An introduction to time series analysis with ARIMA

LSHTM R Users Group

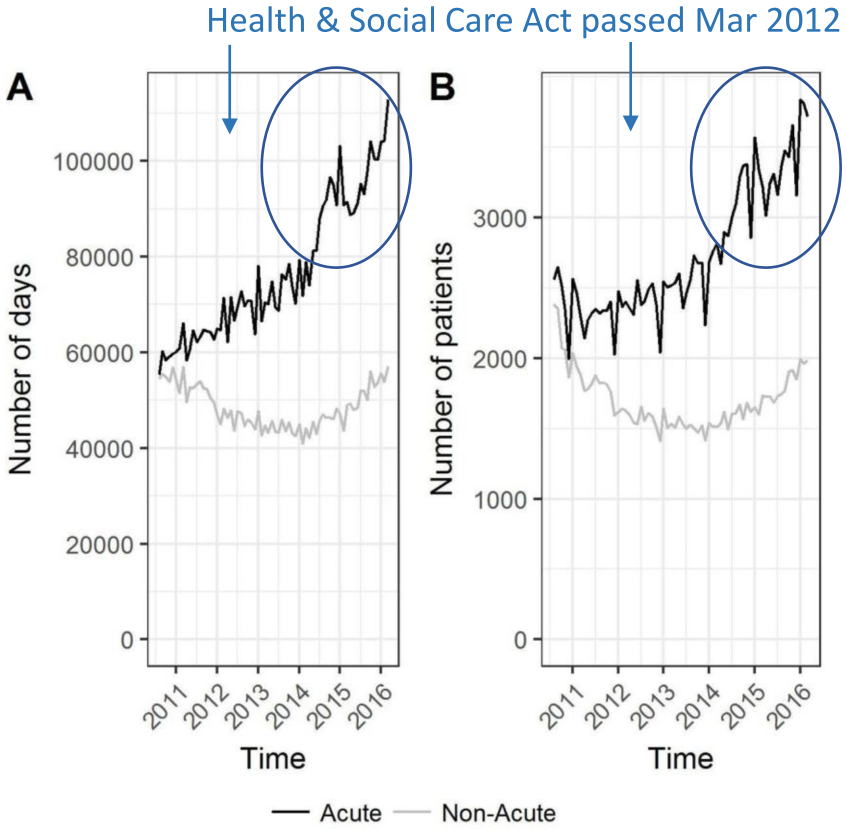
17 December 2020

Julia Shen, julia.shen1@lshtm.ac.uk

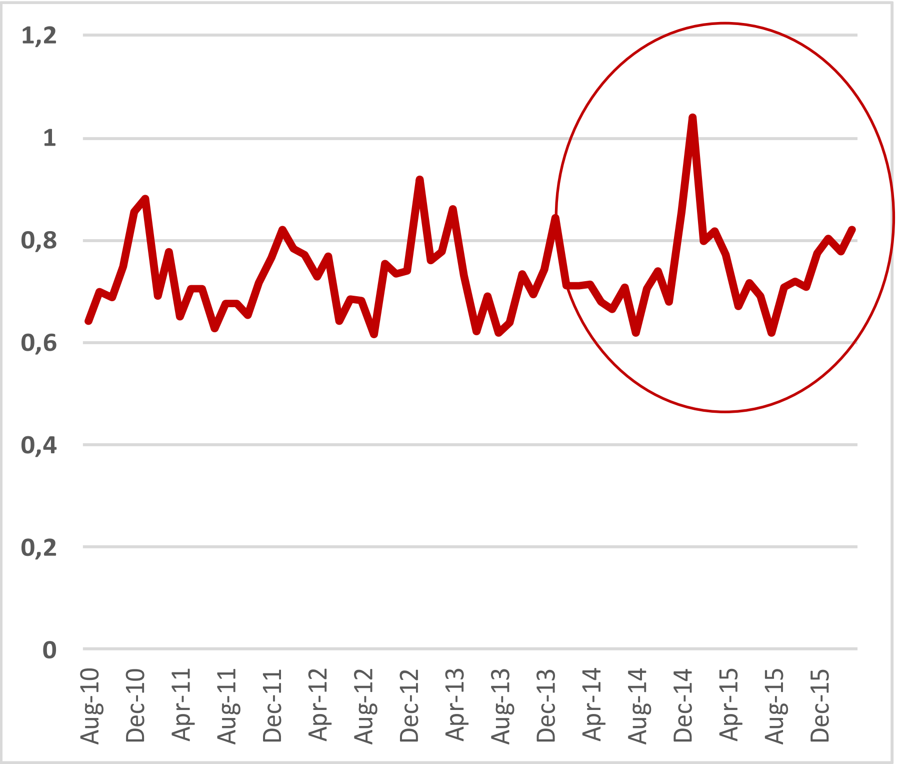
Context & Theory

Does one time series explain another? (motivating case)

Delayed discharge / transfers of care in NHS England[1]



Approximate crude mortality rate/1,000 in England[2]



1 Green MA et al, Could the rise in mortality rates since 2015 be explained by changes in the number of delayed discharges of NHS patients? *J Epi & Comm Hlth* 71(11), 16 Oct 2017.
2 Office for National Statistics, *Monthly figures on deaths registered by area of usual residence in England and Wales* and *England population mid-year estimate*, Aug 2010 – Mar 2016

Does one time series explain another?

Methods Office for National Statistics monthly data of death counts and mortality rates for the period August 2010–March 2016 were compared with delays in discharges from National Health Service (NHS) England data on transfers of care for acute and non-acute patients in England. Autoregressive Integrated Moving Average regression models were used in the analysis.

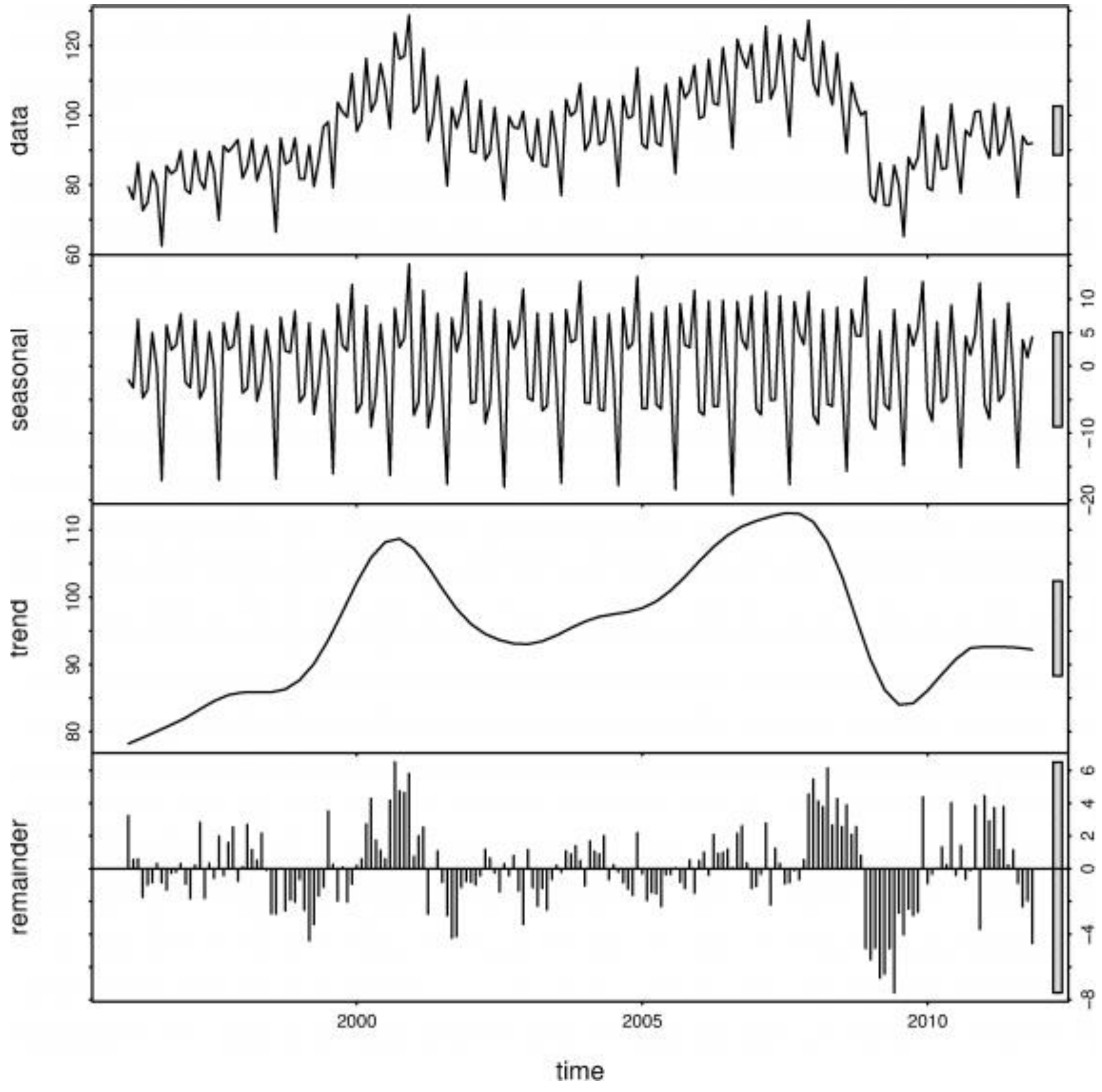
“ ”

Results We estimate that each additional day an acute admission was late being discharged was associated with an increase in 0.394 deaths (95% CIs 0.220 to 0.569). For each additional acute patient delayed being discharged, we found an increase of 7.322 deaths (95% CIs 1.754 to 12.890). Findings for non-acute admissions were mixed.

Conclusion The increased prevalence of patients being delayed in discharge from hospital in 2015 was associated with increases in mortality, accounting for up to a fifth of mortality increases. Our study provides evidence that a lower quality of performance of the NHS and adult social care as a result of austerity may be having an adverse impact on population health.

Green MA et al, Could the rise in mortality rates since 2015 be explained by changes in the number of delayed discharges of NHS patients? *J Epi & Comm Hlth* 71(11), 16 Oct 2017.

Basic framework for time series analysis



y_t (future forecast) data

$=$

S_t seasonal (fixed-period) fluctuations

$+$ or X

T_t **underlying trend**

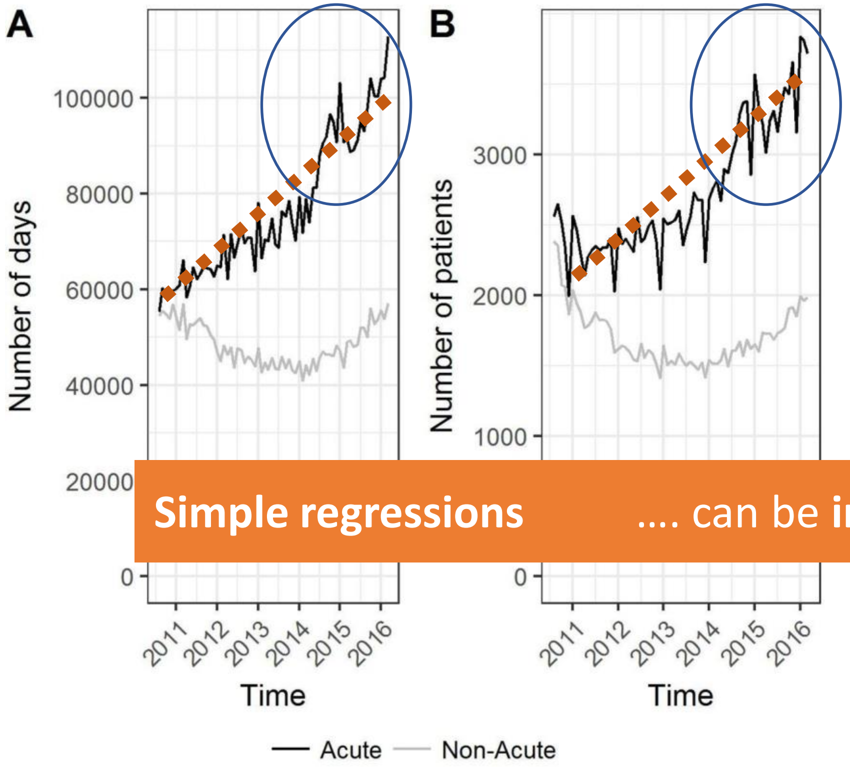
$+$ or X

ε_t random error or varying-period cycle

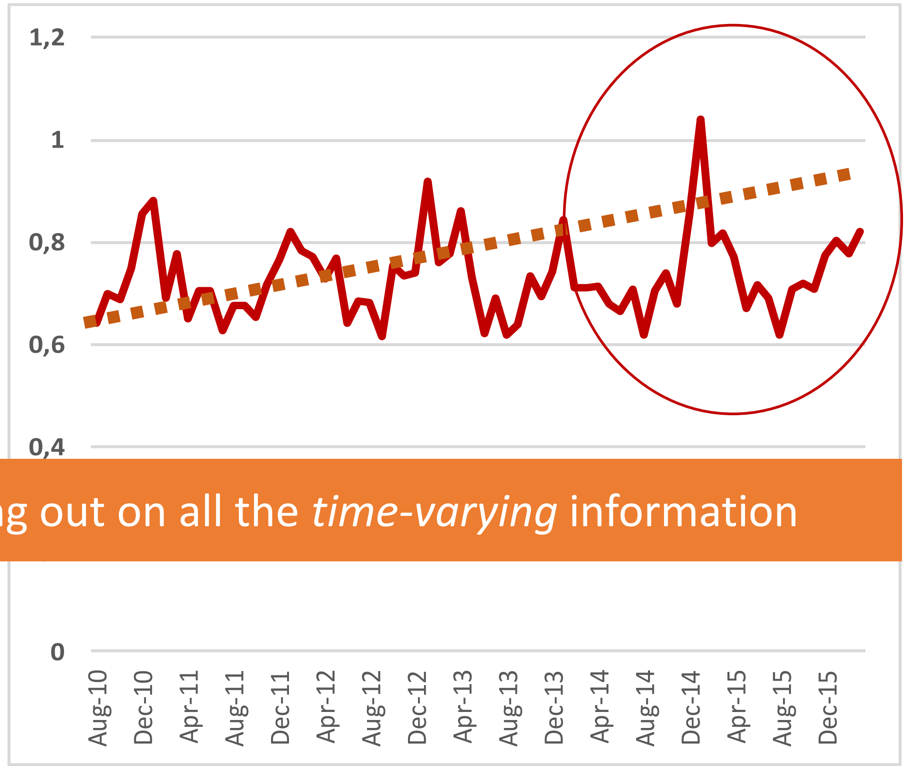
Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

Returning to the motivating case

Delayed discharge / transfers of care in NHS England



Approximate crude mortality rate/1,000 in England



Simple regressions .... can be **inefficient**, missing out on all the *time-varying* information

Green MA et al, Could the rise in mortality rates since 2015 be explained by changes in the number of delayed discharges of NHS patients? *J Epi & Comm Hlth* 71(11), 16 Oct 2017.

What is **ARIMA**?

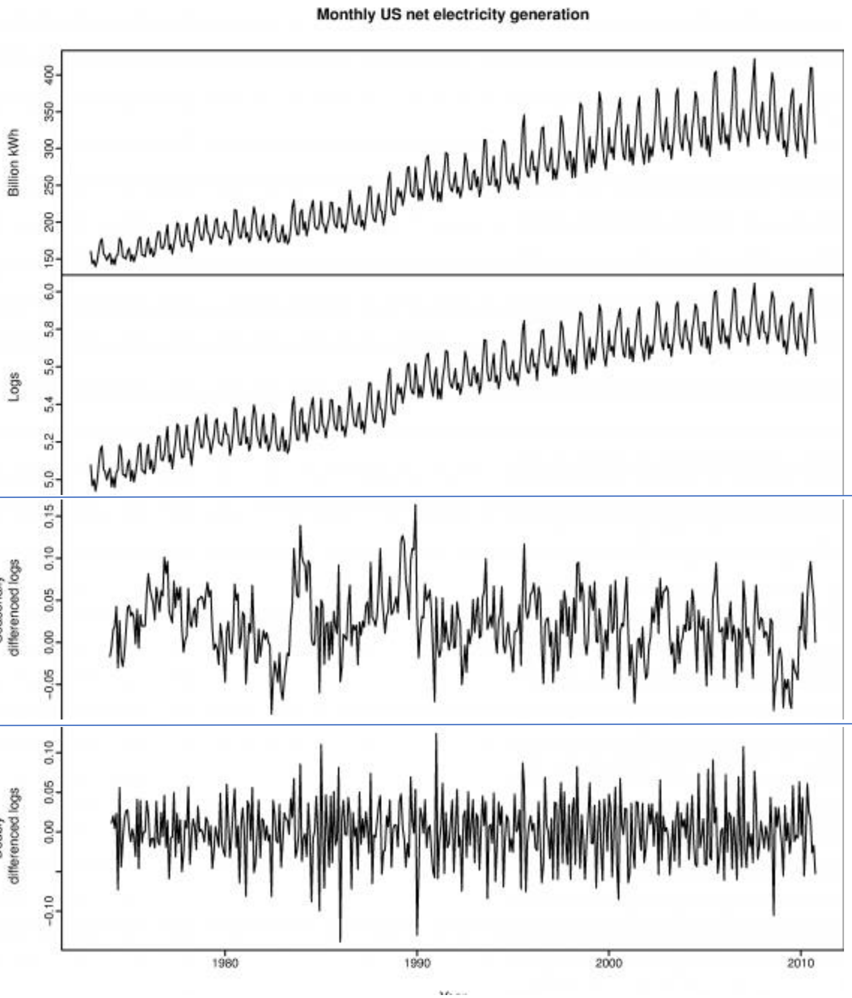
- A standard method for time series analysis, alongside exponential smoothing
- Applicable across econometrics, finance/business, operational modelling, ecology, etc.

AR	I	MA
Auto-Regressive	Integrated	Moving Average

$$(\; p \; , \; d \; , \; q \;)$$

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

What is ARIMA?



'Differencing' of data for stationarity, or integrating across orders of differences

I

Integrated

y_t data

$\ln(y_t)$ (data transformed by logs, or other common methods, e.g. Box-Cox)

d

$$d_1 = y_t - y_{t-1}$$

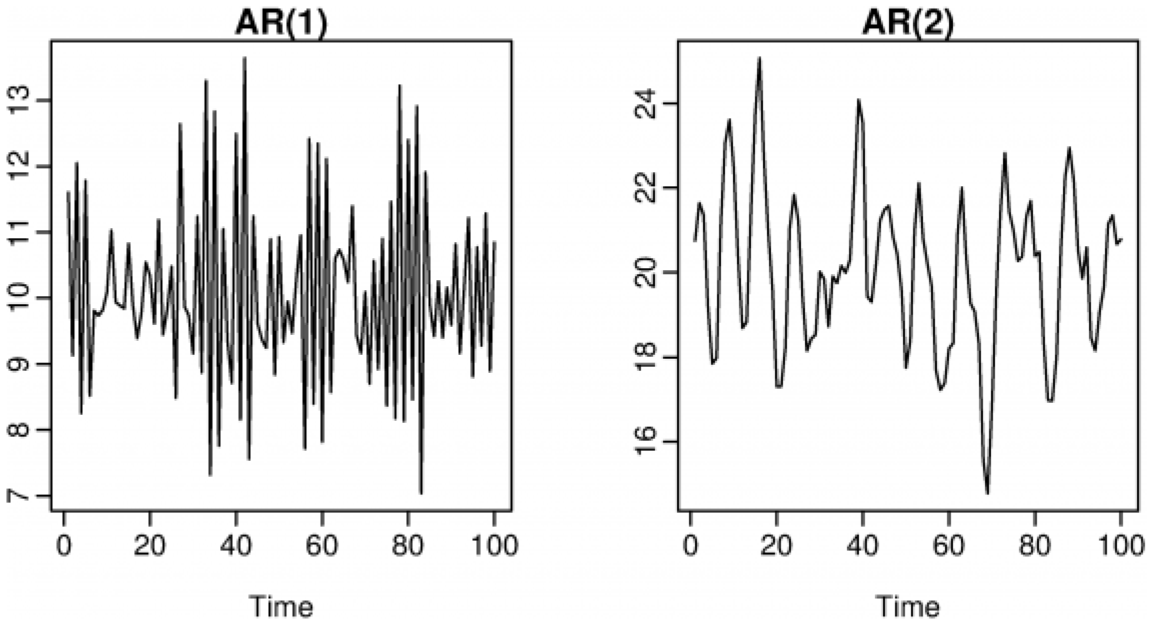
$$d_2 = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$
$$= y_t - 2y_{t-1} + y_{t-2}$$

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

What is ARIMA?

AR

Auto-Regressive



AR(1)
$y_t = 18 - 0.8y_{t-1} + \varepsilon_t$

AR(2)
$y_t = 8 + 1.3y_{t-1} - 0.7y_{t-2} + \varepsilon_t$

For both, $\varepsilon_t \sim$ Norm (0,1)

p

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_n y_{t-n} + \varepsilon_t$$

data

constant

weight 1 • data 1 time period ago

weight 2 • data 2 time periods ago

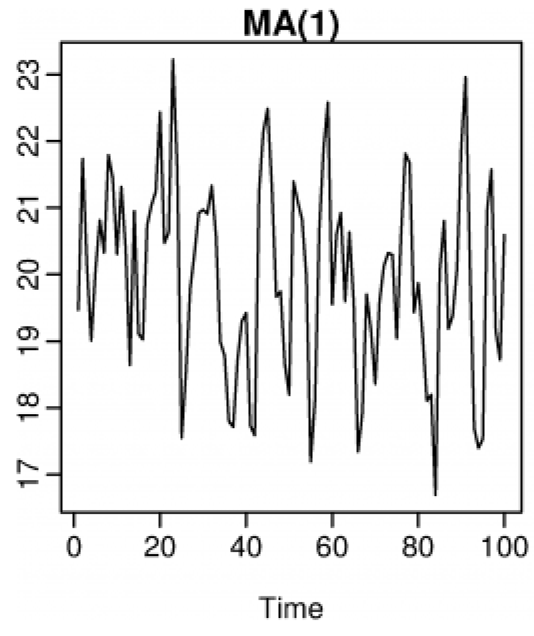
weight *p* • data *p* time periods ago

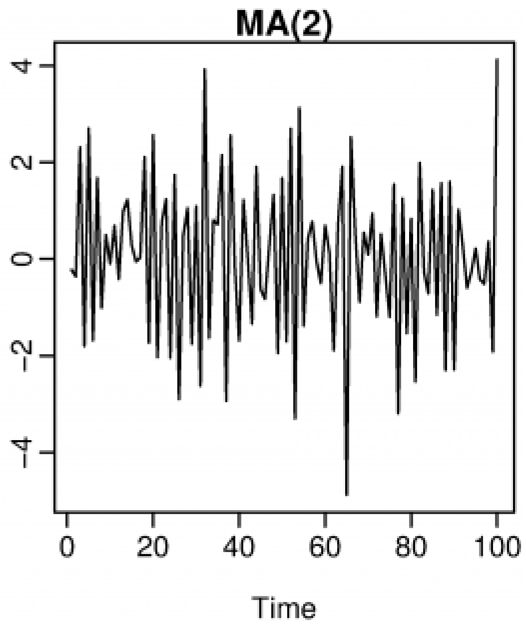
random error

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

What is ARIMA?

MA





MA(1)
$y_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$

MA(2)
$y_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$

For both, $\varepsilon_t \sim$ Norm (0,1)

Moving Average
of error terms

q

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \dots + \theta_q\varepsilon_{t-q}$$

data	constant	random error	weight 1 • error 1 time period ago	weight 2 • error 2 time period ago	weight *q* • random error *q* time periods ago

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

What do we mean by autocorrelation (ACF) and partial ACF? (1/2)

Just as correlation measures the extent of a linear relationship between two variables, **autocorrelation measures the linear relationship between *lagged values* of a time series.**

“ ”

The value of r_k can be written as

$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2},$$

where T is the length of the time series.

When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size… When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags…

When data are both trended and seasonal, you see a combination of these effects.

…. Each partial autocorrelation can be estimated as the last coefficient in an autoregressive model.

Hyndman RJ & Athanasopoulos G, [Forecasting: principles and practice](). 2016.

What do we mean by autocorrelation (ACF) and partial ACF? (2/2)

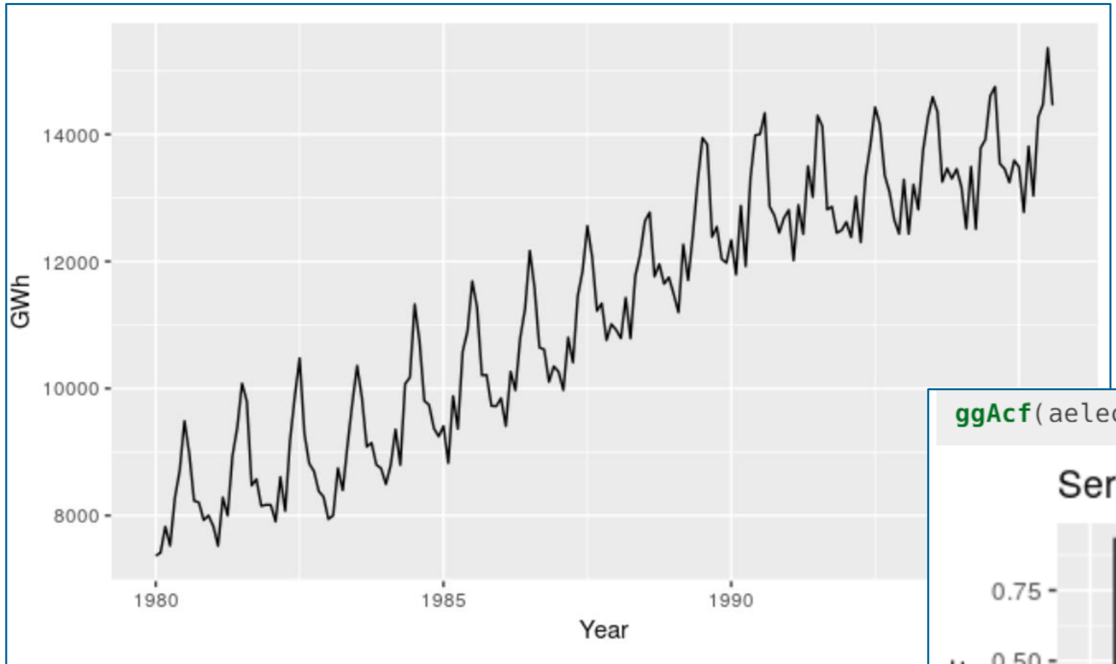


Figure 2.15: Monthly Australian electricity demand from 1980–1995.

```
ggAcf(aelec, lag=48)
```

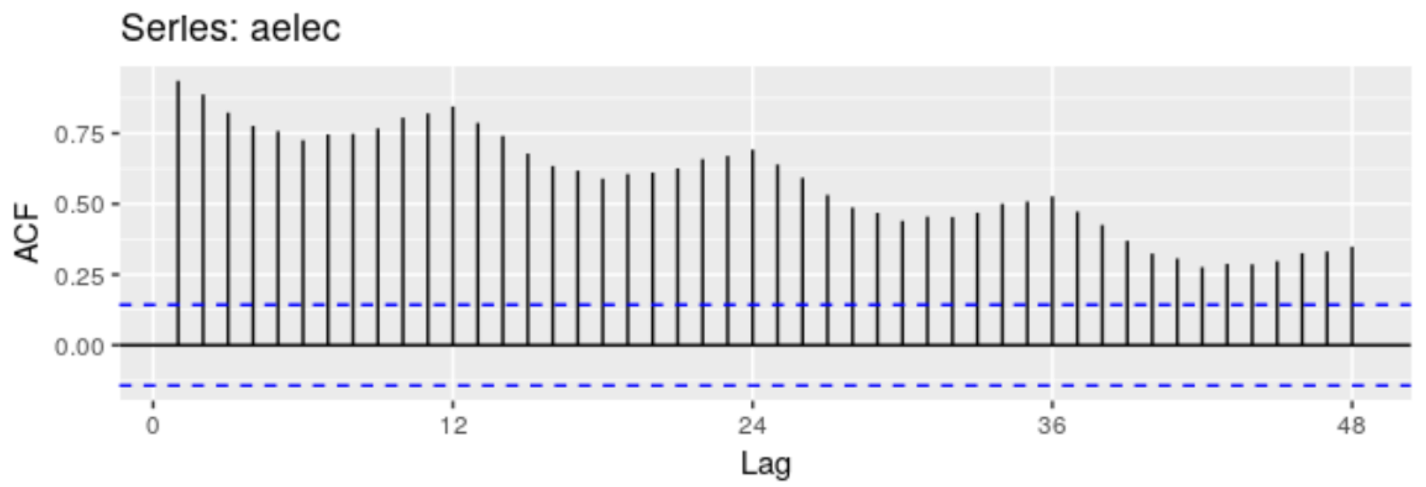


Figure 2.16: ACF of monthly Australian electricity demand.

The slow decrease in the ACF as the lags increase is due to the trend, while the "scalloped" shape is due the seasonality.

Hyndman RJ & Athanasopoulos G, [*Forecasting: principles and practice*]. 2016.

Common ARIMA specifications and seasonality

White noise	**ARIMA(0,0,0)**
Random walk	**ARIMA(0,1,0)**
Random walk with drift	**ARIMA(0,1,0) with c**
Autoregression	**ARIMA(*p*,0,0)**
Moving average	**ARIMA(0,0,*q*)**

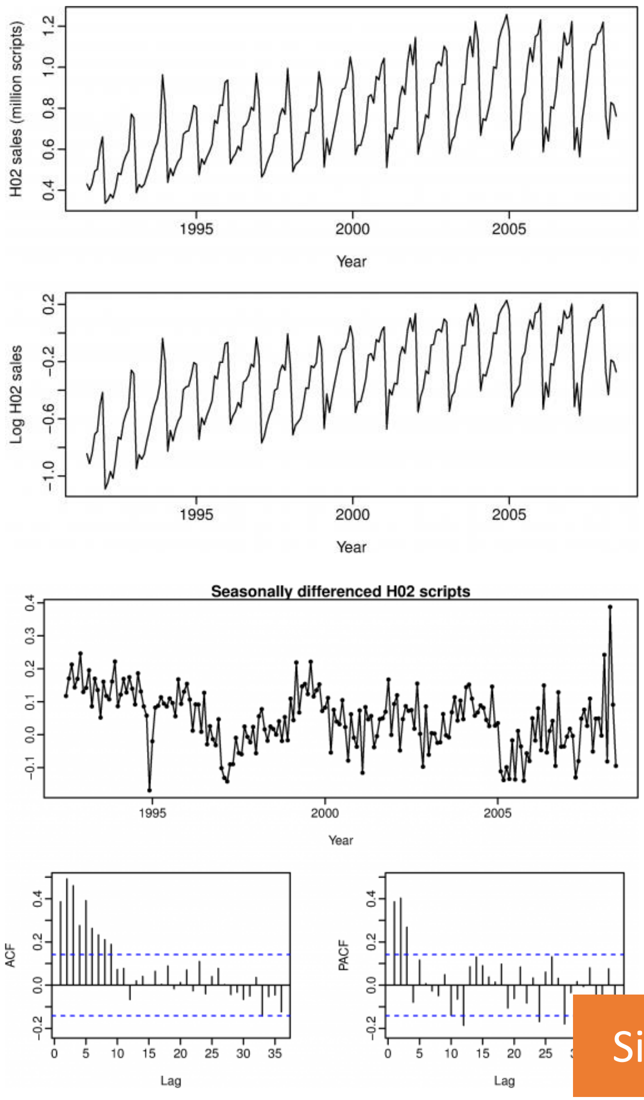
Seasonality

$$\underbrace{\mathrm{ARIMA}(p, d, q)}_{\text{non-seasonal part}}\ \underbrace{(P, D, Q)}_{\text{seasonal part}}{}_{\mathrm{m}}$$

time interval, e.g.
4 for quarters
12 for months

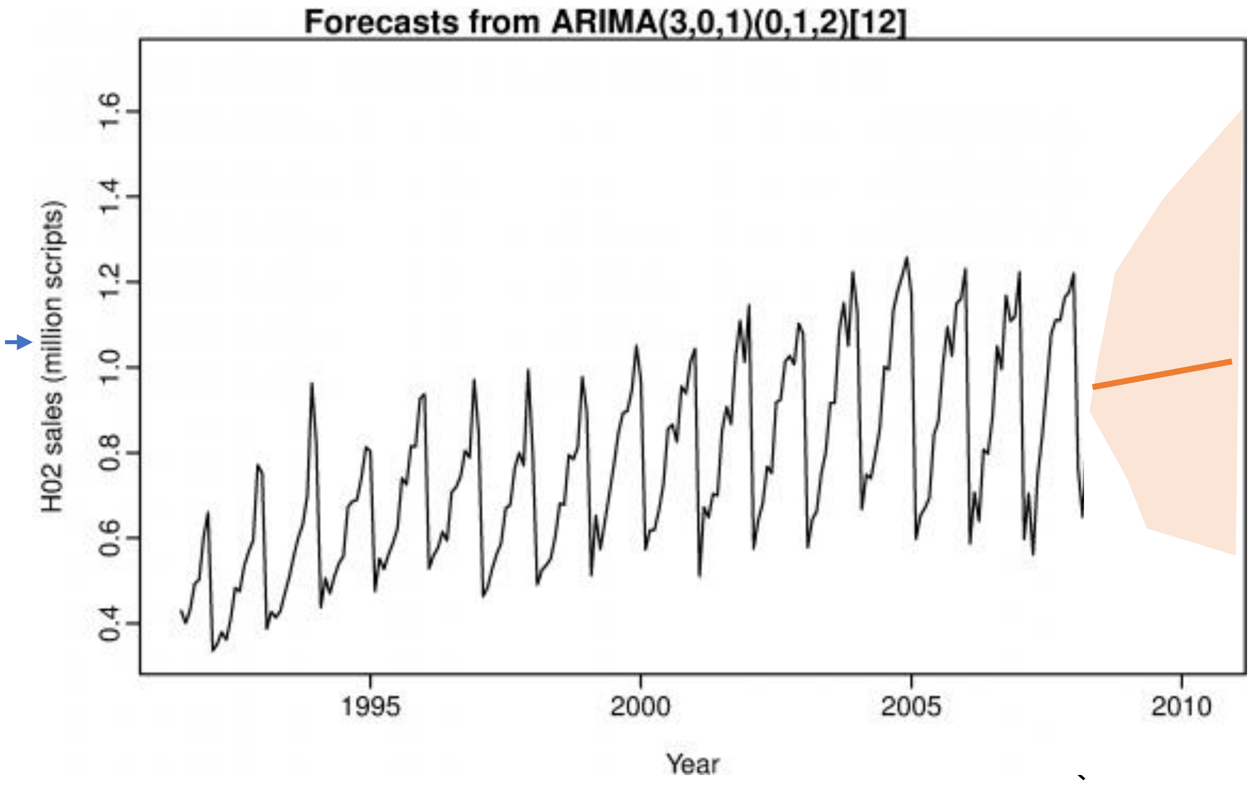
Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

Example of forecasting in seasonal ARIMA (1/2)



Monthly corticosteroid drug sales in Australia

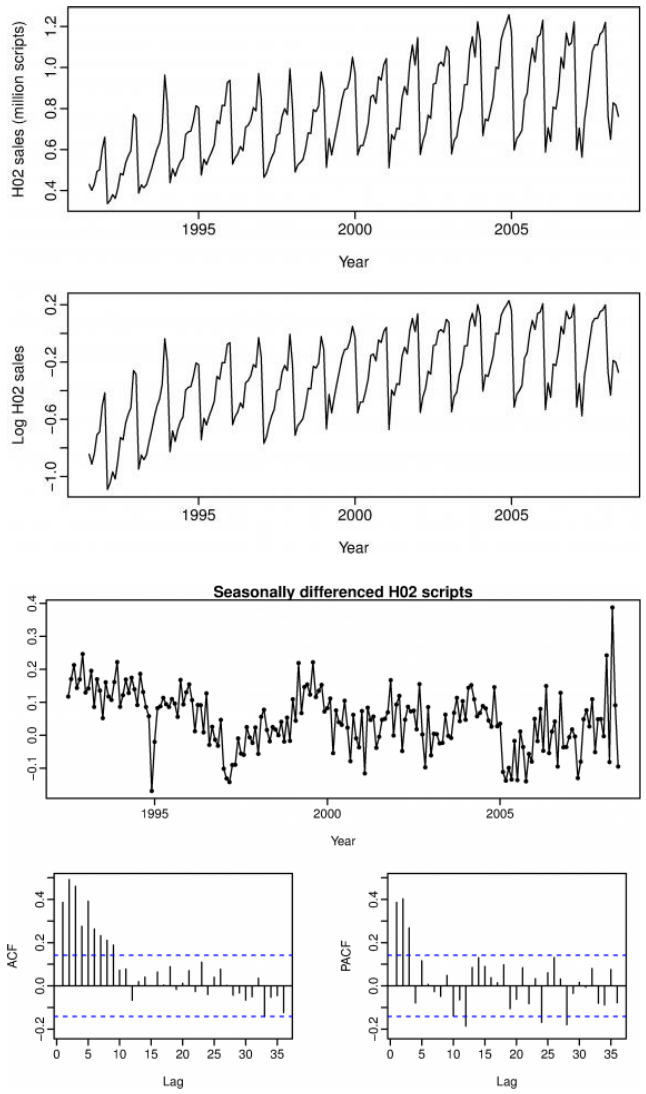
(Corticosteroid sales are highly seasonal as they are used for asthma & allergies)



Simple regressions can be **inefficient**, missing out on all the *time-varying* information

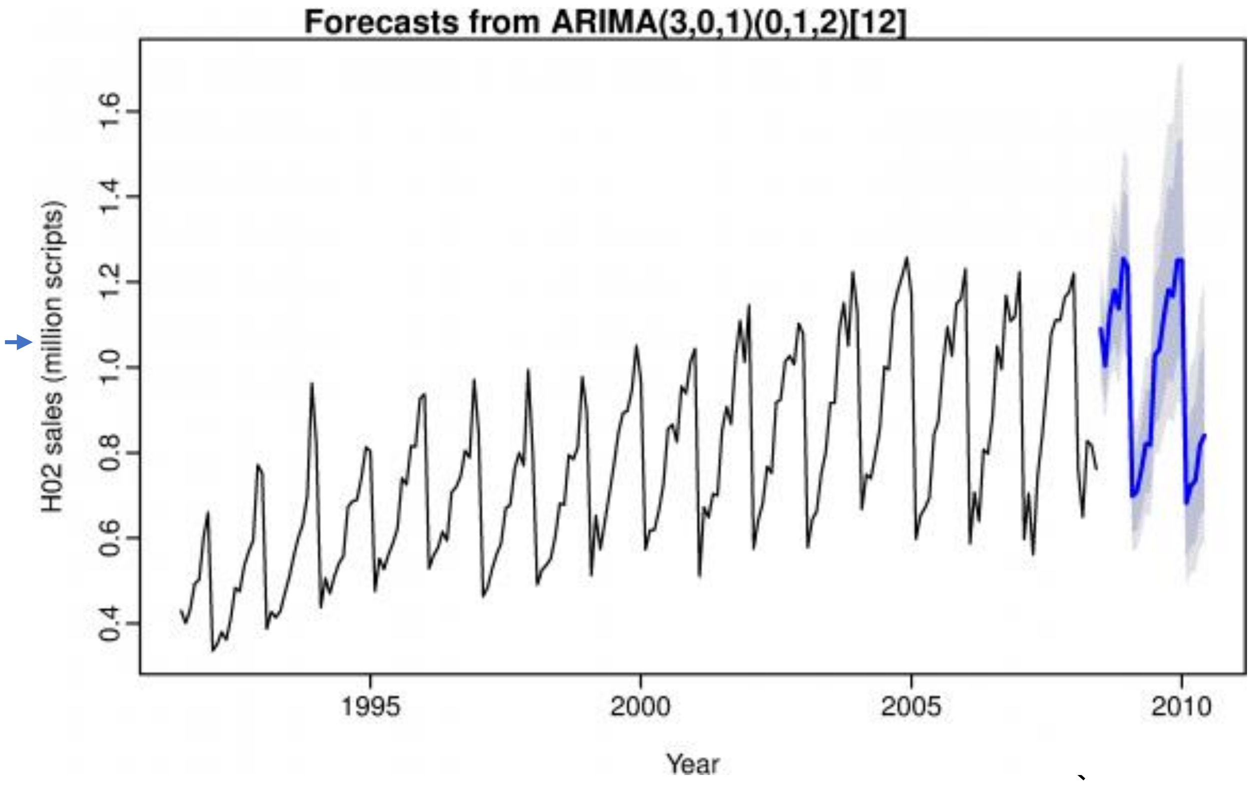
Hyndman RJ & Athanasopoulos G, [Forecasting: principles and practice](). 2016.

Example of forecasting in seasonal ARIMA (2/2)



Monthly corticosteroid drug sales in Australia

(Corticosteroid sales are highly seasonal as they are used for asthma & allergies)



By contrast ARIMA allows the residual time-based data to be meaningfully used.

Hyndman RJ & Athanasopoulos G, [Forecasting: principles and practice](). 2016.

Statistical analysis steps

As usual, it is critical to clarify your design and data assumptions before jumping into the tools and results

1. **Tidy your data,** especially date format
2. **Summarise descriptives**, including any new analysis variables of interest
3. **Explore visual plots** of the raw data across overlapping exposure(s) and outcome(s)
4. **Create time series objects and examine** for seasonality
5. **Decompose** the time series of interest
6. **Evaluate stationarity assumption** of decompositions
7. **Fit univariate ARIMA models**, including future forecasted trends
8. **Check forecasts for fit and predictiveness**, refining and updating as needed
9. **Extend to ARIMA-based regression** between exposures and outcomes

Fitting procedures for reference & practice beyond this session

In R, try the `forecast` package and functions `Arima()` and `auto.arima()`

Iterate ARIMA

i. Identify and **specify** the model

- Make sure variables are stationary
- Identify seasonality in dependent series
- Use plots of autocorrelation and partial autocorrelation functions (ACF & PACF) to determine if/how AR, I, and MA apply

ii. Estimate the parameters

- Minimised conditional sum-of-squares (`method="CSS"`)
- Maximum likelihood (`method="ML"`)
- Default if no missing values (`method="CSS-ML"`)

iii. Check the model

- Check independence of residuals
- Look for constant mean and variance over time
- Carry out Ljung-Box test
- Compare AICc and BIC of different models for ***p*** and ***q***
- Plot ACF and PACF of residuals for ***p*** and ***q*** if either is 0

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

Fitting procedures in this practice case

`auto.arima()` automates the latter tedious steps of modelling using a variation of the Hyndman-Khandakar algorithm

Hyndman-Khandakar algorithm for automatic ARIMA modelling
1. The number of differences $0 \le d \le 2$ is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
a. Four initial models are fitted: ◦ ARIMA$(0, d, 0)$, ◦ ARIMA$(2, d, 2)$, ◦ ARIMA$(1, d, 0)$, ◦ ARIMA$(0, d, 1)$. A constant is included unless $d = 2$. If $d \le 1$, an additional model is also fitted: ◦ ARIMA$(0, d, 0)$ without a constant.
b. The best model (with the smallest AICc value) fitted in step (a) is set to be the "current model".
c. Variations on the current model are considered: ◦ vary p and/or q from the current model by ± 1; ◦ include/exclude c from the current model. The best model considered so far (either the current model or one of these variations) becomes the new current model.
d. Repeat Step 2(c) until no lower AICc can be found.

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*. 2016.

Fitting procedures in this practice case

`auto.arima()` automates the latter tedious steps of modelling as shown at right

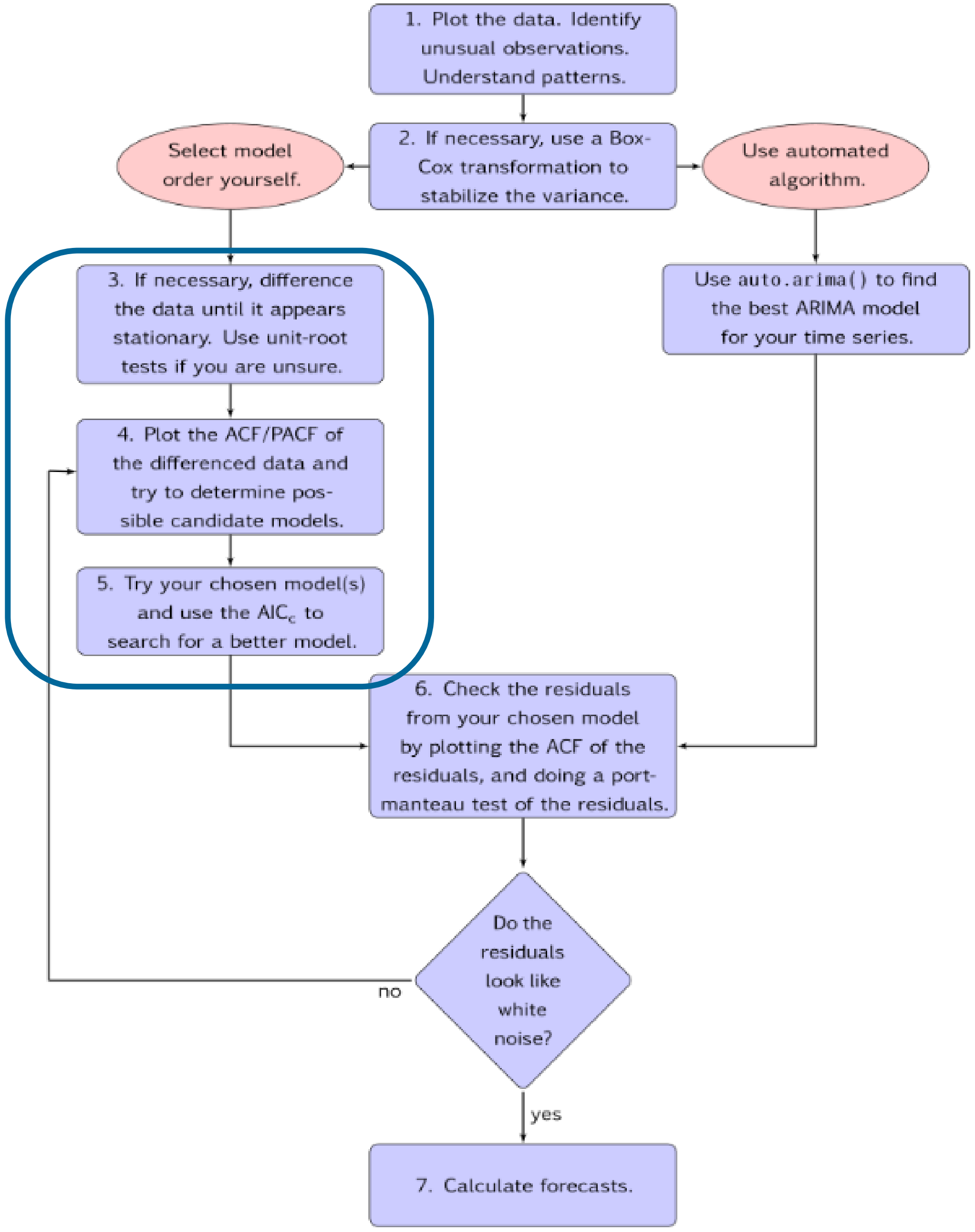


Figure 8.11: General process for forecasting using an ARIMA model.

Hyndman RJ & Athanasopoulos G, [Forecasting: principles and practice](). 2016.

Practice case

Start by loading libraries and packages

```
library(tidyverse)
library(forecast)
library(here)

data_orig <- read.csv(here::here('ARIMA-intro.csv'))
```

ONS_pop_est	out_deaths_total	exp_days_dtoc	exp_days_acute	exp_days_nonacute	exp_patients_dtoc	exp_patients_acute
55692423	33775	109918	55332	54586	3546	1785
55692423	36842	115855	60316	55539	3862	2011
55692423	36159	113246	58362	54884	3653	1883
55692423	39409	113091	59184	53907	3770	1973
55692423	44988	116466	59665	56801	3757	1925
56170927	46835	114346	60125	54221	3689	1940
56170927	36733	112386	60809	51577	4014	2172
56170927	41342	123130	66097	57033	3972	2132
56170927	34521	108064	58407	49657	3602	1947
56170927	37437	113364	60638	52726	3657	1956
56170927	37398	117075	64607	52468	3903	2154
56170927	33275	115517	62140	53377	3726	2005
56170927	35924	117297	63288	54009	3784	2042

1. Tidy your data, especially date format

```
str(data_orig)
levels(data_orig$ï..month)


data_use <- data_orig %>%
  mutate(ï..month=as.character(ï..month)) %>%
  mutate(date=paste0('24', ï..month, year, sep = "", collapse = NULL)) %>%
  mutate(date=as.Date(date, "%d%B%Y")) # convert string to date

data_use <- data_use %>%
select(-ï..month, -year)
View(data_use)
```

	ï..month	year	ONS_pop_est	ou
1	August	2010	55692423	
2	September	2010	55692423	
3	October	2010	55692423	
4	November	2010	55692423	
5	December	2010	55692423	
6	January	2011	56170927	
7	February	2011	56170927	
8	March	2011	56170927	
9	April	2011	56170927	
10	May	2011	56170927	

date
2010-08-24
2010-09-24
2010-10-24
2010-11-24
2010-12-24
2011-01-24
2011-02-24
2011-03-24
2011-04-24
2011-05-24

2. Summarise descriptives, including any new analysis variables of interest

```
data_use <- data_use %>%
 mutate(annual_mort = out_deaths_total * 1000 / ONS_pop_est,
         dtoc_days_rate = exp_days_dtoc * 1000 / ONS_pop_est,
         dtoc_pts_rate = exp_patients_dtoc * 100000 / ONS_pop_est,
         dtoc_days_percapita = exp_days_dtoc / exp_patients_dtoc,
         prop_acute_days = exp_days_acute / exp_days_dtoc,
         prop_acute_patients = exp_patients_acute / exp_patients_dtoc)

summary(data_use)
```

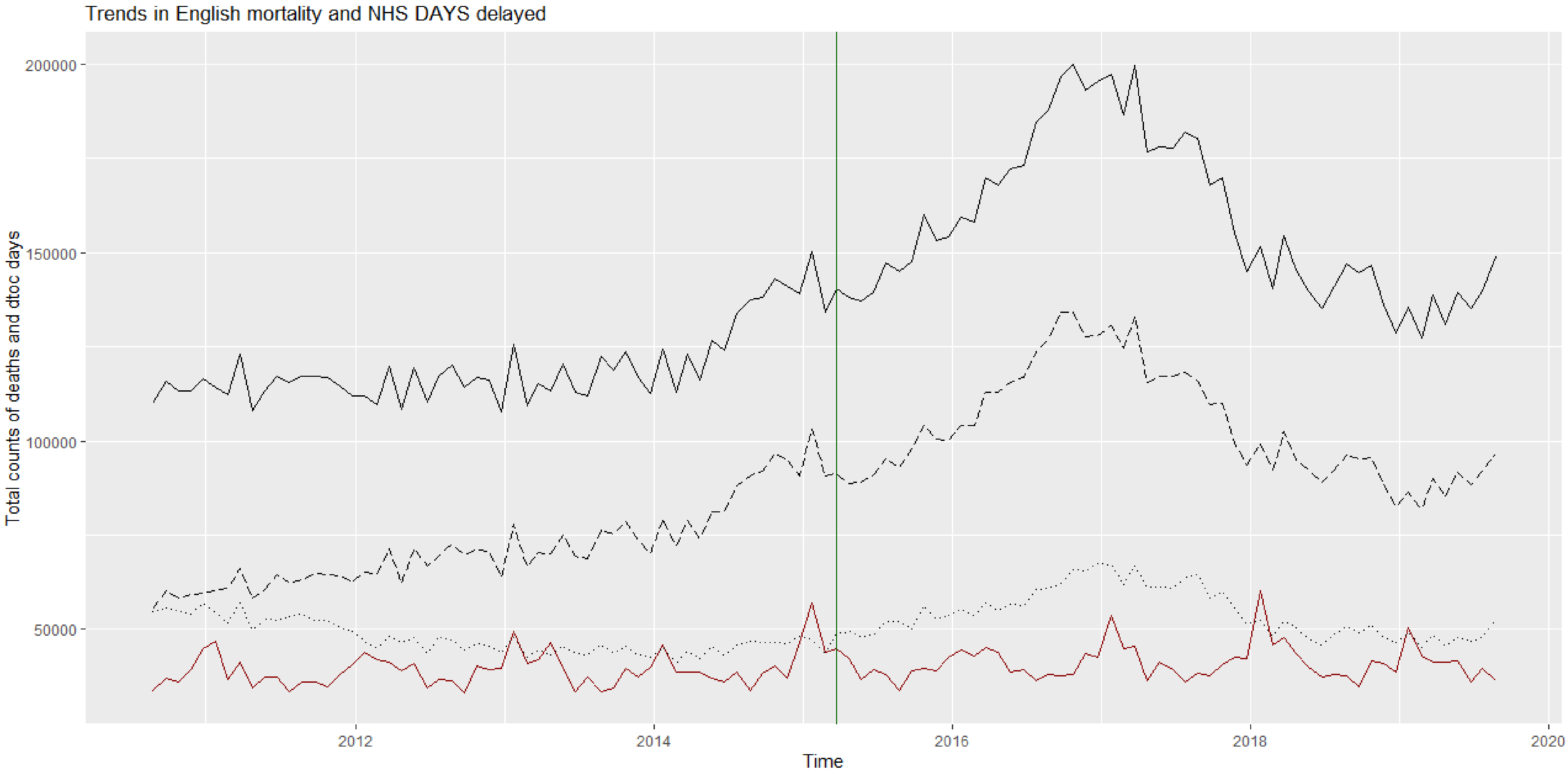
2. Summarise descriptives, including any new analysis variables of interest

```
> summary(data_use) # note data descriptives, and also the NAs. What data are missing?
  ONS_pop_est         out_deaths_total exp_days_dtoc     exp_days_acute    exp_days_nonacute exp_patients_dtoc
 exp_patients_acute
 Min.   :55692423   Min.   :32934    Min.   :107652   Min.   : 55332   Min.   :40973     Min.   :3473
 Min.   :1785
 1st Qu.:56567796   1st Qu.:36967    1st Qu.:116878   1st Qu.: 70028   1st Qu.:46006     1st Qu.:3862
 1st Qu.:2305
 Median :57408654   Median :39378    Median :135428   Median : 88686   Median :49185     Median :4485
 Median :2927
 Mean   :57555780   Mean   :40121    Mean   :138665   Mean   : 87697   Mean   :50968     Mean   :4556
 Mean   :2882
 3rd Qu.:58381217   3rd Qu.:42295    3rd Qu.:151383   3rd Qu.: 99627   3rd Qu.:54586     3rd Qu.:4968
 3rd Qu.:3298
 Max.   :59115809   Max.   :60075    Max.   :200095   Max.   :134256   Max.   :67444     Max.   :6660
 Max.   :4475
 NA's   :8

 exp_patients_nonacute    date                annual_mort       dtoc_days_rate   dtoc_pts_rate     dtoc_days_pe
rcapita
 Min.   :1371          Min.   :2010-08-24   Min.   :0.5822   Min.   :1.903   Min.   : 6.140   Min.   :28.0
0
 1st Qu.:1522          1st Qu.:2012-11-24   1st Qu.:0.6441   1st Qu.:2.057   1st Qu.: 6.804   1st Qu.:30.0
0
 Median :1635          Median :2015-02-24   Median :0.6758   Median :2.321   Median : 7.629   Median :31.0
0
 Mean   :1675          Mean   :2015-02-22   Mean   :0.6955   Mean   :2.406   Mean   : 7.905   Mean   :30.4
4
 3rd Qu.:1797          3rd Qu.:2017-05-24   3rd Qu.:0.7341   3rd Qu.:2.641   3rd Qu.: 8.582   3rd Qu.:31.0
0
 Max.   :2212          Max.   :2019-08-24   Max.   :1.0162   Max.   :3.427   Max.   :11.337   Max.   :31.0
0
                                            NA's   :8        NA's   :8       NA's   :8

 prop_acute_days   prop_acute_patients
 Min.   :0.5034    Min.   :0.5034
 1st Qu.:0.6088    1st Qu.:0.6087
 Median :0.6464    Median :0.6464
 Mean   :0.6269    Mean   :0.6269
 3rd Qu.:0.6575    3rd Qu.:0.6576
 Max.   :0.6860    Max.   :0.6860
```

3. **Explore visual plots** of the raw data across overlapping exposure(s) and outcome(s)

```
plot_counts <- ggplot(data=data_use, aes(x=date)) +
  ggtitle("Trends in English mortality and NHS DAYS delayed") +
    geom_line(aes(y=out_deaths_total), color="darkred") +
  geom_line(aes(y=exp_days_dtoc), color="black") +
  geom_line(aes(y=exp_days_acute), color="black", linetype="longdash") +
  geom_line(aes(y=exp_days_nonacute), color="black", linetype="dotted") +
  xlab("Time") + ylab("Total counts of deaths and dtoc days") +
  geom_vline(aes(xintercept = as.numeric(as.Date("2015/03/24"))), color="darkgreen")
windows(); plot_counts
```

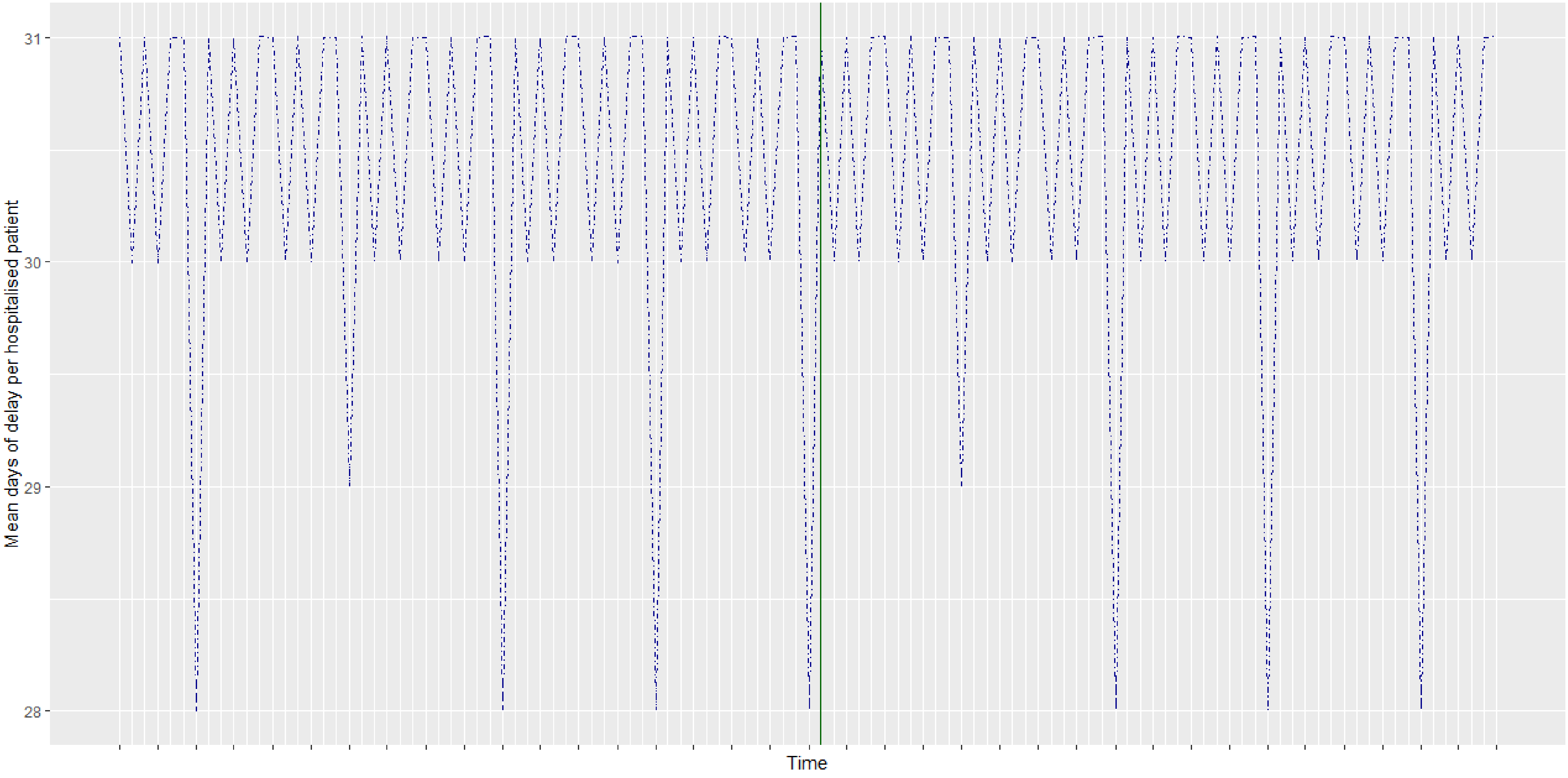
3. Explore visual plots of the raw data across overlapping exposure(s) and outcome(s)



3. **Explore visual plots** of the raw data across overlapping exposure(s) and outcome(s)

```
plot_dtoc_percap <- ggplot(data=data_use, aes(x=date)) +
  geom_line(aes(y=dtoc_days_percapita), color="darkblue", linetype="dotdash") +
  xlab("Time") + ylab("Mean days of delay per hospitalised patient") +
  scale_x_date(minor_breaks=seq(as.Date("2010/08/24"), as.Date("2019/08/24"),
                                by="months"), breaks=seq(as.Date("2010/08/24"),
                                                         as.Date("2019/08/24"),
                                                        by="quarter")) +
 geom_vline(aes(xintercept = as.numeric(as.Date("2015/03/24"))), color="darkgreen") +
 theme(axis.text.x = element_blank())
windows(); plot_dtoc_percap by="quarter")) +
```

3. **Explore visual plots** of the raw data across overlapping exposure(s) and outcome(s)



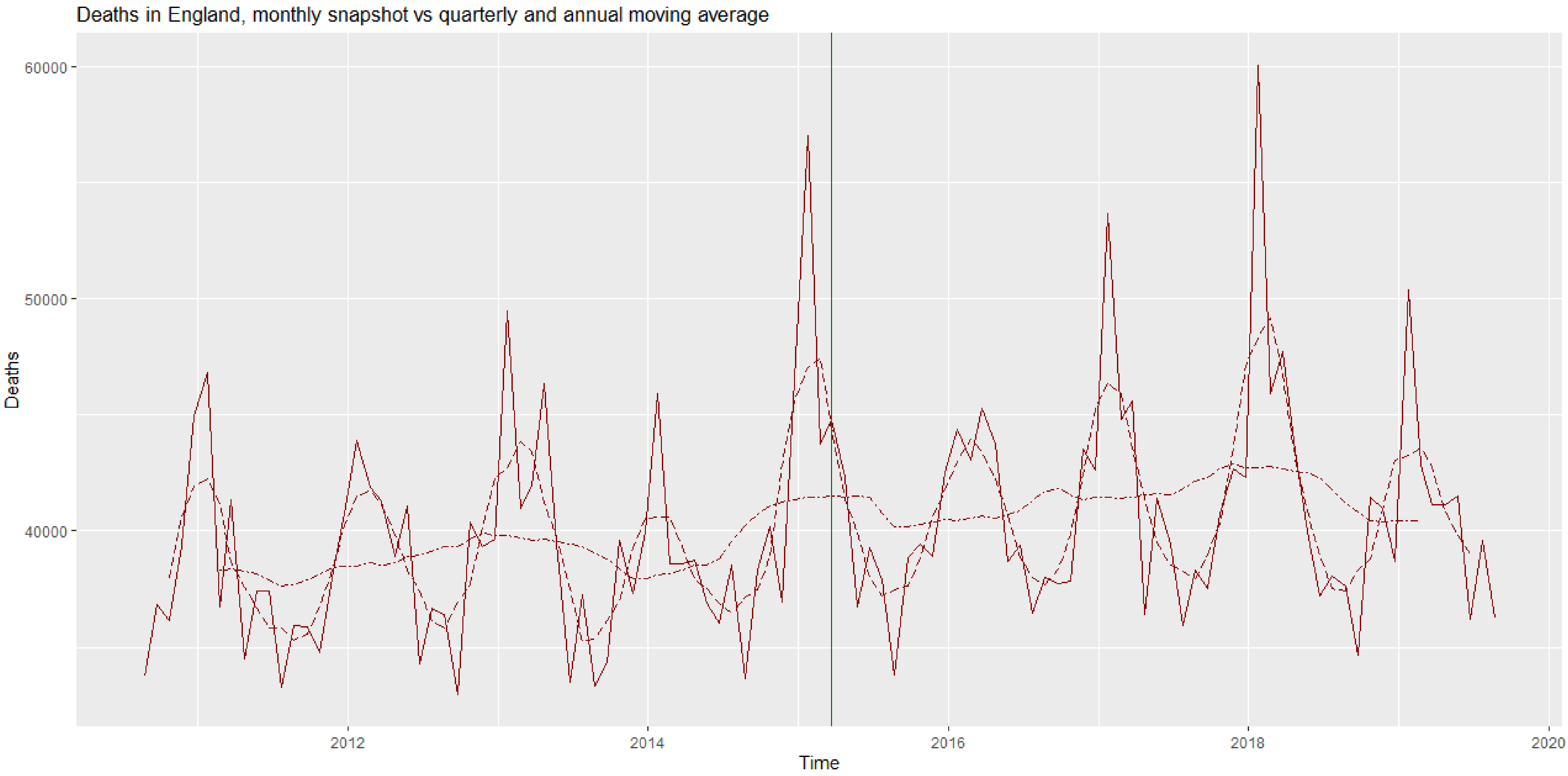
4. **Create time series objects and examine** for seasonality

```
ts_dtocdays_all <- ts(data_use$exp_days_dtoc, frequency=12, start=c(2010, 8))
ts_deaths_all <- ts(data_use$out_deaths_total, frequency=12, start=c(2010, 8))
```

4. Create time series objects and examine for seasonality

```
data_use$deaths_ma_q = ma(data_use$out_deaths_total, order=4)
data_use$deaths_ma_y = ma(data_use$out_deaths_total, order=12)

plot_ts_deaths <- ggplot(data=data_use, aes(x=date)) +
  ggtitle("Deaths in England, monthly snapshot vs quarterly and annual moving average") +

geom_line(aes(y=out_deaths_total), color="darkred") +
 geom_line(aes(y=deaths_ma_q), color="darkred", linetype="longdash") +
 geom_line(aes(y=deaths_ma_y), color="darkred", linetype="twodash") +
 xlab("Time") + ylab("Deaths") +
 geom_vline(aes(xintercept = as.numeric(as.Date("2015/03/24"))), color="darkgreen")
windows(); plot_ts_deaths
```

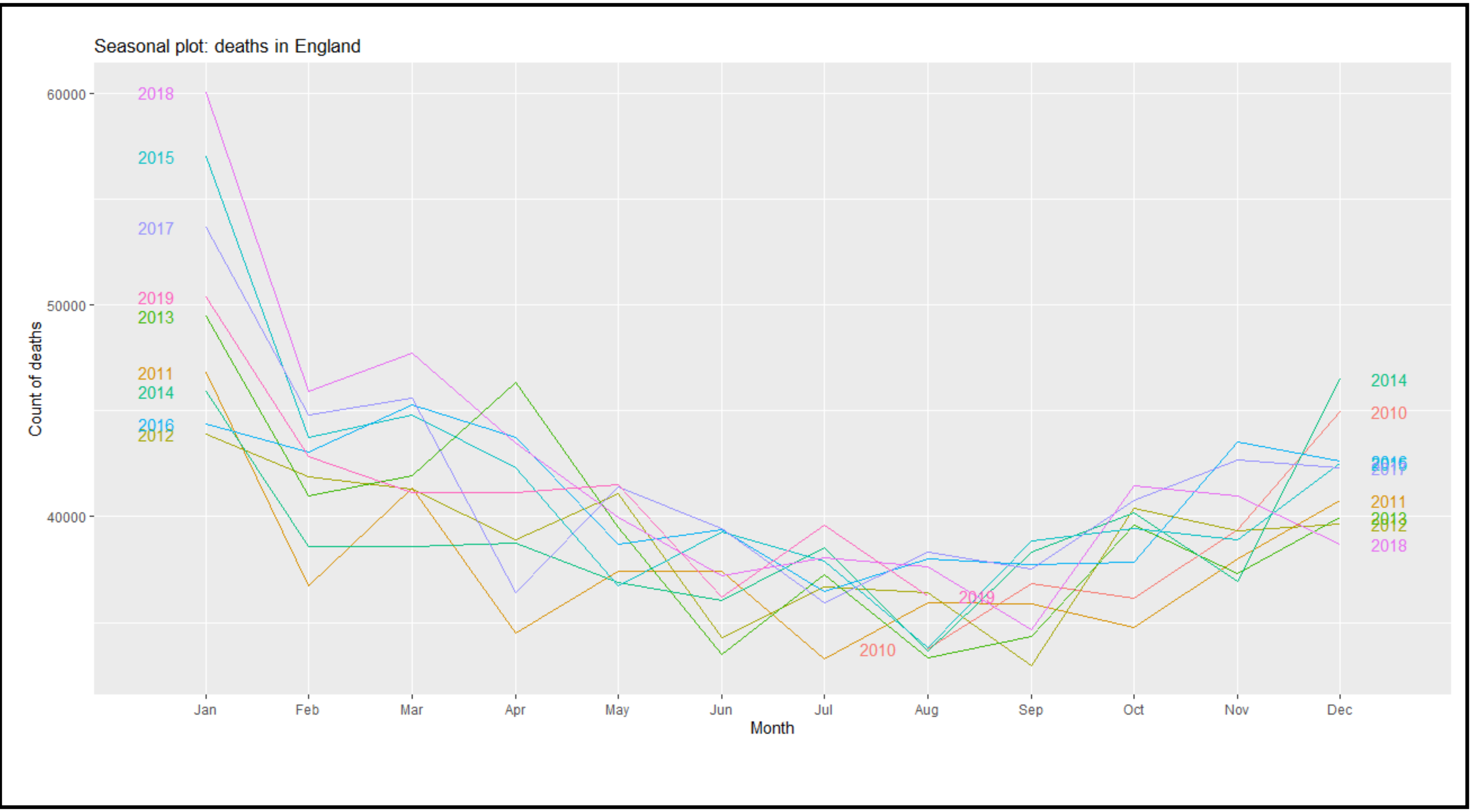
4. Create time series objects and examine for seasonality

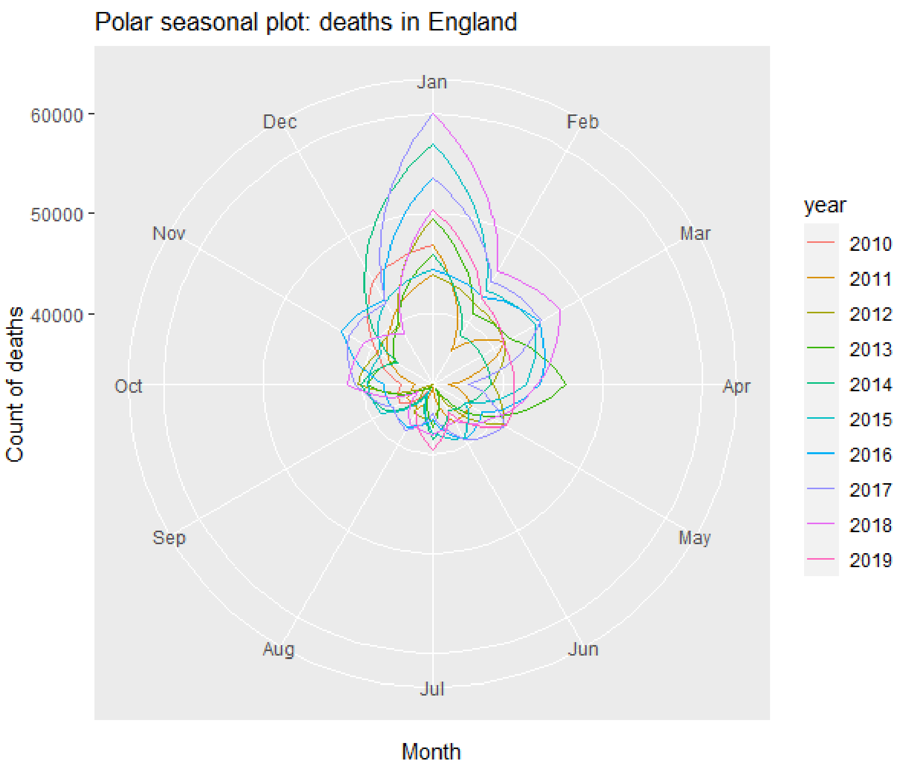


4. **Create time series objects and examine** for seasonality

Deaths in England [linear seasonal plot code]

```
plot_season_deaths <- ggseasonplot(ts_deaths_all, year.labels=TRUE,
                                   year.labels.left=TRUE) +
  ylab("Count of deaths") + ggtitle("Seasonal plot: deaths in England")
windows(); plot_season_deaths
```

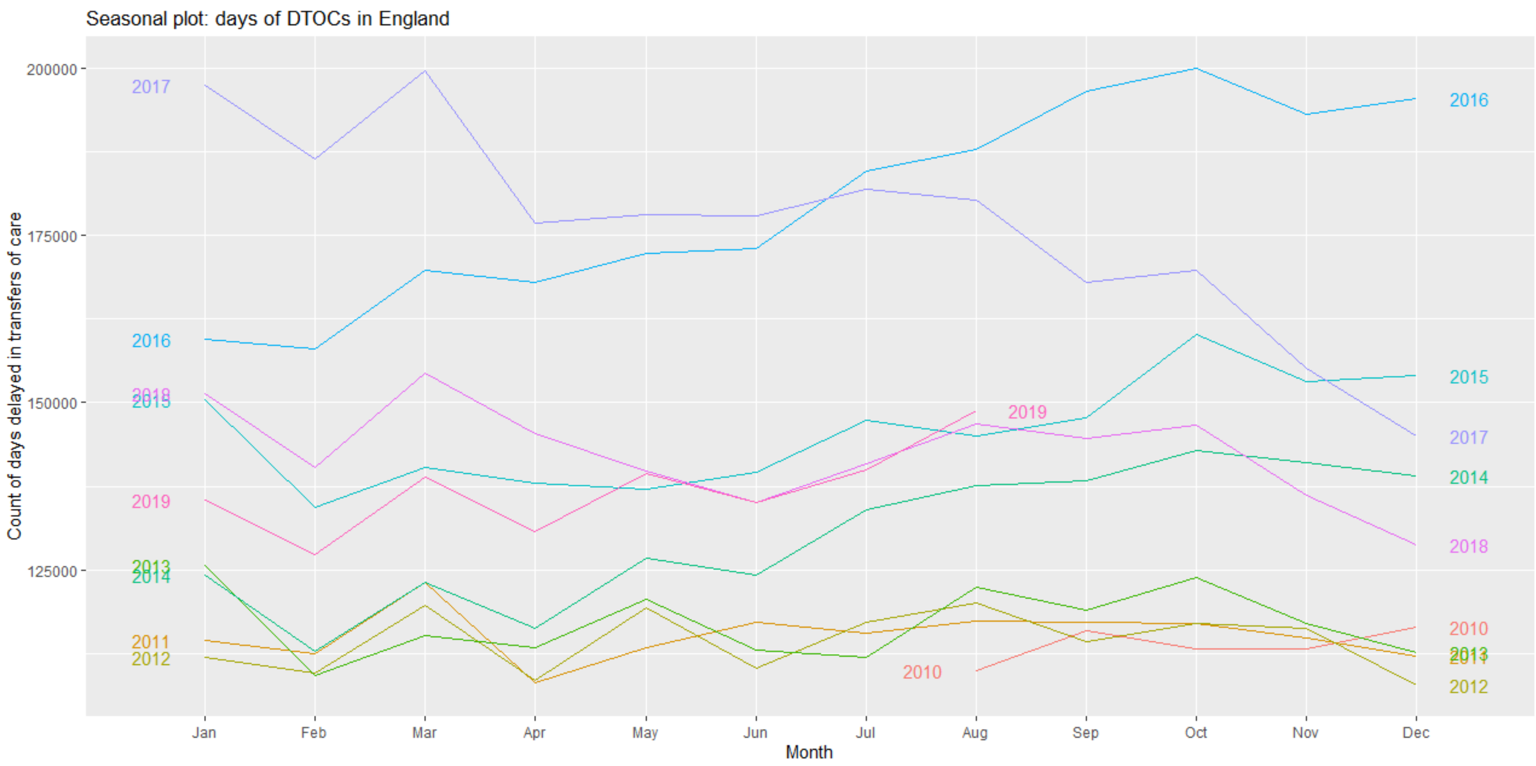


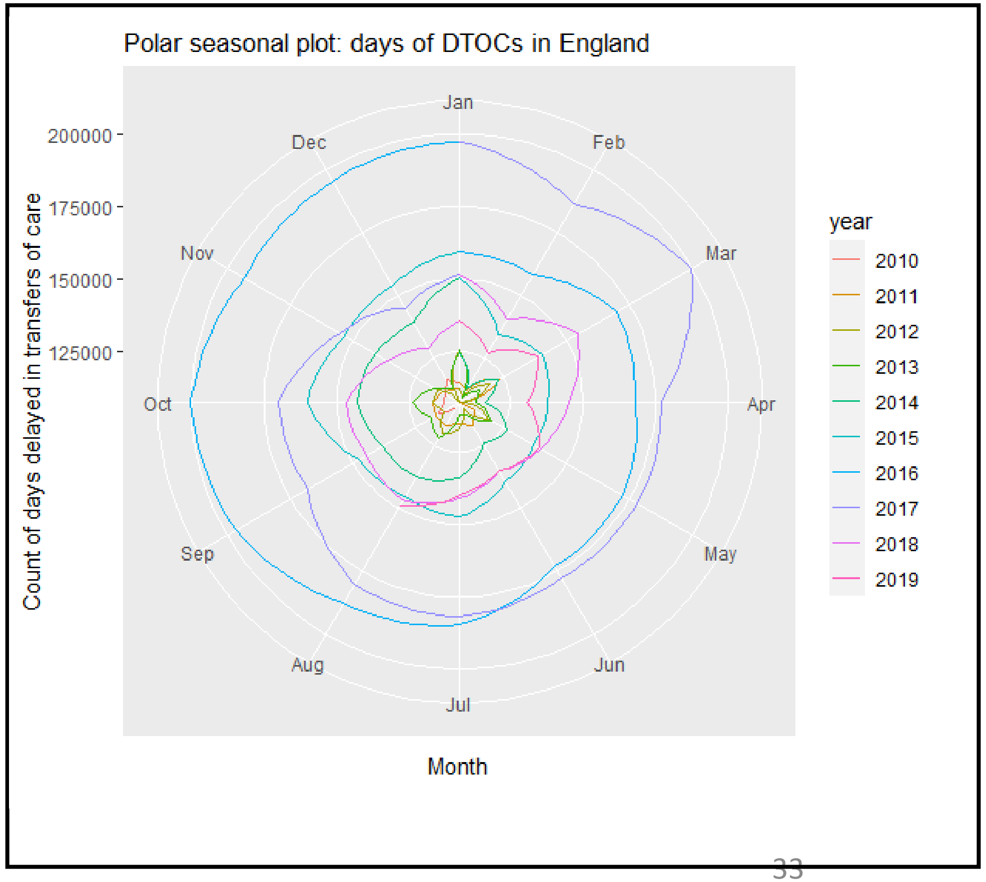


4. **Create time series objects and examine** for seasonality

Days of delayed transfers of care in NHS England [polar seasonal plot code]

```
plot_polar_dtocs <- ggseasonplot(ts_dtocdays_all, polar=TRUE) +
  ylab("Count of days delayed in transfers of care") +
  ggtitle("Polar seasonal plot: days of DTOCs in England")
windows(); plot_polar_dtocs
```





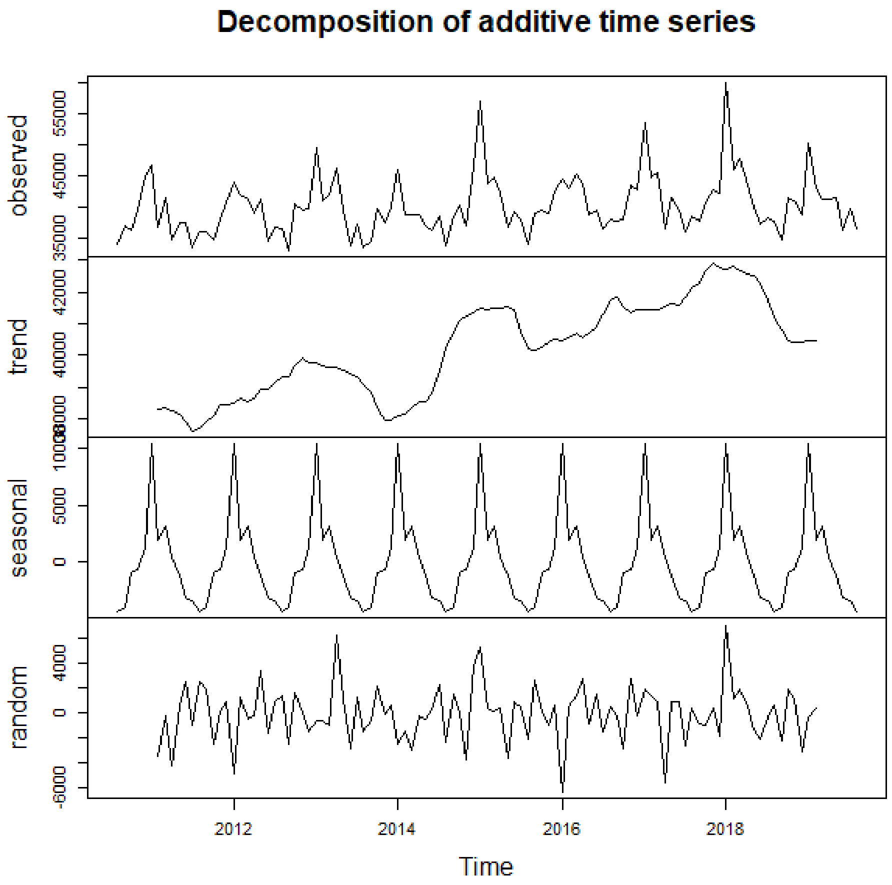
5. **Decompose** the time series of interest

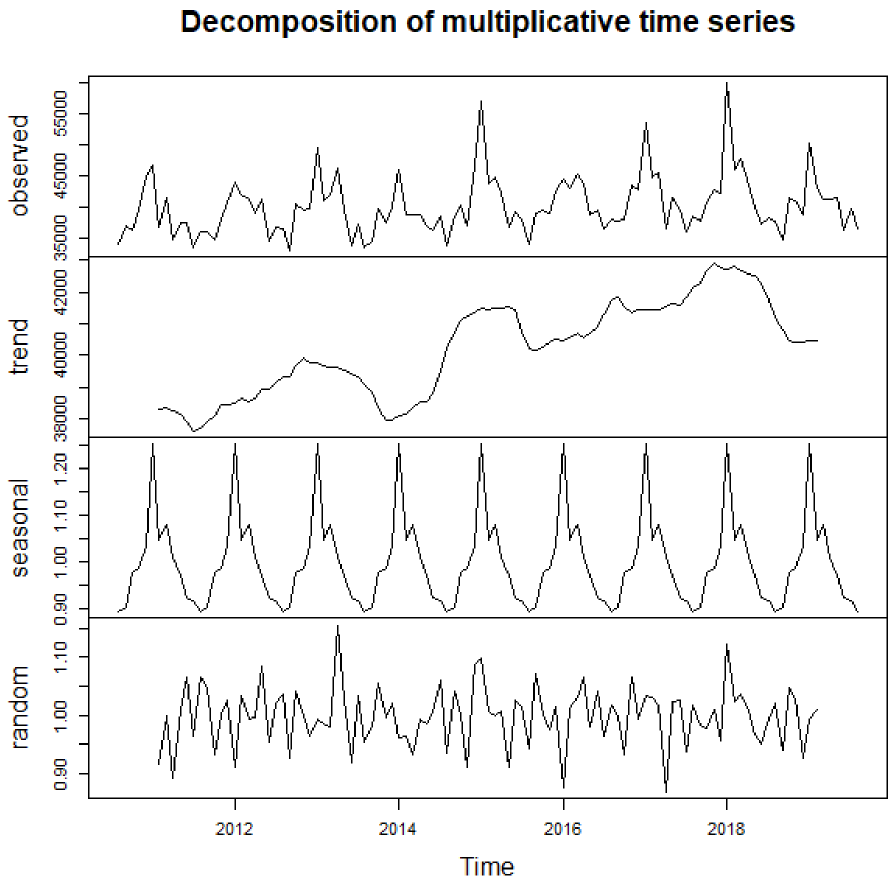
Deaths in England [additive, multiplicative, and periodic i.e. loess decomposition]

```
decomp_deaths_add <-
  decompose(ts_deaths_all, type="additive")
windows(); plot(decomp_deaths_add)

decomp_deaths_mult <-
  decompose(ts_deaths_all, type="multiplicative")
windows(); plot(decomp_deaths_mult)

stl_deaths <-
  stl(ts_deaths_all, s.window="periodic")
windows(); autoplot(stl_deaths)
```

5. **Decompose** the time series of interest

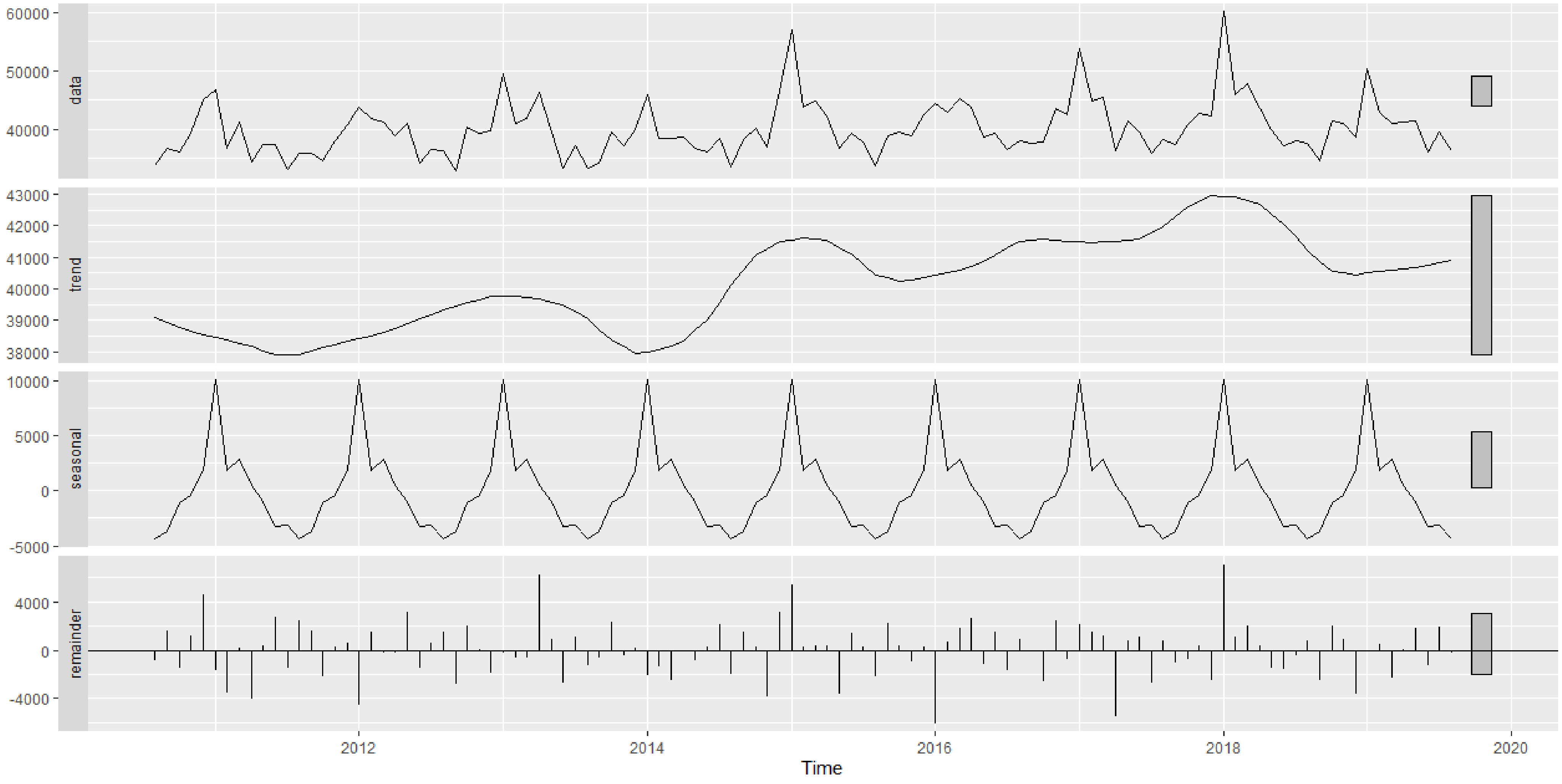
Deaths in England [additive and multiplicative models are consistent]





5. **Decompose** the time series of interest

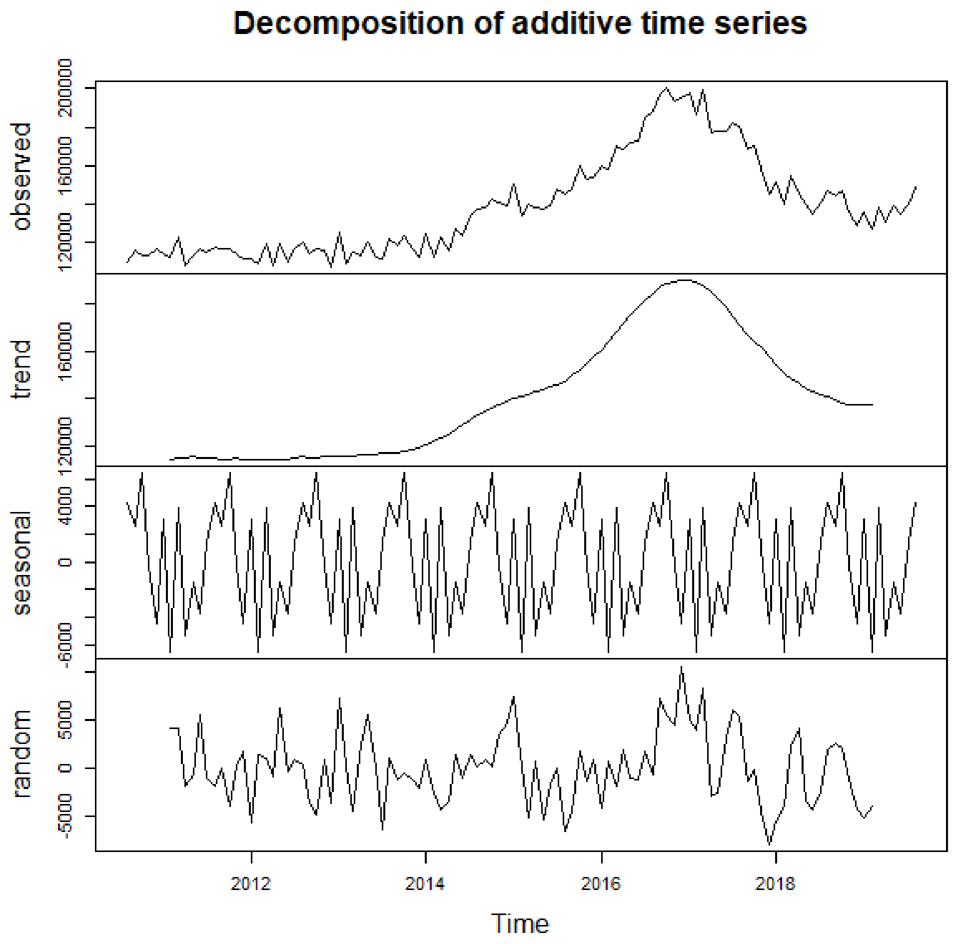
Deaths in England [periodic smoothing in **`stl`** command by loess regression]



5. Decompose the time series of interest

Days of delayed transfers of care in NHS England

```
decomp_dtocdays_all <-
    decompose(ts_dtocdays_all, type="additive")
plot(decomp_dtocdays_all)
```

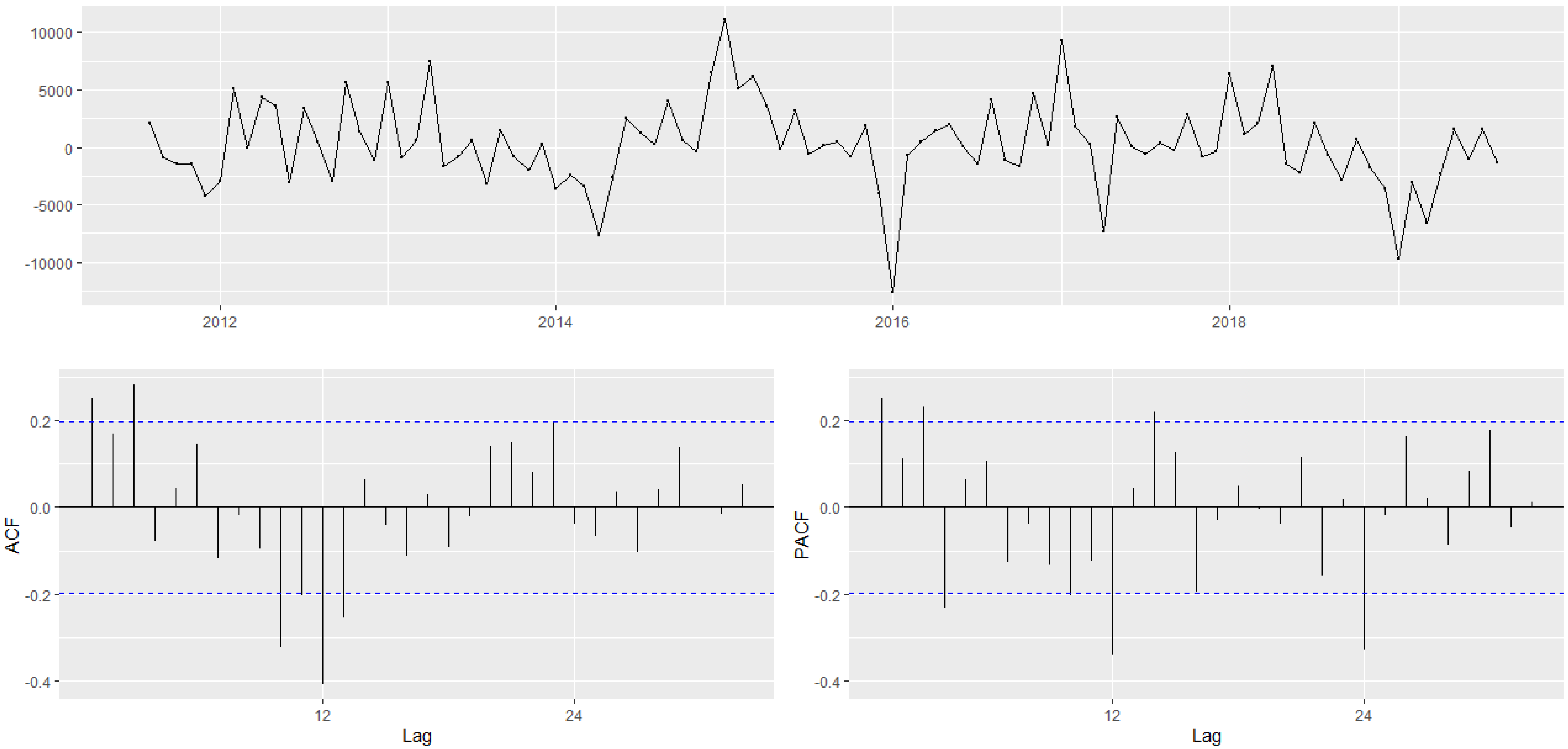


6. **Evaluate stationarity assumption** of decompositions

Deaths in England

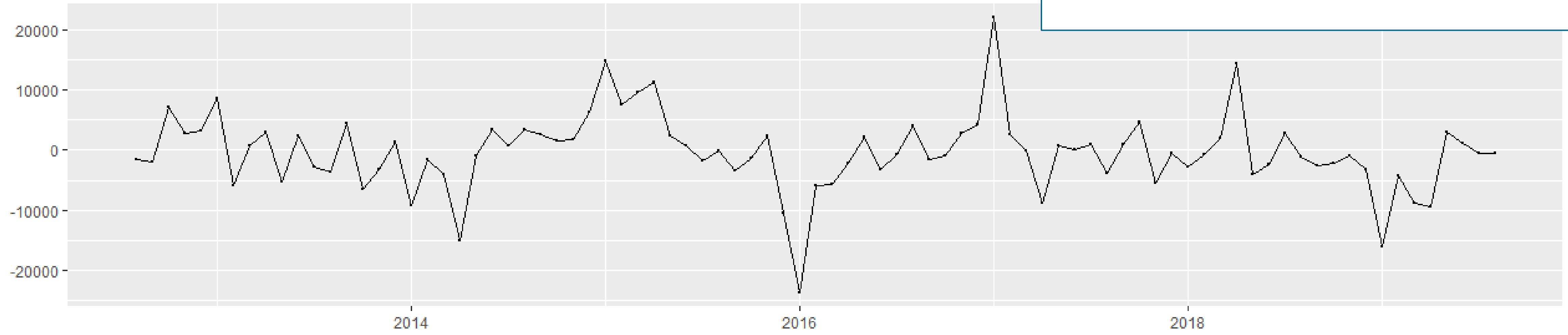
```
windows(); Acf(ts_deaths_all, lag.max = NULL, type = c("correlation", "covariance",
                                        "partial"),
    plot = TRUE, na.action = na.contiguous, demean = TRUE)

windows(); Pacf(ts_deaths_all, lag.max = NULL, plot = TRUE, na.action =
na.contiguous, demean = TRUE)

windows(); ts_deaths_all %>% ggtsdisplay() # summary view across data, Acf, and Pacf

Box.test(diff(ts_deaths_all), lag=12, type="Ljung-Box")


windows(); ts_deaths_all %>% diff(lag=12) %>%
ggtsdisplay()

windows(); ts_deaths_all %>% diff(lag=12) %>% diff(lag=12) %>%
 ggtsdisplay()
```

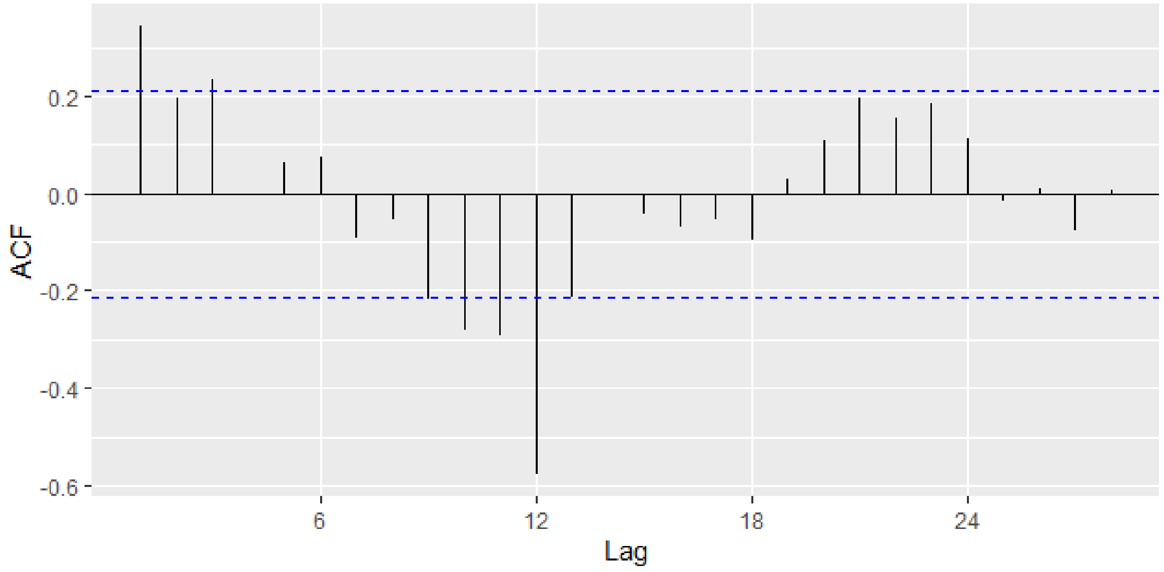
6. **Evaluate stationarity assumption** of decompositions
Deaths in England – raw data (i.e. differencing = 0)

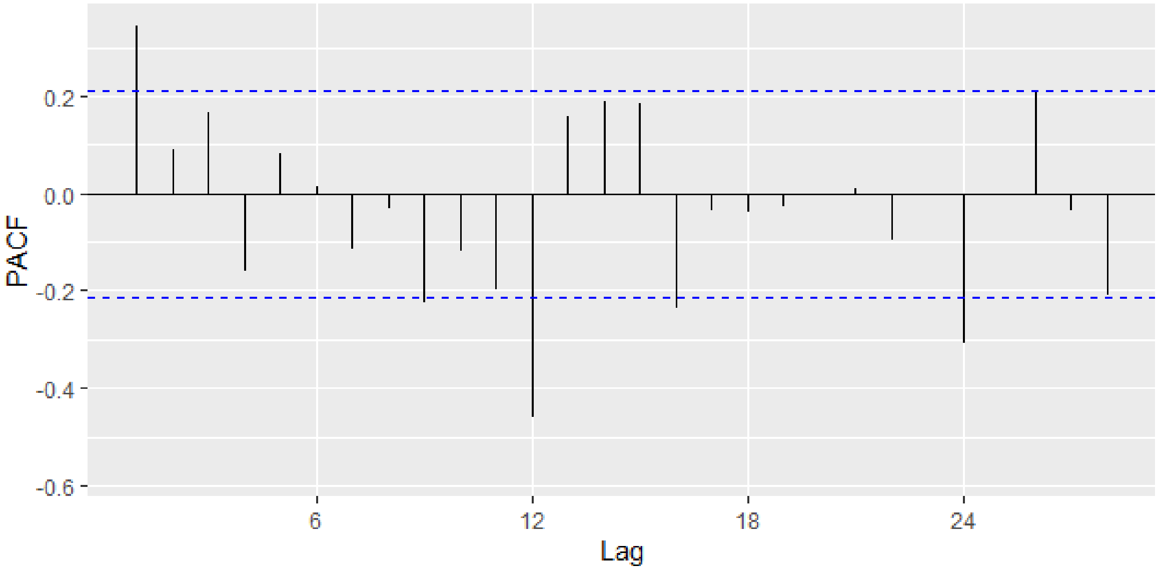


6. **Evaluate stationarity assumption** of decompositions
Deaths in England, after differencing=1

```
        Box-Ljung test

data:  diff(ts_deaths_all)
X-squared = 79.14, df = 12, p-value = 6.019e-12
```





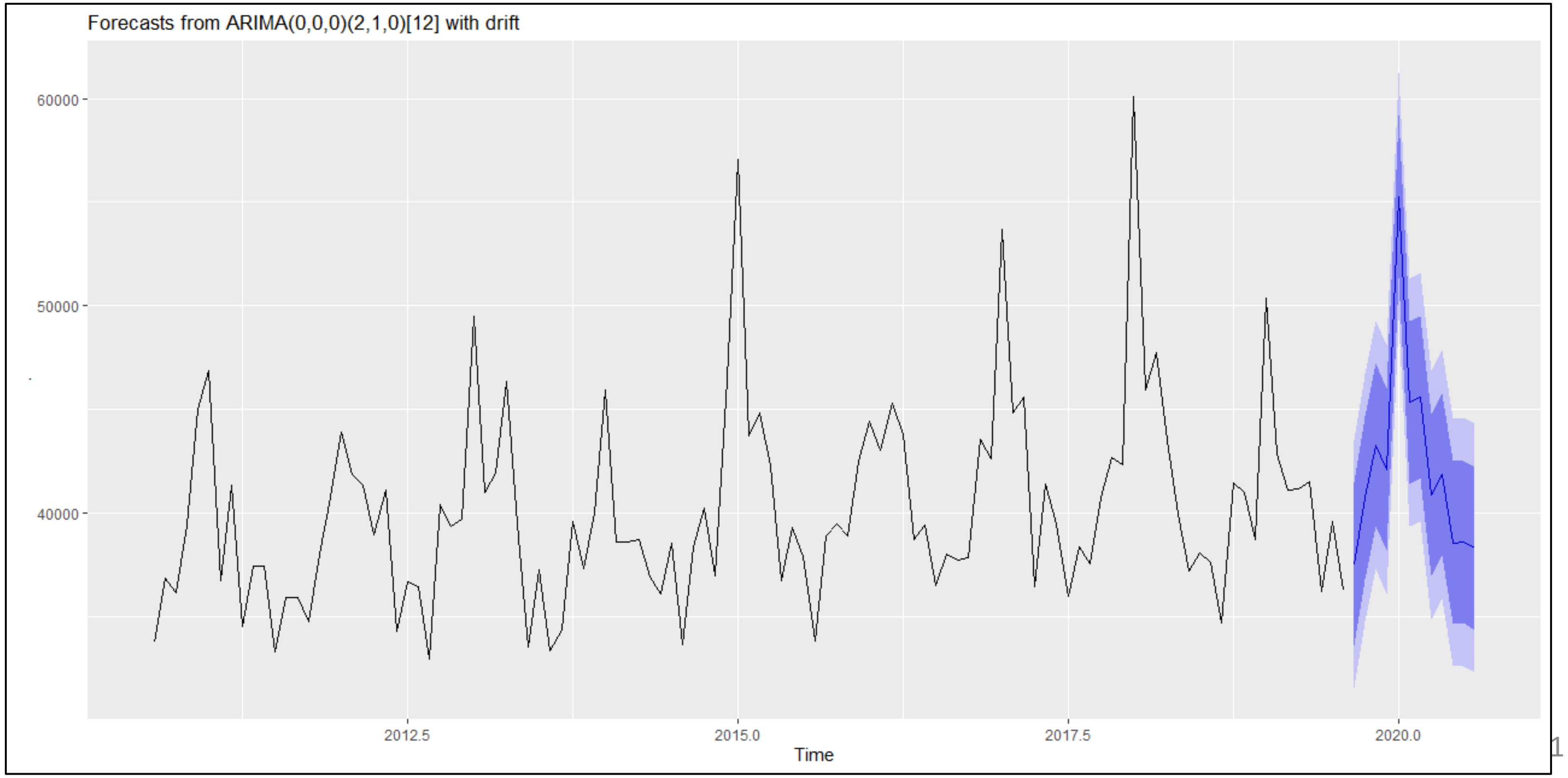


7. **Fit univariate ARIMA models**, with forecast using `auto.arima()`

Deaths in England – full data available now, through August 2019

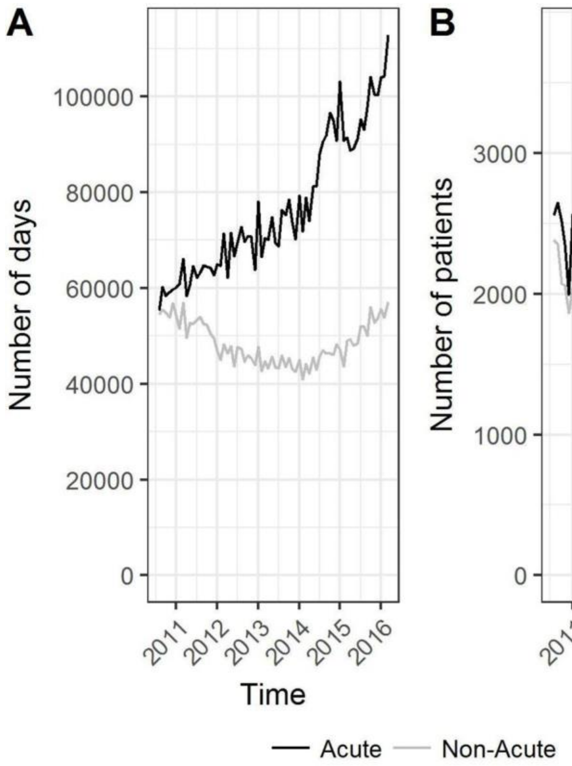
```
fit_deaths_all <- ts_deaths_all %>%
  auto.arima()

windows(); fit_deaths_all %>%
 forecast(h=12) %>% autoplot()
```

```
Series: .
ARIMA(0,0,0)(2,1,0)[12] with drift

Coefficients:
         sar1     sar2    drift
      -0.6697  -0.3793  33.9801
s.e.   0.1047   0.1024  13.7082

sigma^2 estimated as 9354336:  log likelihood=-918.08
AIC=1844.16   AICc=1844.6   BIC=1854.46
```

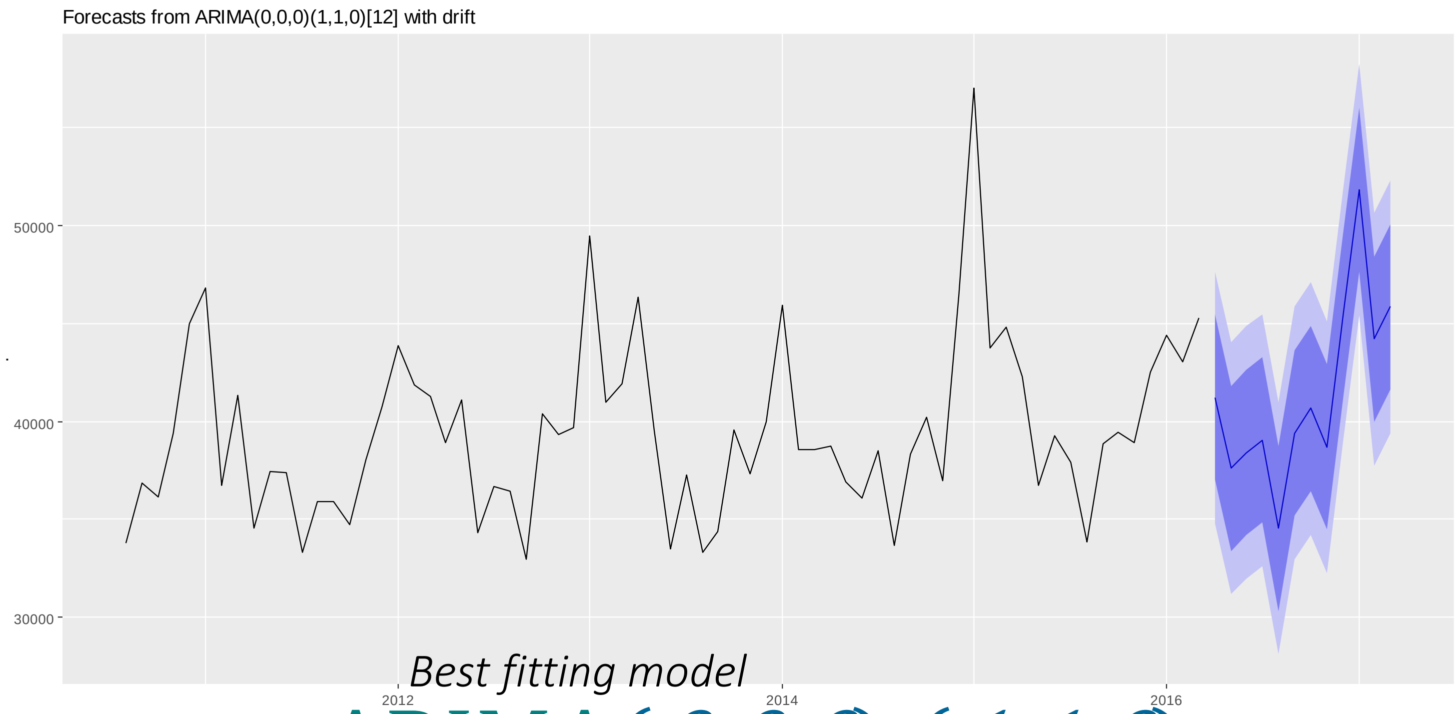


Returning to the applied case – one data point is replicated

Delayed discharge / transfers of care in NHS England



Mortality in England, including 12-month forecast



Best fitting model

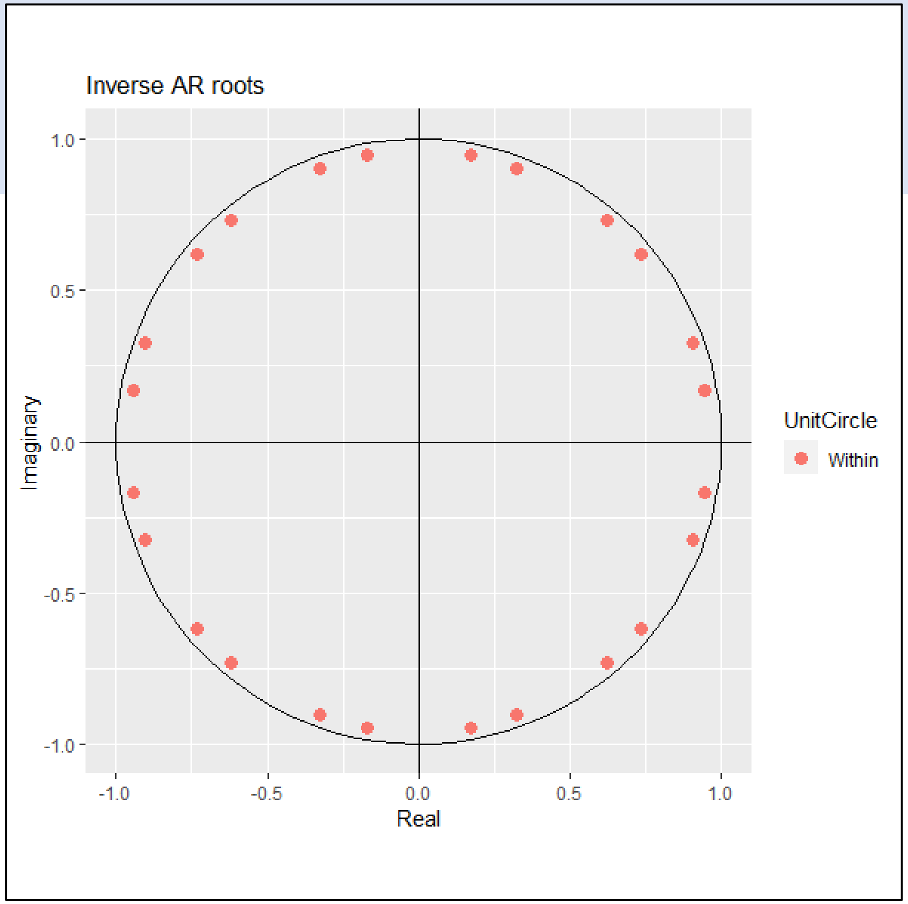
$$\mathrm{ARIMA}(0,0,0)\,(1,1,0)_{12} + c$$

A white-noise general trend with clear **seasonal pattern of** upward drift apparent in the autoregressive and integration (differencing) aspects

Green MA et al, Could the rise in mortality rates since 2015 be explained by changes in the number of delayed discharges of NHS patients? *J Epi & Comm Hlth* 71(11), 16 Oct 2017.

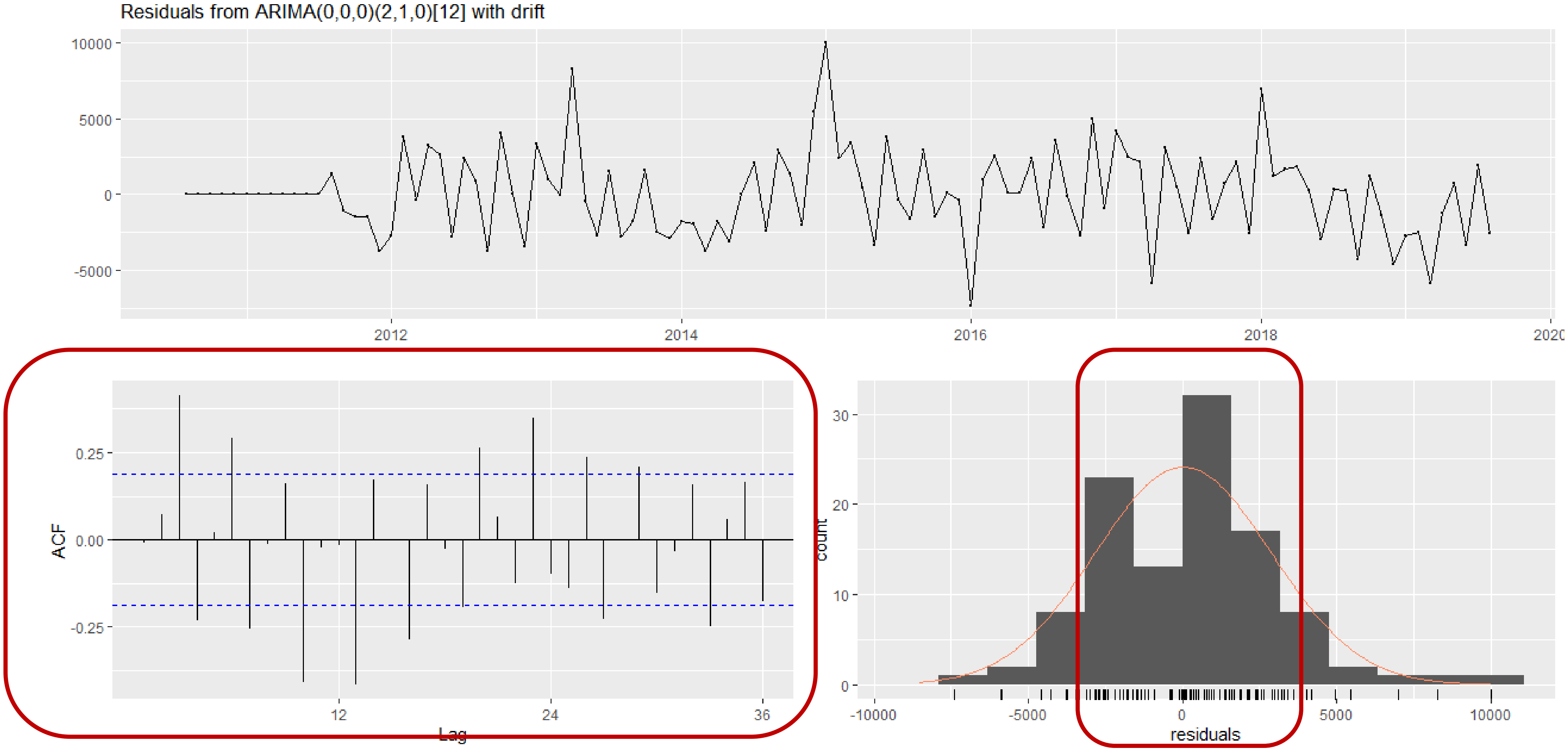
8. **Check forecasts for fit and predictiveness**, refining and updating as needed

```
windows(); checkresiduals(fit_deaths_all)

refit_deaths_all <- ts_deaths_all %>%
  auto.arima(approximation=FALSE)
refit_deaths_all

windows(); autoplot(refit_deaths_all)
```



8. **Check forecasts for fit and predictiveness**, refining and updating as needed

Deaths in England – full data available now, through August 2019



9. **Extend to ARIMA-based regression** between exposures and outcomes

```
266 ts_acutedays_all <- ts(data_use$exp_days_acute, frequency=12, start=c(2010, 8))
```

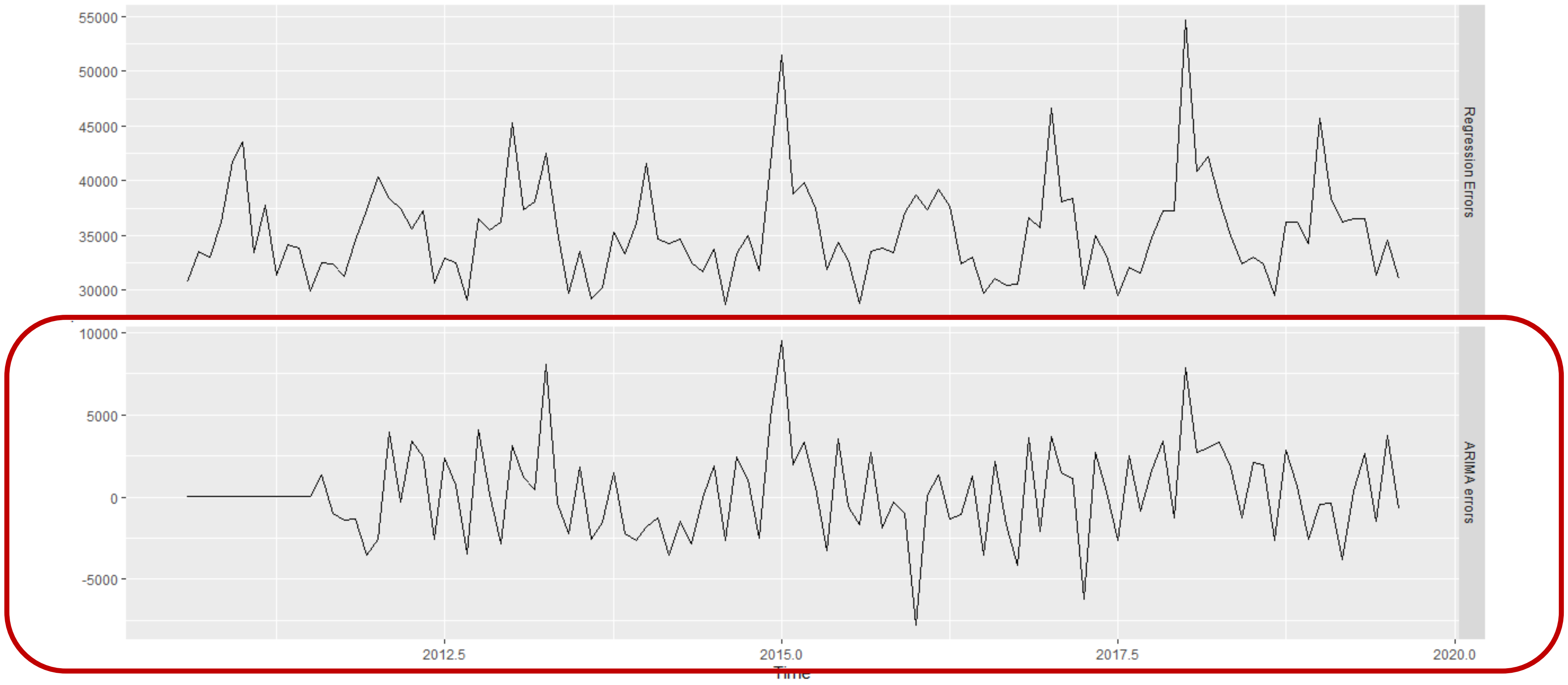
```
271 regress_all_auto <- auto.arima(ts_deaths_all, xreg=ts_acutedays_all,
allowdrift=TRUE)
272 regress_all_auto
```

What is the observed relationship between DTOCs and deaths across 41 months of data?

```
Series: ts_deaths_all
Regression with ARIMA(0,0,0)(2,1,0)[12] errors

Coefficients:
         sar1     sar2    xreg
       -0.654  -0.4210  0.0542
s.e.    0.099   0.1002  0.0146

sigma^2 estimated as 8665612:  log likelihood=-914.66
AIC=1837.33   AICc=1837.76   BIC=1847.63
```

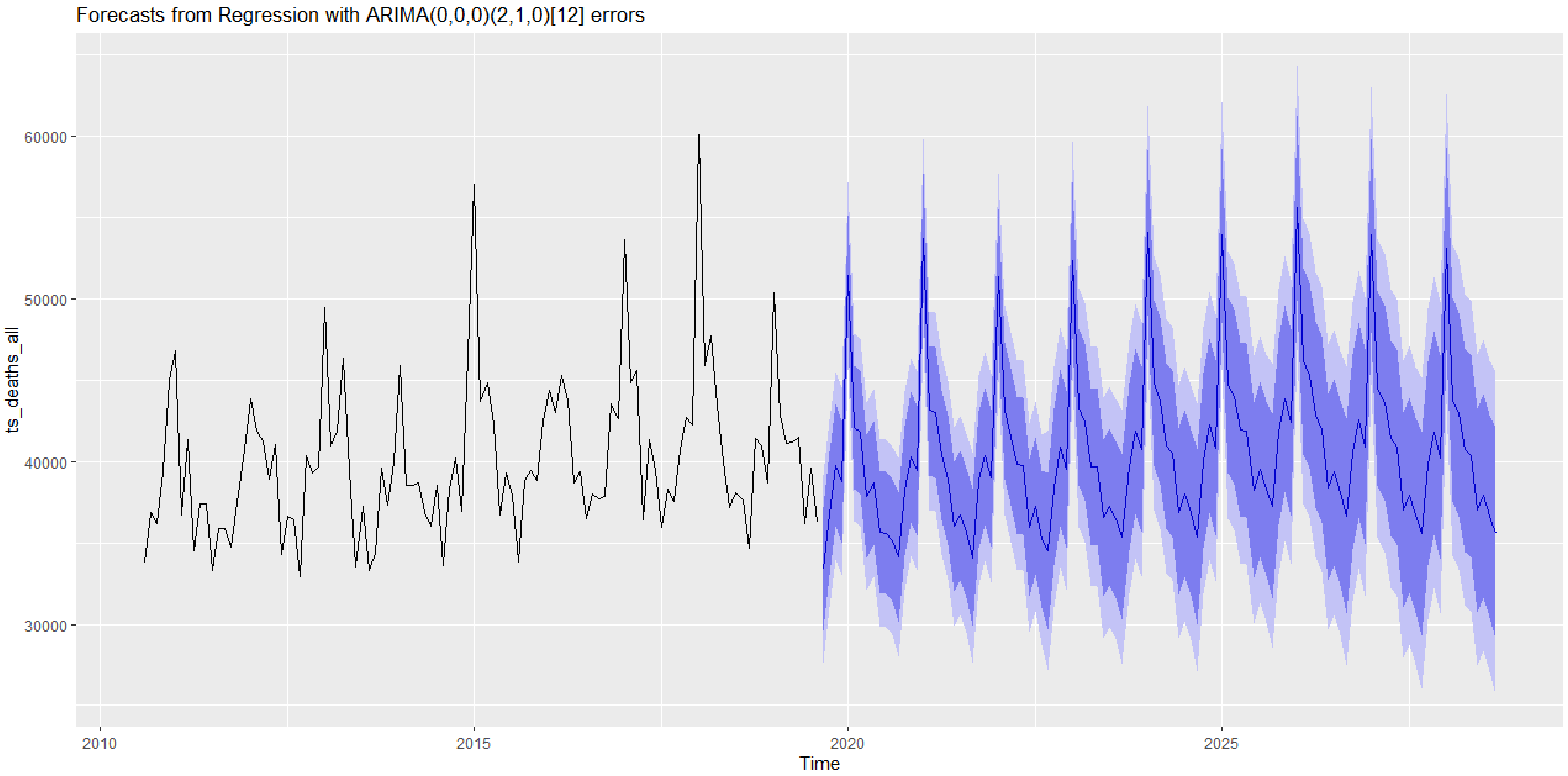
9. Extend to ARIMA-based regression between exposures and outcomes

```
windows(); cbind("Regression Errors" = residuals(regress_all_auto,
type="regression"),
         "ARIMA errors" = residuals(regress_all_auto, type="innovation")) %>%
 autoplot(facets=TRUE)
```



9. **Extend to ARIMA-based regression** between exposures and outcomes

```
286 windows(); regress_all_auto %>% forecast(xreg = ts_acutedays_all, h=12) %>%
autoplot()
```



Discussion

Advantages and limitations of ARIMA

Advantages

- Enables useful, robust predictions and exploration of univariate + multivariable associations in time series data
- Captures nuanced time variations without needing to know underlying predictors
- Parametrically parsimonious
- Extremely flexible

Disadvantages

- Very limited causal strength (cf. Granger causality vs. more strict Pearl/Rubin sense of counterfactual causality)
- Cannot capture exogenous shocks or information
- Risk of model over-specification
- Data hungry, requiring constant updates
- Limited generalisability

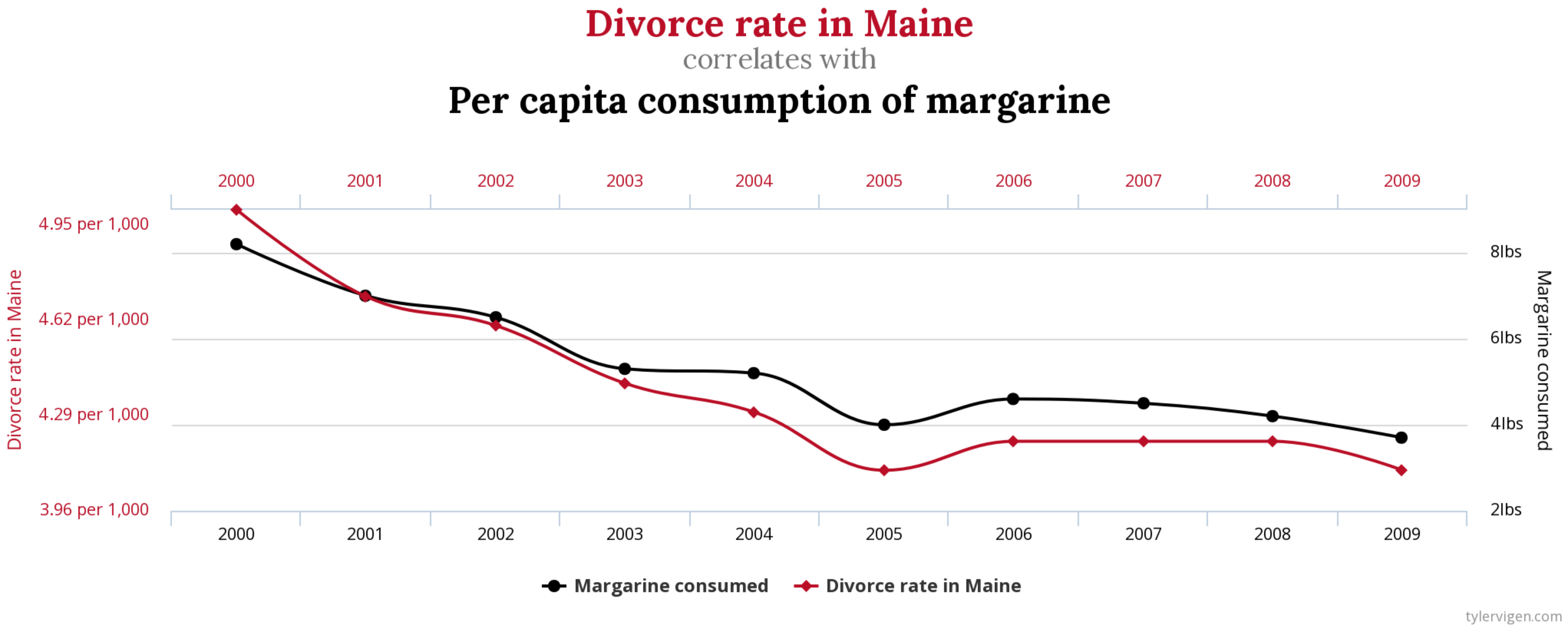
Do NOT use ARIMA to…

- **Make long-term predictions**
  - Always update time series data wherever possible
  - In general, avoid ever making predictions for a longer timeline than the historical data you have available
  - Be aware that predictions will always become more uncertain the farther they are in the future
  - ARIMA is not good at all at handling exogenous shocks (cf. this practice case!)

- **Evaluate time series for which stationarity cannot be achieved**, e.g. data too sparse
  - Consider alternate methods, e.g. exponential smoothing

- **'Skip ahead' in science:** In most cases of complex reality, ARIMA is not very suited to **test (hypotheses)** of generalisable or externally valid **causal** relationships
  - Instead, consider research design as well as analytical strategies, **e.g. Survival analysis** based on individual exposures and outcome censoring over time

Caveat emptor



Tyler Vigen, http://www.tylervigen.com/spurious-correlations, citing U.S. National Vital Statistics Reports and US Department of Agriculture

ARIMA is useful to…

- Assess historic time trends and make near-term predictions
  - Relatively elegant parameterization makes for very nice internal validity across many different kinds of time series
  - Strength of internal validity in decomposing noisy data is particularly well-suited to ongoing monitoring of deterministic systems (e.g. sales forecasts)
  - Availability of quick, user-friendly ML and visualization tools allows for nifty widgets

- **Build an evidence base toward tricky scientific inference:** Explore associations and surface hypotheses for potential causal links after descriptive comparison and expert intuition suggest 'there is something there'
  - Time-series analysis is particularly well-suited to ecological studies
  - Consistency or robustness of findings across space as well as time adds compelling evidence (e.g. regression effects observed in different local areas, among sub-populations of different ages)

Further references & reading

Chambers JC et al. "How to choose the right forecasting technique", *Harvard Business Review,* July 1971.

Coghlan A. *A little book of R for time series*, v0.2, Sept 2018.

Dalinina R. "Introduction to forecasting with ARIMA in R", Jan 2017.

Hyndman RJ. *R documentation:* **`forecast`** package, v8.9.

Hyndman RJ & Athanasopoulos G, *Forecasting: principles and practice*, *2nd ed.* 2018.

Jones RH. "Maximum likelihood fitting of ARMA models to time series with missing observations", *Technometrics*, Mar 2012.

SAS/ETS® 9.2 Users' Guide. "X-11 ARIMA method" (for SAS vs. R implementation)

Thank you