Time series analysis
and forecasting

Erik-Jan van Kesteren
(based on materials by Rob Hyndman & Oisín Ryan)

Welcome back!

Before the break

- Regularization, high-dimensional data
- Dimension reduction
- Deep learning
- Clustering

After the break

- Time series (today)
- Text mining & natural language processing
  (taught by Ayoub Bagheri)

- Deadline assignment 2: next week Friday at 11:00!

- Final exam: February 2, 14:00 – 16:00
  Example questions can be found on infomda2.nl

Last week

- MNIST digit recognition
- Feed forward neural networks
- Convolutional neural networks
- Keras
- Curse of dimensionality exists there too!

Today

- Time series & forecasting
- Nonparametric decomposition of timeseries (STL)
- Autocorrelation & partial autocorrelation
- Autoregression-based models (SARIMA)

- Model selection & generalization
- Fable package

Time series & Forecasting

What is a time series?

Repeated measures from a single unit

Time series data

- The daily price of oil since 1960
- Annual deaths in the Netherlands for 100 years
- Maximum temperature per day/month/year for a period of one year/decade/century
- Daily measures of your happiness for 60 days

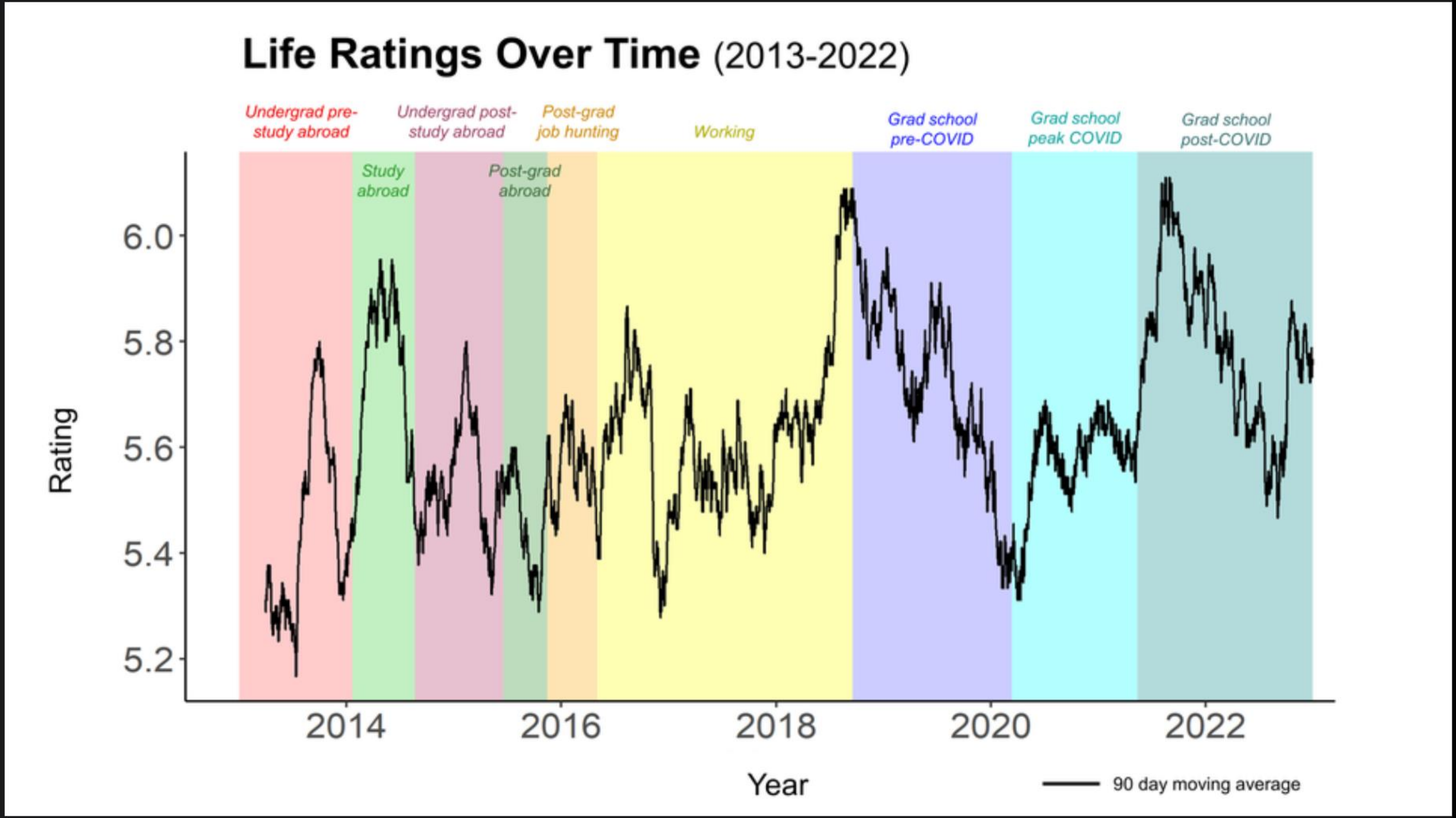
Typically, a time series is a vector of numbers.

r/dataisbeautiful · Posted by u/tkpkid 2 days ago

2.3k

[OC] I've been keeping a daily journal and rating each day on a scale from 1-10 since 2013. Here is a moving average of my happiness over the past 10 years

OC



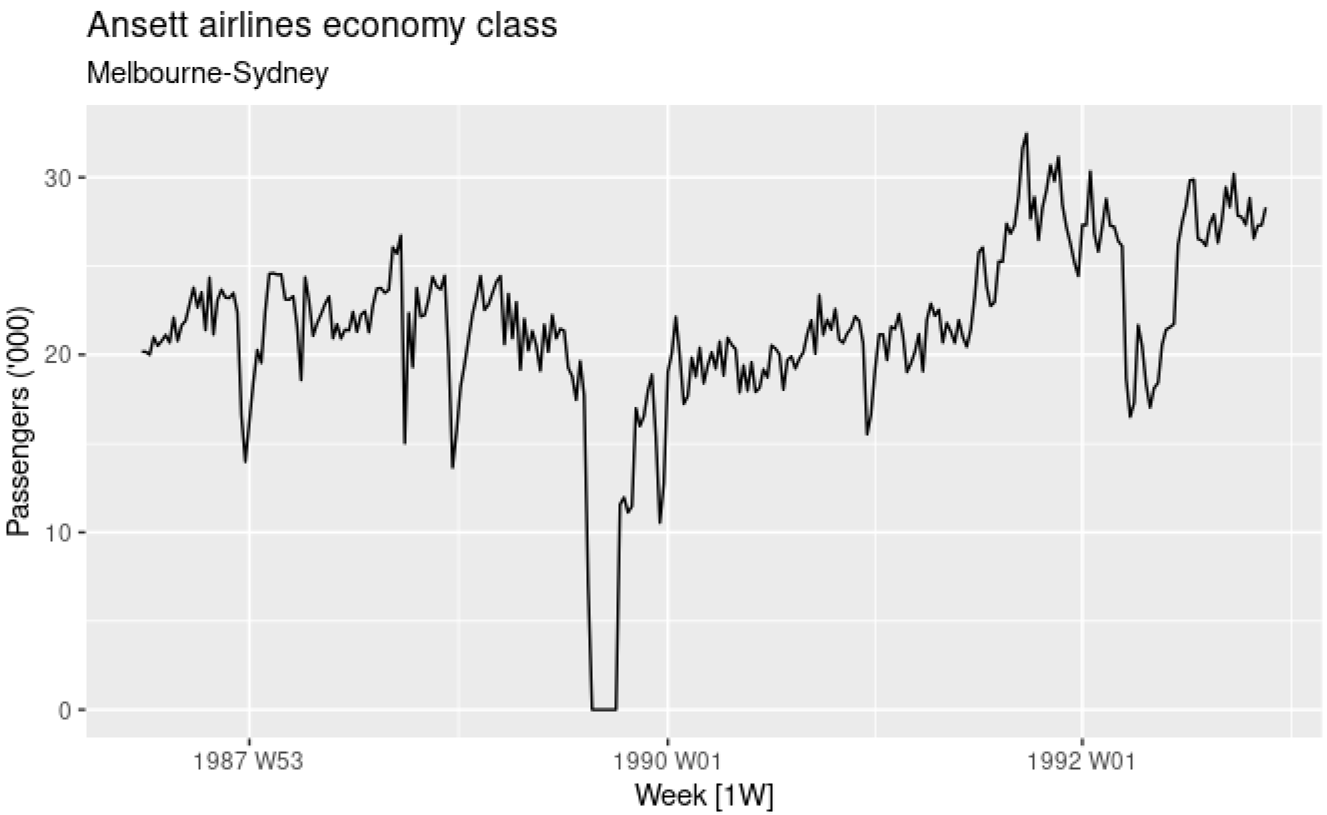


Figure 2.1: Weekly economy passenger load on Ansett Airlines.

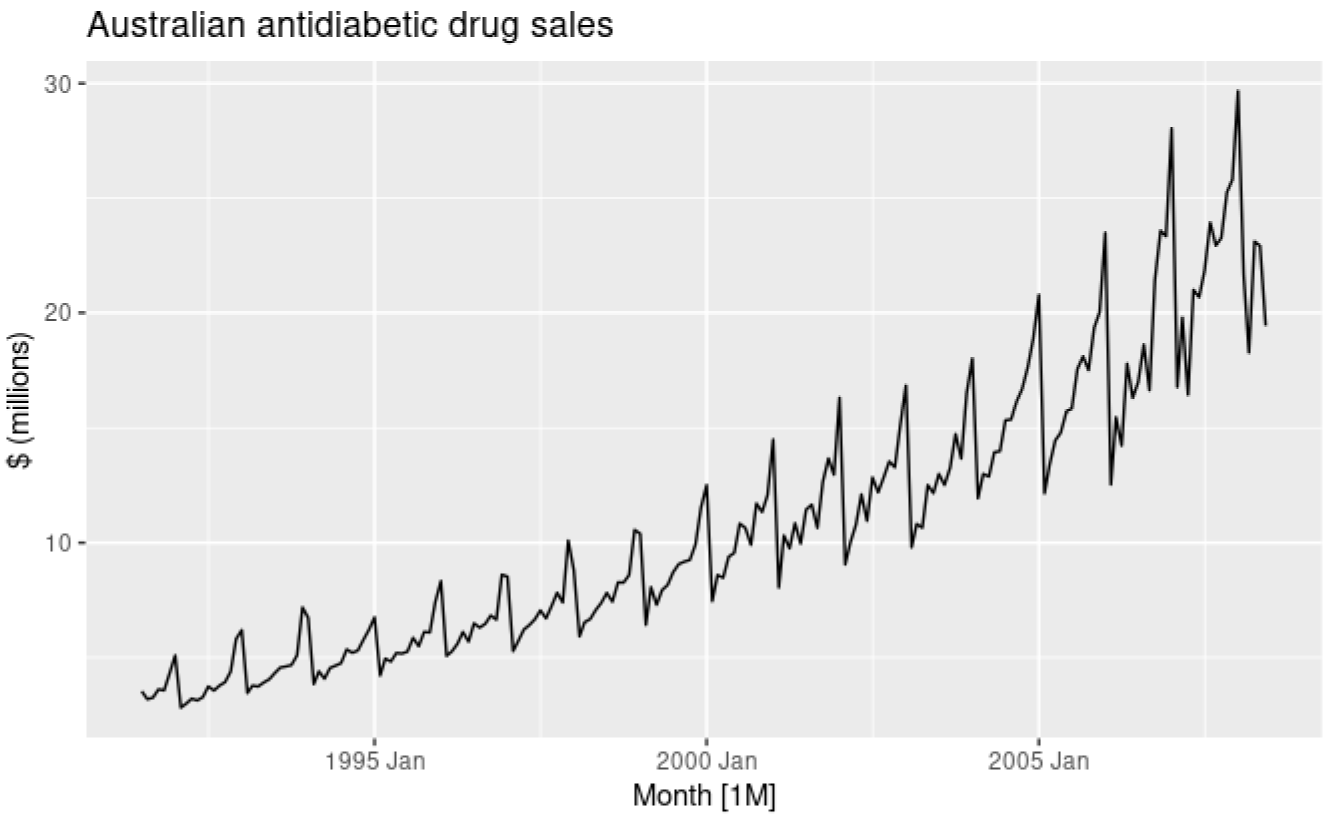


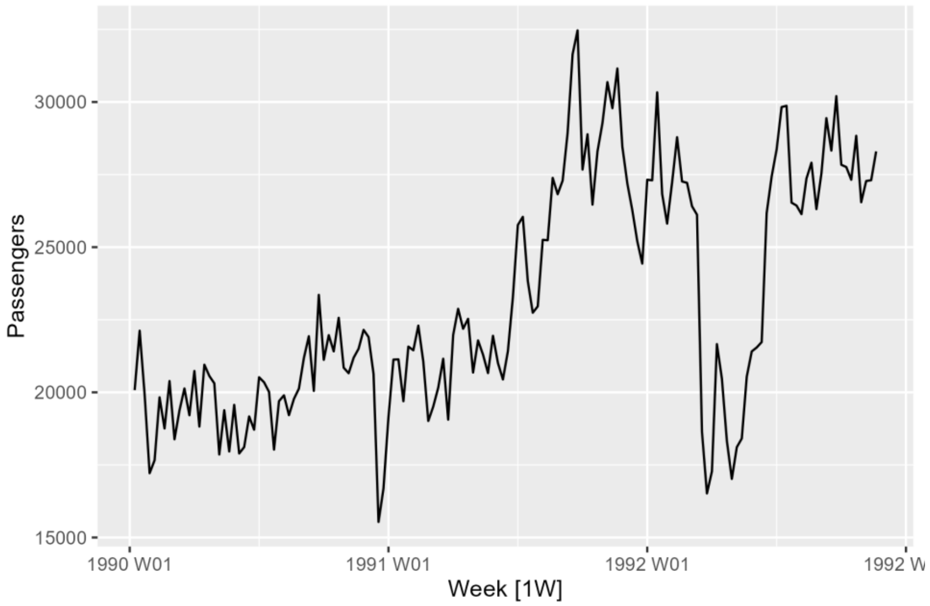
Figure 2.2: Monthly sales of antidiabetic drugs in Australia.

Time series data

In R

- tsibble object type
- It's a tibble, with special "index" column
- Makes it easy to work with times and dates
- (also by default groups by index)

```
ansett |>
  filter(
    Week > yearweek("1990-01-01"),
    Airports == "MEL-SYD",
    Class == "Economy"
  ) |>
  autoplot(Passengers)
```



Time series analysis

Time series data for wide range of research questions

- Econometrics, seismology, meteorology, control engineering, signal processing …

Time series analysis often focuses on the task of **forecasting**

- Forward / future predictions
- Goal: learn a model from data that forecasts well

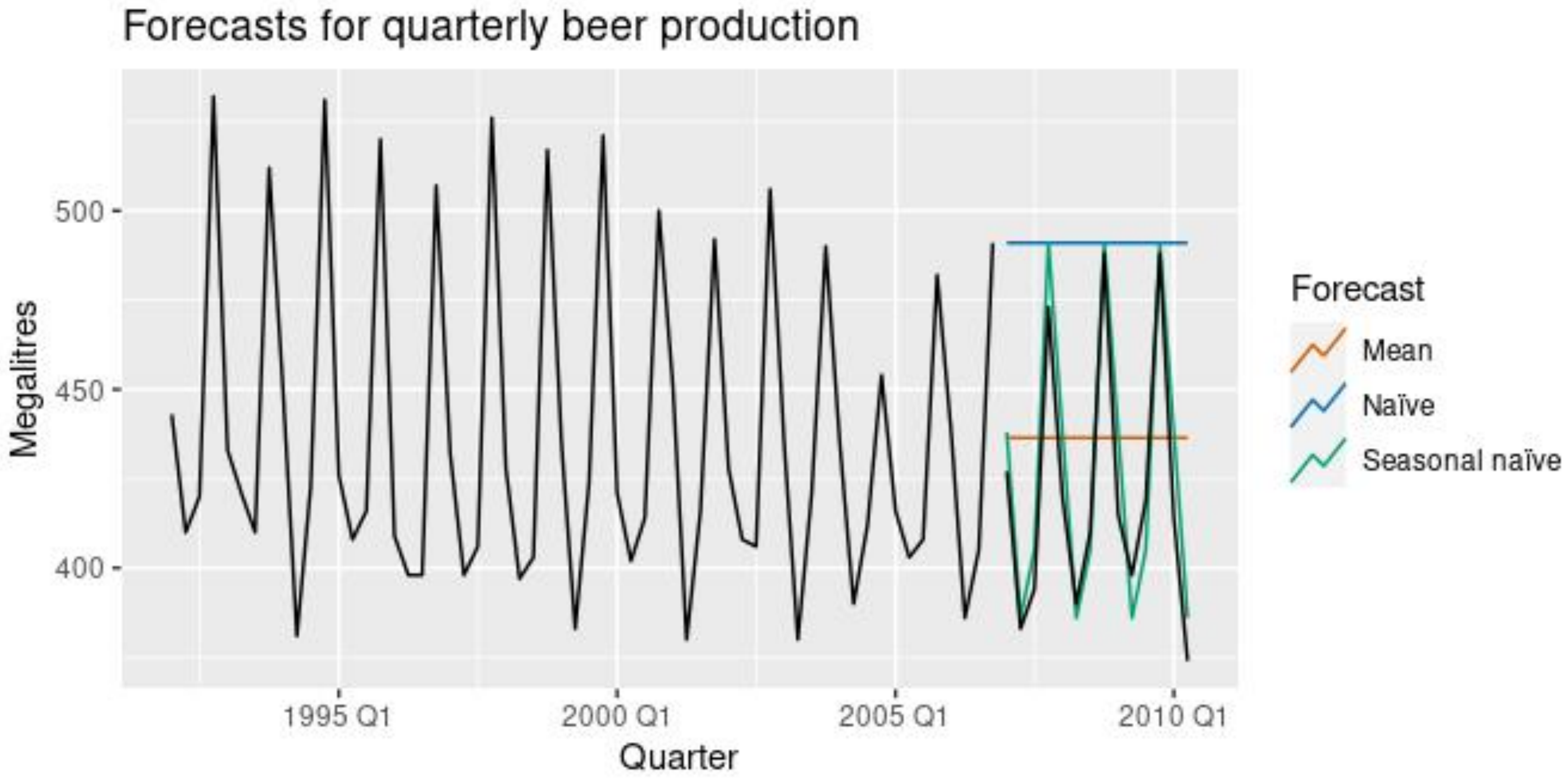


Figure 5.7: Forecasts of Australian quarterly beer production.

Time series analysis

Basic problem

How to best use the past of a time series to make predictions about the future?

Many different methods

- Simple / naive
- Nonparametric
- Auto-regressive
- Complicated Bayesian state-space models

Most of these **decompose** time-series into components

Time series analysis

Basic problem

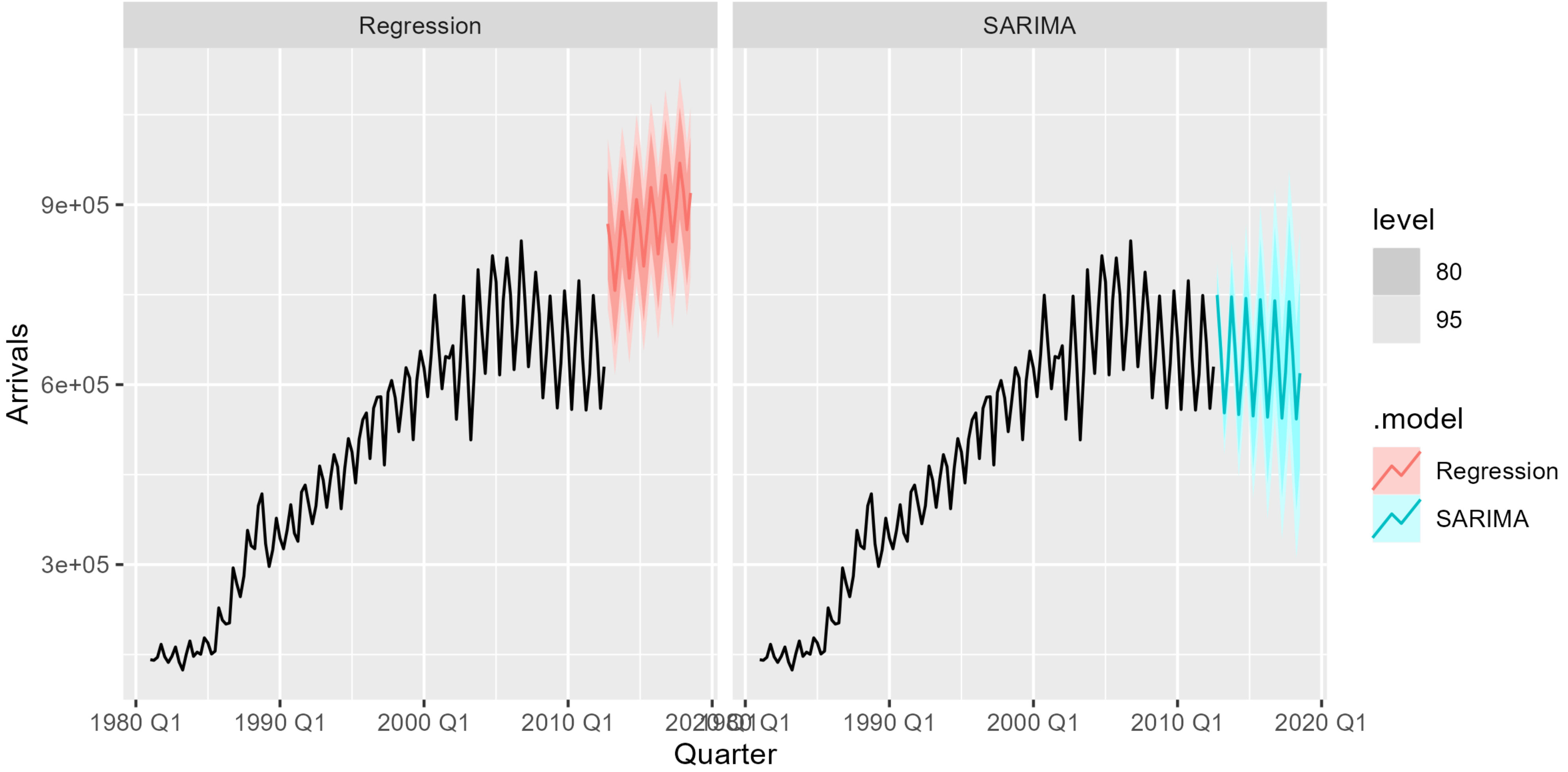
How to best use the past of a time series to make predictions about the future?

Many different methods

- Simple / naive
- **Nonparametric**
- **Auto-regressive**
- Complicated Bayesian state-space models

Most of these **decompose** time-series into components

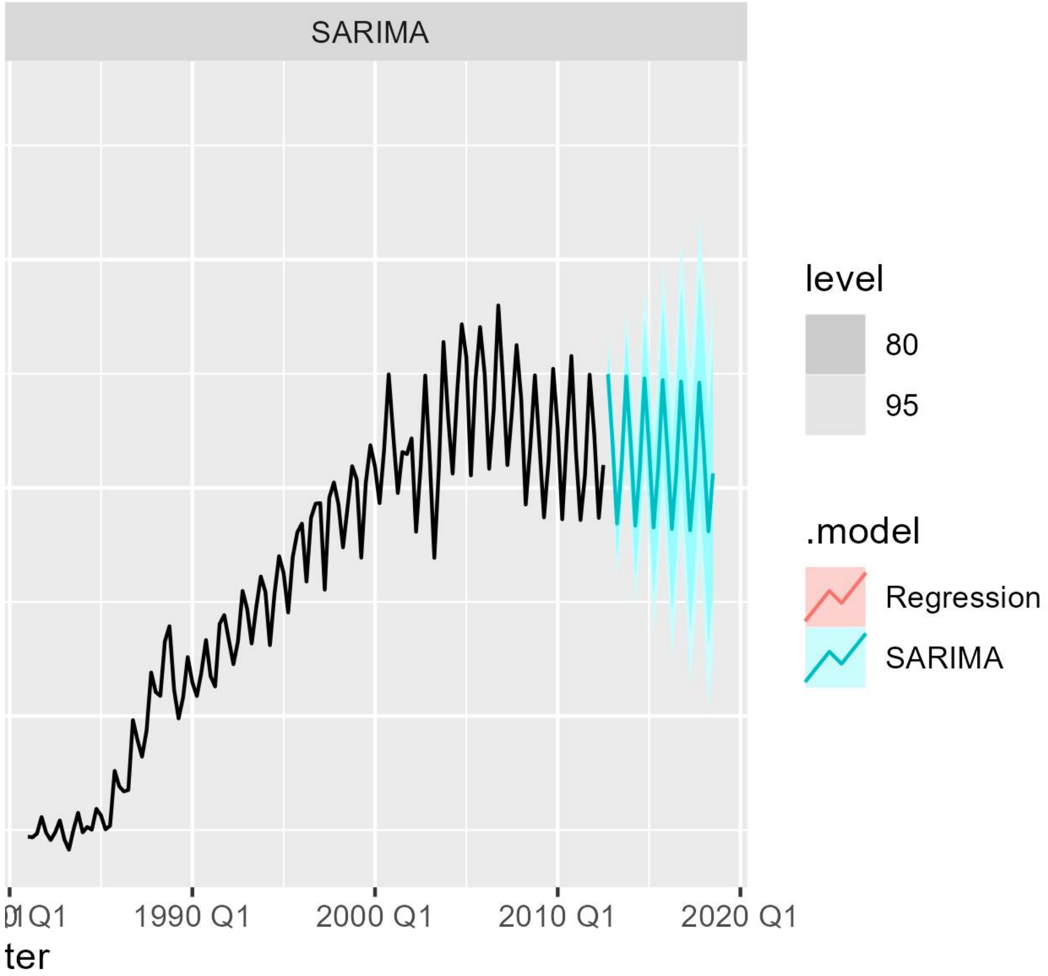
Why not just lm(y ~ f(t))?



Why not just lm(y ~ f(t))?

Why is SARIMA better?

- No sudden jump
- Slight downward trend
- Looks more like recent observations
- Uncertainty increases as we forecast further away (!!!)



Decomposing time series

STL decomposition

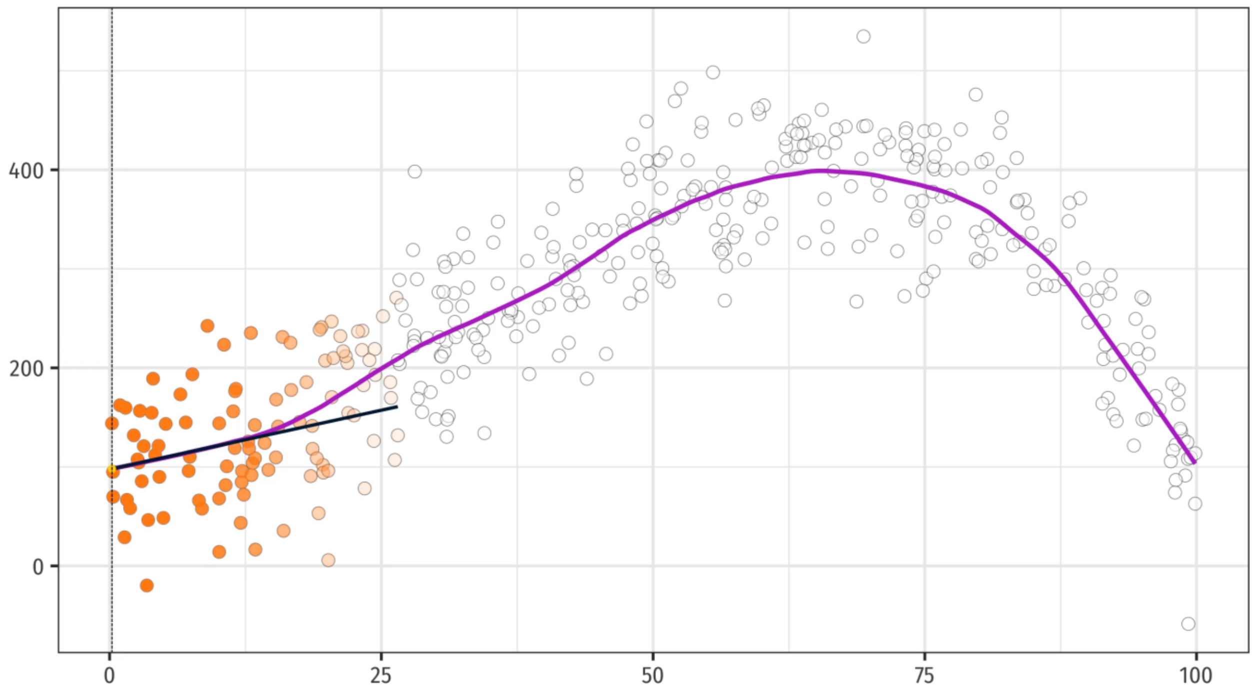
- *Seasonal and Trend decomposition using Loess*
- Use **nonparametric** regression to decompose time series into:
  - Season
  - Trend
  - Remainder

LOESS

- For each point x_i, weighted regression using nearby data points

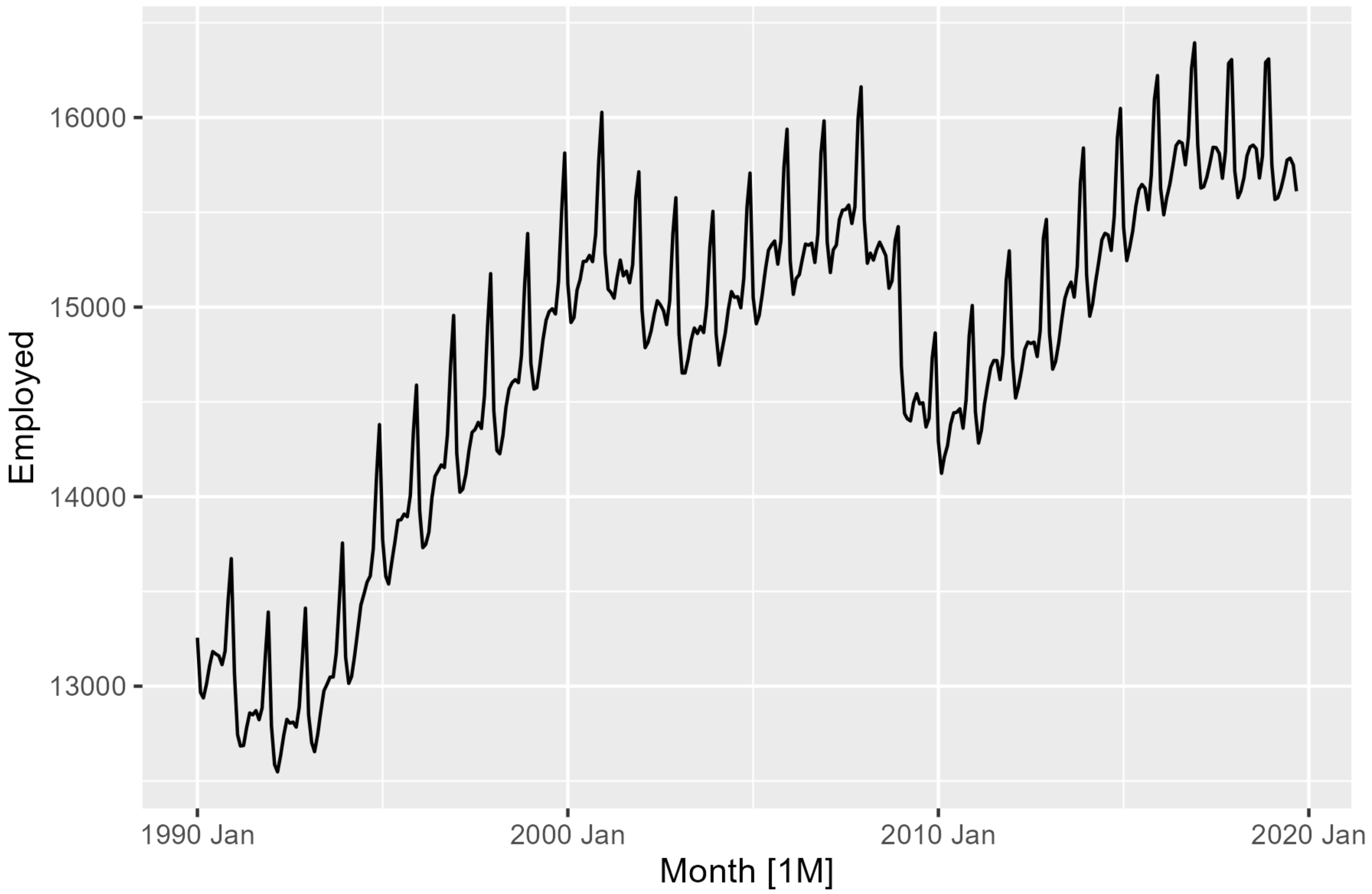
$$\text{weights} \propto f(distance)$$

- Hyperparameters
  - maximum distance
  - regression model type
  - Weighing function

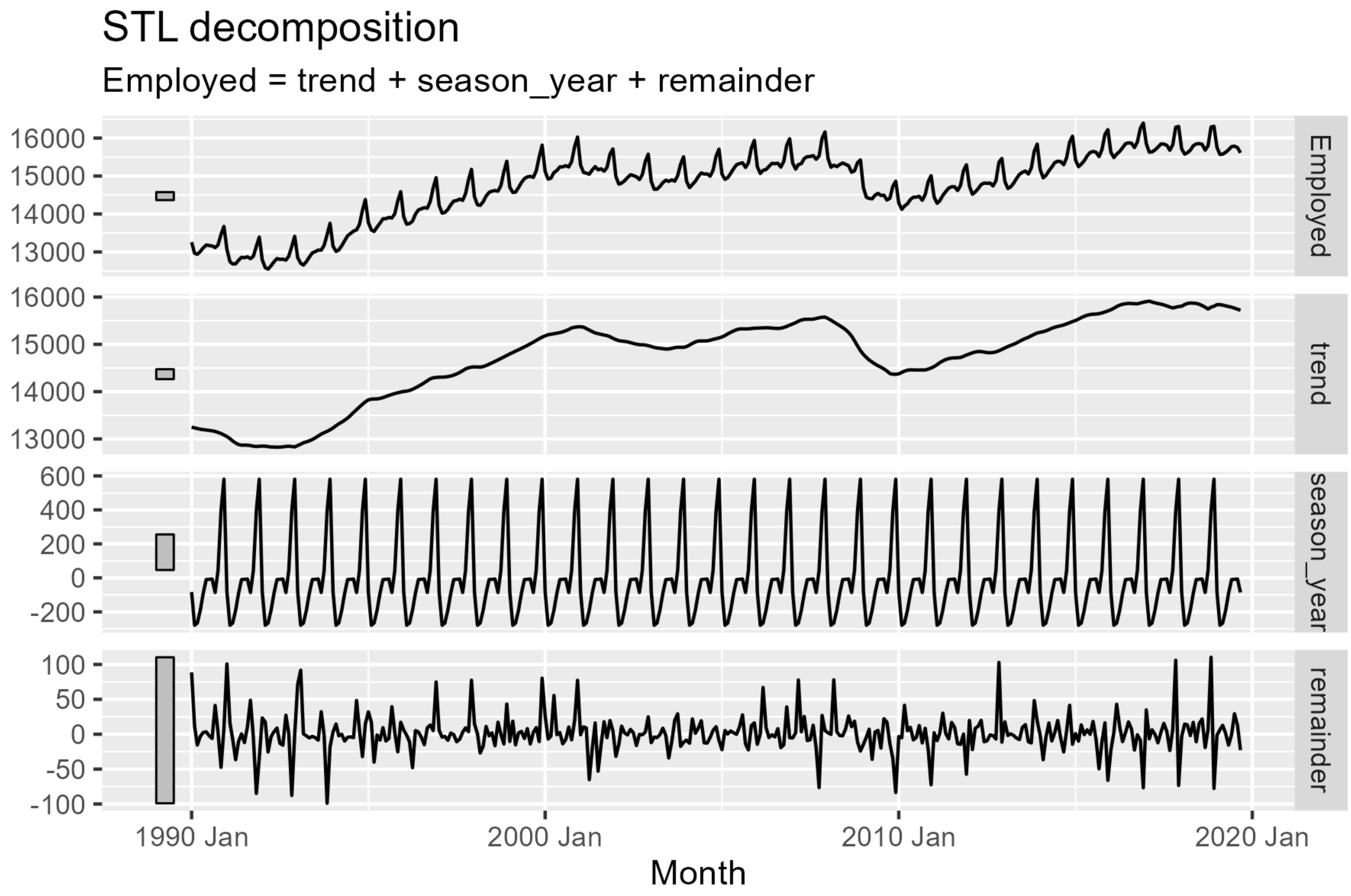


https://evalf22.classes.andrewheiss.com/slides/10-slides.html

STL decomposition



STL decomposition

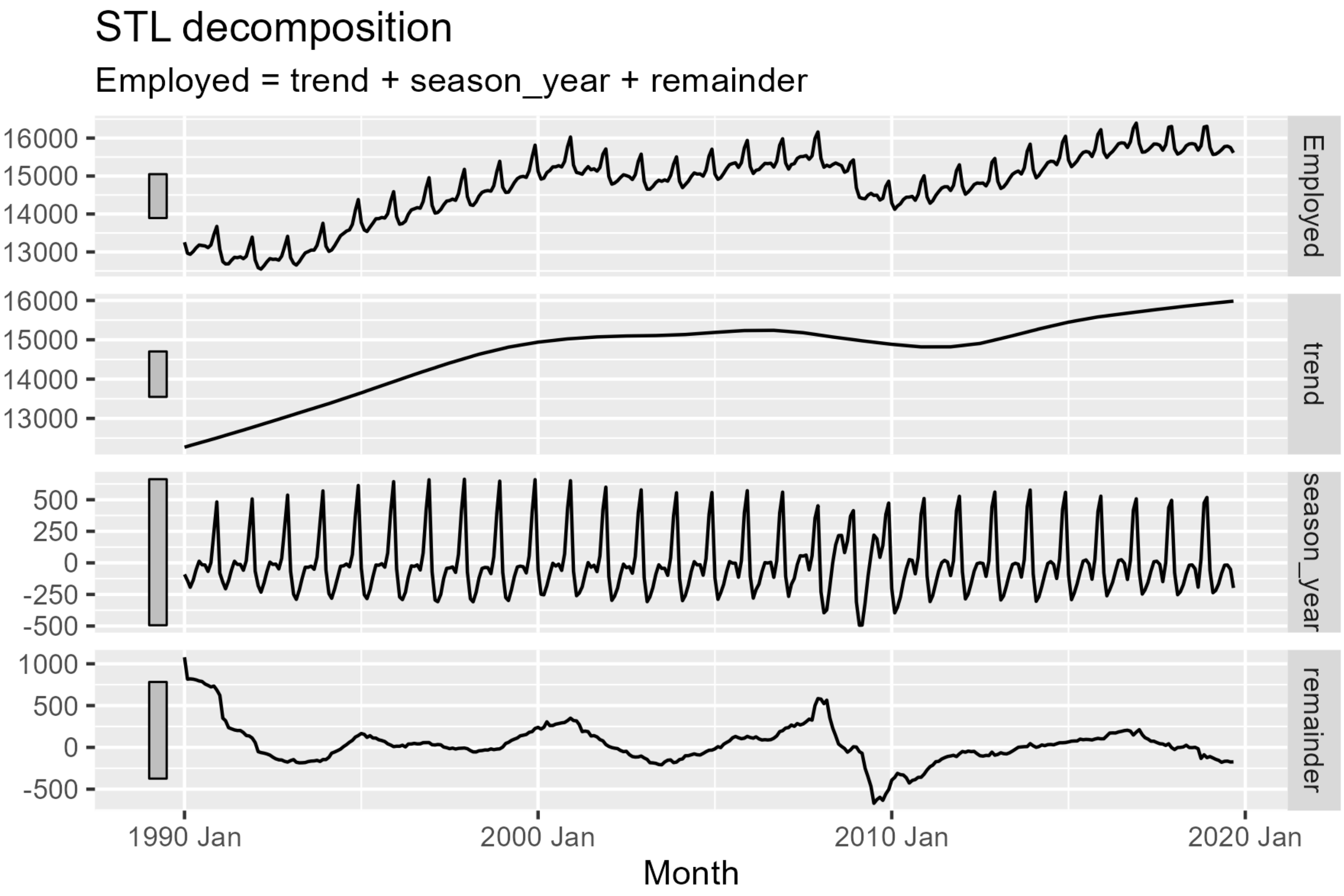


STL decomposition

Need to perform model selection

- How wiggly is my trend?
- How stable is my seasonality effect?

STL decomposition

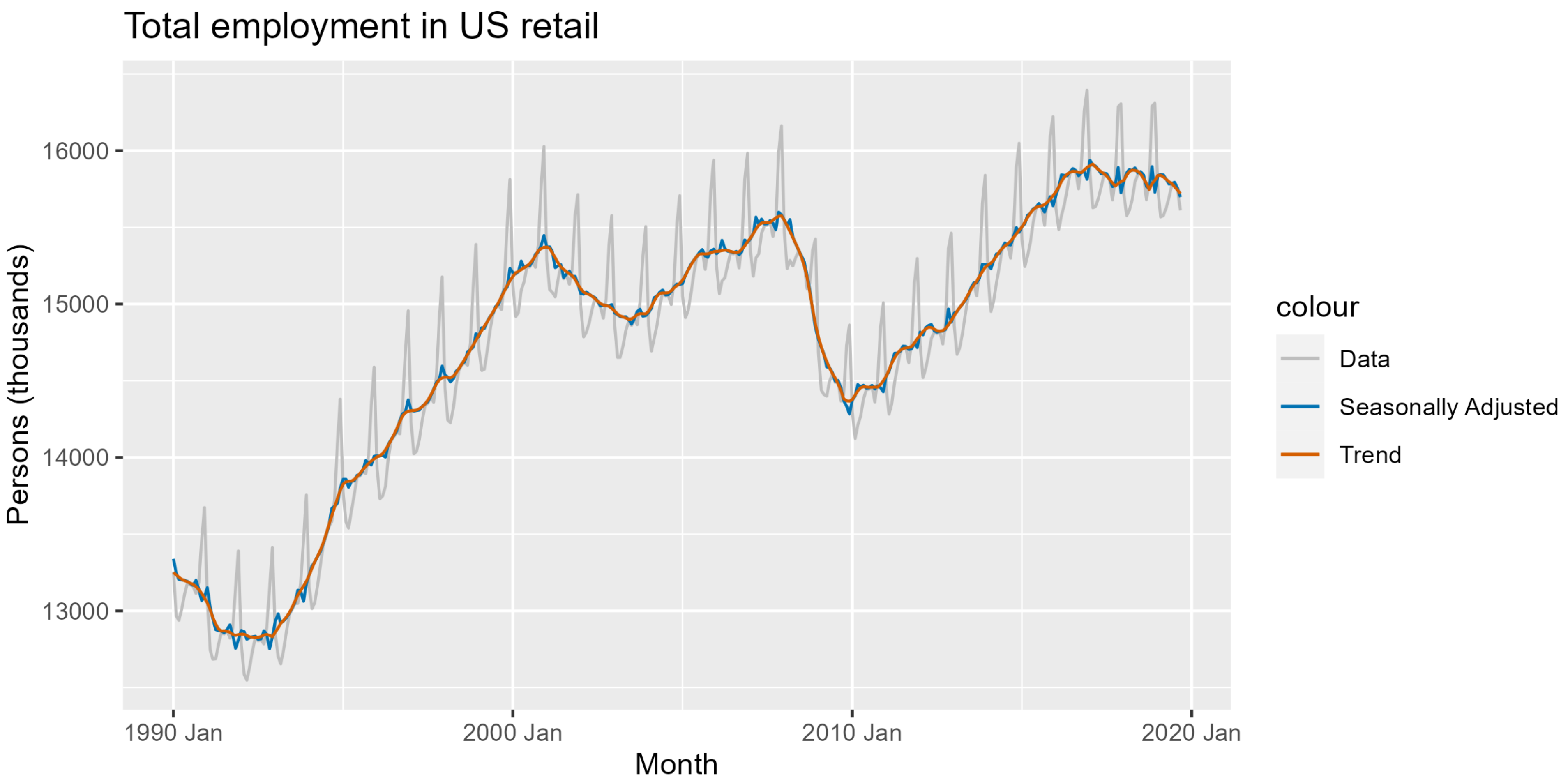


STL decomposition

After finding a good model, you can

- Produce seasonally adjusted estimates (national statistical agencies tend to output these)
  - $y_t^{sa} = trend_t + \epsilon_t$
- Create forecasts
  - $\hat{y}_{t+1} = trend_{t+1} + season_{t+1} + \epsilon_{t+1}$

Seasonal adjustment



Forecast

US retail employment
16000
15000
14000
13000
Number of people
1990 Jan
2000 Jan
2010 Jan
2020 Jan
Month
level
80
95

Auto-regressive models

$$Y_t = f_1(T) + f_2(S) + f_3(Y_{t-1}, \dots, Y_{t-s}) + \epsilon_t$$

- Deterministic trend T
  - Ice cream brand is becoming popular so sales go up
- Seasonal effects S
  - Ice cream sales are higher in summer than in winter
- Past values of our process of interest: **auto-correlation structure**
  - Conditional on T and S, my best guess of sales in August is sales in July
- Random / noise / residual / innovation / shock ϵ_t

Autoregressive model selection

Traditional view: Box-Jenkins method (1970)

We should aim to obtain a model which explains all **serial dependency** present in the data.

Non-traditional view: Data Science® method™

We should aim to obtain a model which creates good **out-of-sample forecasts**

Autoregressive model selection

- Serial dependency: all statistical dependency between current and past values of the time-series
- Errors of the regression model ϵ_t should contain no information about future observations
  - Why? If it did, we could improve our forecasts
  - To check: the residuals should be white noise (i.e. uncorrelated over time)

Box-Jenkins method

- Check for trends and seasonal components. Remove them if present.
- Choose order / lag of autocorrelation part of model
- Estimate parameters / fit your model
- Check if your residuals are white noise!

Break

Box-Jenkins method

- Check for trends and seasonal components. Remove them if present.
- **Choose order / lag of autocorrelation part of model**
- Estimate parameters / fit your model
- Check if your residuals are white noise!

Choosing AR terms

$$Y_t = f_1(T) + f_2(S) + f_3(Y_{t-1}, \ldots, Y_{t-s}) + \epsilon_t$$

- Should current Y be predicted by
  - Y an hour ago
  - Y two hours ago?
  - Three hours ago?
  - 24 hours ago?

- What **lag** should we consider?

Choosing AR terms

AR(1) model:

$$y_t = c + \phi y_{t-1} + \epsilon_t$$

AR(2) model:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t$$

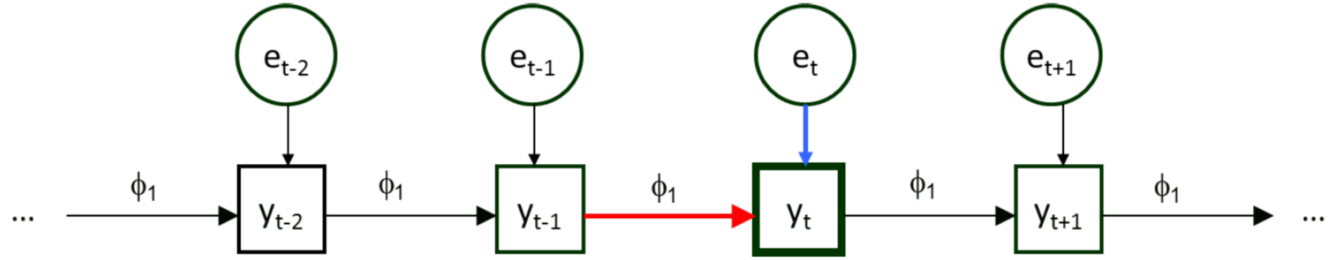
Diagnostic tools

Autocorrelation function

Correlation between lagged version of our variable y_t and y_{t-k}

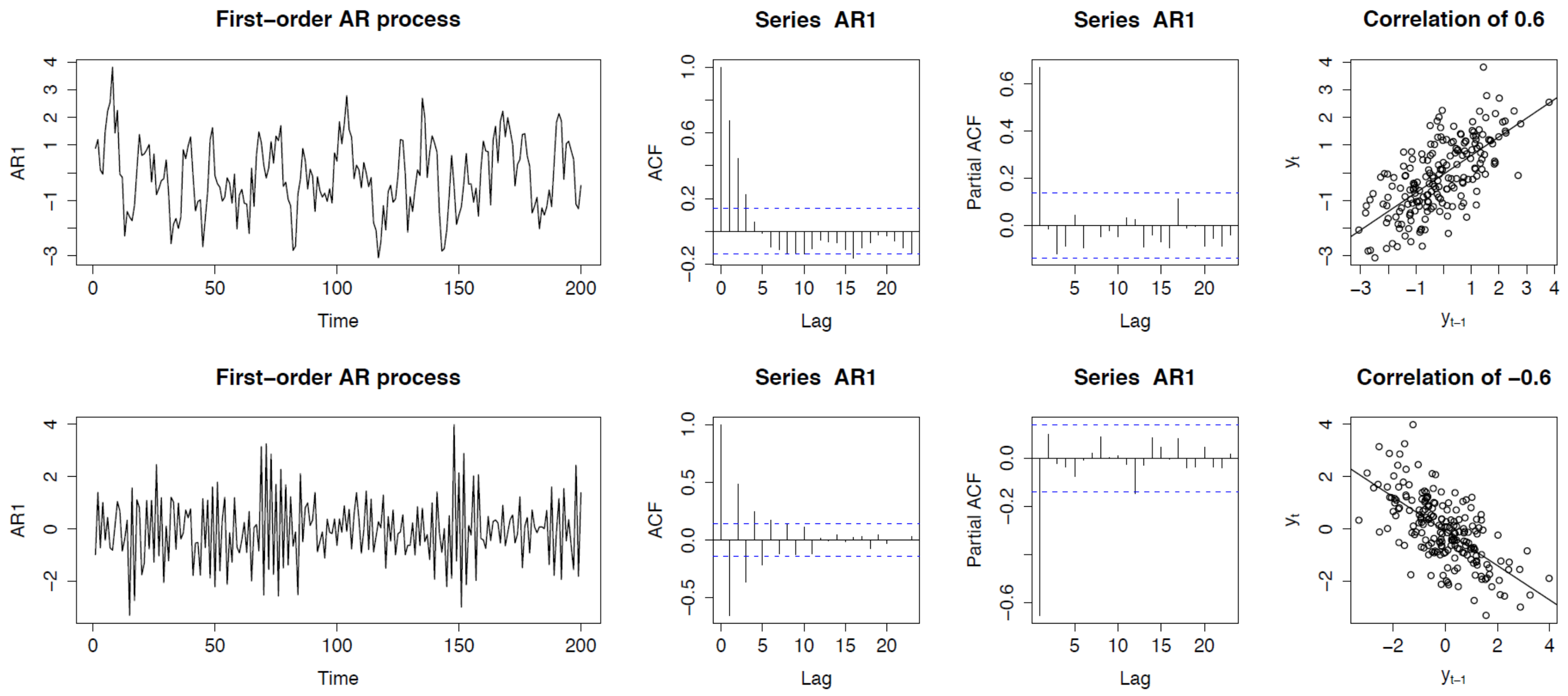
Partial autocorrelation function

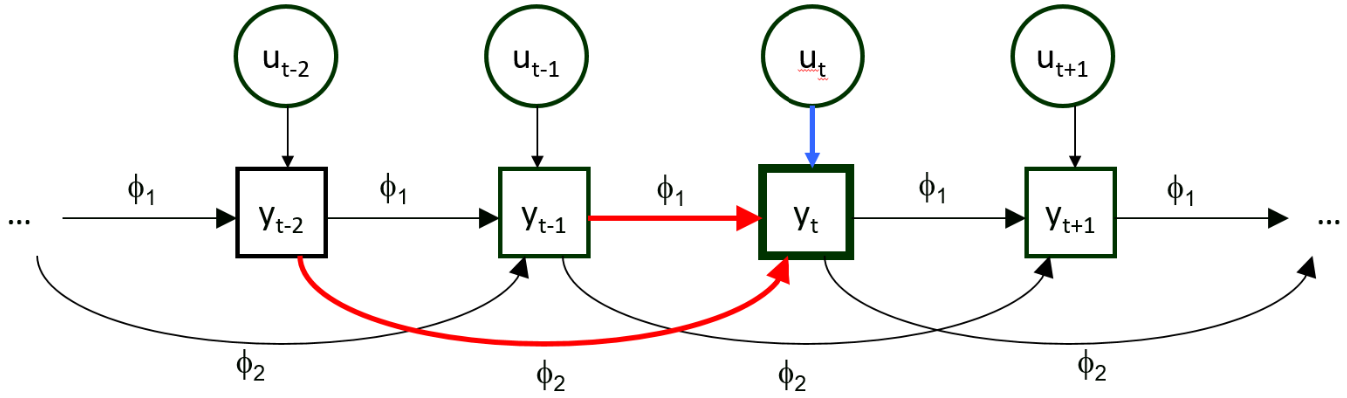
Partial correlation between y_t and y_{t-k}, controlling for intermediate observations y_{t-1} to y_{t-k+1}.



$$y_t = \phi_1 y_{t-1} + e_t$$

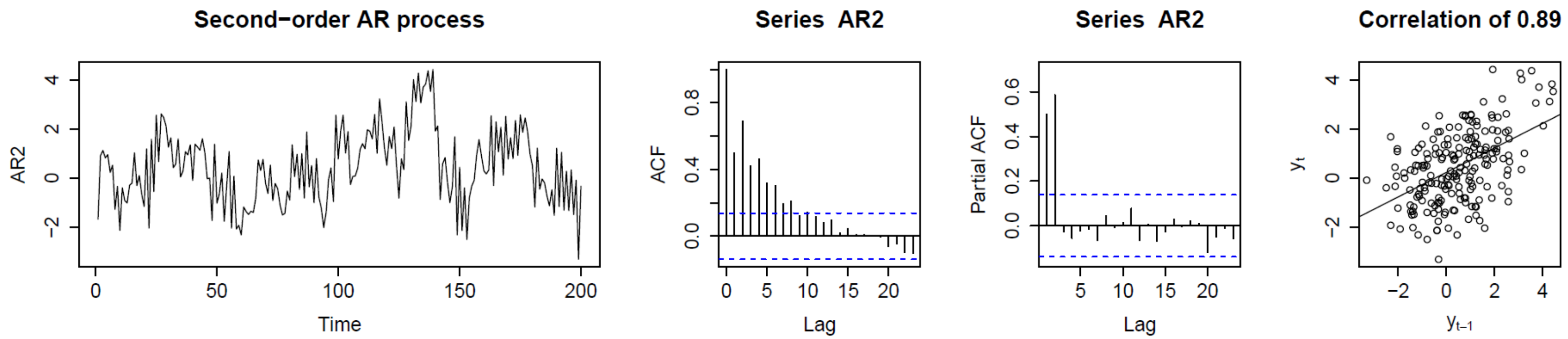
$$\phi_1 = \{0.6; -.6\}$$





$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t$$

$$\phi_1 = 0.2;\ \phi_2 = 0.6$$



Box-Jenkins method

- Check for trends and seasonal components. Remove them if present.
- Choose order / lag of autocorrelation part of model
- **Estimate parameters / fit your model**
- Check if your residuals are white noise!

Lags in design matrix

To estimate AR(2) model in base R:

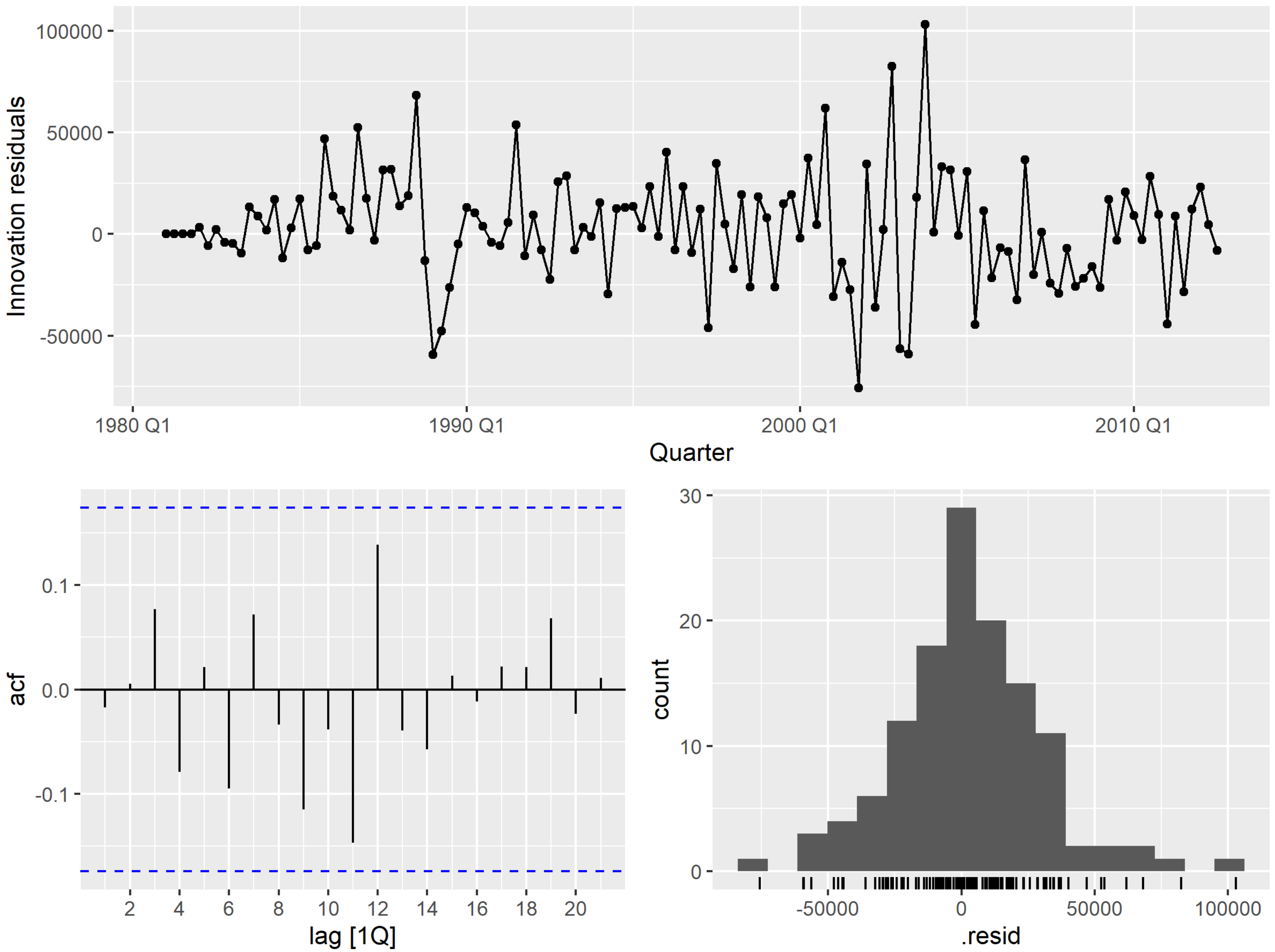
```
lm(y ~ lag(y, 1) + lag(y, 2))
```

y_t	y_{t-1}	y_{t-2}
y_1		
y_2	y_1	
y_3	y_2	y_1
y_4	y_3	y_2
y_5	y_4	y_3
…	…	…
y_T	y_{T-1}	y_{T-2}
	y_T	y_{T-1}
		y_T

Box-Jenkins method

- Check for trends and seasonal components. Remove them if present.
- Choose order / lag of autocorrelation part of model
- Estimate parameters / fit your model
- **Check if your residuals are white noise!**

```
arrival_data |> model(ARIMA(Arrivals)) |> gg_tsresiduals()
```



Moving Average terms

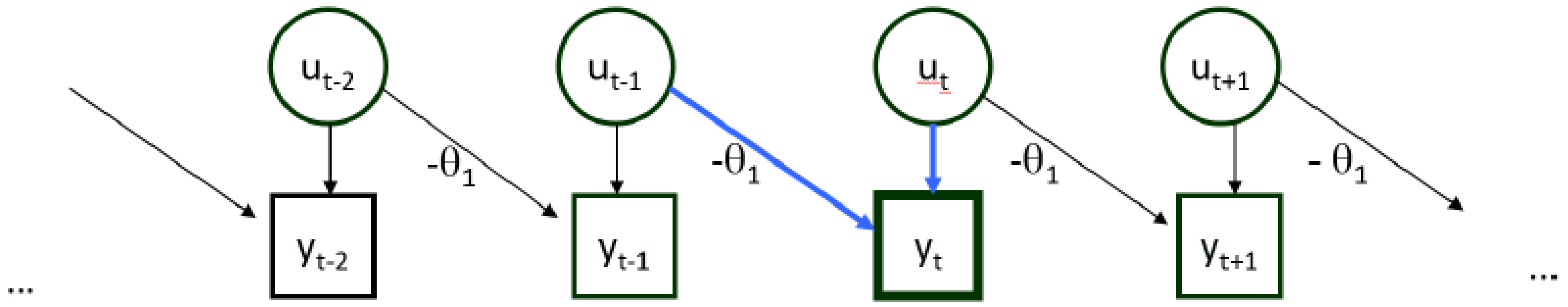
Moving average terms

AR
Current value y_t is dependent on y_{t-k}

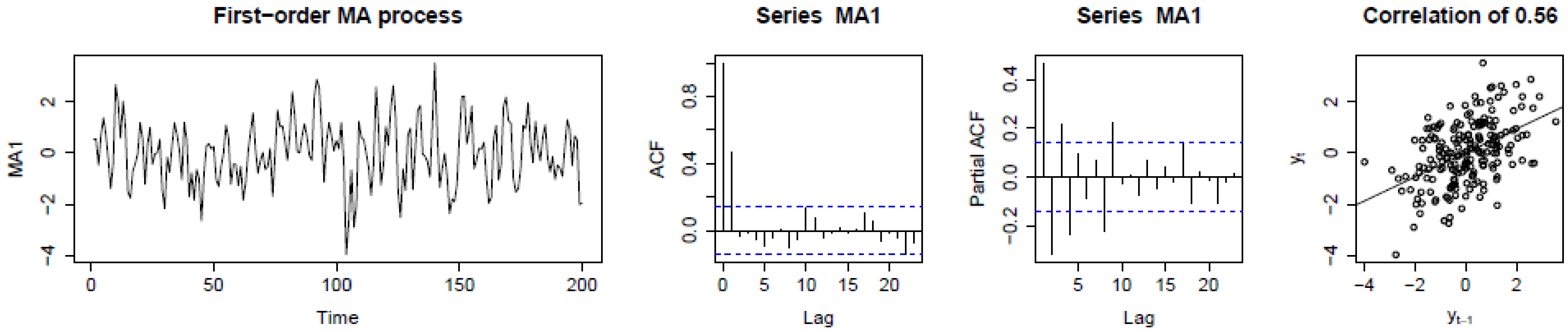
MA
Current value y_t is dependent on ϵ_{t-k}
Dependency on past **shock terms**

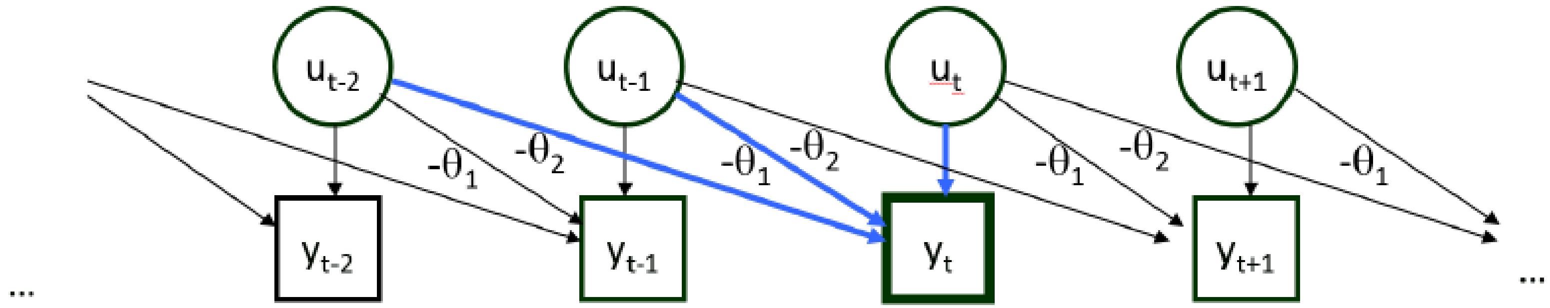
MA(1) model:

$$y_t = c + \epsilon_t + \theta\epsilon_{t-1}$$

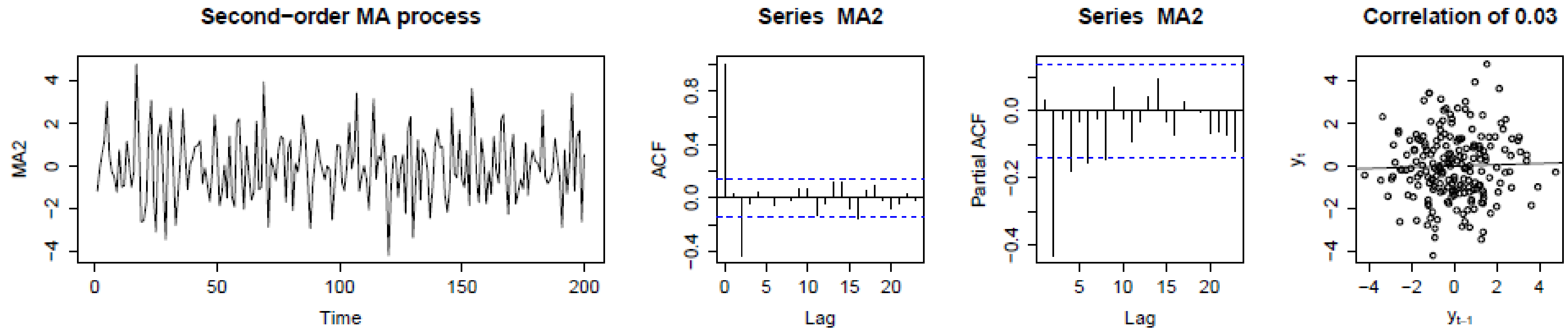


Example with $\theta_1 = 0.9$:





Example with $\theta_1 = 0.7$ and $\theta_2 = -0.9$:



AR / MA interpretation

Granger and Morris (1976):

- An **AR process** is a momentum effect in a random variable that varies smoothly over time

- A **MA process** is a variable in equilibrium, which is perturbed by random shocks with delayed effects

ARMA

ARMA(1, 1) model:

$$y_t = c + \phi y_{t-1} + \epsilon_t + \theta \epsilon_{t-1}$$

ARMA(p,q) models are basic building block of many different time-series techniques.

But: crucial assumption when building ARMA models is **stationarity**, something we have ignored until now

Stationarity

In order to choose appropriate ARMA model for data, we need our time-series to be **stationary**

Definition:

A stationary time series is one whose statistical properties do not depend on the time at which the series is observed.

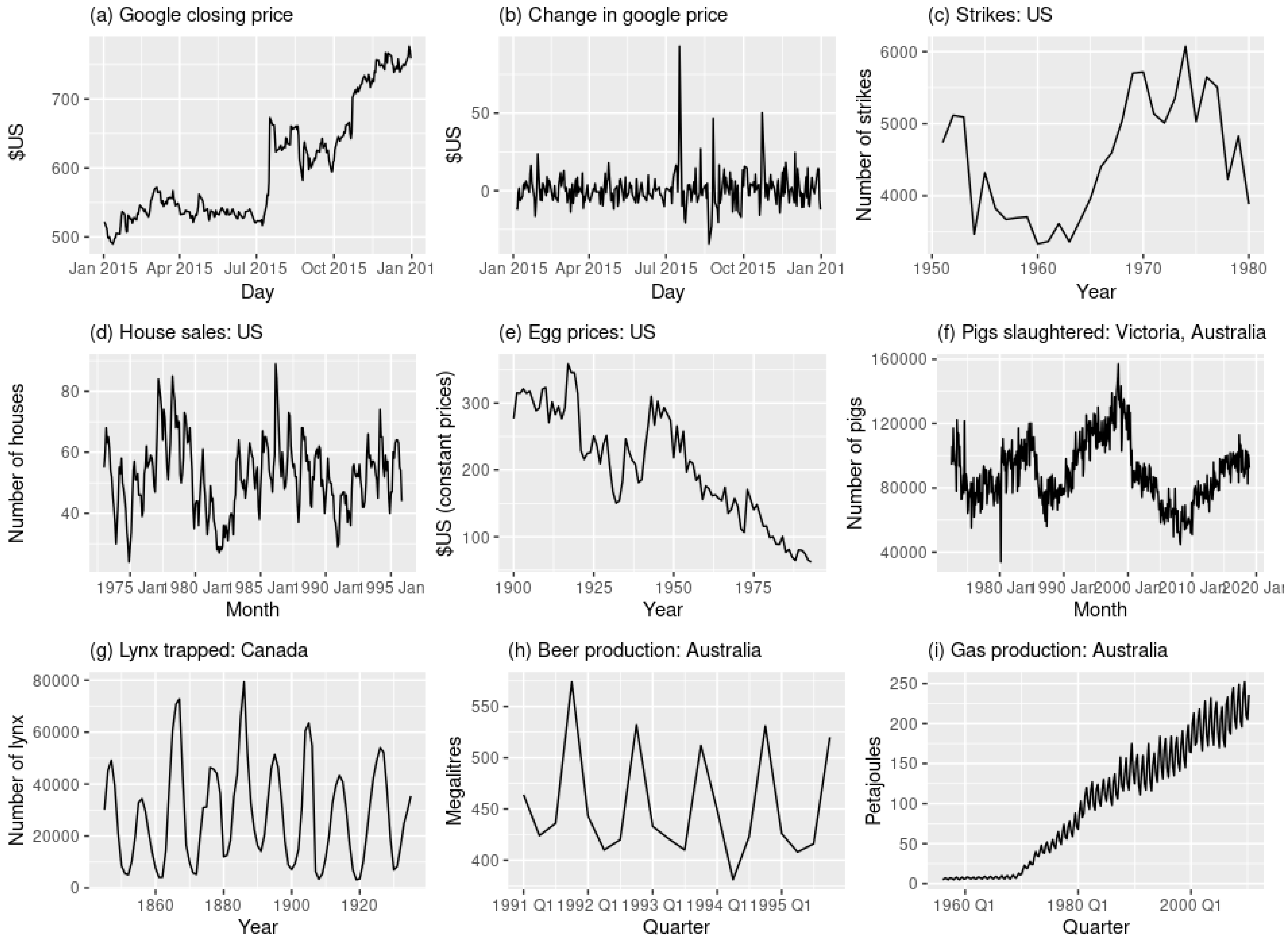
FPP3, section 9.1

Stationarity

Stationary processes are those for which the mean, variance, and auto-correlation structure stay the same over our window of observation.

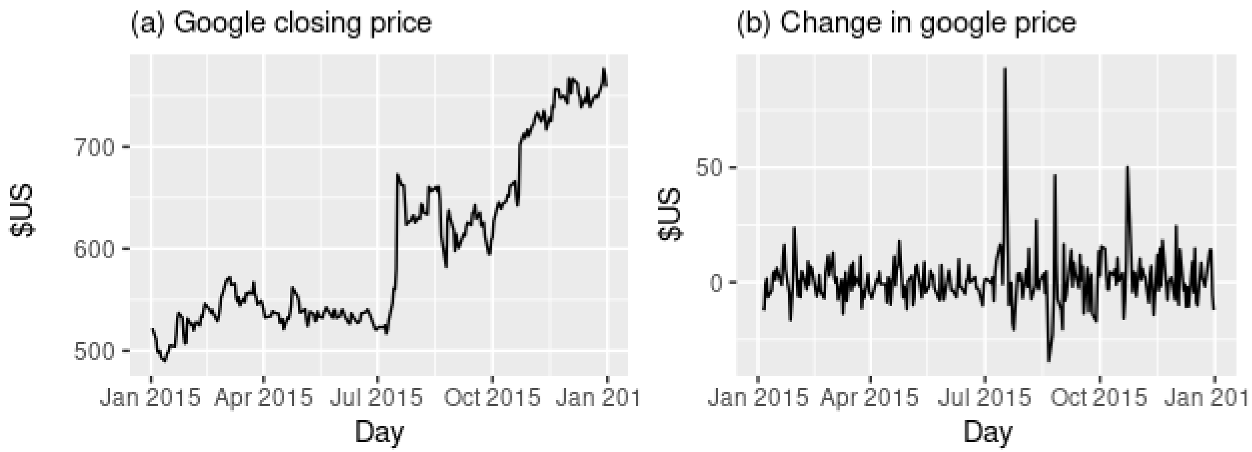
Implication

- Lagged regression parameters stay fixed across waves
- Otherwise ARMA is impossible to estimate!


(a) Google closing price
$US
700
600
500
Jan 2015
Apr 2015
Jul 2015
Oct 2015
Jan 201
Day
(b) Change in google price
$US
50
0
Jan 2015
Apr 2015
Jul 2015
Oct 2015
Jan 201
Day
(c) Strikes: US
Number of strikes
6000
5000
4000
1950
1960
1970
1980
Year
(d) House sales: US
Number of houses
80
60
40
1975 Jan
1980 Jan
1985 Jan
1990 Jan
1995 Jan
Month
(e) Egg prices: US
$US (constant prices)
300
200
100
1900
1925
1950
1975
Year
(f) Pigs slaughtered: Victoria, Australia
Number of pigs
160000
120000
80000
40000
1980 Jan
1990 Jan
2000 Jan
2010 Jan
2020 Jan
Month
(g) Lynx trapped: Canada
Number of lynx
80000
60000
40000
20000
0
1860
1880
1900
1920
Year
(h) Beer production: Australia
Megalitres
550
500
450
400
1991 Q1
1992 Q1
1993 Q1
1994 Q1
1995 Q1
Quarter
(i) Gas production: Australia
Petajoules
250
200
150
100
50
0
1960 Q1
1980 Q1
2000 Q1
Quarter

Achieving stationarity

- Differencing: ARIMA(p, d, q) models
- Essentially: **difference the data** to obtain stationarity



- Doesn't "remove" any information, since the differencing is accounted for when making forecasts

Achieving stationarity

- Basic idea: Even if we have explosive growth, the relationship between current values and change in values is constant over time
- Model selection challenge: still need to select the appropriate order of differencing

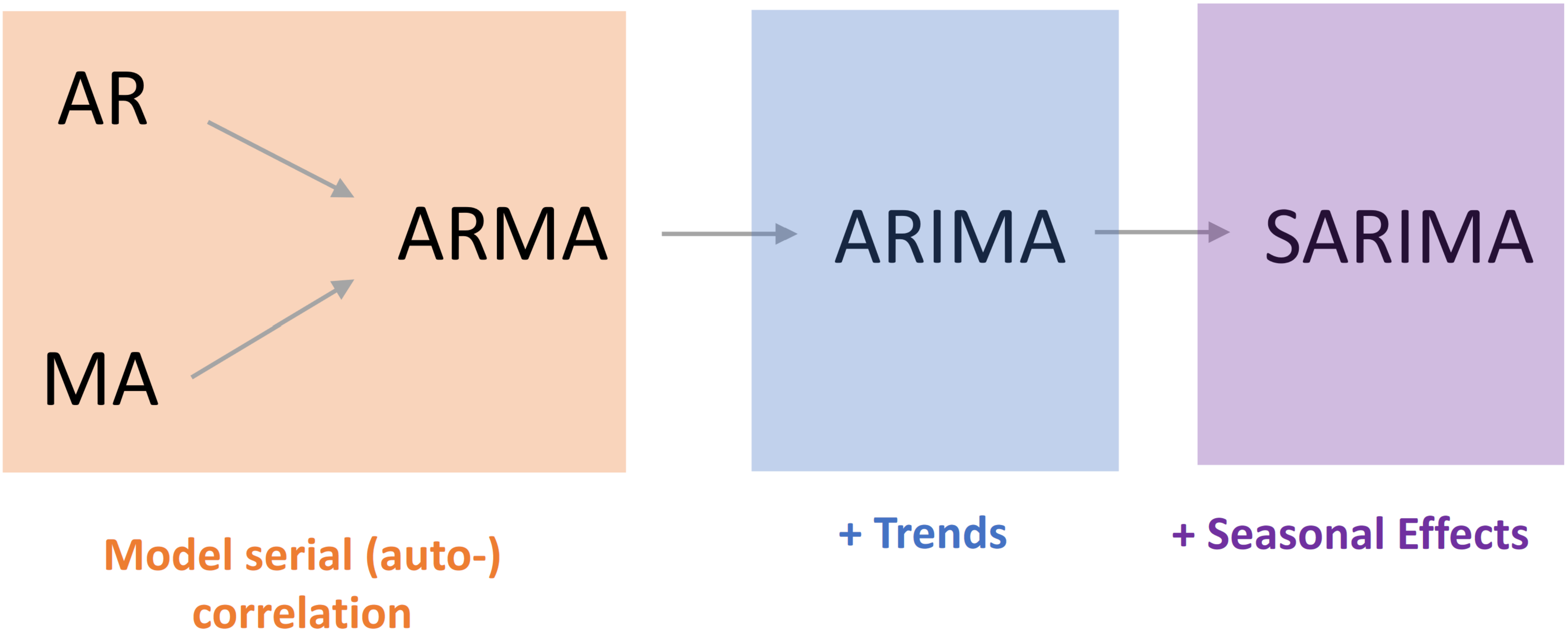
ARIMA(1, 1, 0) model:

$$y_t - y_{t-1} = c + \phi y_{t-1} + \epsilon_t$$

Achieving stationarity

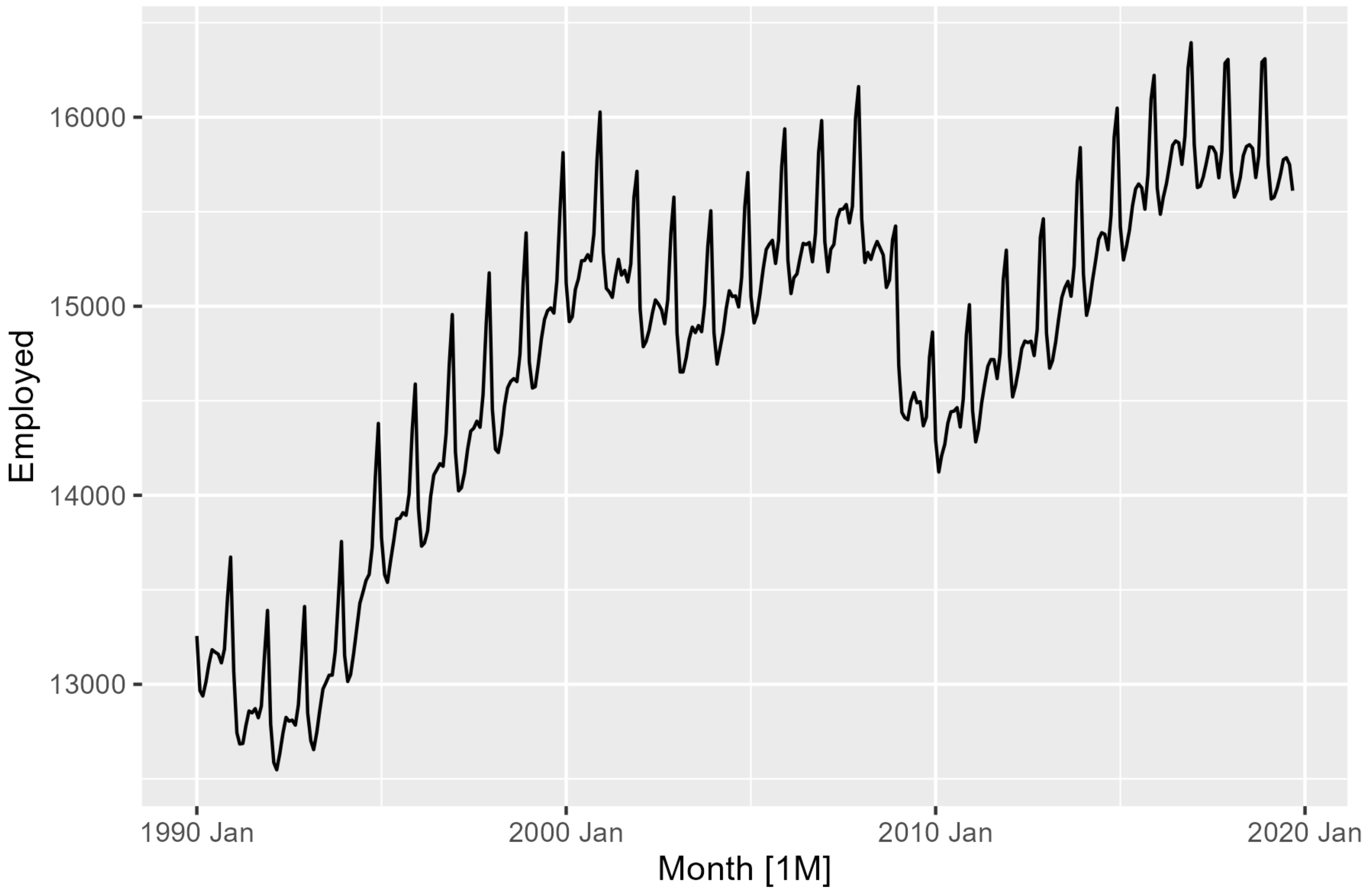
- Using differencing, you can also account for seasonality
- This gets complicated quickly, because you need to back-transform it as well
  - Combination of normal and seasonal differencing
  - 2nd, 3rd order differencing

- Luckily, fable (R package) does all of this for us


AR
MA
ARMA
ARIMA
SARIMA
Model serial (auto-) correlation
+ Trends
+ Seasonal Effects

SARIMA modeling in fable

US retail employment data



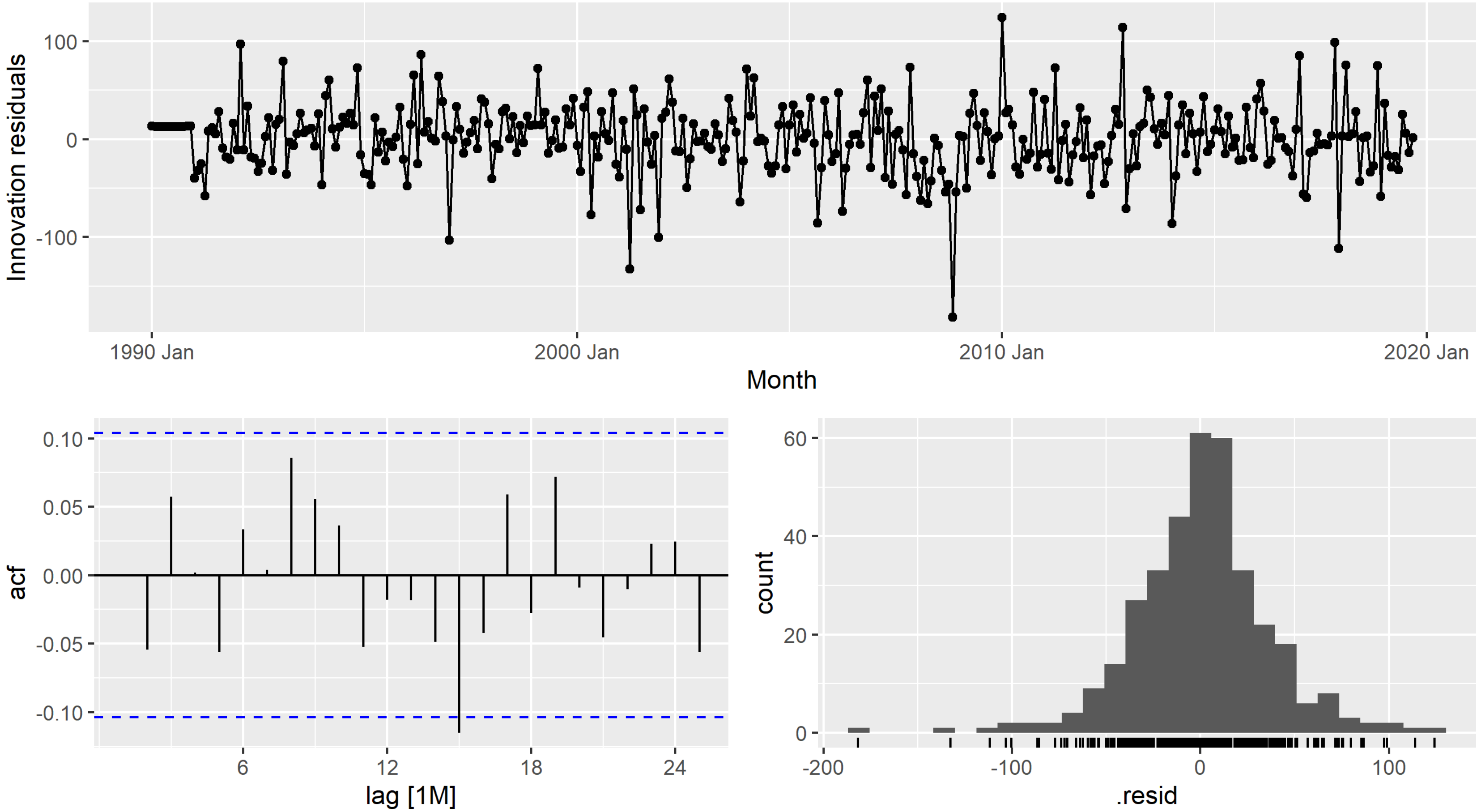
SARIMA modeling in fable

```
# regular ARIMA pdq and seasonal PDQ
frm <- Employed ~ pdq(3, 0, 1) + PDQ(1, 1, 1)

# estimate model
mod <- model(us_retail_employment, ARIMA(frm))

# check residuals
mod |> gg_tsresiduals()
```

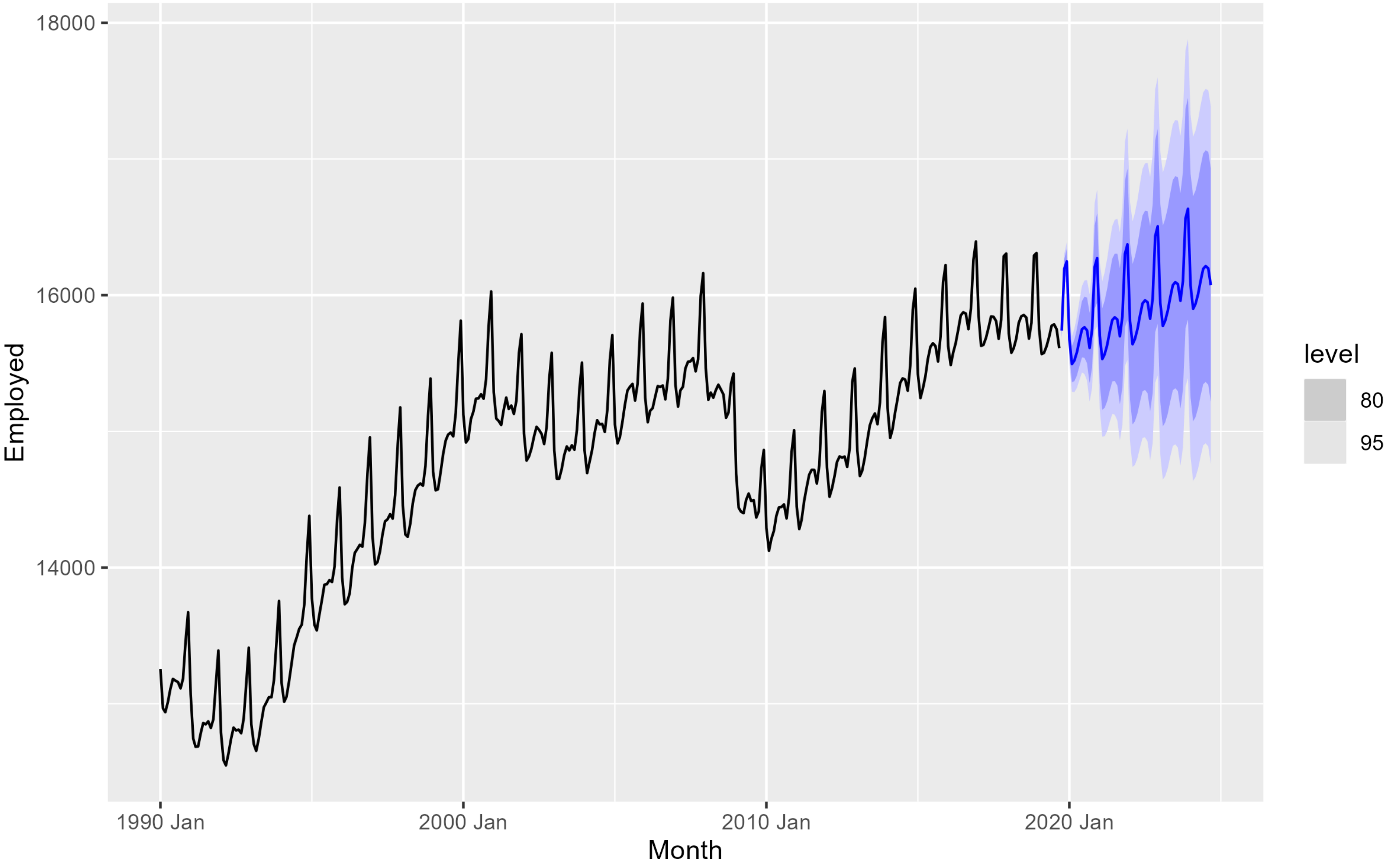
SARIMA modeling in fable



SARIMA modeling in fable

```
# create and plot the forecasts
mod |>
  forecast(h = 60) |> # 60 months ahead
  autoplot(us_retail_employment)
```

SARIMA modeling in fable

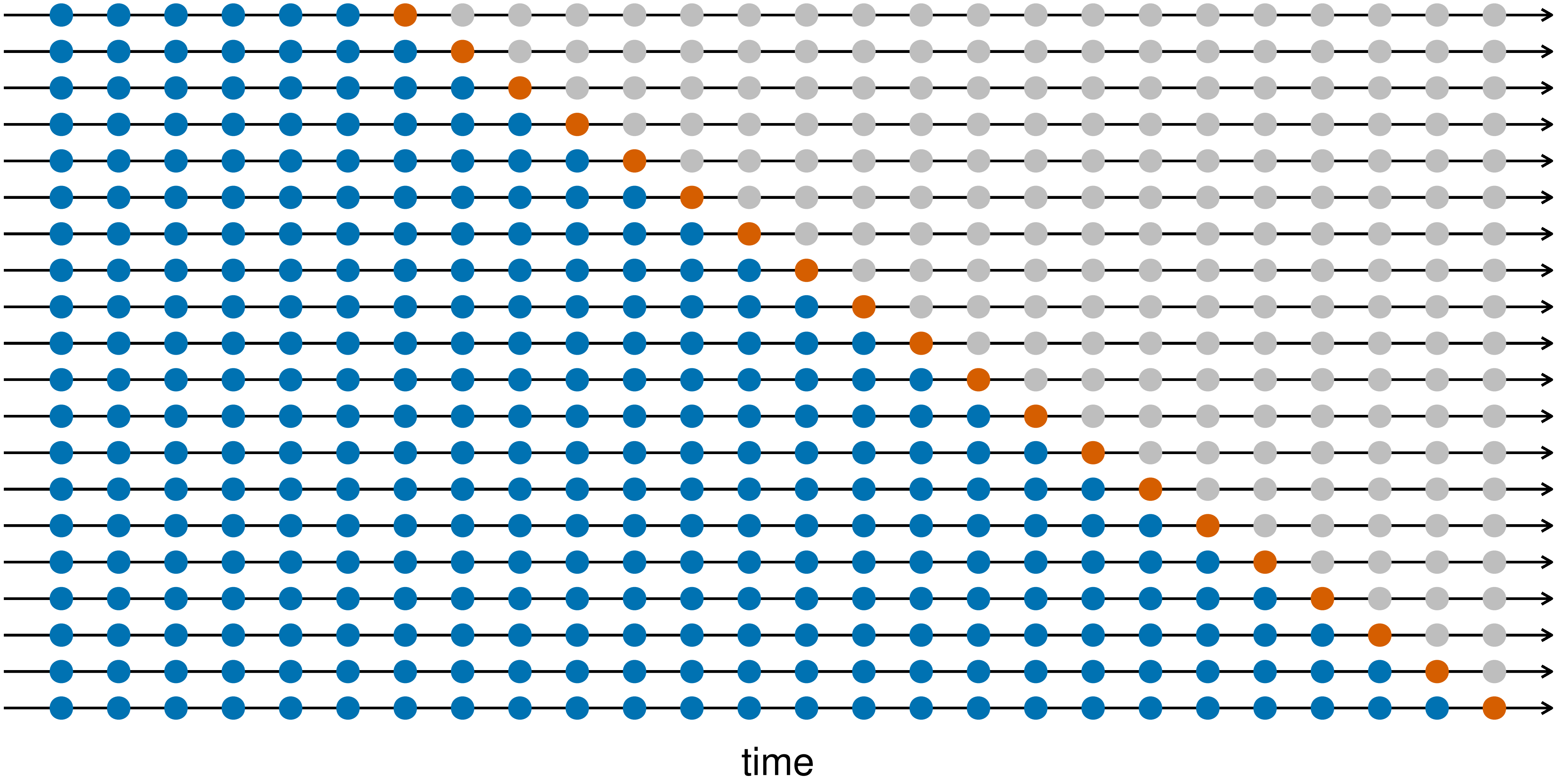


Other topics in time series models

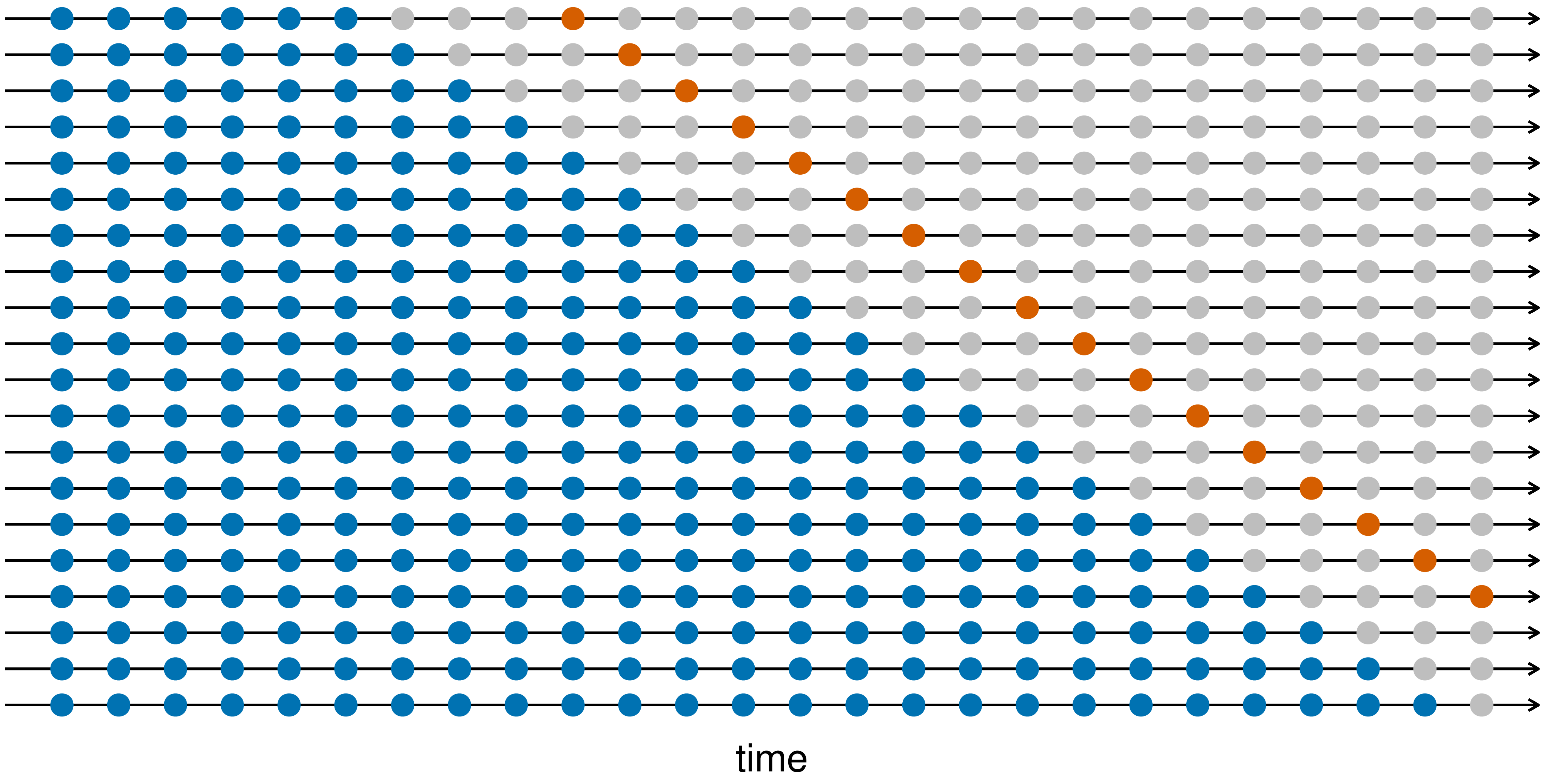
Other topics

- How to do cross-validation
  - (!?) can we even do cross-validation?
- deep autoregression / neural networks
- Causal impact analysis

“cross”-validation



“cross”-validation

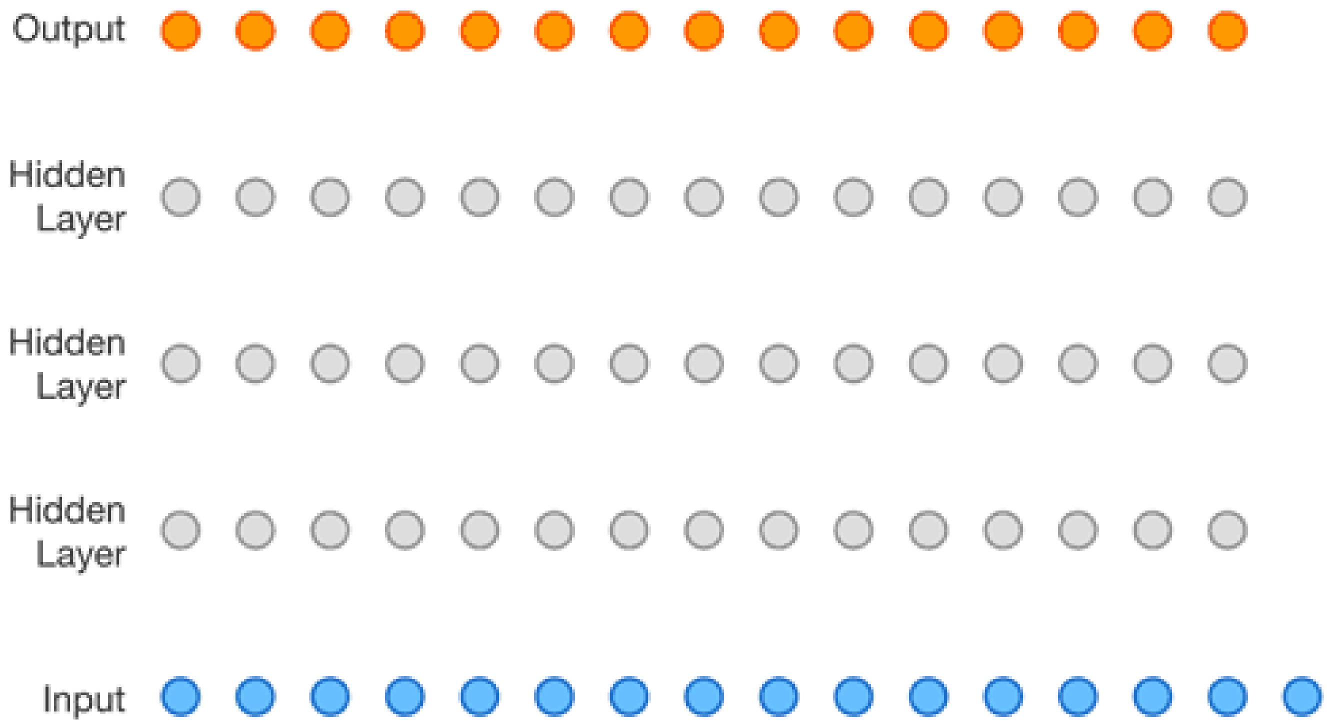


Deep auto-regressive models

Neural networks can also be used for forecasting models

- In fable: NNAR(p, k)
- p is lag and k is number of hidden neurons
- Same benefits as all NN: non-linear relations

Deep auto-regressive models



https://ml.berkeley.edu/blog/posts/AR_intro/

Causal impact

To assess the impact of an event or intervention:

- effect of marketing campaign on sales
- effect of primary school intervention on student results

Steps:

- Train a **good** forecasting model on **pre-intervention data**
- Make a **forecast** for **post-intervention** data
- **Compare** forecast to observed post-intervention data

The Annals of Applied Statistics
2015, Vol. 9, No. 1, 247–274
DOI: 10.1214/14-AOAS788

INFERRING CAUSAL IMPACT USING BAYESIAN STRUCTURAL TIME-SERIES MODELS

BY KAY H. BRODERSEN, FABIAN GALLUSSER, JIM KOEHLER, NICOLAS REMY AND STEVEN L. SCOTT

Google, Inc.

An important problem in econometrics and marketing is to infer the causal impact that a designed market intervention has exerted on an outcome metric over time. This paper proposes to infer causal impact on the basis of a diffusion-regression state-space model that predicts the counterfactual market response in a synthetic control that would have occurred had no intervention taken place. In contrast to classical difference-in-differences schemes, state-space models make it possible to (i) infer the temporal evolution of attributable impact, (ii) incorporate empirical priors on the parameters in a fully Bayesian treatment, and (iii) flexibly accommodate multiple

Causal impact

- causalimpact R package
- Developed at Google in 2015
- Uses “Bayesian Structural Time Series”
    - Essentially, ARIMA model with some extra fanciness

Elon Musk @elonmusk · 1h

Tesla stock price is too high imo

6.2K 8.7K 59K

Elvis
@TradeLikeElvis

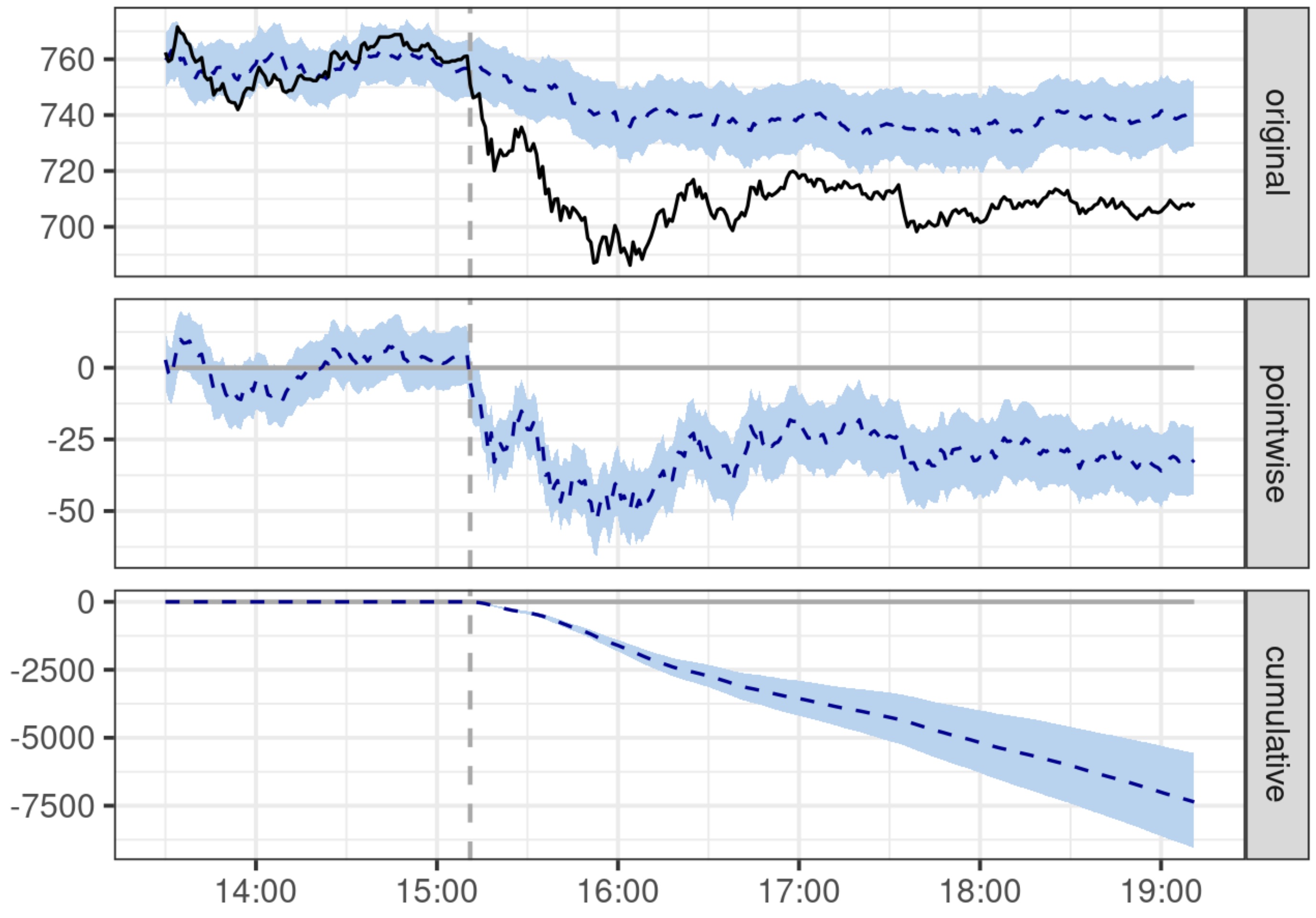
Replying to @elonmusk

Dude...I just lost $10k because of this tweet. Wtf is wrong with u

10:18 AM · May 1, 2020 · Twitter for Android

104 Retweets **1.3K** Likes

https://www.alexpghayes.com/post/2020-05-01_elon-musk-send-tweet/



https://www.alexpghayes.com/post/2020-05-01_elon-musk-send-tweet/