

# Timeseries analysis with R



**JIGSAW ACADEMY**

Analytics for Professionals

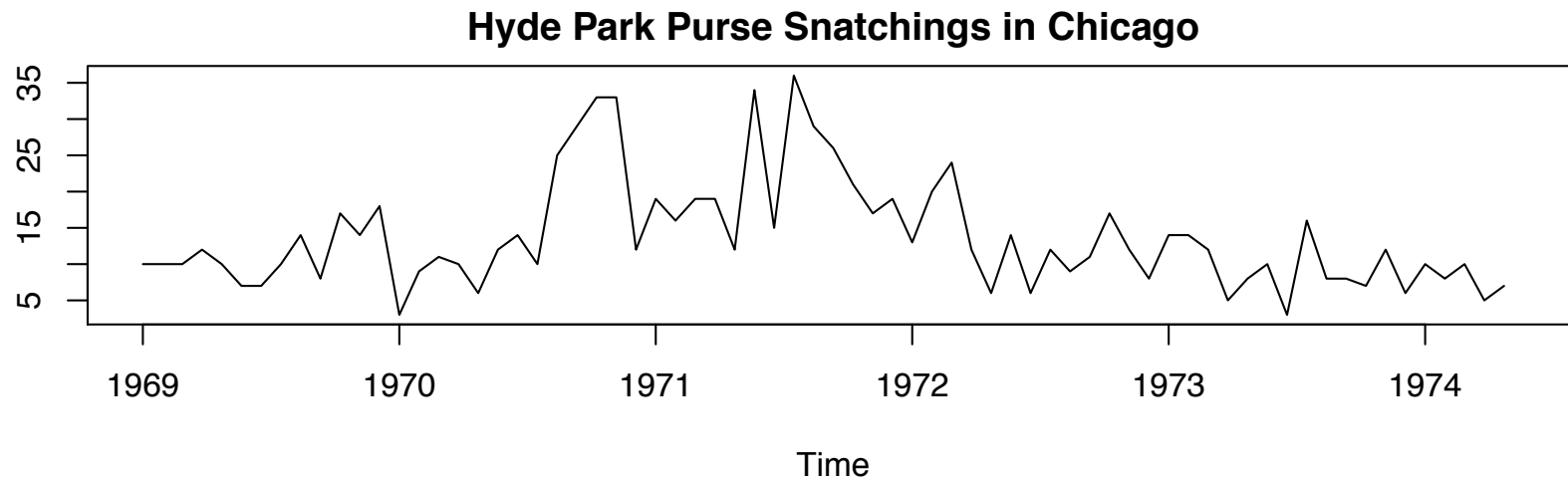


# Outline

1. What is a timeseries?
2. Timeseries characteristics
3. Approaches to modeling timeseries
4. Automation in timeseries modeling

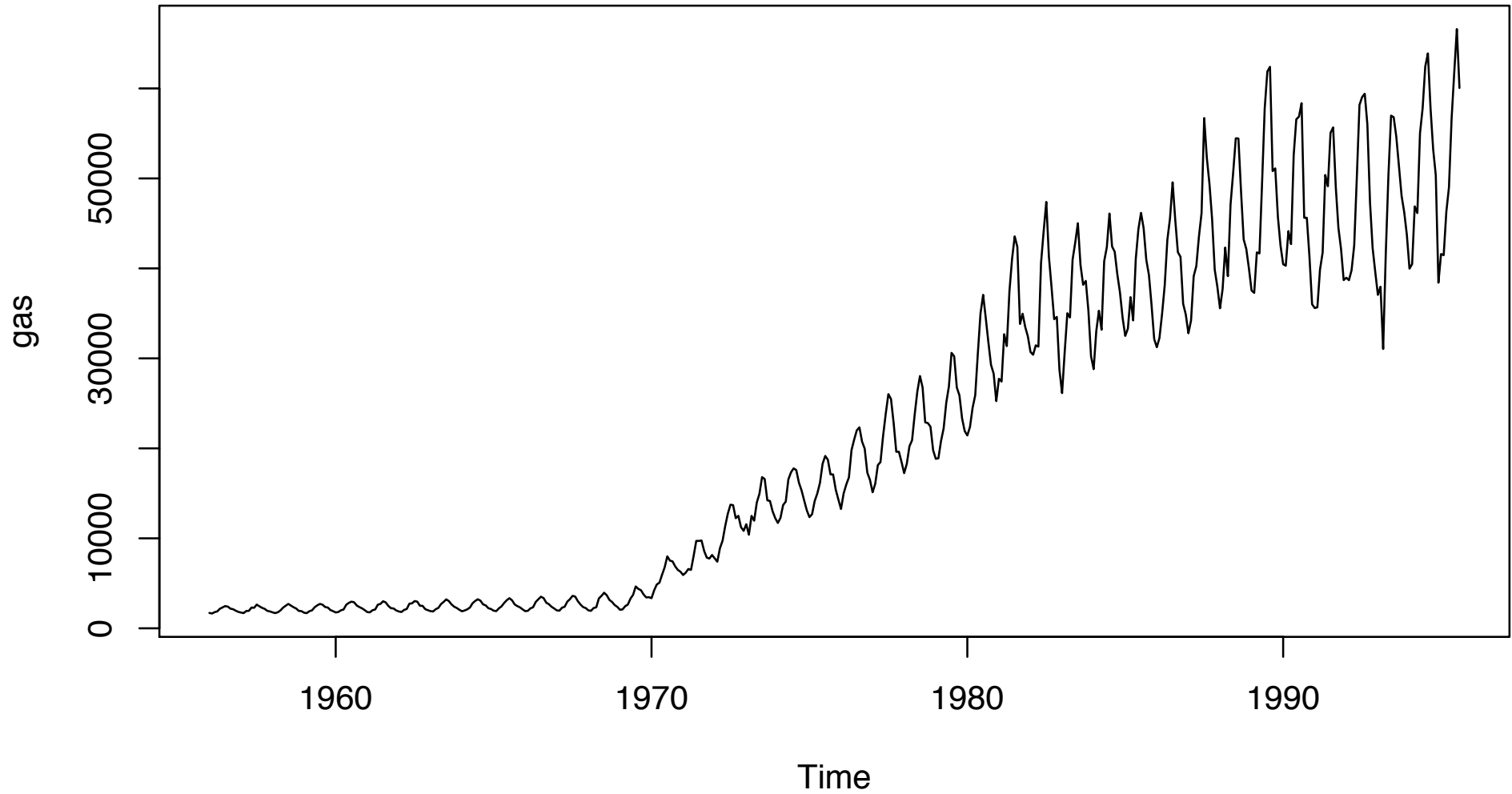
# What is a timeseries?

- A phenomenon observed over time
- Observations on the same unit, system etc over time
- Therefore the observations are not independent
- Regular or irregular time intervals
  - temperature at 2 pm
  - stock price at 2 pm is actually last traded price, which itself is a random event



# Timeseries characteristics

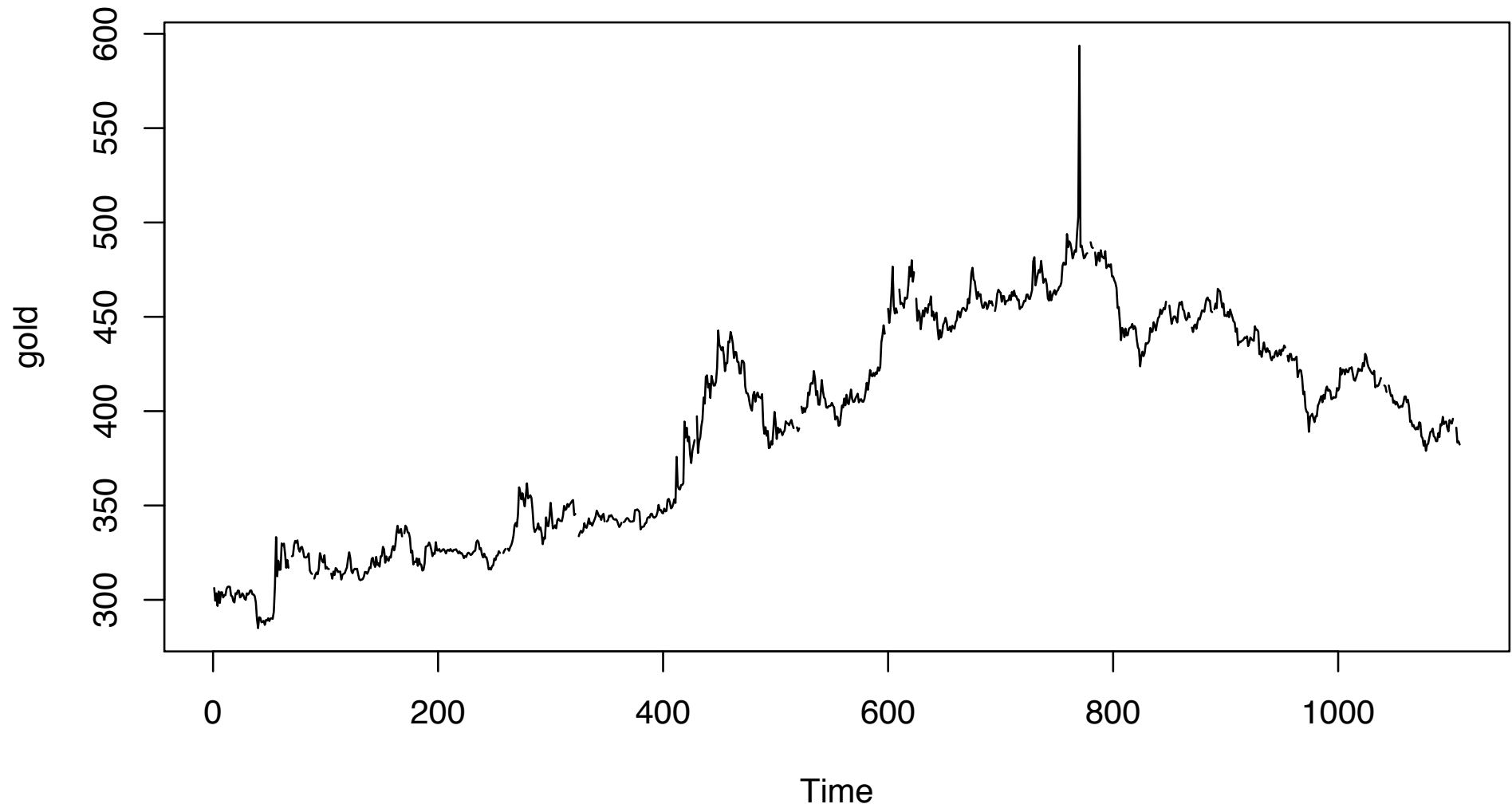
## 1. Australian monthly gas production





# Timeseries characteristics

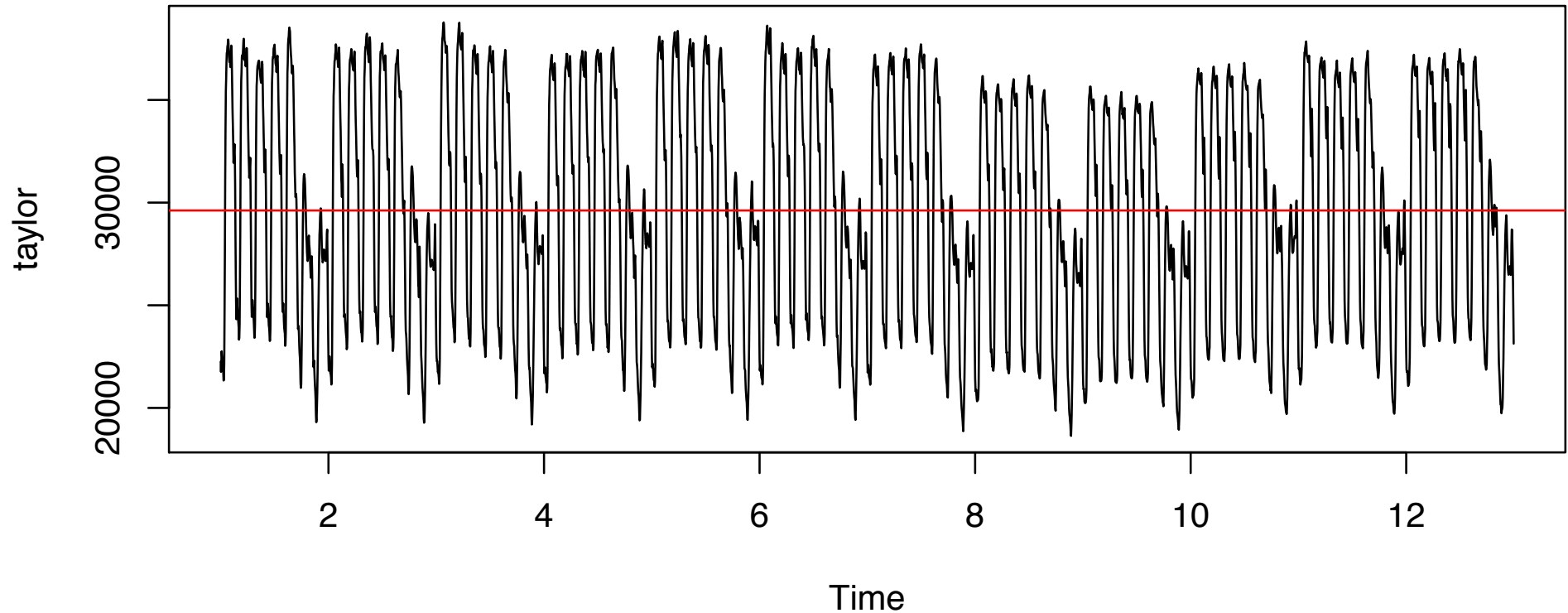
## 2. Daily morning gold prices





# Timeseries characteristics

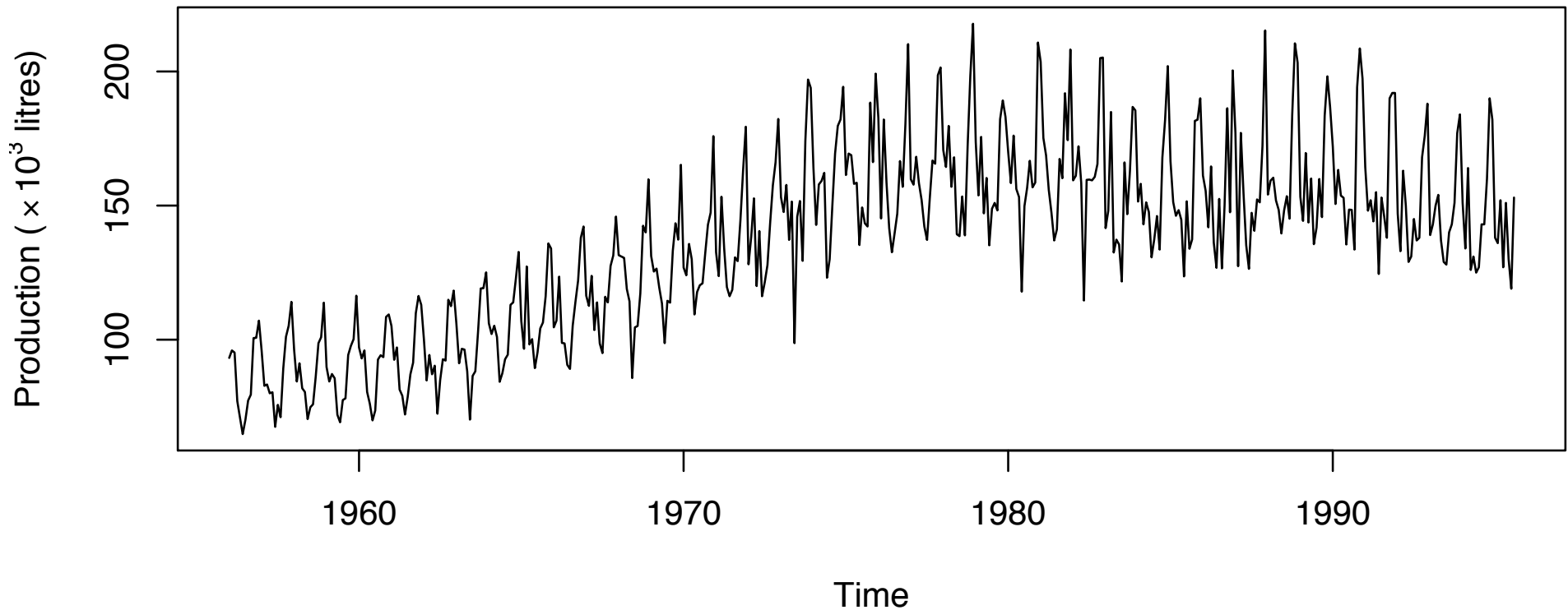
## 3. Half-hourly electricity demand



- Trend
- Seasonality

# Timeseries object in R

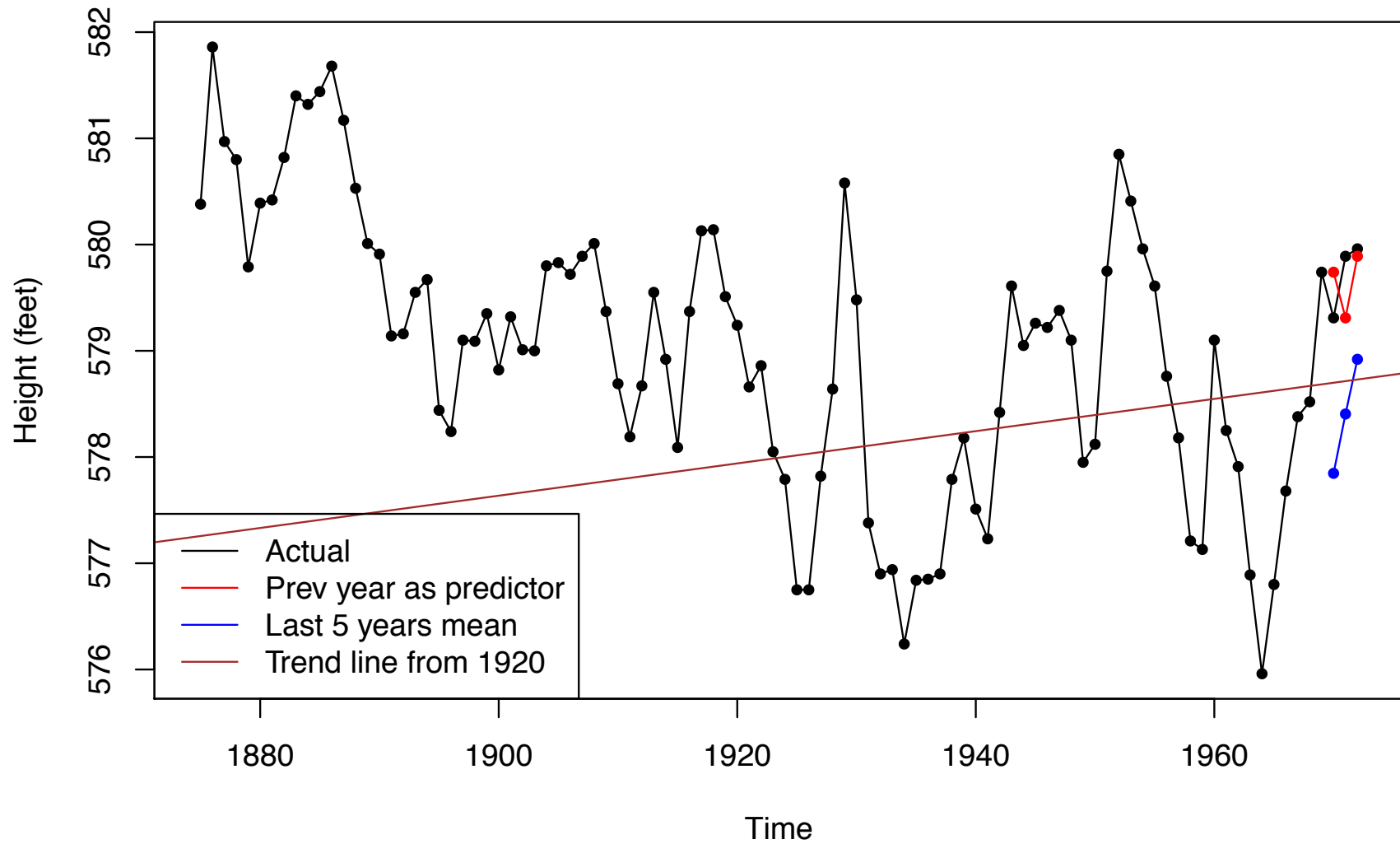
- Beer production in Australia from Jan. 1956 to Aug. 1995
- <http://134.76.173.220/beer.zip>





# Naive predictors

Lake Huron water level







# Exponential smoothing

- Average of previous values, with the most recent values weighted more strongly.

$$y_t = \alpha(y_{t-1}) + \alpha(1 - \alpha)y_{t-2} + \alpha(1 - \alpha)^2y_{t-3} + \cdots, \quad 0 < \alpha < 1$$

- Exponentially Weighted Moving Average (EWMA)
- `HoltWinters(x, beta=FALSE, gamma=FALSE)`



# HoltWinters

HoltWinters offers following ways of modeling trend and seasonality

- Trend: None or Additive
- Seasonality: None, Additive or Multiplicative
- Any combination of these is possible
- Only non-negative terms may be modelled by multiplicative seasonality
- Error term is always additive



# ExponenTial Smoothing

ETS family of models offer the following

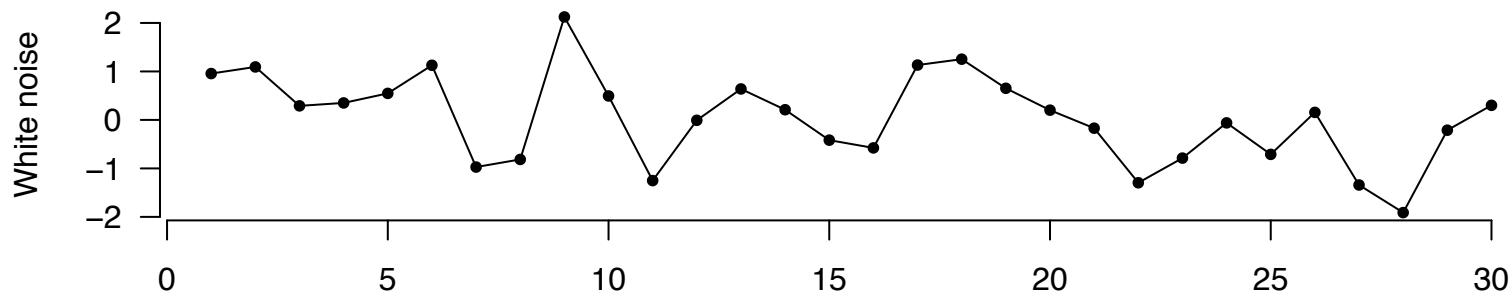
- Trend: None, Additive, Additive-damped, Multiplicative, Multiplicative-damped
- Seasonality: None, Additive or Multiplicative
- Error: Additive or Multiplicative
- In all 24 models are possible
- Identified using 3-letter combination like (M Ad M)

```
ets(y, model="ZZZ", damped=NULL)
```

# Stationary timeseries

A timeseries (or a process as it is also called) is stationary if

1. The mean is constant over time
2. The variance is constant over time
3. Covariance between  $y_t$  and  $y_{t-h}$ , where  $h$  is the lag, depends only on  $h$ , and not on  $t$ , the time at which the covariance is measured.



- Better known as 'Covariance stationary'
- Augmented Dickey-Fuller test for stationarity: `tseries::adf.test`

# Transformations for Stationarity

- Box-Cox transformation

$$y_t^{(\lambda)} = \begin{cases} \log(y_t), & \lambda = 0 \\ \frac{y_t^\lambda - 1}{\lambda}, & \lambda \neq 0 \end{cases}$$

MASS: :boxcox

- Differencing

$$\nabla y_t = y_t - y_{t-1} \dots \text{first difference}$$

# Box-Jenkins (ARIMA) Models

- Autoregressive process: Next term is a sum of some past terms

$$y_t = \eta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \epsilon$$

- Stationarity requirement:  $\sum \phi_i < 1$
- Moving average process: Next term is a sum of random shocks over time

$$y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

- Invertibility requirement: MA process  $\sim$  AR process
- Real process is a combination of AR & MA processes
- **order** is the number of terms in the model
- ARIMA: Auto Regressive Integrated Moving Average



# ARIMA modeling

- stats:: ar, arma, arima
- forecast:: Arima

```
Arima(x, order=c(0,0,0),  
      seasonal=list(order=c(0,0,0), period=NA))
```

- forecast:: Acf, Pacf

# ARIMA model identification

Parameters	ACF	PACF
AR(1)	exponential decay	spike at lag 1, no correlation for other lags
AR(2)	sine-wave shape pattern or a set of exponential decays	spikes at lags 1 and 2; no correlation for other lags
MA(1)	spike at lag 1, no correlation for other lags	damps out exponentially
MA(2)	spikes at lags 1 and 2, no correlation for other lags	sine-wave shape pattern or a set of exponential decays
AR(1), MA(1)	exponential decay starting at lag 1	exponential decay starting at lag 1





# Automated Modeling

- Large number of models, repeatedly, every month or so
- Difficulty in choosing model
- Limited skill for bulk work
- `ets(y, model="ZZZ")`
- `auto.arima(y)`

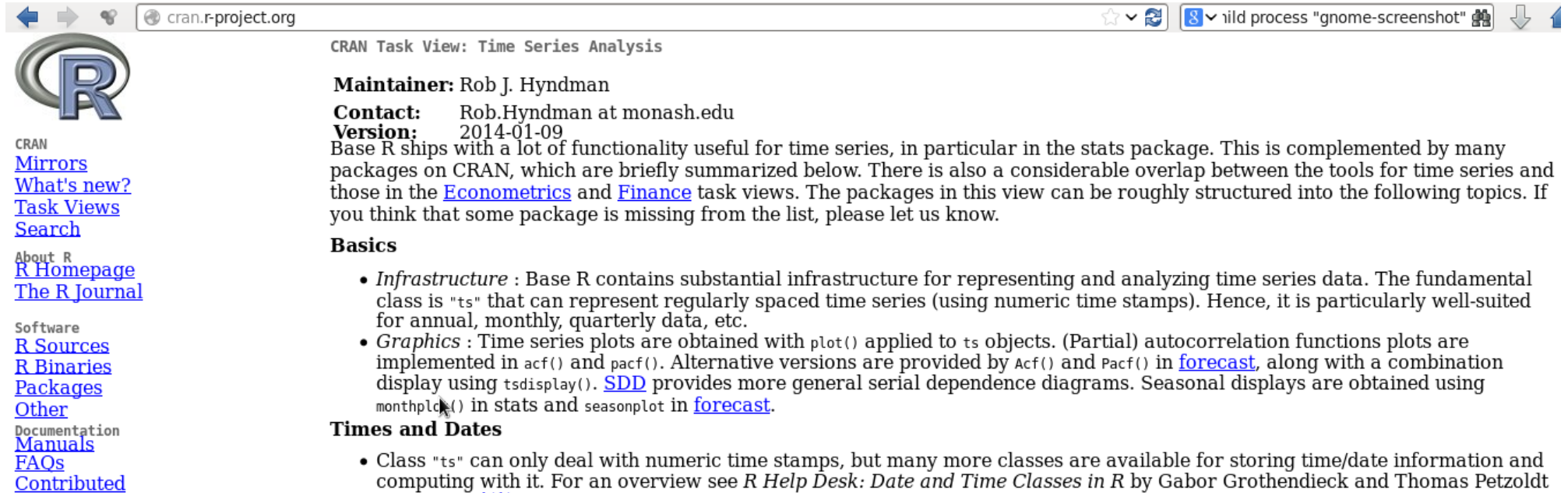
# Advanced Topics

- Packages: zoo, xts
  - irregular timeseries
  - multivariate & financial
- quantmod
  - financial timeseries
  - fetching series
  - custom plots
  - Re-purposed techniques & models



# Cran: TaskViews

<http://cran.r-project.org/web/views/TimeSeries.html>



CRAN Task View: Time Series Analysis

**Maintainer:** Rob J. Hyndman  
**Contact:** Rob.Hyndman at monash.edu  
**Version:** 2014-01-09

Base R ships with a lot of functionality useful for time series, in particular in the stats package. This is complemented by many packages on CRAN, which are briefly summarized below. There is also a considerable overlap between the tools for time series and those in the [Econometrics](#) and [Finance](#) task views. The packages in this view can be roughly structured into the following topics. If you think that some package is missing from the list, please let us know.

**Basics**

- *Infrastructure* : Base R contains substantial infrastructure for representing and analyzing time series data. The fundamental class is "ts" that can represent regularly spaced time series (using numeric time stamps). Hence, it is particularly well-suited for annual, monthly, quarterly data, etc.
- *Graphics* : Time series plots are obtained with `plot()` applied to `ts` objects. (Partial) autocorrelation functions plots are implemented in `acf()` and `pacf()`. Alternative versions are provided by `Acf()` and `Pacf()` in [forecast](#), along with a combination display using `tsdisplay()`. [SDD](#) provides more general serial dependence diagrams. Seasonal displays are obtained using `monthplot()` in stats and `seasonplot` in [forecast](#).

**Times and Dates**

- Class "ts" can only deal with numeric time stamps, but many more classes are available for storing time/date information and computing with it. For an overview see *R Help Desk: Date and Time Classes in R* by Gabor Grothendieck and Thomas Petzoldt in [R News 4\(1\)](#), 29-32.
- Classes "yearmon" and "yearqtr" from [zoo](#) allow for more convenient computation with monthly and quarterly observations, respectively.
- Class "Date" from the base package is the basic class for dealing with dates in daily data. The dates are internally stored as the number of days since 1970-01-01.
- The [chron](#) package provides classes for `dates()`, `hours()` and date/time (intra-day) in `chron()`. There is no support for time zones and daylight saving time. Internally, "chron" objects are (fractional) days since 1970-01-01.

CRAN  
[Mirrors](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)  
About R  
[R Homepage](#)  
[The R Journal](#)  
Software  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Other](#)  
Documentation  
[Manuals](#)  
[FAQs](#)  
[Contributed](#)