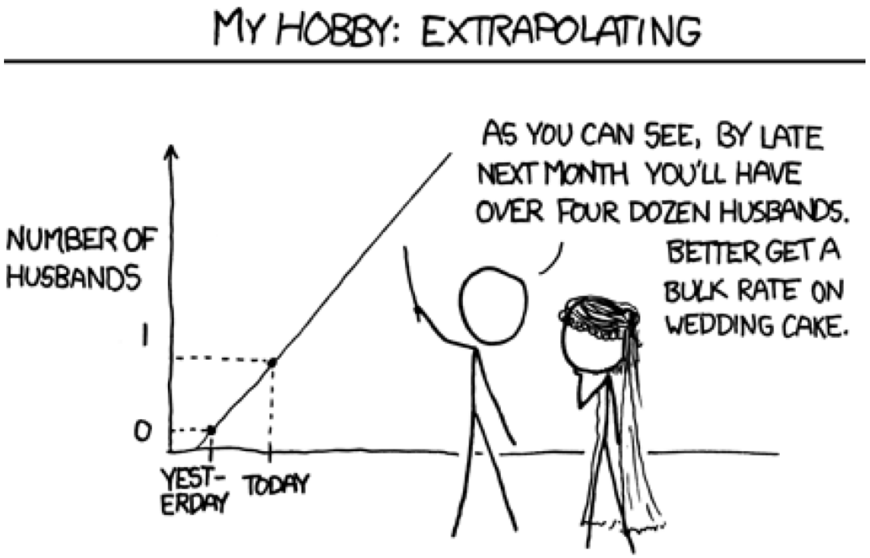
Time Series (TS) Forecasting



How meaningful is your forecasting?

Nagiza F. Samatova, samatova@csc.ncsu.edu
Professor, Department of Computer Science
North Carolina State University

Senior Scientist, Computer Science & Mathematics Division
Oak Ridge National Laboratory

Oak Ridge National Laboratory
NC STATE UNIVERSITY Department of Computer Science

Outline

- Motivational Use Cases
  - Packages for TS data analysis
  - Data Sets for TS analysis and forecasting
- TS Forecasting Problem Statement & Baseline Methods
  - Average
  - Naive
  - Seasonal Naive
  - Drift
- TS Systematic and Non-Systematic Parts
  - Systematic: Level, Trend, Seasonality, Cyclicity, Autocorrelation
  - Non-Systematic: Noise
- Autocorrelation and Seasonality Detection
  - Autocorrelation: Definition, metrics, correlogram
  - Seasonality and Trend detection with Acf analysis
  - Residuals quantification with Acf

Overall Learning Objectives for TS Forecasting

- **Introduce the principles of forecasting**
- **Learn how to use forecasting effectively**
- **Learn when to use forecasting**

Recommended Resources

- Books
  - Free and online (otexts.com/fpp): Forecasting Principles & Practice by R. Hyndman, G. Athanasopoulos ← Excellent Book!!!
  - Practical Time Series Forecasting with R: A Hand-on Guide by Shmueli & Lichtendahl
- Packages
  - R: fpp (install.packages ("fpp", dependencies=TRUE))s

Get Familiar with the Package

```r
install.packages("fpp", dependencies=TRUE)
library(fpp)

help.search("forecasting")
help(forecast)
example("forecast.ar")

# similar names
apropos("forecast")

help(package="fpp")
```

Dependency Packages for "fpp"

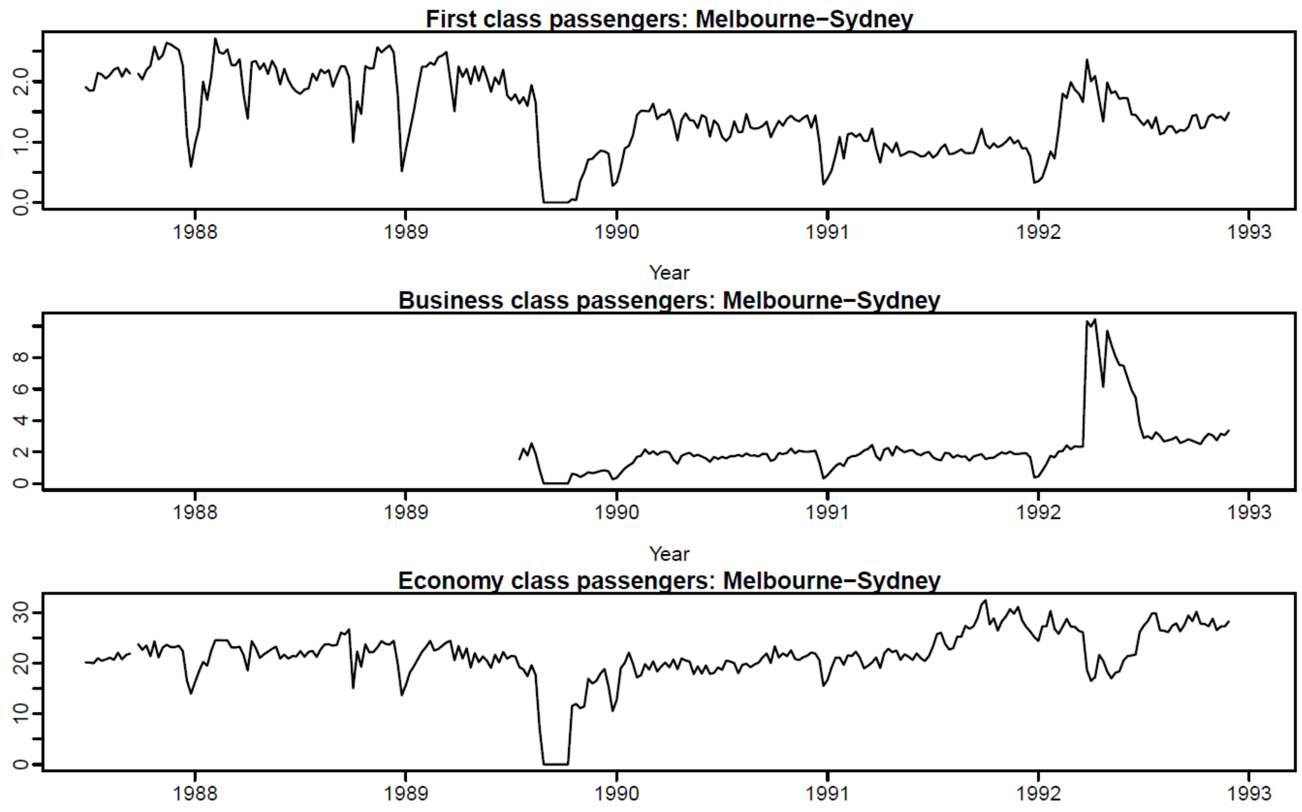
- **library (fpp): will load the following**
  - data sets and examples
  - **forecast** package: forecasting functions
  - **tseries** packages: some time series functions
  - **fma** package: lots of time series data
  - **expsmooth** package: ts data
  - **lmtest** package: for some regression functions

- **data (package="fpp"): show available ts data**

- **data (package="fma"): show available ts data**

Use Case: Airline Passenger Traffic Forecast

- **Problem: Forecast passenger traffic on major airlines**



- Large amount of data on previous routes is available.
- Traffic is affected by school holidays, special events, advertising campaigns, competition, etc.

Other Use Cases for TS Forecasting

- Daily Stock Prices (e.g., Dow Jones Index)
- Monthly, quarterly and annual profits
- Monthly, quarterly and annual product demands
- Quarterly beer production
- Monthly rainfall
- Monthly residential electricity sales

- **library (fpp): will load the following**
- **data (package="fpp"): show available ts use cases**
- **data (package="fma"): show available ts use cases**

TS Forecasting

PROBLEM & BASELINE METHODS

Time Series Forecasting: Problem

- Time series:
  - A sequence(s) of observations collected over time.
- Assumptions:
  - The time periods are equally spaced (e.g., not always true: Bitcoin data was changed to fit this assumption).

Forecasting is estimating how the sequence of observations will continue into the future.

Basic Notation

Symbol	Definition
$\boldsymbol{t = 1, 2, 3, \dots,}$	An index for the time period of interest; e.g., for a *daily* time period, $t = 1$ means day 1, $t = 2$ means day 2, etc.
$\boldsymbol{y_1, y_2, \dots, y_T}$	A series of T values measure over T time periods; e.g., for the annual average stock price, y_1 denotes the price for year 1, y_2 denotes the price for year 2, etc.
$\boldsymbol{F_t}$ or $\boldsymbol{\hat{y}_t}$	The forecast value for time period t
$\boldsymbol{F_{t+k}}$ or $\boldsymbol{\widehat{y_{t+k}}}$	The k-step-ahead forecast when forecasting time is t; e.g., F_{t+1} is the forecast for time period $(t + 1)$ made during the time period t
$\boldsymbol{e_t = y_t - F_t}$	The forecast error for time period t

Baseline: Simple Forecasting Methods

- **Average**: meanf (ts.data, h=20)
  - Forecast of all future values is the mean of historical data $\{y_1, \dots, y_T\}$
  - $F_{T+h} = \hat{y}_{T+h} = \bar{y} = (y_1 + \cdots + y_T)/T$

- **Naive:** naive (ts.data, h=20) or rwf (ts.data, h=20)
  - Forecast is equal to the last observed value
  - $F_{T+h|T} = \hat{y}_{T+h|T} = y_T$

- **Seasonal naive:** snaive (ts.data, h=20)
  - Forecast is equal to the last value from the same season
  - $\hat{y}_{T+h|T} = y_{T+h-km}$, where m is the seasonal period and $k = round\left(\frac{h-1}{m}\right) + 1$

- **Drift:** rwf (ts.data, drift=TRUE, h=20)
  - Forecast is equal to the last value plus the average change
  - Equivalent to extrapolating a line between the first and last observation
  - $F_{T+h|T} = \hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + \frac{h}{T-1}(y_T - y_1)$

Ex: Show Baseline Forecasts for TS Data

- library(fpp)
- data (package="fma")
- data (package="fpp")

- ts.data <- data (beer)

Visual Analysis of Time Series (TS) Data

- **Time plots: plot or plot.ts (e.g., plot (a10))**
- **Seasonal plots: seasonplot (e.g., seasonplot(a10))**
  - Data from each season is overlapped.
  - To view the underlying seasonal patterns.
- **Seasonal subseries plots: monthplot (e.g., monthplot(a10))**
  - Data for each season collected together in time plot as separate time series.
  - To view the underlying seasonal patterns and the changes in seasonality over time
- **Lag plots: lag.plot**
- **ACF plots: Acf**

```
beer <- window(ausbeer,start=1992)

plot(beer)

seasonplot(beer,year.labels=TRUE)

monthplot(beer)
```

Ex: Visually Explore Different TS Data Sets

- library(fpp)
- data (package="fma")
- data (package="fpp")

- ts.data <- data (beer)
- ***plot***(ts.data)

Time Series

SYSTEMATIC VS. NON-SYSTEMATIC PARTS

TS Parts: Systematic vs Non-systematic

TS Part	Definition	Detection	How to deal w/
Level	Average value of ts		
Trend	Long-term increase decrease in the data	lag.plot	De-trend via lag-1 differencing
Seasonality	Variations occurring during known periods of the year (monthly, quarterly, holidays)	lag.plot, Acf plots	De-seasonalize via lag-k differencing
Cycles	Other oscillating patterns about the trend (e.g., business or economic conditions)		
Auto-correlation	Correlation between neighboring points in ts	Acf, lag.plot	
Noise	Residuals after level, trend, seasonality, and cycles are removed	Normality tests	

Non-systematic Part: Noise, or Residuals

Residuals: difference between observed value & its forecast based on all previous observations: $e_t = y_t - \hat{y}_{t|t-1}$

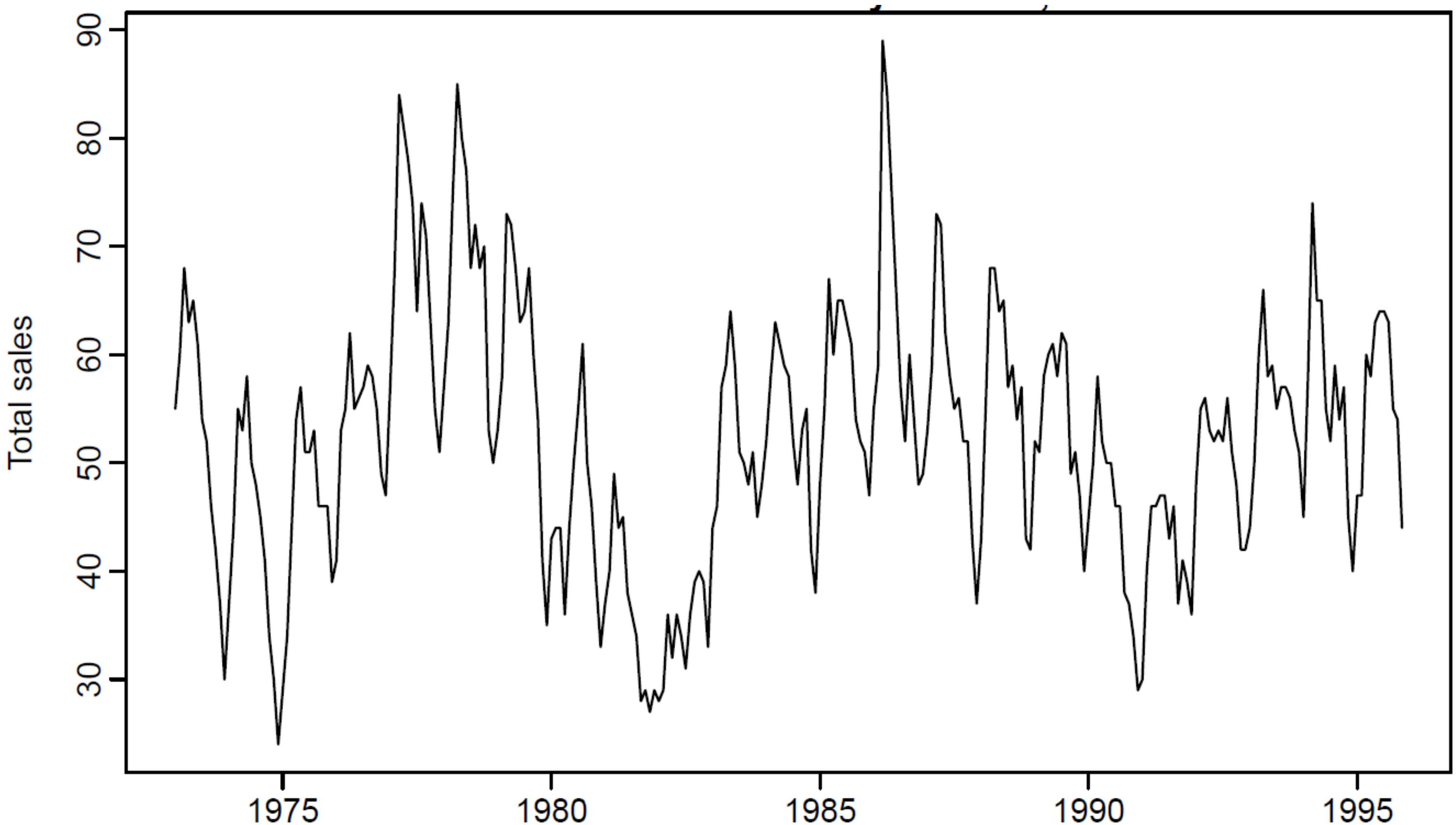
- Assumptions
  - $\{e_t\}$ are uncorrelated: otherwise, information is left in residuals that should be used in computing forecasts
  - $\{e_t\}$ have zero means; otherwise, forecasts are biased
  - $\{e_t\}$ have constant variance
  - $\{e_t\}$ are normally distributed

Seasonal vs. Cyclic Patterns

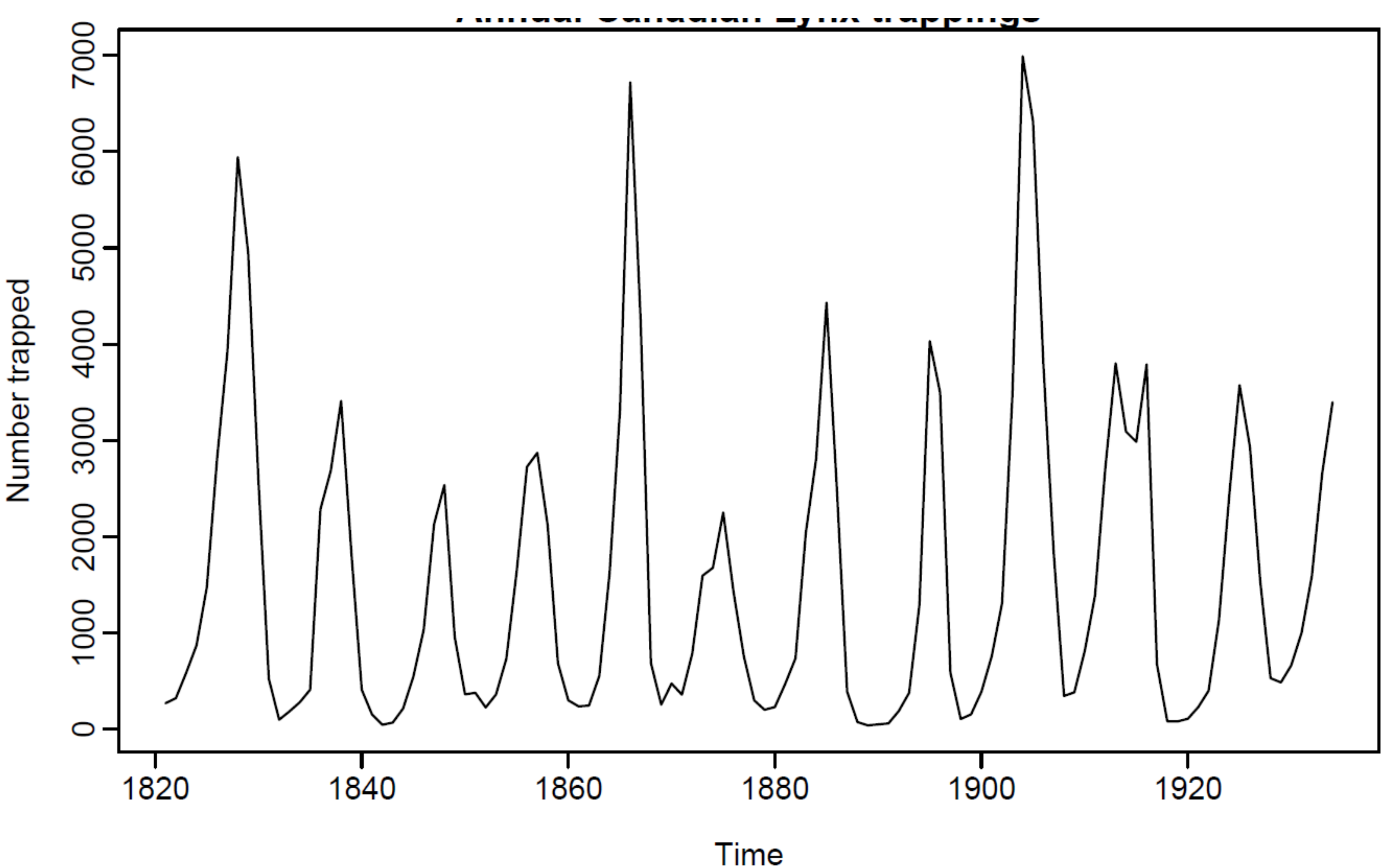
- Seasonal pattern exists when a time-series is influence by seasonal factors
  - quarter of the year, month, day of the week, holidays
- Cyclic patterns exist when data exhibit rises & falls that are not of fixed period
  - usually of at least 2 years

Pattern	Seasonal	Cyclic
Length	Constant length	Variable length
Average length	Shorter	Longer
Magnitude	Magnitude of seasonal pattern is less variable	More variable
Predictability of peaks and troughs	Predictable	Hard to predict for long-term cyclic data

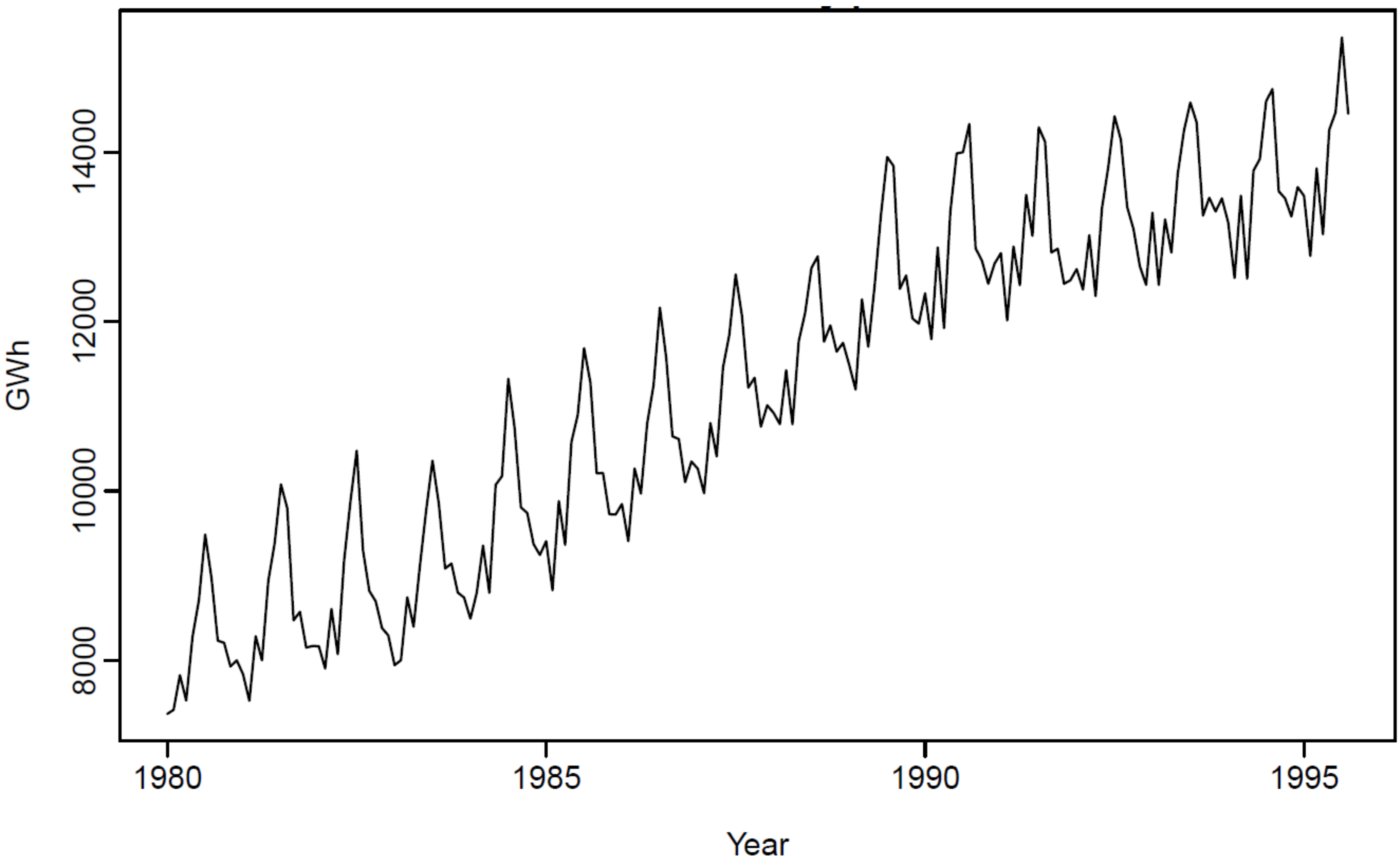
Cyclic: Sales of new one-family houses in US



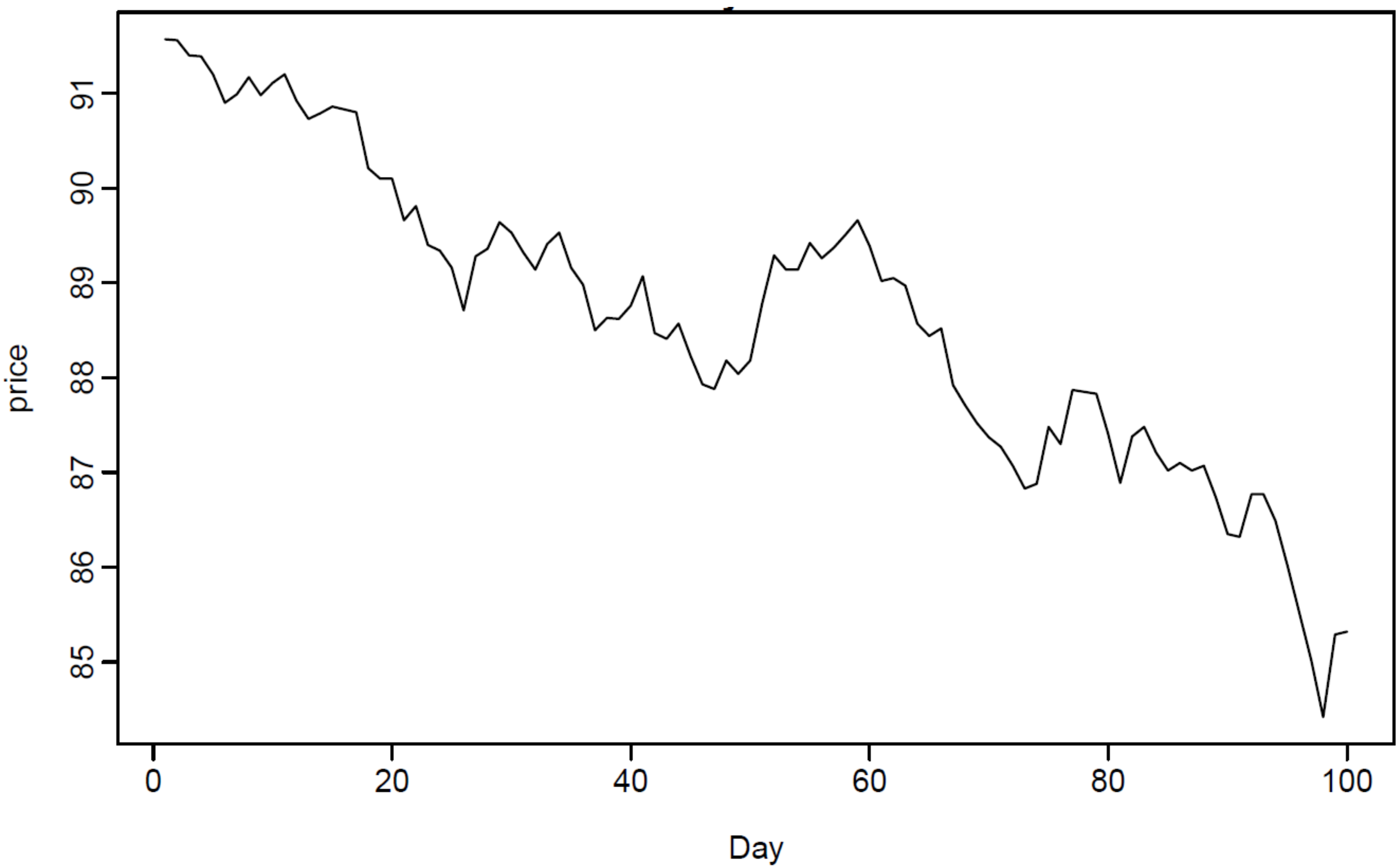
Seasonal: Annual Lynx Trappings in Canada



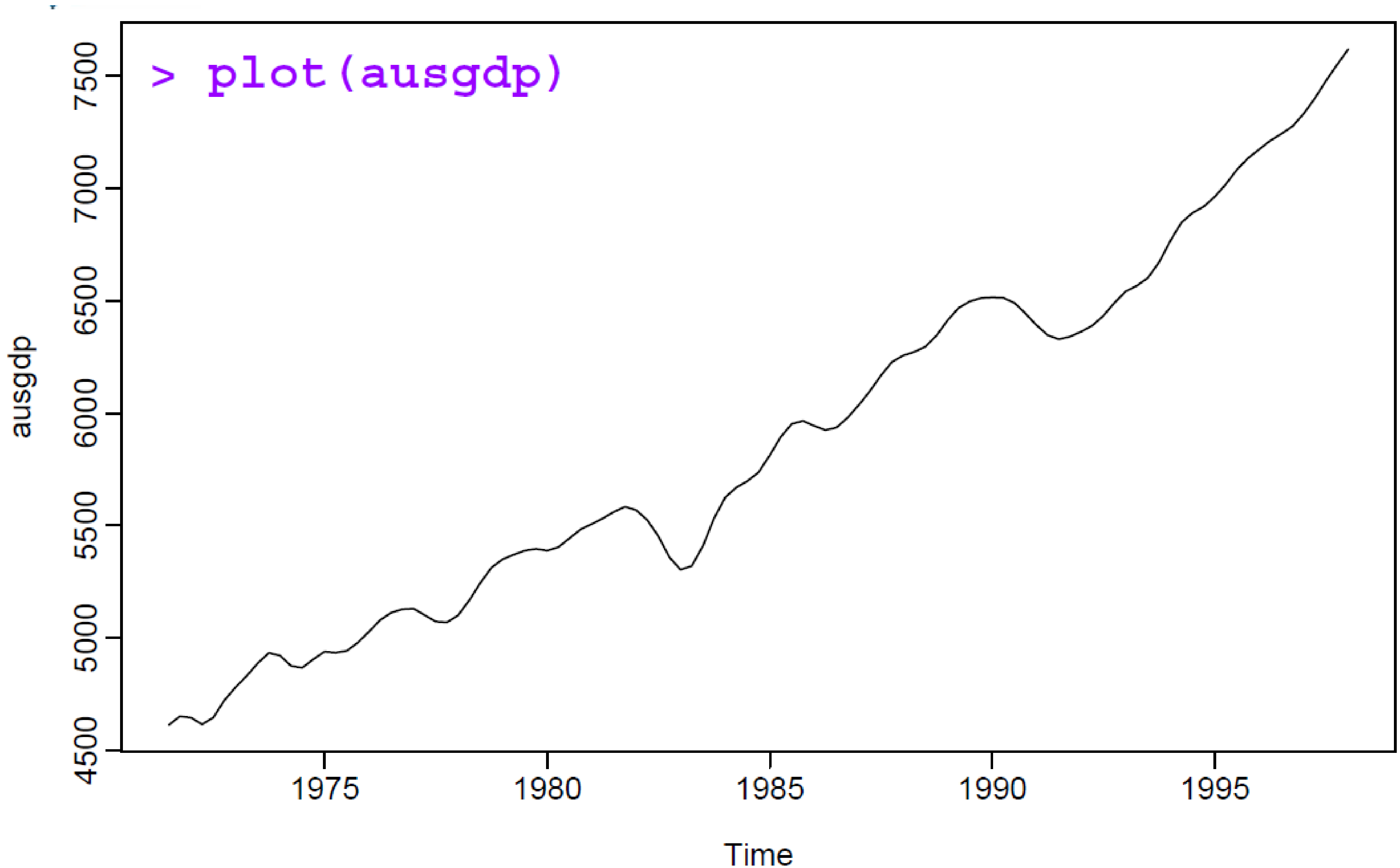
Trend & Seasonality: Electricity Production



Trend Pattern: US Treasury Bill Contracts



Upward Trend: Australian GDP



Additive and Multiplicative TS Components

- A time series with additive components can be modeled as:

$$y_t = Level + Trend + Seasonality/Cycles + Noise$$

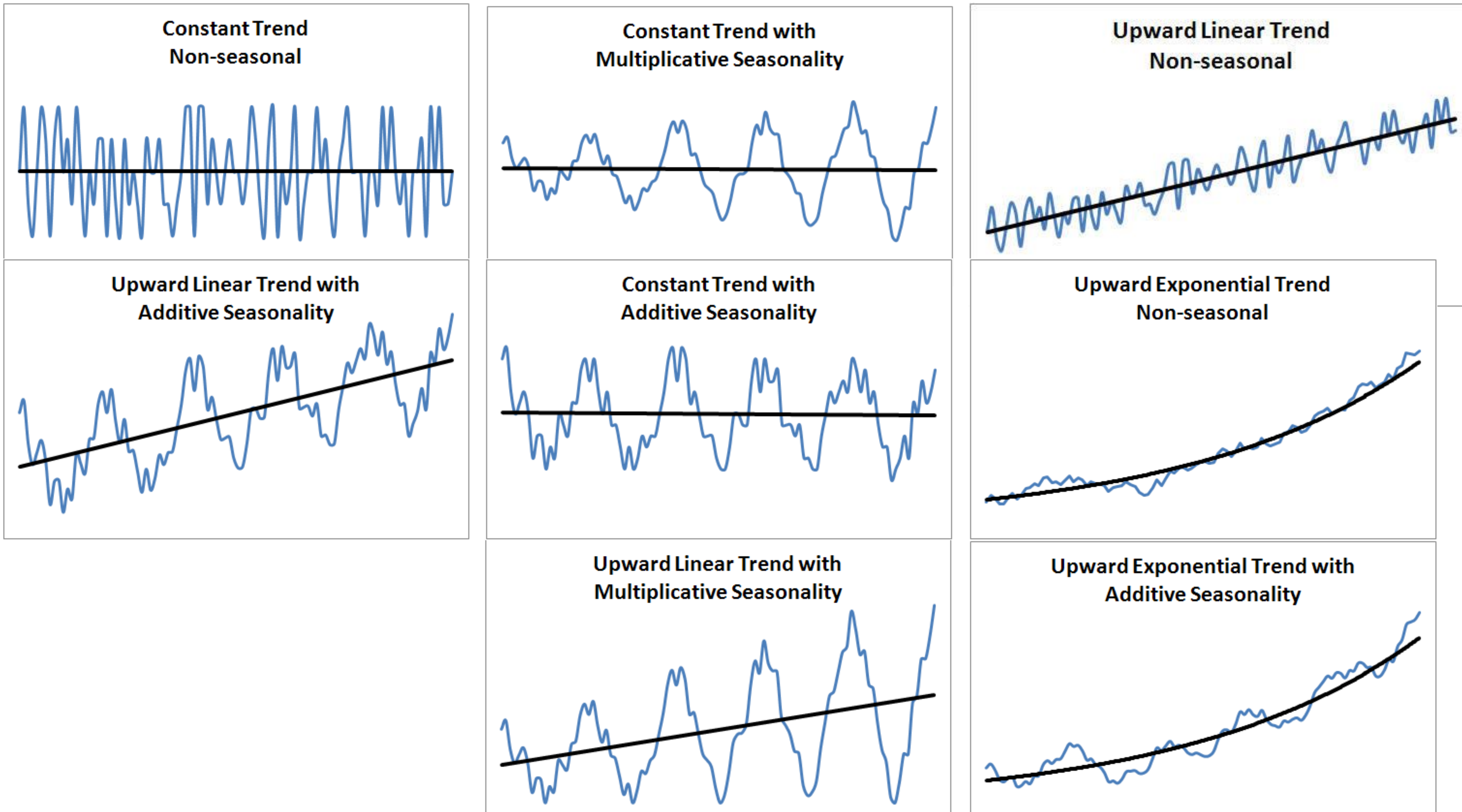
- A time series with multiplicative components is modeled as:

$$y_t = Level \times Trend \times Seasonality/Cycles \times Noise$$

- Forecasting methods attempt to isolate the systematic part and quantify the noise level.
  - The systematic part is used for generating point forecasts
  - The level of noise helps assess the uncertainty associated with the point forecasts

Trend & Seasonality Patterns

- **Trend patterns are commonly approximated by linear, exponential and other mathematical functions**



http://jcflowers1.iweb.bsu.edu/rlo/trends.htm

Seasonality Detection

AUTO-CORRELATION

Autocorrelation: Motivation

- When we use linear regression for time series forecasting, we are able to account for patterns such as trend and seasonality.
- However, ordinary regression models do not account for dependence between observations, which in cross-sectional data is assumed to be absent.
- Yet, in the time series context, observations in neighboring periods tend to be correlated.
- Such correlation or autocorrelation, is informative and can help in improving forecasts.
- How do you compute autocorrelation and how do you best utilize the information for improving forecasting?

Autocorrelation

Autocorrelation: measures linear relationship between lagged values of the time series

- Lag-1: y_t and y_{t-1}
- Lag-2: y_t and y_{t-2}

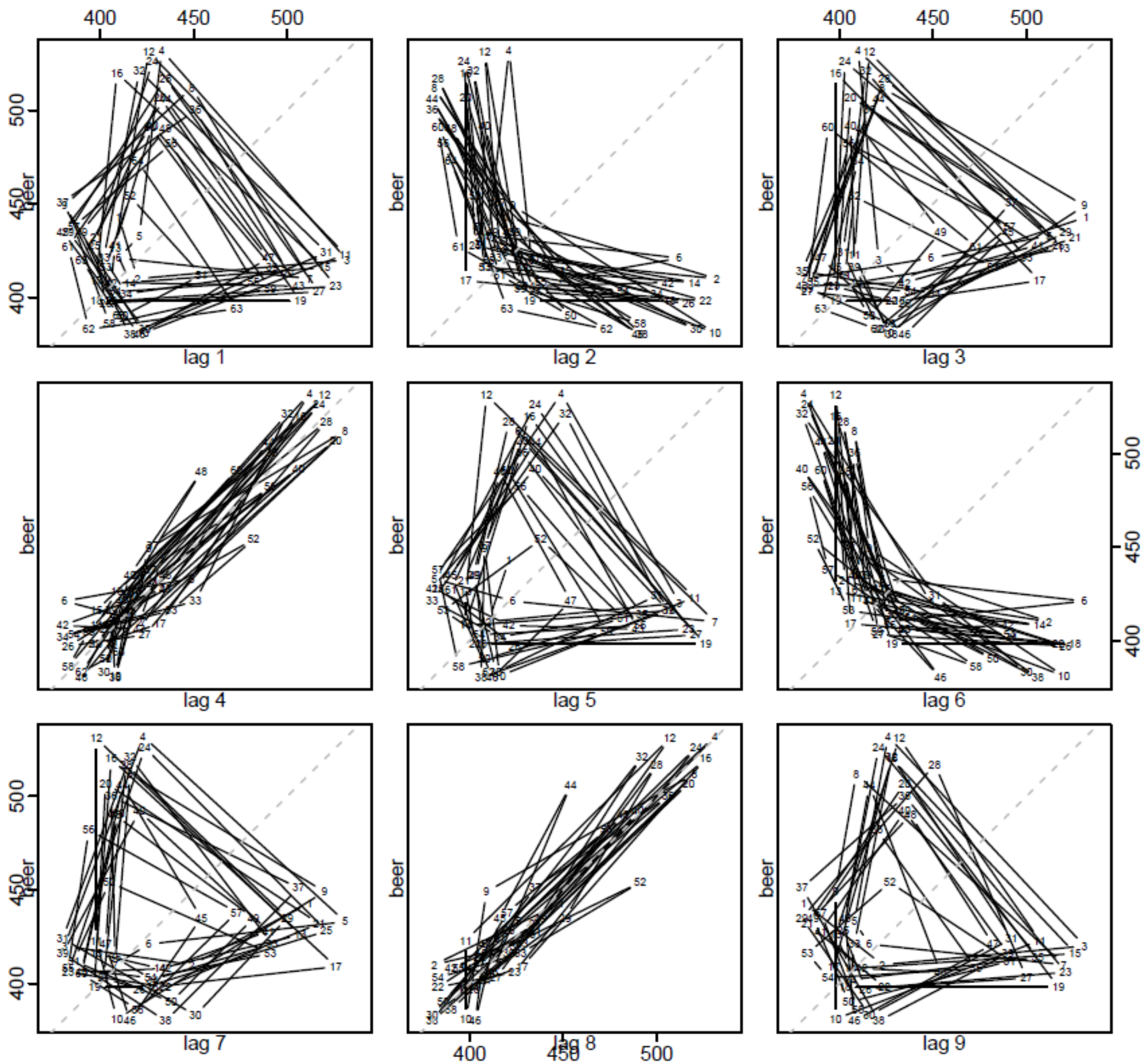
- Lag-6: y_t and y_{t-6}

Computing Autocorrelation

- **Correlation between values of a time series in neighboring periods is called autocorrelation because it describes a relationship between the series and itself.**
- **To compute autocorrelation, we compute the correlation between the series and a lagged version of the series.**
  - **A lagged series is a copy of the original series which is moved forward one or more time periods.**
  - **A lagged series with lag-1 is the original series moved forward one time period. A lagged series with lag-2 is the original series moved forward two time periods, and so on.**
- **The lag-1 autocorrelation is the correlation that**
  - **measures the linear relationship between values in consecutive time periods**
  - **is computed as correlation between the original series and the lag-1 series**

Linear Relationship between Lagged Values

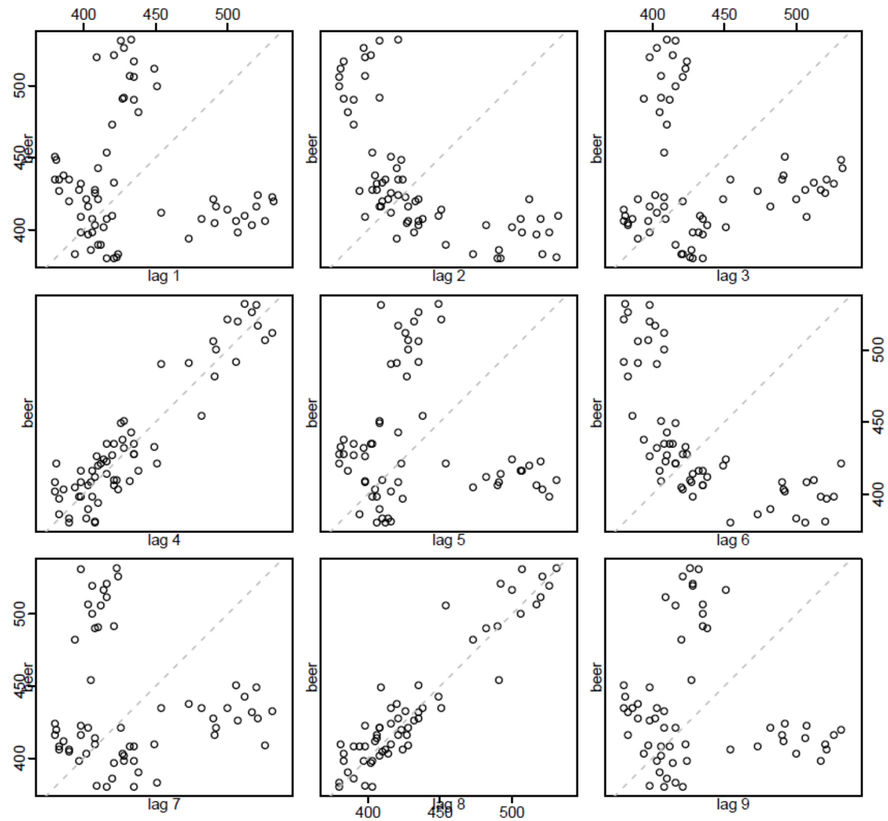
```
> lag.plot(beer,lags=9)
```



Lagged Scatterplots

- Each graph shows y_t plotted against y_{t-k} for different values of k.
- The autocorrelations are the correlations associated with these scatterplots.

```
> lag.plot(beer,lags=9,do.lines=FALSE)
```



Autocorrelation and Autocovariance

- **Autocovariance at lag k**

$$c_k = \frac{1}{T}\sum_{t=k+1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y})$$

- **Autocorelation at lag k**

$$r_k = c_k / c_0$$

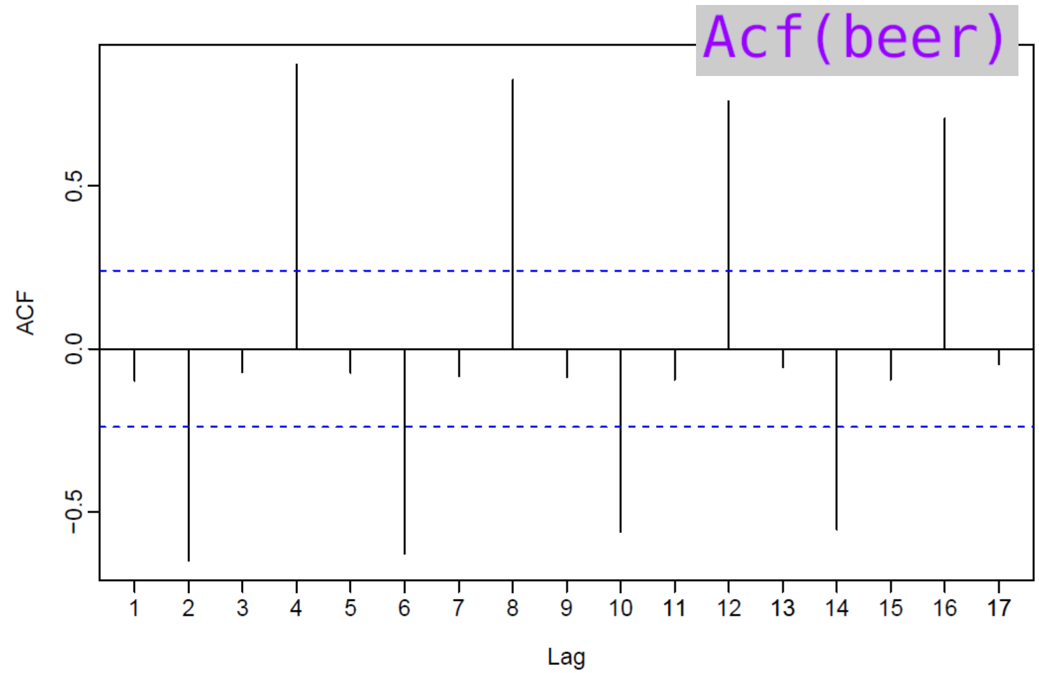
  - r_1 indicates how successive values of y relate to each other
  - r_2 indicates how y values two periods apart relate to each other
  - r_k is like the correlation between y_t and y_{t-k}

Example: Autocorrelation & Correlogram

Results for first 9 lags for beer data:

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
−0.126	−0.650	−0.094	0.863	−0.099	−0.642	−0.098	0.834	−0.116

- r_4 higher than for the other lags:
  - due to the seasonal pattern in the data: the peaks tend to be 4 quarters apart and the troughs tend to be 2 quarters apart.
- r_2 is more negative than for the other lags
  - because troughs tend to be 2 quarters behind peaks



Detecting Trend & Seasonality via ACF

- If there is seasonality, the **ACF at the seasonal lag** (e.g., 12 for monthly data) will be **large** and **positive**:
  - For seasonal ***monthly*** data, a large ACF value will be seen at lag 12 and possibly also at lags 24, 36, . . .
  - For seasonal ***quarterly*** data, a large ACF value will be seen at lag 4 and possibly also at lags 8, 12, . . .
- The **slowly decaying ACF indicates trend**

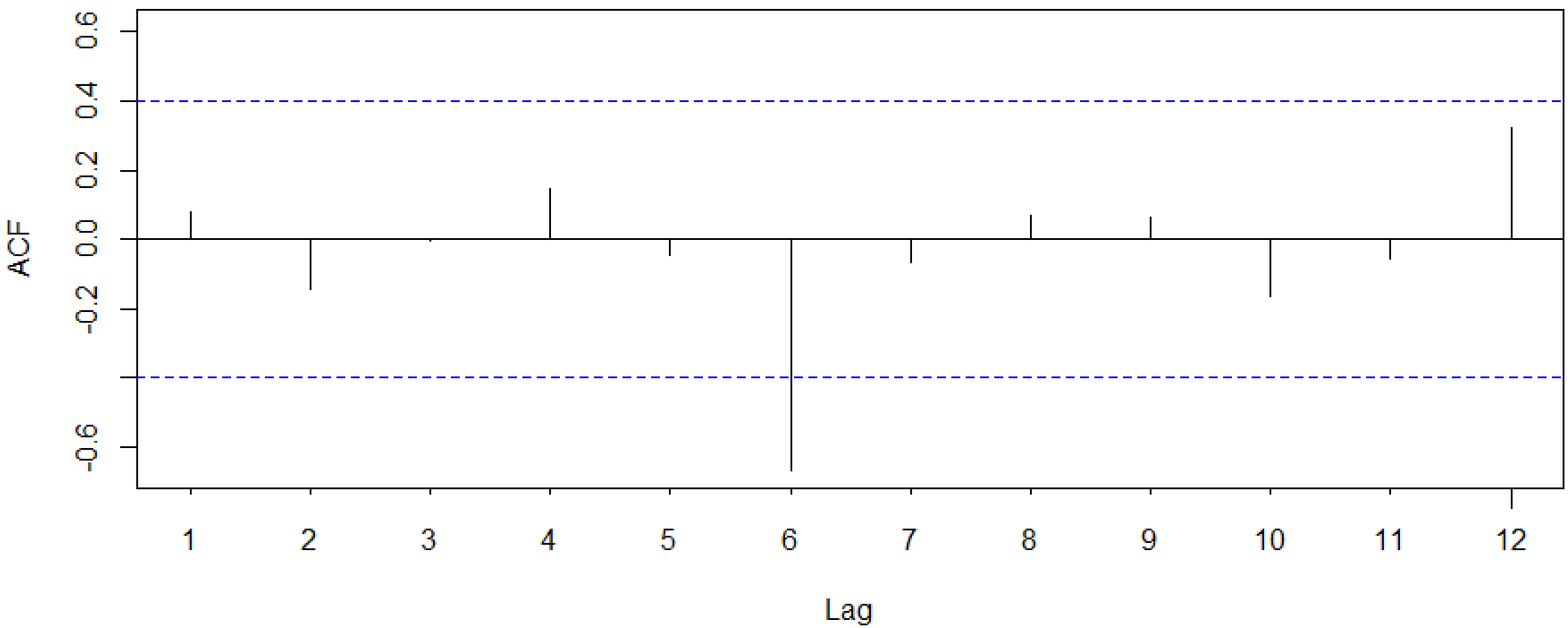
Autocorrelation with Acf

- In R forecast package **Acf()** function is used to compute and plot autocorrelations of a series at different lags

```
library("forecast")
library("zoo")

Amtrak.data <- read.csv("Amtrak data.csv")
ridership.ts <- ts(Amtrak.data$Ridership, start = c(1991, 1), end = c(2004, 3), freq = 12)
ridership.24.ts <- window(ridership.ts, start = c(1991, 1), end = c(1991, 24))

Acf(ridership.24.ts, lag.max = 12, main = "")
```



Autocorrelation Interpretation

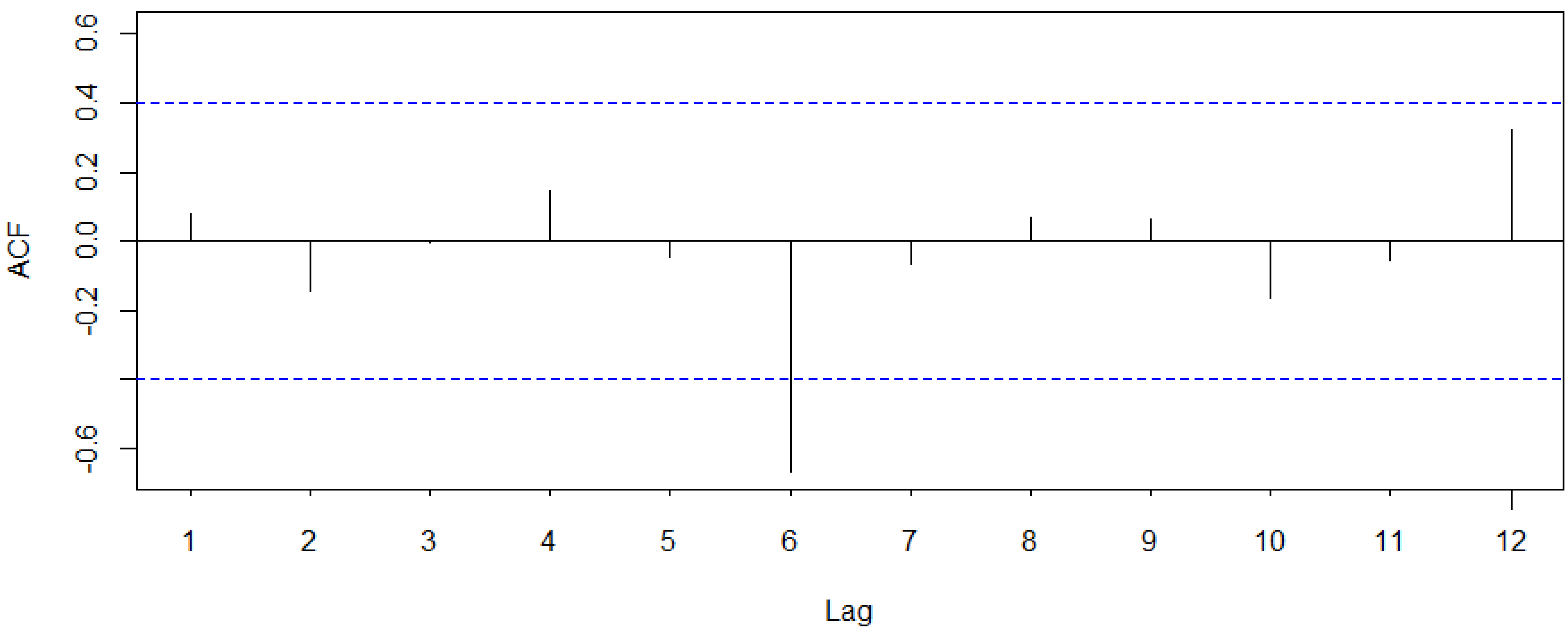
- Strong autocorrelation (positive or negative) at a lag larger than 1 typically reflects a cyclical pattern.
  - strong positive autocorrelation at lag-12 in monthly data reflects an annual seasonality where values during a given month each year are positively correlated

- Positive lag-1 autocorrelation (called "stickiness") describes a series where consecutive values move generally in the same direction.
  - In the presence of a strong linear trend, we would expect to see a strong and positive lag-1 autocorrelation

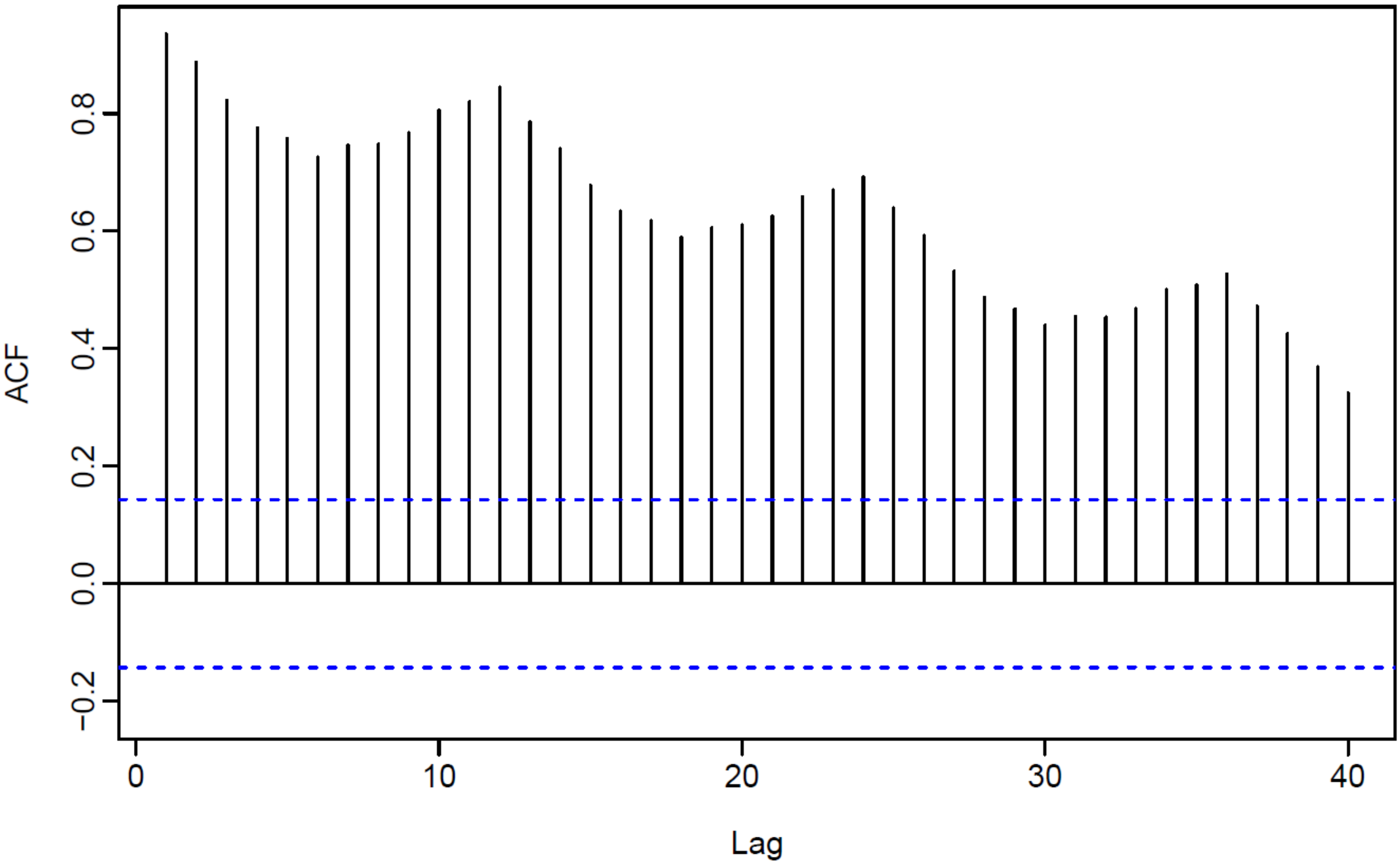
- Negative lag-1 autocorrelation reflects swings in the series, where high values are immediately followed by low values and vice versa

Autocorrelation Detection of Seasonality Patterns

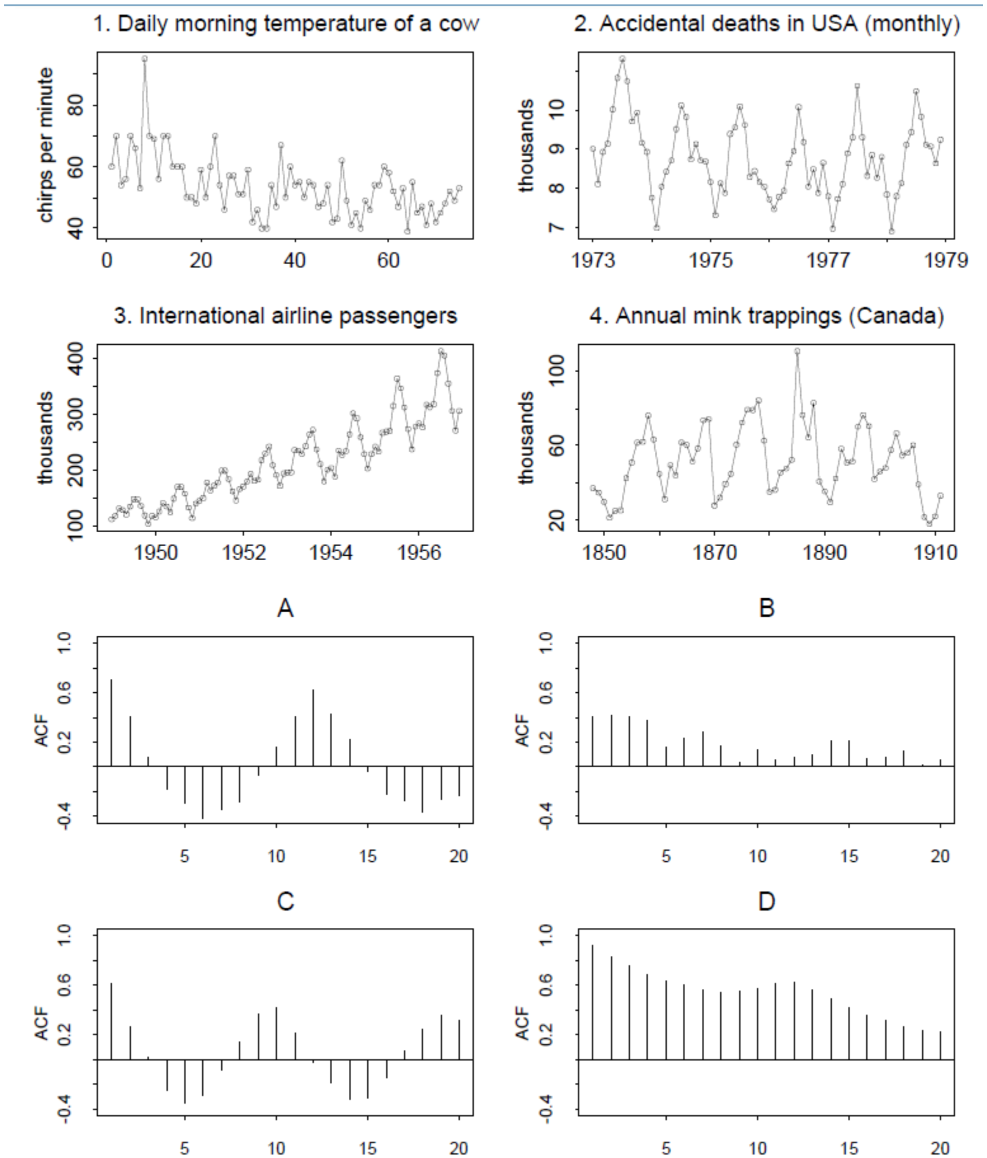
- **The strongest negative correlation at lag-6 indicates a biannual pattern in ridership, with 6-month switches from high to low**
  - **high summer and low-winter patter**



Ex: What ACF says about Monthly Electricity Production in Australia?



What ACF tells us about TS data sets?



Autocorrelation: Residual Series

- It is useful to look at autocorrelations of residual series.
  - e.g. after fitting a regression model we can examine the autocorrelation of the series of residuals
- If we adequately modeled the seasonal pattern, then the residual series should show no autocorrelation at the season's lag.
  - 6 month and 12 month cyclical behavior is not there in the residual series from the regression with seasonality and quadratic trend