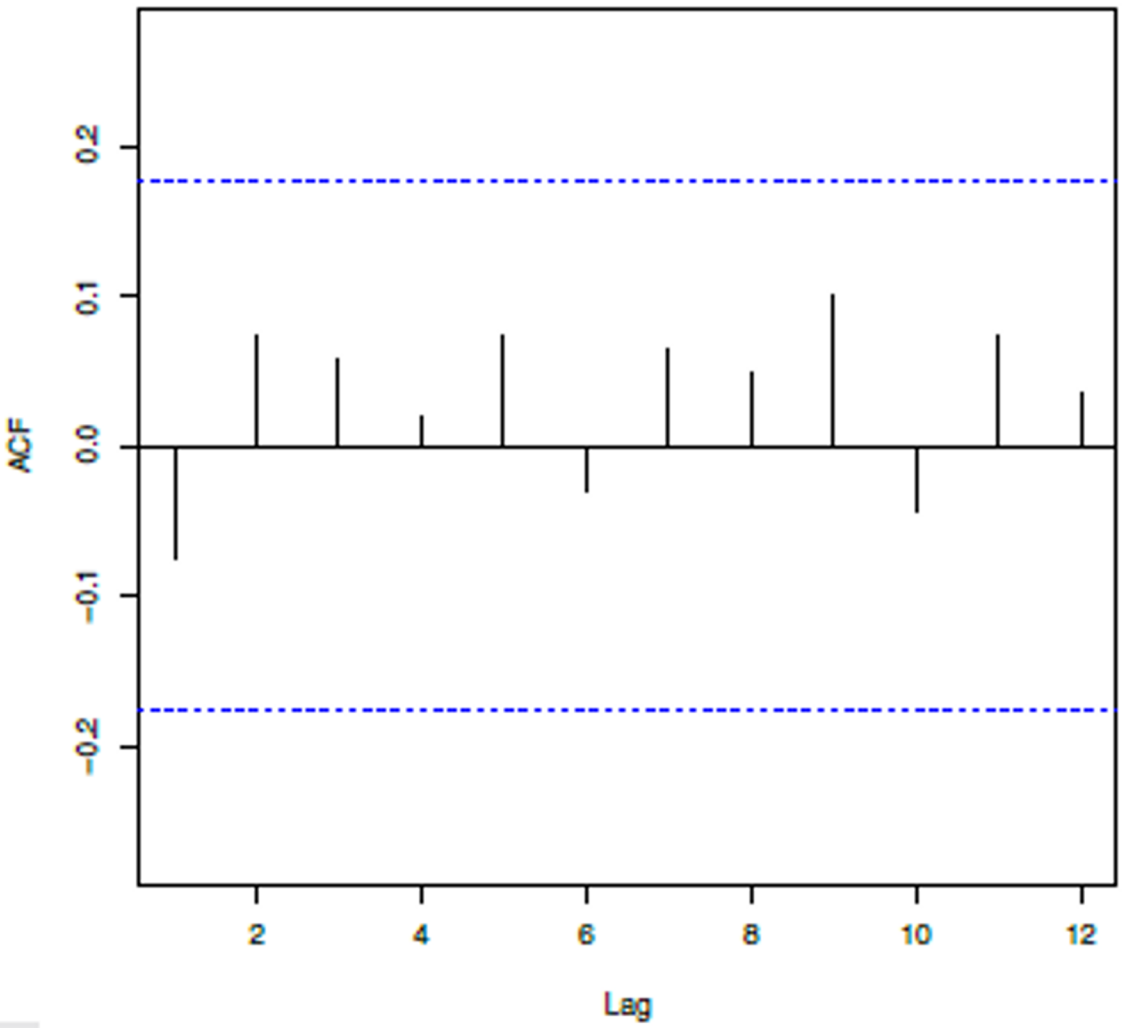
Where E_t is residual (forecast error) at time t

Adding ARIMA lag 1 - AR(1) - to earlier model…

```
# fit linear regression with quadratic trend and seasonality
# to Ridership

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

# fit AR(1) model to training residuals
# use Arima() in the forecast package to fit an ARIMA model
# (that includes AR models); order = c(1,0,0) gives an AR(1).

train.res.arima <- Arima(train.lm.trend.season$residuals,
     order = c(1,0,0))
valid.res.arima.pred <- forecast(train.res.arima, h = 1)
```

Plot autocorrelation of “residuals of residuals”

Autocorrelation is mostly gone – AR(1) has adequately captured the autocorrelation in the data:



Random walks

- Before forecasting, consider “is the time series predictable?”
- Or is it a random walk?
- Do a statistical hypothesis test that slope = 1 in an AR(1) model (i.e. that the forecast for a period is the most recently-observed value)
- If hypothesis cannot be rejected, series is statistically equivalent to a random walk (i.e. we have not shown that it is predictable).

Summary – Regression Based Forecasting

- Can use linear regression for exponential models (use logs) and polynomials (exponentiation)
- For seasonality, use categorical variable (make dummies)
- Incorporate autocorrelation by modeling it, then using those error forecasts in the main model

An Example

17.1 Impact of September 11 on Air Travel in the United States. The Research and Innovative Technology Administration's Bureau of Transportation Statistics conducted a study to evaluate the impact of the September 11, 2001 terrorist attack on US transportation. The 2006 study report and the data can be found at http://goo.gl/w2lJPV. The goal of the study was stated as follows:

> The purpose of this study is to provide a greater understanding of the passenger travel behavior patterns of persons making long distance trips before and after 9/11.

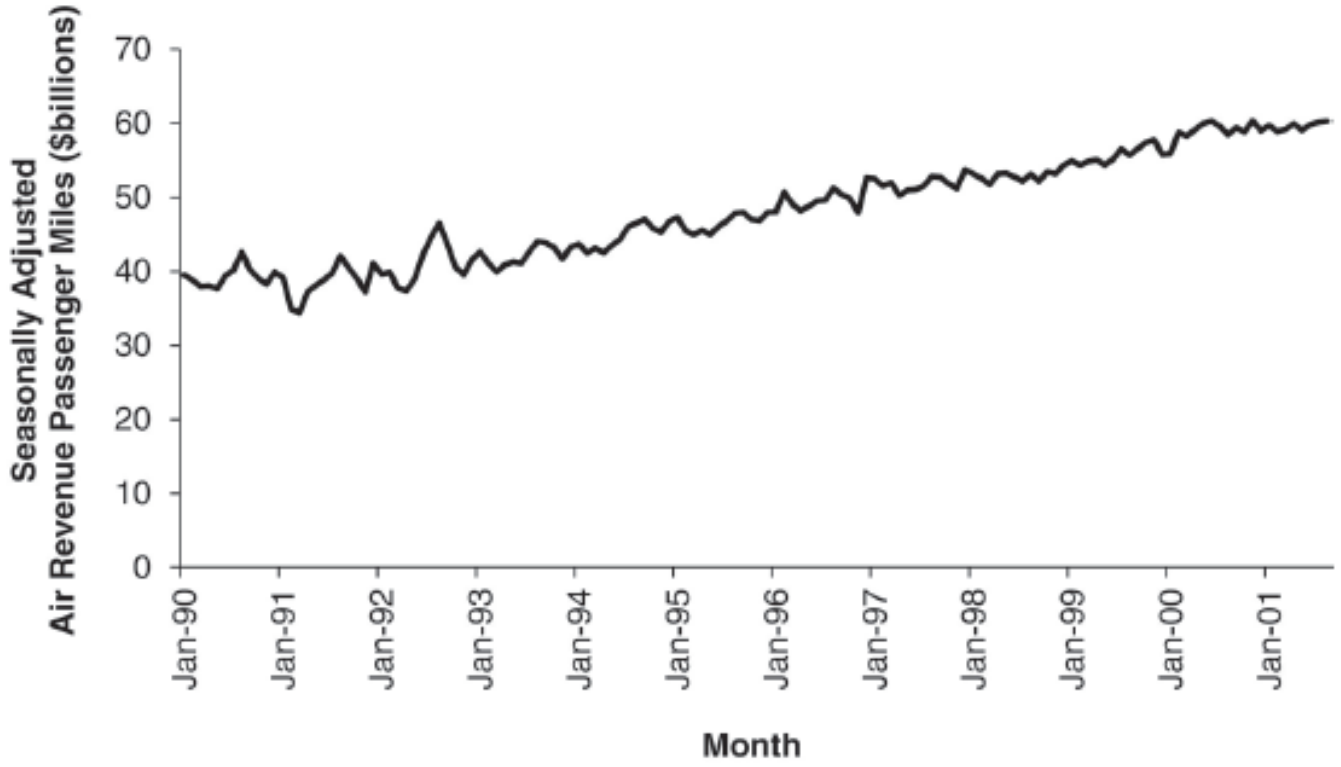
The report analyzes monthly passenger movement data between January 1990 and May 2004. Data on three monthly time series are given in file *Sept11Travel.csv* for this period: (1) Actual airline revenue passenger miles (Air), (2) rail passenger miles (Rail), and (3) vehicle miles traveled (Car).

In order to assess the impact of September 11, BTS took the following approach: using data before September 11, they forecasted future data (under the assumption of no terrorist attack). Then, they compared the forecasted series with the actual data to assess the impact of the event. Our first step, therefore, is to split each of the time series into two parts: pre- and post September 11. We now concentrate only on the earlier time series.

a. Plot the pre-event AIR time series. What time series components appear?

b. Figure 17.11 shows a time plot of the **seasonally adjusted** pre-September-11 AIR series. Which of the following methods would be adequate for forecasting the series shown in the figure?

- Linear regression model seasonality
- Linear regression model with trend
- Linear regression model with trend and seasonality



c. Specify a linear regression model for the AIR series that would produce a seasonally adjusted series similar to the one shown in Figure 17.11, with multiplicative seasonality. What is the outcome variable? What are the predictors?

d. Run the regression model from (c). Remember to use only pre-event data.

- **i.** What can we learn from the statistical insignificance of the coefficients for October and September?
- **ii.** The actual value of AIR (air revenue passenger miles) in January 1990 was 35.153577 billion. What is the residual for this month, using the regression model? Report the residual in terms of air revenue passenger miles.

e. Create an ACF (autocorrelation) plot of the regression residuals.

- **i.** What does the ACF plot tell us about the regression model's forecasts?
- **ii.** How can this information be used to improve the model?

f. Fit linear regression models to Air, Rail, and to Auto with additive seasonality and an appropriate trend. For Air and Rail, fit a linear trend. For Rail, use a quadratic trend. Remember to use only pre-event data. Once the models are estimated, use them to forecast each of the three post-event series.

- **i.** For each series (Air, Rail, Auto), plot the complete pre-event and post-event actual series overlayed with the predicted series.
- **ii.** What can be said about the effect of the September 11 terrorist attack on the three modes of transportation? Discuss the magnitude of the effect, its time span, and any other relevant aspects.