Time Series Forecasting

Github

https://github.com/mshanker1/60095_201810.git

Time Series Data

- Collected over time
- Time frequency can be short or long, or both
- Match scale of required forecast to level of noise in the data
- Individual Series versus Multiple Series

Descriptive versus Predictive Modeling

- In descriptive, or *time series analysis*, a time series is modeled to the determine the components in terms of seasonal patterns, trends, and relation to external factors.
- In *time series forecasting*, in contrast, we use current information to forecast future values.

Forecasting Methods

1. Regression-based methods
2. Data-driven, where methods learn from data
3. Recurrent neural networks

Combining Methods

We combine forecasts from multiple methods. For example, the first method uses the original time series for forecasts of future values, and then the second method revises the estimate by using the residual to generate forecast of future forecast errors.

Another approach is to use ensembles, where we average the forecast from different methods.

Time Series Components

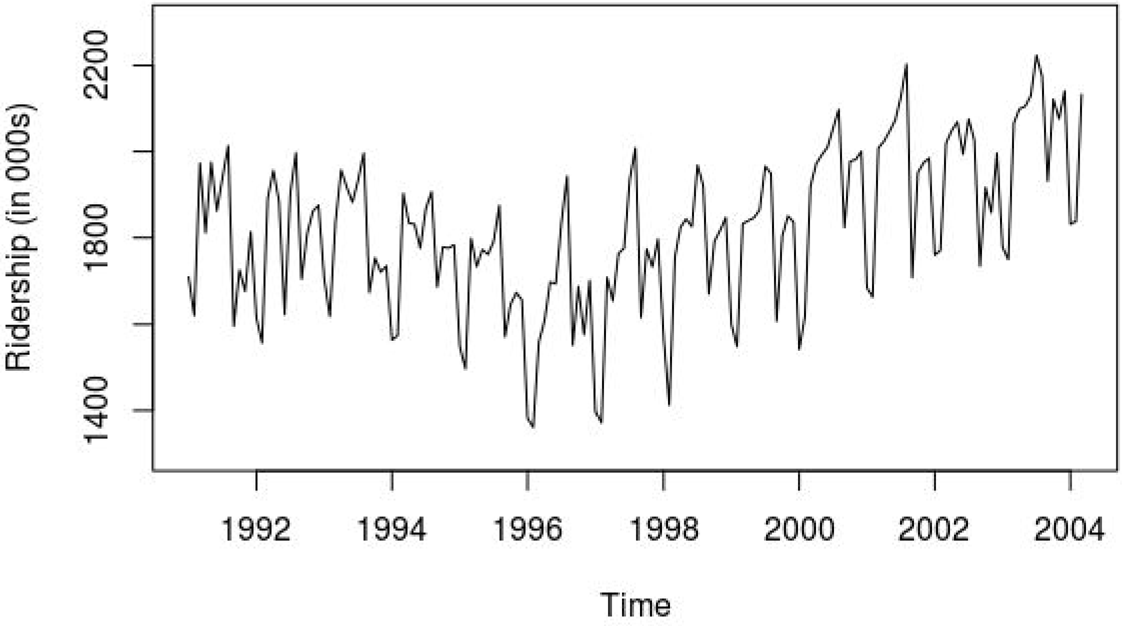
1. Level
2. Trend
3. Seasonality
4. Noise

Example - Amtrak Data

Amtrak, a US railway company, routinely collects data on ridership. Here we focus on forecasting future ridership using the series of monthly ridership between January 1991 and March 2004. These data are publicly available at www.forecastingprinciples.com

Amtrak Output

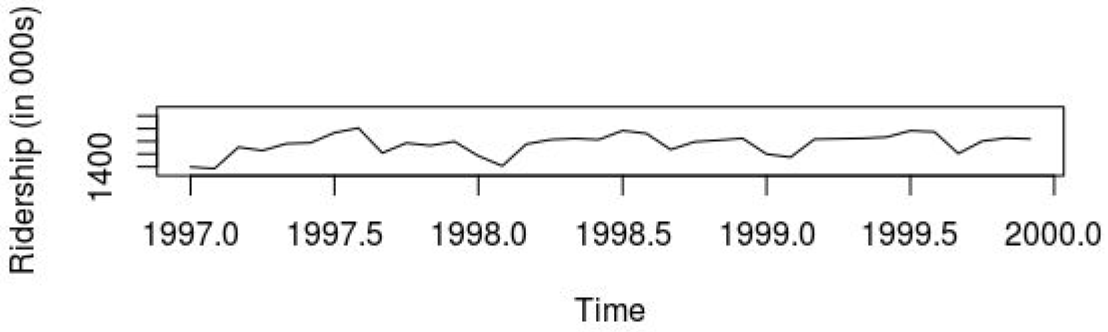
1. What is the overall level?
2. What is the trend?
3. Are there any seasonality?

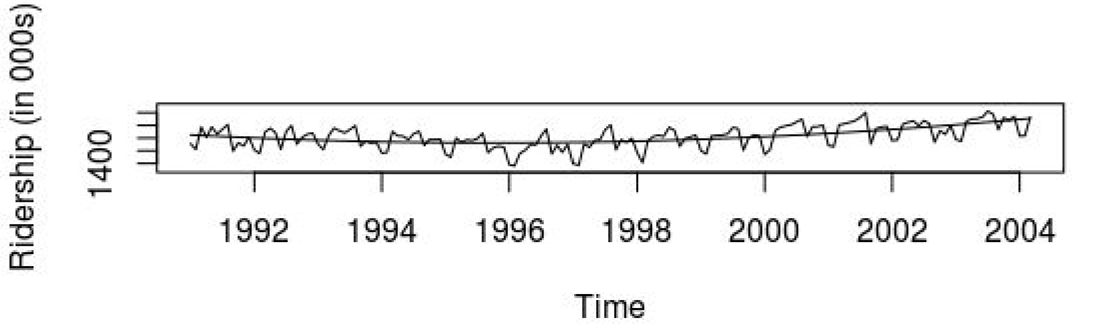


Enhanced Plots

Assumptions

- Trend: Linear, exponential,...
- Noise: Normality?





Model Driven Versus Data Driven

Model Driven

- Generally preferable when looking at global patterns, i.e., patterns that are relatively constant throughout the series
- When assumptions are met
- Uses all the data

Data Driven

- Preferable for local patterns
- Require fewer assumptions

Data Partitioning and Performance Evaluation

We use the same metrics that we use for cross-sectional data. For example, MAE, MAPE, or RMSELet the prediction error be defined as $e_i = y_i - \hat{y_i}$. Then,

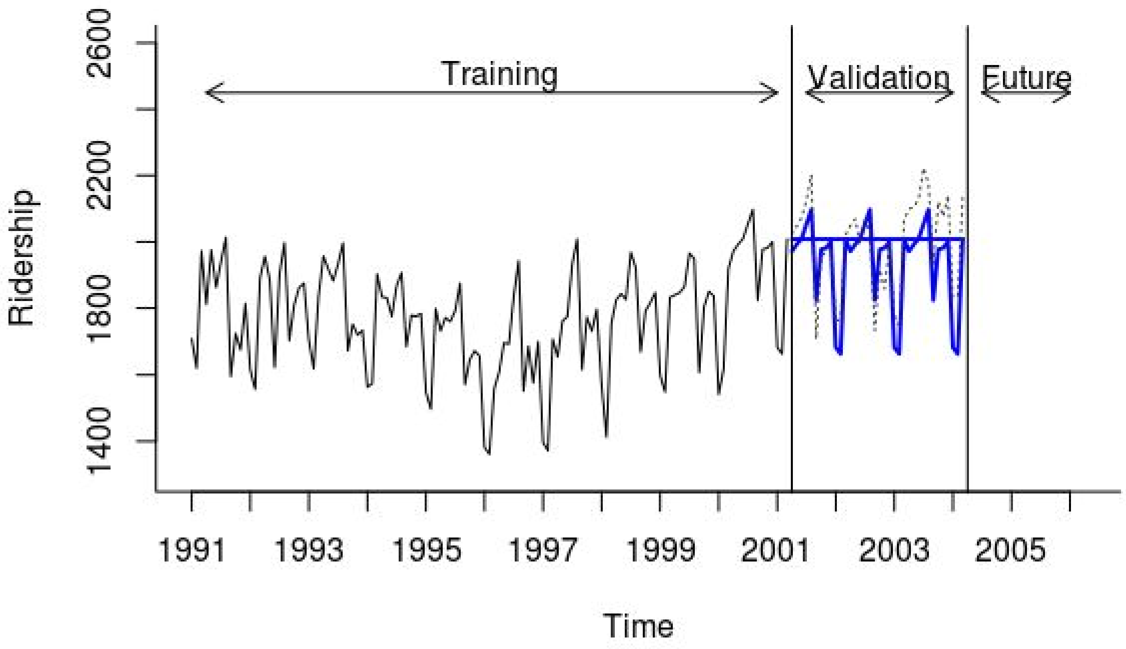
- MAE, MAD, or AAE (mean absolute error/deviation, or average absolute error) = $1/n \sum_{i=1}^{n} |e_i|$
- RMSE (root mean square error) = $\sqrt{1/n \sum_{i=1}^{n} e_i^2}$

But, how do we validate?

Benchmark: Naive Forecasts

A naive forecast is simply the most recent value of the series. That is, a forecast for time $t+k$ is simply the value at time t. For a time series with seasonality, a seasonal naive forecast can be generated. It is simply the last value in the season. For example, to forecast April 2001 for Amtrak ridership, we use the ridership of the most recent April, April 2000.

Naive and Seasonal Naive Forecast



Performance

- forecast::accuracy(naive.pred, valid.ts)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	2.45091	168.1470	125.2975	-0.3460027	7.271393	1.518906	-0.2472621
Test set	-14.71772	142.7551	115.9234	-1.2749992	6.021396	1.405269	0.2764480

- forecast::accuracy(snaive.pred, valid.ts)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	13.93991	99.26557	82.49196	0.5850656	4.715251	1.000000	0.6400044
Test set	54.72961	95.62433	84.09406	2.6527928	4.247656	1.019421	0.6373346

Generating Future Forecasts

Once a model has been determined, the training and validation sets are then recombined into a single set, and the model is rerun on this dataset. The advantage of doing this are:

- The validation set normally contains the most recent data
- With larger data, models might be more accurate
- If only training data is used, future forecasts are for a larger time period, than with the recombined data. This introduces greater variability and possibly inaccuracy.

Regression Based Forecasting

Trends

- Linear
- Exponential
- Polynomial

Seasonality

- Additive
- Multiplicative

Autocorrelation

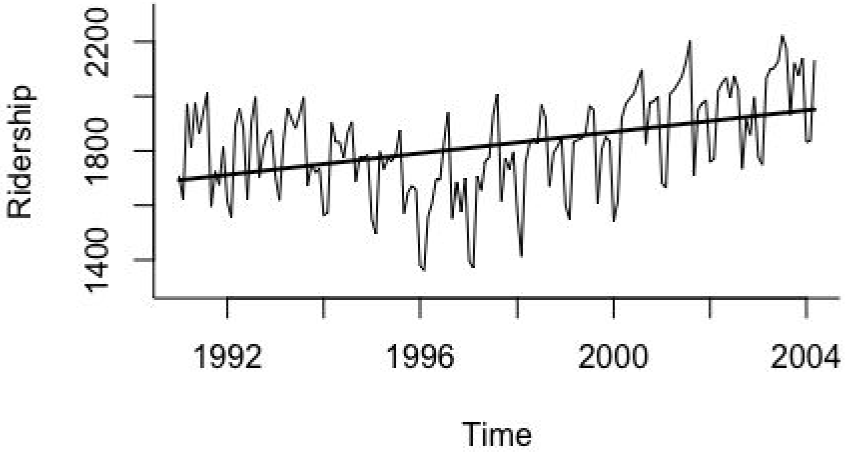
Trends

Linear Trend: Set the outcome variable Y as the time-series values, and X as the index of time.

Model: $Y_t = \beta_0 + \beta_1 t + \epsilon$

$\beta_0 = \text{Level}$
$\beta_1 = \text{Trend}$
$\epsilon = \text{Noise}$

Month	Ridership (Y_t)	t
Jan 91	1709	1
Feb 91	1621	2
Mar 91	1973	3
Apr 91	1812	4
May 91	1975	5
Jun 91	1862	6
Jul 91	1940	7



Validation

Call:

tslm(formula = train.ts ~ trend)

Residuals:

```
    Min      1Q  Median      3Q     Max
-411.29 -114.02   16.06  129.28  306.35
```

Coefficients:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1750.3595    29.0729  60.206   <2e-16 ***
trend          0.3514     0.4069   0.864     0.39
---
```

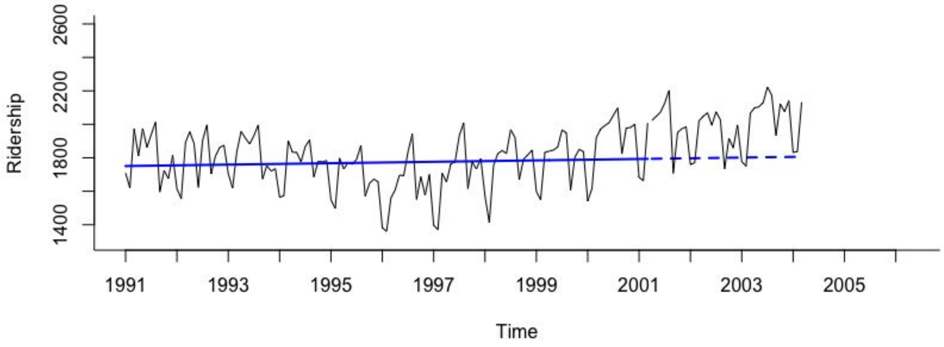
Signif. codes:

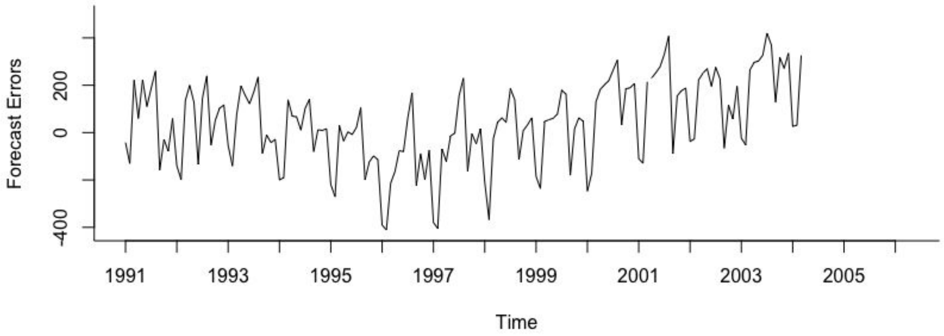
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 160.2 on 121 degrees of freedom

Multiple R-squared: 0.006125,	Adjusted R-squared: -0.002089

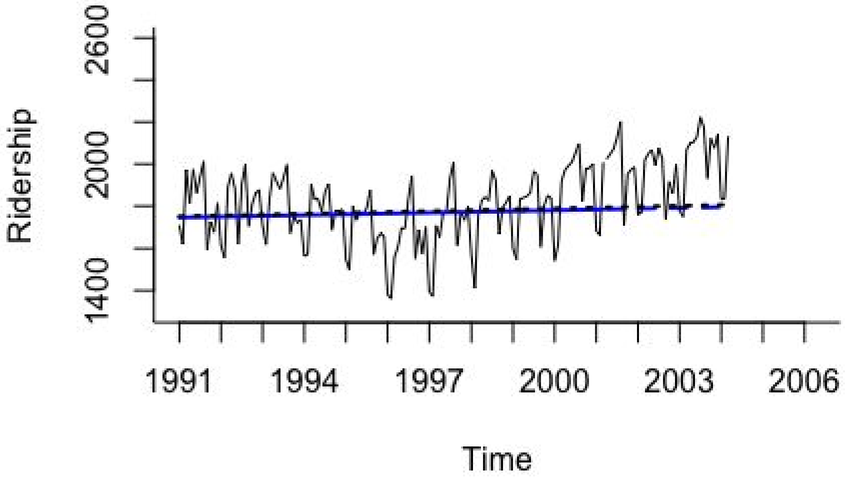
F-statistic: 0.7456 on 1 and 121 DF, p-value: 0.3896





Exponential Trend

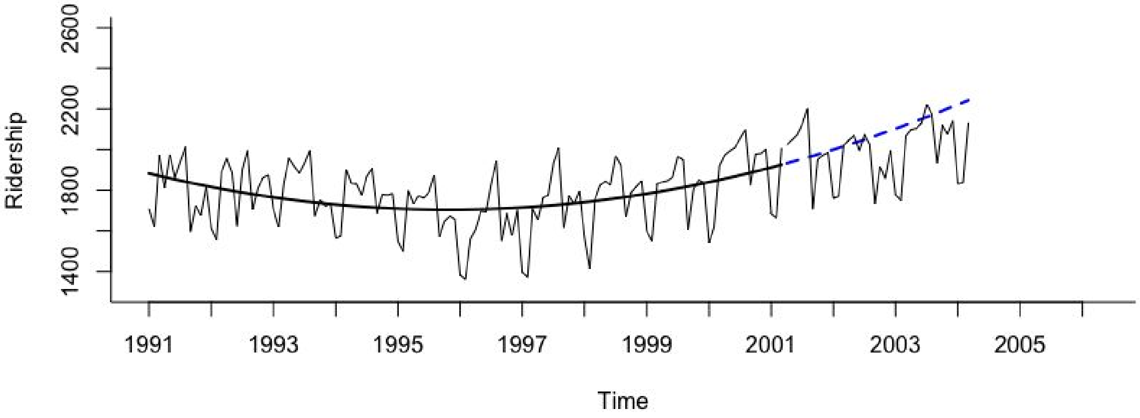
$$Y_t = e^{\beta_0 + \beta_1 t + \epsilon}$$
$$log(Y_t) = \beta_0 + \beta_1 t + \epsilon$$

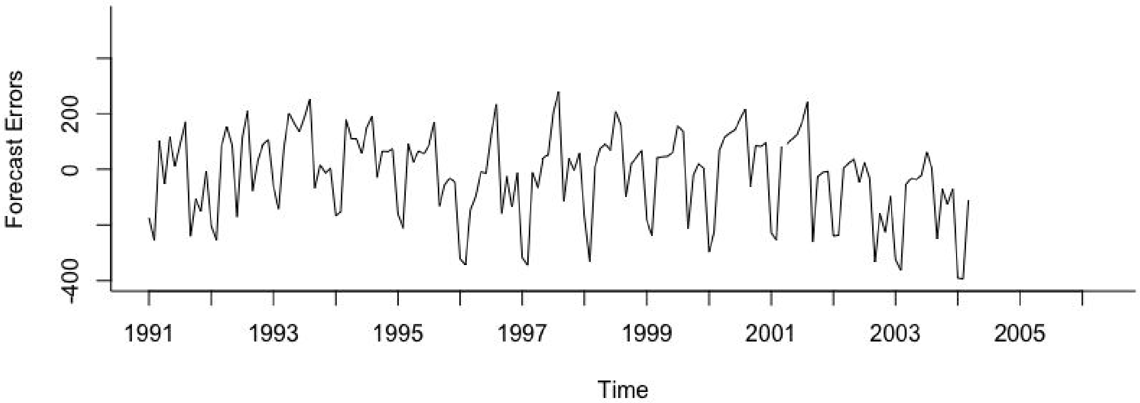


Polynomial Trend

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon$$

```
Call:
tslm(formula = train.ts ~ trend + I(trend^2))
Residuals:
    Min      1Q  Median      3Q     Max
-344.79 -101.86   40.89   98.54  279.81
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 1888.88401   40.91521  46.166  < 2e-16 ***
trend         -6.29780    1.52327  -4.134 6.63e-05 ***
I(trend^2)     0.05362    0.01190   4.506 1.55e-05 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 148.8 on 120 degrees of
freedom
Multiple R-squared:  0.1499,            Adjusted
R-squared:  0.1358
F-statistic: 10.58 on 2 and 120 DF,  p-value: 5.844e-05
```



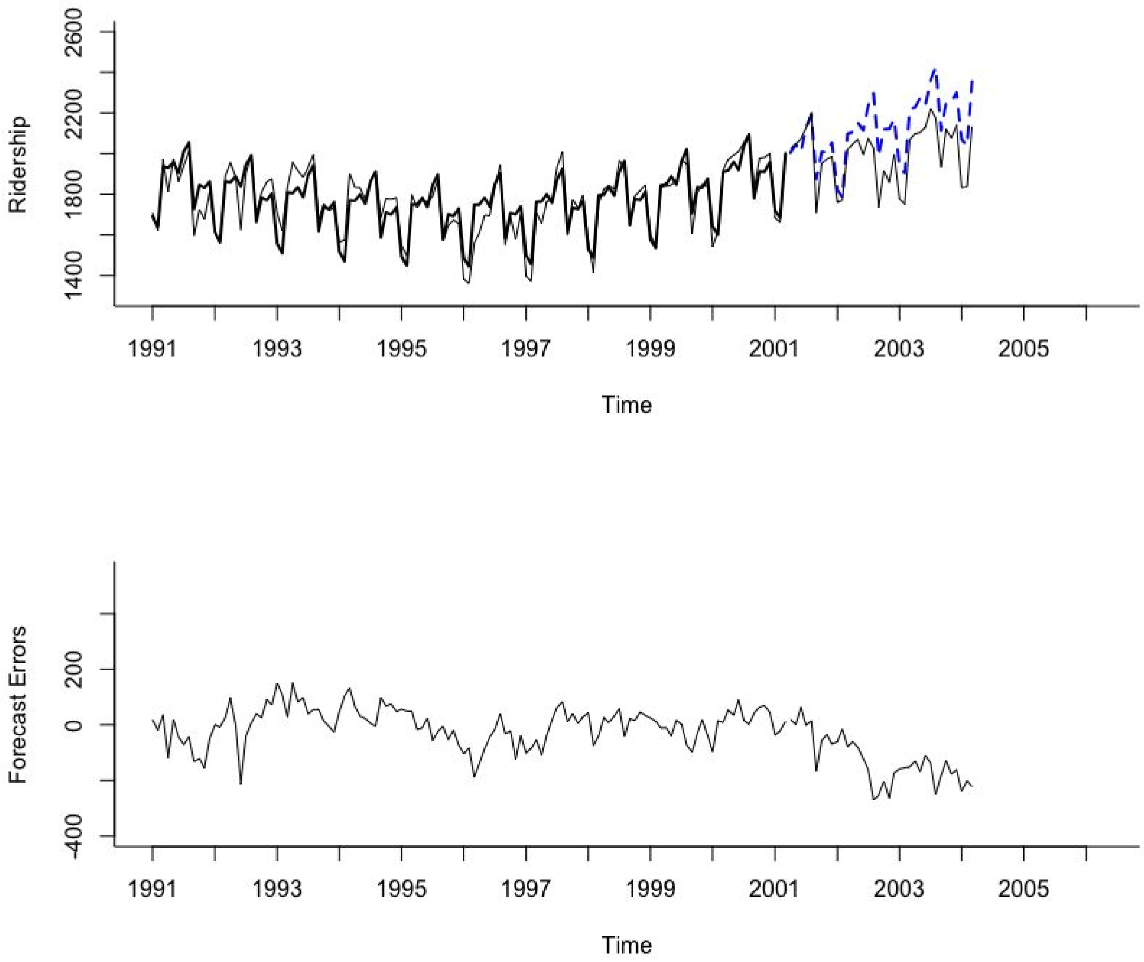


Seasonality

1. Seasonality means that values changes based on some seasons. The Amtrak data shows strong changes by month.
2. Create a new categorical variable for Seasons. Turn this into a dummy variable.
3. If Seasons has 12 categories, you will need 11 dummy variables
4. In R, function tslm() uses ts() which automatically creates the categorical Season column (called season) and converts it into dummy variables.

Month	Ridership	Season
Jan 91	1709	Jan
Feb 91	1621	Feb
Mar 91	1973	Mar
Apr 91	1812	Apr
May 91	1975	May
Jun 91	1862	Jun
Jul 91	1940	Jul

Results



Ridership
2600
2200
1800
1400
1991
1993
1995
1997
1999
2001
2003
2005
Time
Forecast Errors
200
0
-400
1991
1993
1995
1997
1999
2001
2003
2005
Time

Autocorrelation and ARIMA Models

Ordinary regression models, while accounting for trends and season, do not account for the dependence between periods. In regression, cross-sectional data, we assume that values in adjacent periods are independent of each other. This is not true with time series data. The strength of the relationship (correlation) between neighboring values is called *autocorrelation*.

Autocorrelation

Is the correlation calculated between values in a period, and the same set of values *lagged* by *k* periods. For example, the table below shows values lagged when *k = 1* and *k = 2*.

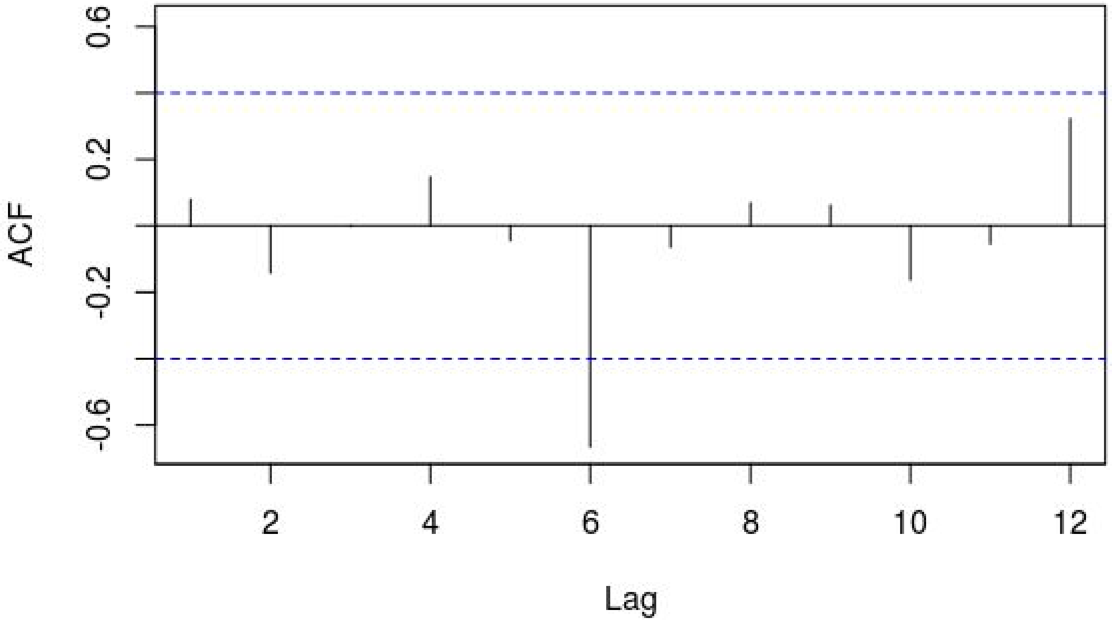
$$R(t) = E[(X_t - \mu_t)(X_{t+k} - \mu_{t+k})]/\sigma^2$$

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

Example

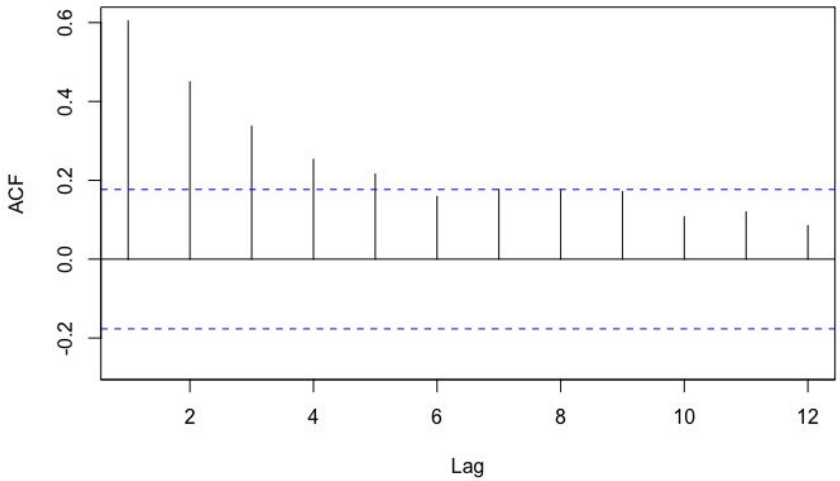
Here is the autocorrelation at different lags for a 24-month ridership (Amtrak data)

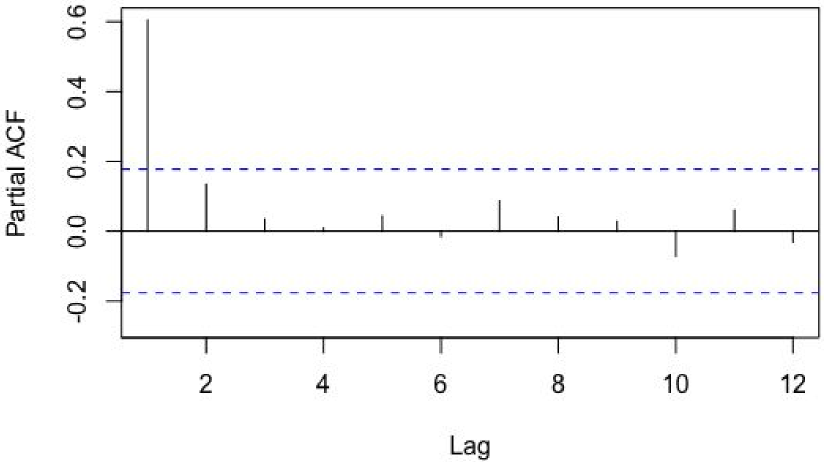
What do you notice?



Autocorrelation of Residuals

Did we correctly model the seasons? Here is the autocorrelation, and partial autocorrelation plots of the residuals from the model using season and quadratic trends.





Improving Forecasts - Using Autocorrelation

- Incorporate autocorrelation directly into the regression model
  - Autoregressive (AR) models
  - Autoregressive Integrated Moving Averages
- Construct a second-level forecasting model on the residual series

Box-Jenkins Methodology

This methodology for time series involves the following steps:

1. Condition data and select a model
   a. Identify and account for any trends or seasonality
   b. Examine the remaining time series and determine a suitable model
2. Estimate the model parameters
3. Assess the model, and return to Step 1 if necessary

ARIMA Model

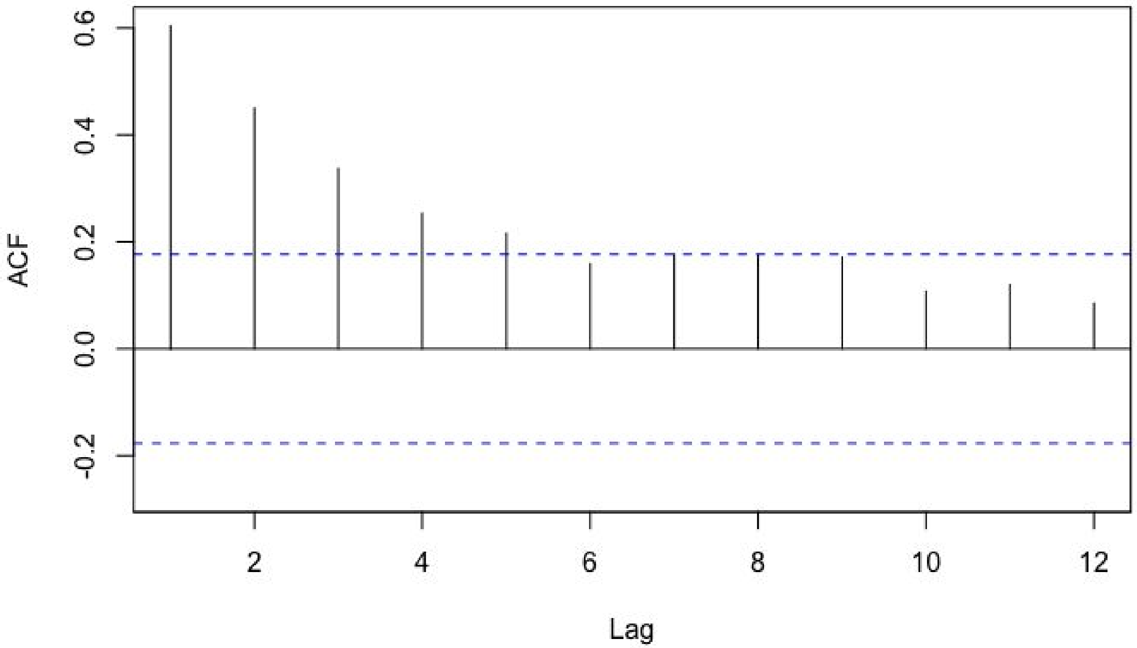
Data should be a stationary time series. Three conditions need to be met:

1. The expected value (mean) of Y_t is constant over time t.
2. The variance of Y_t is finite
3. The covariance of Y_t and Y_{t+h} depends only on the values of $h=0, 1, 2, \ldots$ for all t.

Essentially, remove all trends and seasonality from the data, thus hopefully leading to constant variance and fixed mean over the time series.

Autocorrelation Function

What provides information about the covariance of the variables in time series?



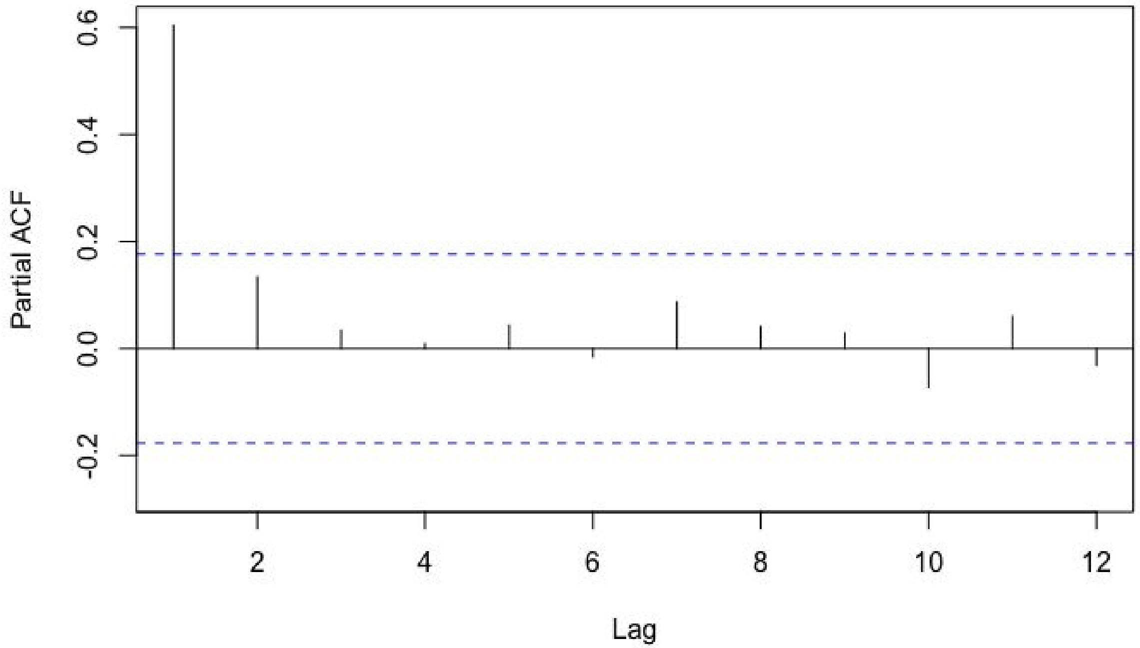
Autoregressive Models

For a stationary time series y_t *t = 1, 2, 3, …*an autoregressive model of order *p*, denoted as AR(p), is expressed as

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + \epsilon_t$$

Partial Autocorrelation Function

It can be shown that because of the dependent on prior values, even a simple AR(1) model can show significant autocorrelation for values of lags more than 1. Thus, sometimes, we use the Partial Autocorrelation Function (PCH) which shows the correlation after removing all effects between variables for lags under than 1.



Moving Average Models

For a time series y_t, centered around 0, a moving average model of order *q*, denoted MA(q) is

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q}$$

In MA(q), the value of the time series is a linear combination of the current white noise term and the prior *q* white noise terms.

ARMA and ARIMA

- No need to choose between AR(p) and MA(q) models
- ARMA(p, q) combines both into a single model

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + ... + \beta_p Y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q}$$

The ARMA (p, q) is valid only when applied to a stationary time series. However, many time series exhibit trend over time. Some approaches are:

- Run a regression, and then subtract the value of the fitted regression line from each y value
- Compute the difference between success y values. This is known as *differencing*.

Differencing

$d_t = y_t - y_{t-1}$ for t = 2, 3, ..., n

Differencing can also be applied multiple times to get a stationary time series. Once we have a stationary time series, we can then apply an ARMA (p, q) model.

ARIMA (p, d, q)

The Autoregressive Integrated Moving Average model is similar to the structure of the ARMA model, with the ARMA (p, q) model applied to the time series y_t after applying differencing d times.

Seasonal ARIMA (p, d, q) Model

We can also account for seasonal patterns by using a seasonal autoregressive integrated moving average model, denoted by ARIMA (p, d, q) X (P, D, Q), where

- p, d, q are as before
- s denotes the seasonal period
  - 52 for weekly data; 12 for monthly data; 7 for daily data
- P is the number of terms in the AR model across s periods
- D is the number of differences applied across s periods
- Q is the number of terms in the MA model across s periods

Example

For a large country, the monthly gasoline production measured in millions of barrels has been obtained for the past 240 months. A market research firm requires some short-term gasoline production forecast to assess the industry’s ability to deliver future gasoline supplies and the effect on gasoline prices.

https://github.com/mshanker1/60095_201810/blob/master/Gas_prod.nb.html

Comparing Fitted Time Series Models

There are several measures that are used for comparing fitted models. These include:

- AIC
- AICc
- BIC

Choose the fitted model with the smallest AIC, AICc, or BIC values. But, our real interest is in forecasting....

Fitting AR Models to Residuals

- ARIMA modeling is less robust and requires significant expertise
- AR models applied to residuals provides a clearer approach to forecasting, and can significantly improve short-term forecasts

Approach

1. Generate a k-step-ahead forecast of the the series (F_{t+k}) using any method
2. Generate a k-step-ahead forecast of the residual (E_{t+k}) using an AR model
3. Improve the initial k-step-ahead forecast of the series by adjusting it according to its forecasted error as $F_{t+k}^* = F_{t+k} + E_{t+k}$

Fitting an AR Model to the Residuals

1. Examine the autocorrelation of the residuals
2. Choose the AR model according to the lags
3. Usually an AR(1) model will suffice