

An introduction to time series analysis with ARIMA

R Ladies London
24 October 2019

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

Welcome to LSHTM! Some logistics

Wi-fi network

eduroam

or

lshtm-conf

password: “louispasteur”

All materials

<https://github.com/7j7j/R-ladies-ARIMA-intro>

```
install.packages(c(“tidyverse”, “forecast”, “here”))
```

R-ladies

Who are we?

- The R community suffers from underrepresentation of minority genders (including but not limited to cis/trans women, trans men, non-binary, genderqueer, agender) in every role and area of participation
- R-Ladies is a worldwide organization whose mission is to promote gender diversity in the R community.
- Currently 169 chapters in 46 countries and 57162 members.

Hello! My name is Jules and I am an applied social scientist.

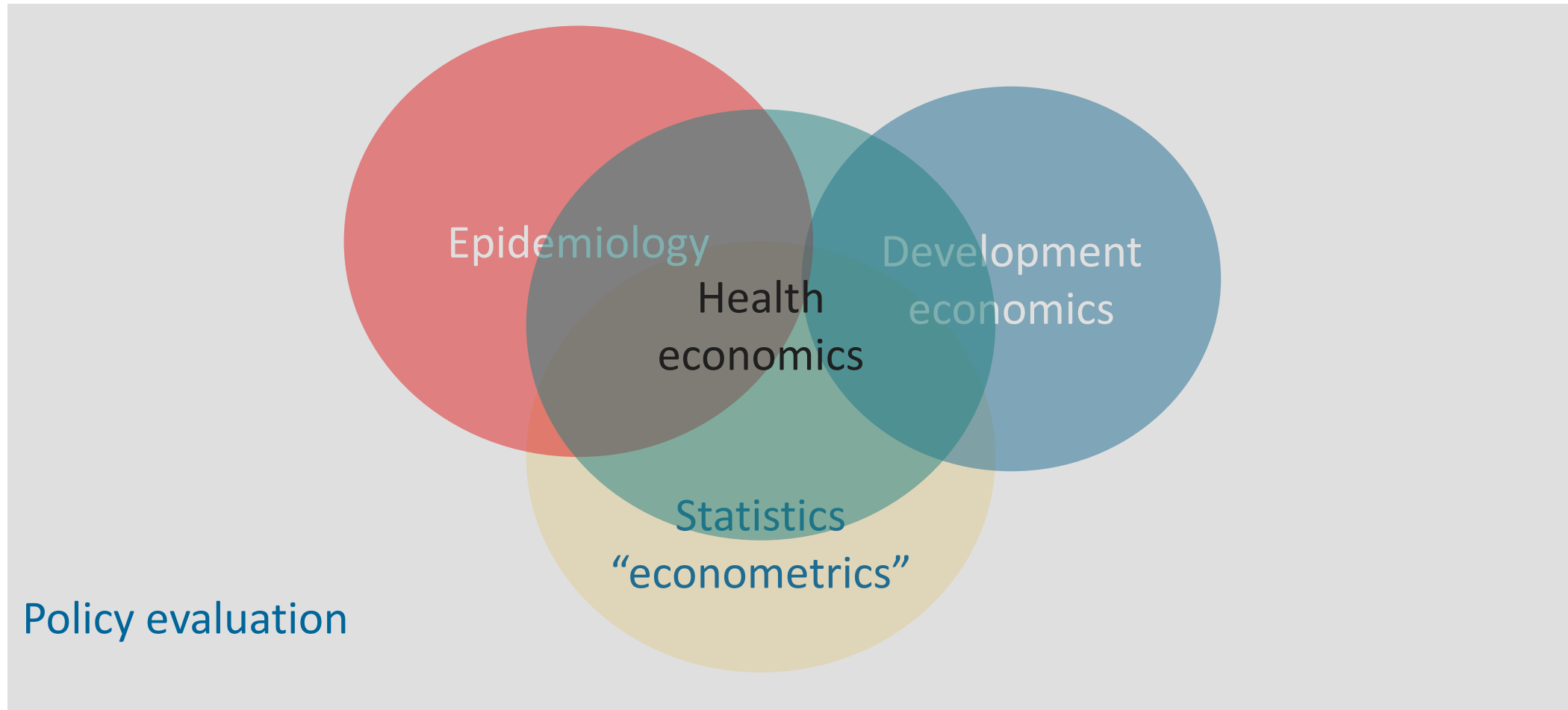
LONDON
SCHOOL of
HYGIENE
& TROPICAL
MEDICINE



Economic
and Social
Research Council



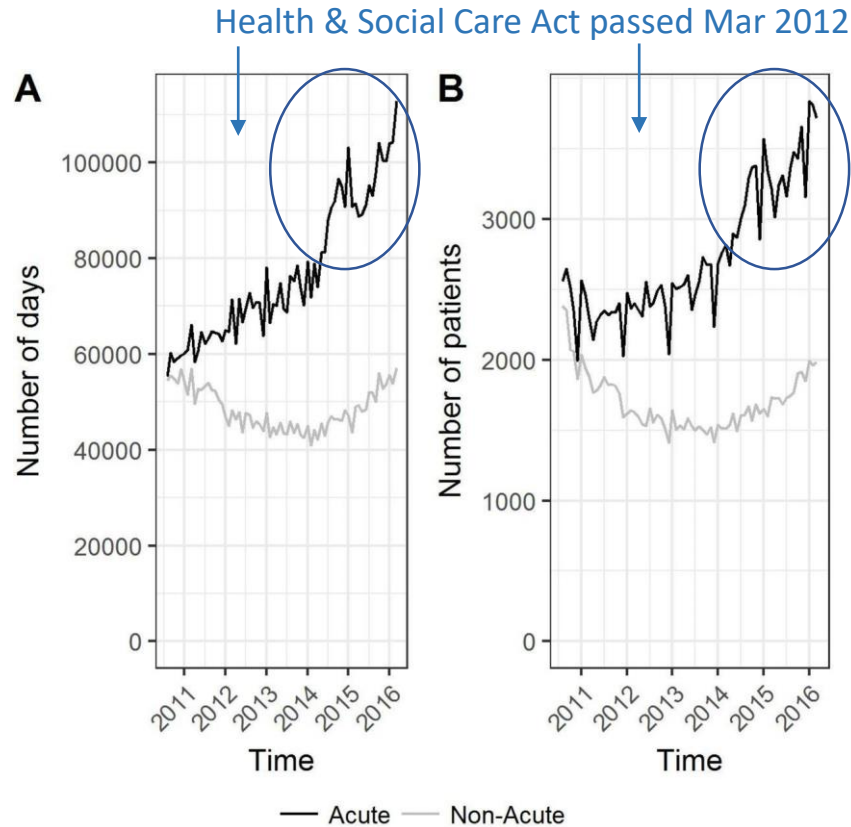
@jules__shen



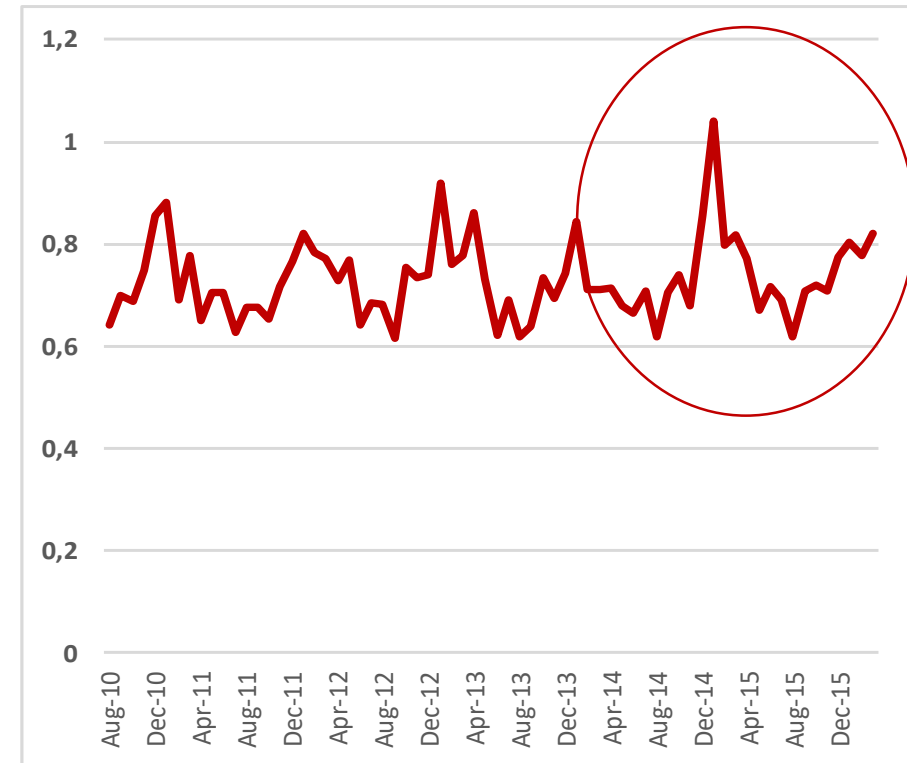
Context & Theory

Does one time series explain another? (motivating case)

Delayed discharge / transfers of care
in NHS England¹



Approximate crude mortality rate/1,000
in England²



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.

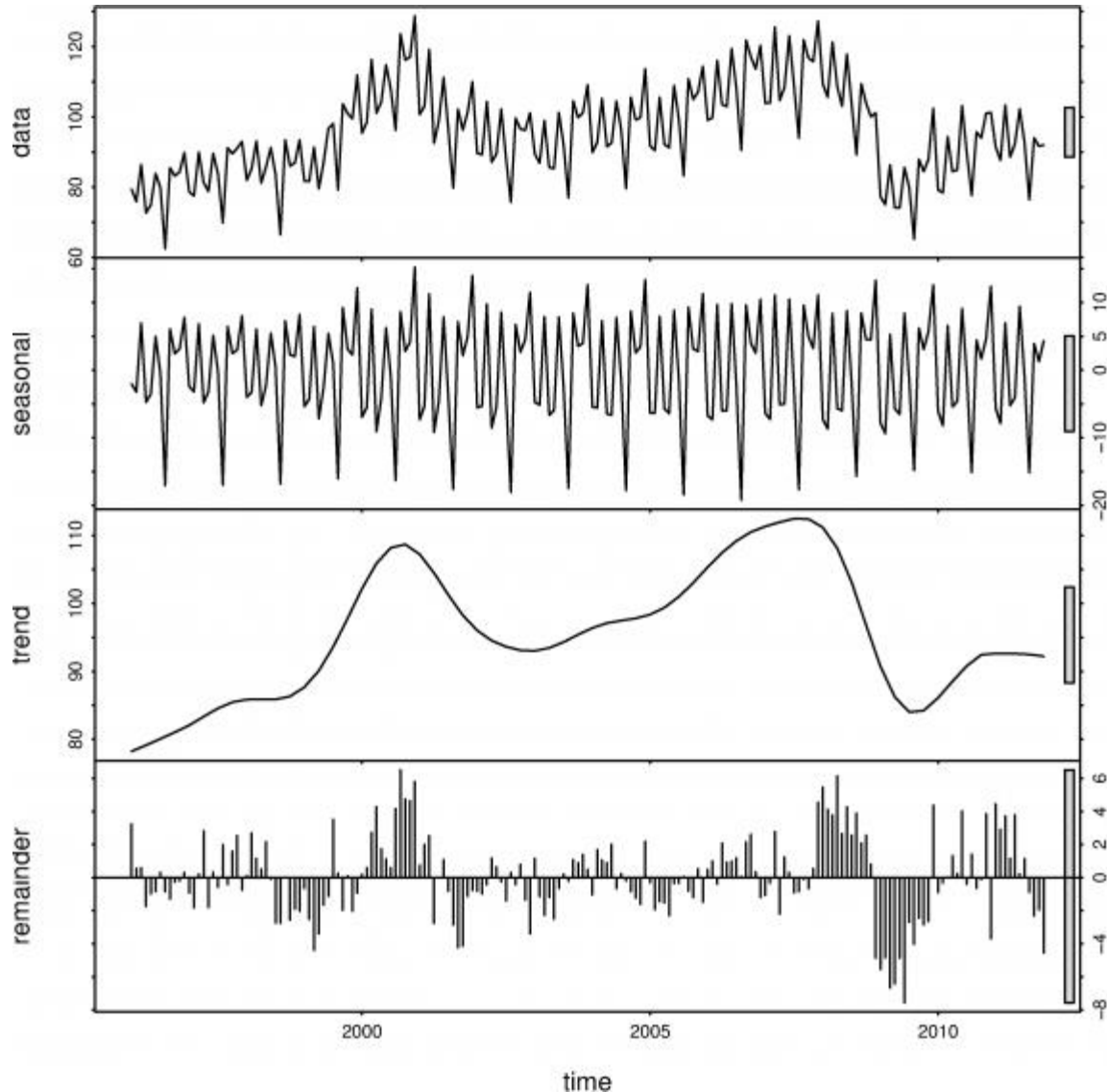
Autoregressive Integrated Moving Average regression models were used in the analysis.

“ ”

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.

Basic framework for time series analysis



y_t (future forecast) data

=

S_t seasonal (fixed-period) fluctuations

+ or X

T_t underlying trend

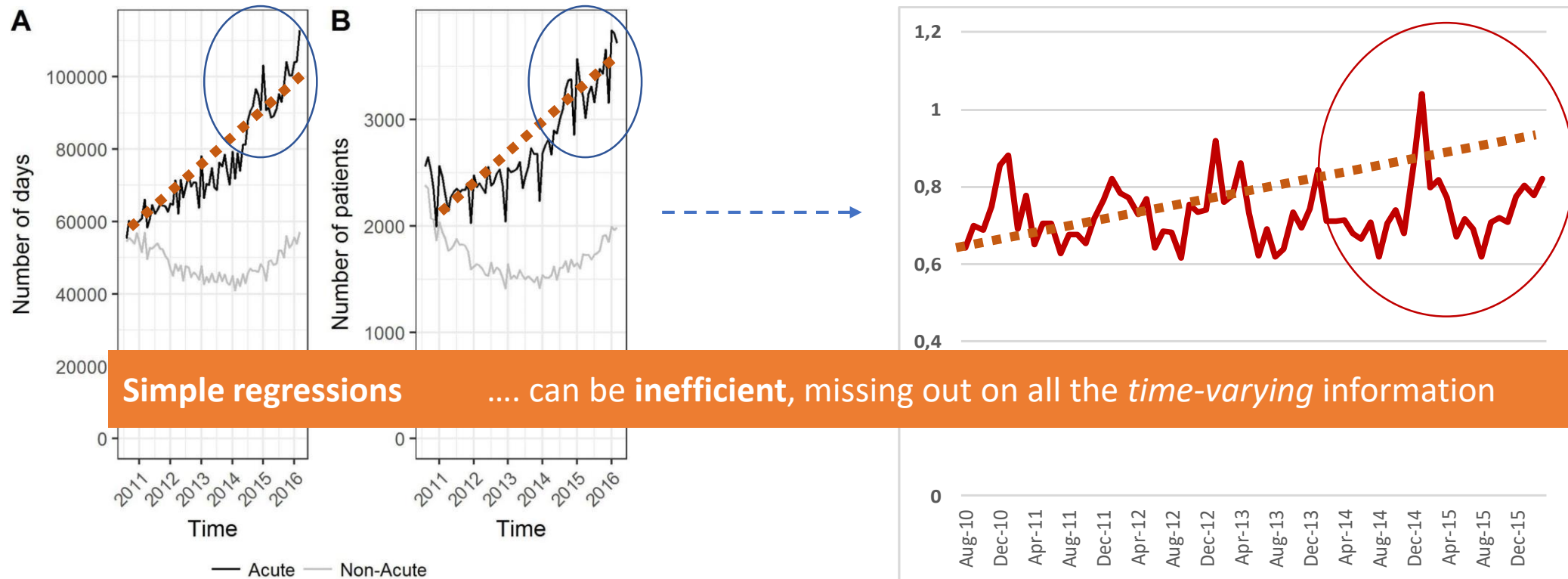
+ or X

ε_t random error or varying-period cycle

Returning to the motivating case

Delayed discharge / transfers of care
in NHS England

Approximate crude mortality rate/1,000
in England



What is ARIMA?

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

AR

Auto-Regressive

I

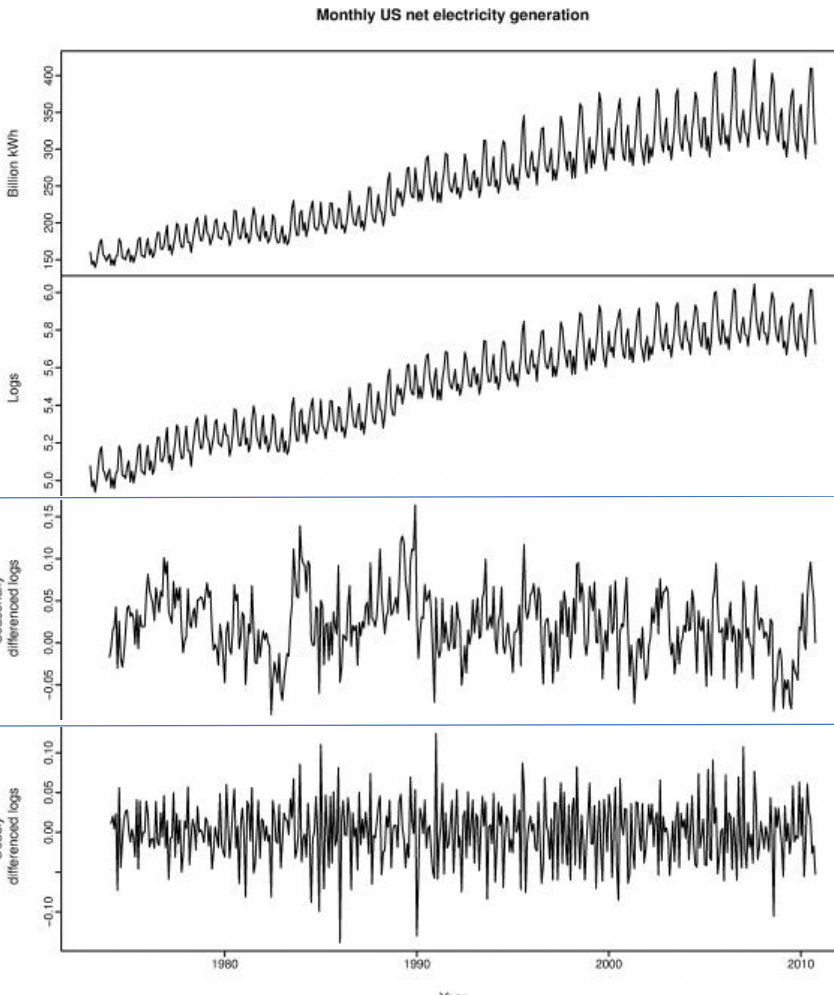
Integrated

MA

Moving Average

(p , d , q)

What is ARIMA?



‘Differencing’ of data for stationarity, or
integrating across orders of differences

y_t data

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

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

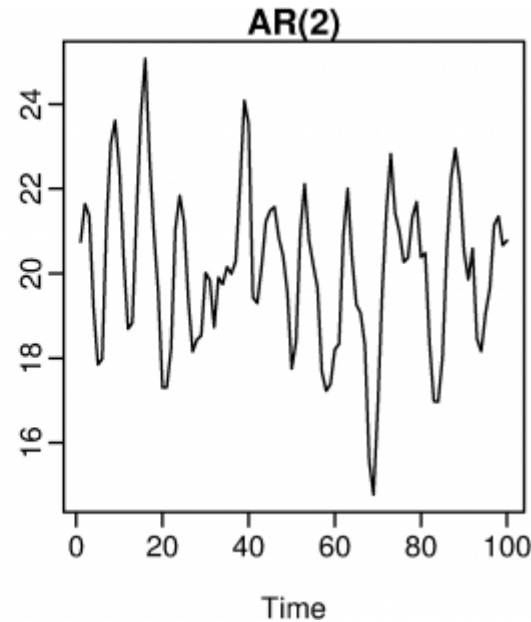
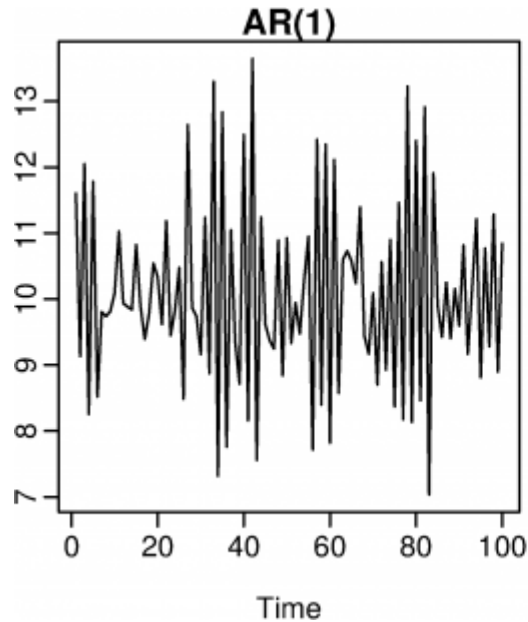
$$\begin{aligned} d_2 &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned}$$

I

Integrated

d

What is ARIMA?



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, $\varepsilon_t \sim \text{Norm}(0,1)$

AR

Auto-Regressive

p

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

data

constant

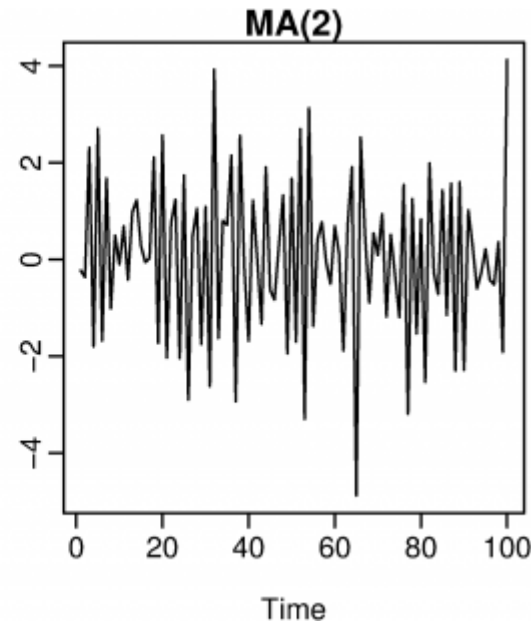
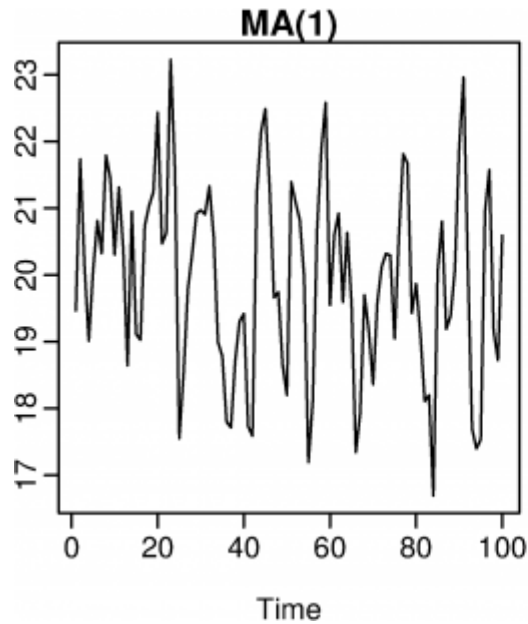
weight 1 •
data 1 time
period ago

weight 2 •
data 2 time
periods
ago

weight p •
data p time
periods
ago

random
error

What is ARIMA?



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, $\varepsilon_t \sim \text{Norm}(0,1)$

MA

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

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.

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

“ ”

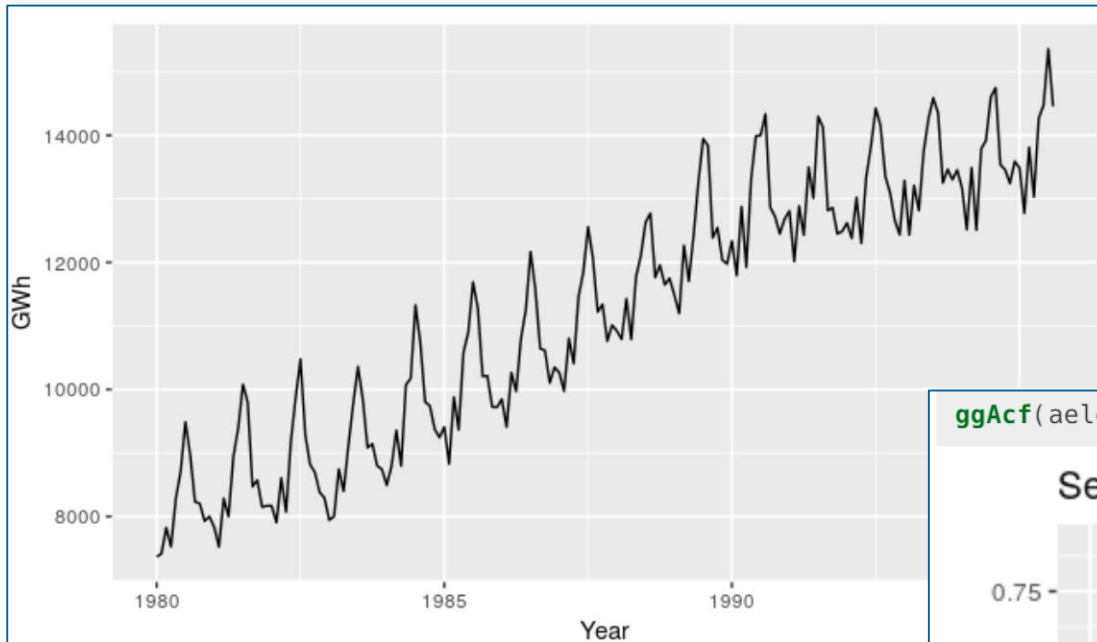


Figure 2.15: Monthly Australian electricity demand from 1980–1995.

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

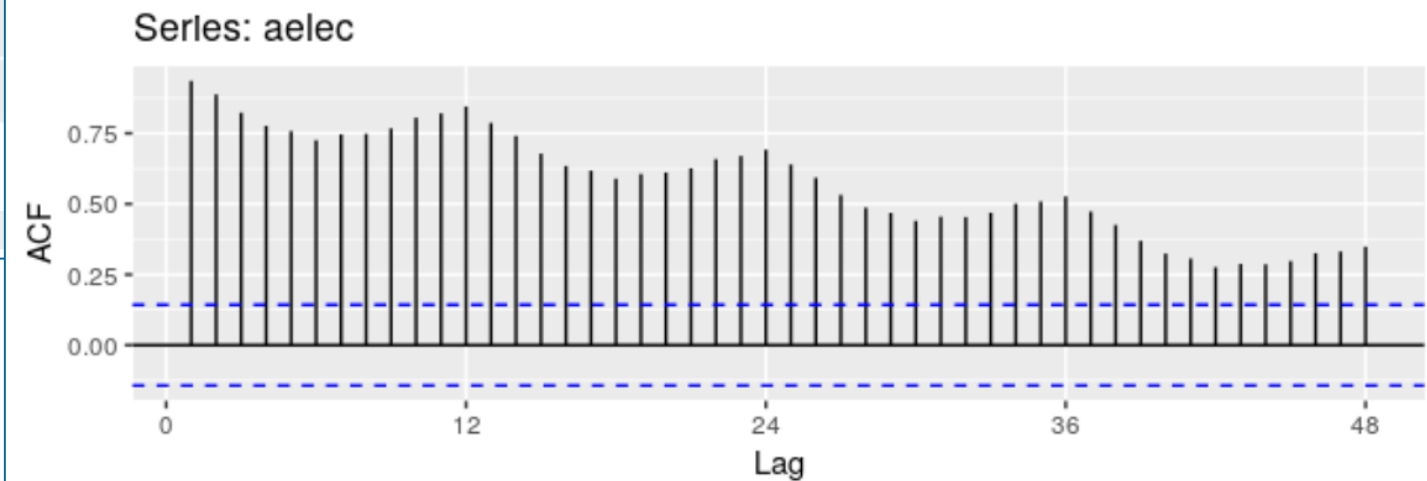


Figure 2.16: ACF of monthly Australian electricity demand.

The slow decrease in the ACF as the lags increase is due to the trend, while the “scalped” shape is due the seasonality.

Common ARIMA specifications and seasonality

<i>White noise</i>	ARIMA(0,0,0)
<i>Random walk</i>	ARIMA(0,1,0)
<i>Random walk with drift</i>	ARIMA(0,1,0) with c
<i>Autoregression</i>	ARIMA(p ,0,0)
<i>Moving average</i>	ARIMA(0,0, q)

Seasonality

$$\text{ARIMA}(p, d, q) (P, D, Q)_m$$

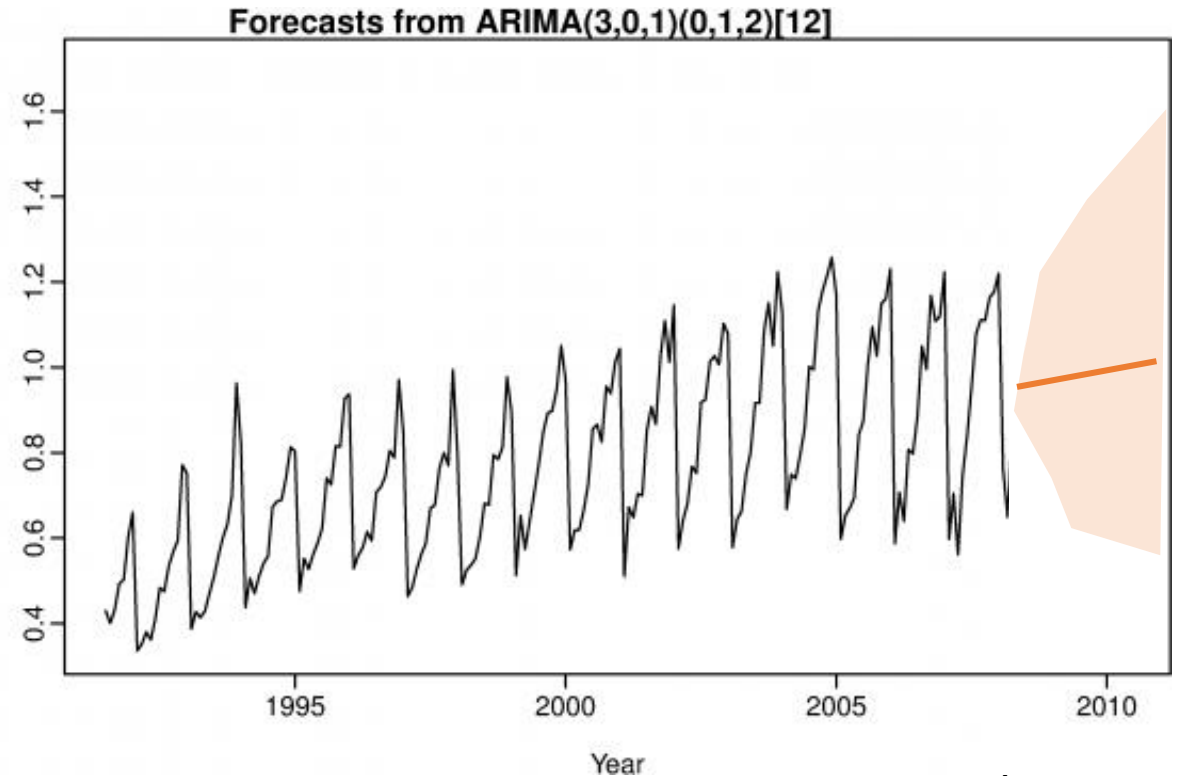
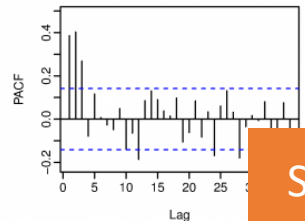
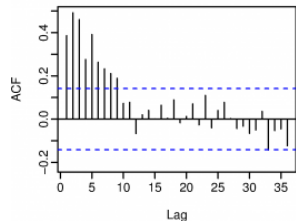
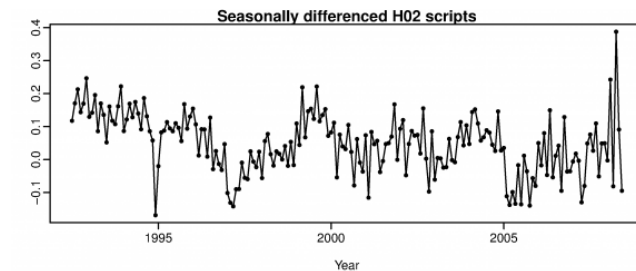
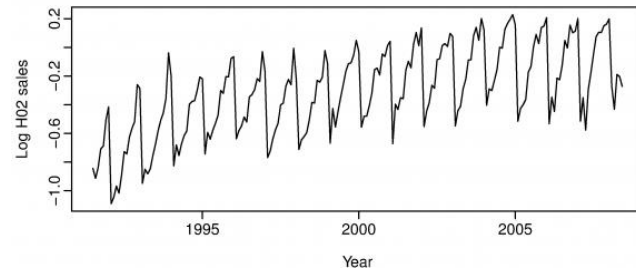
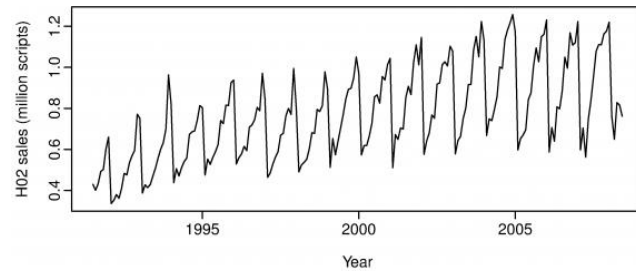
non-seasonal part seasonal part

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

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)

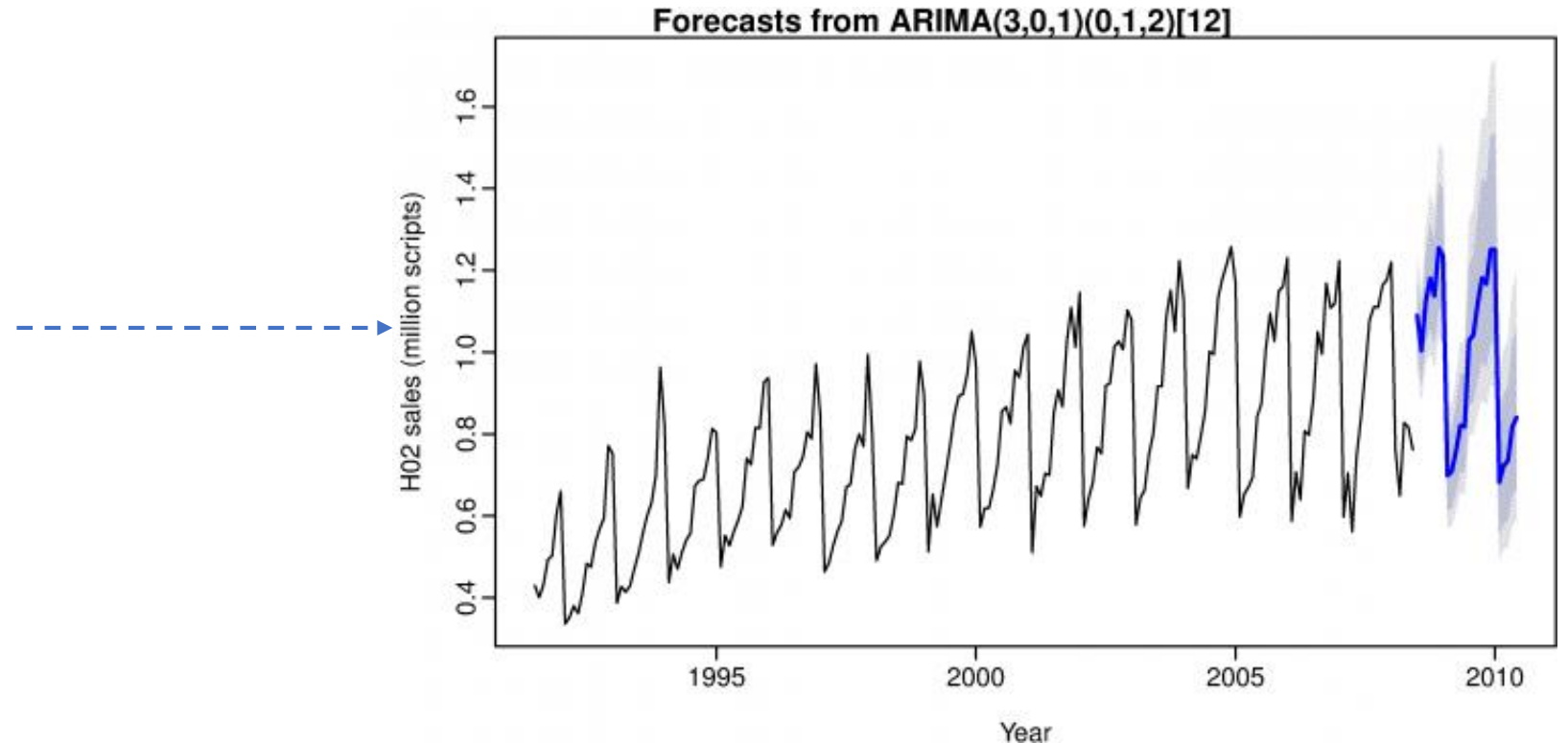
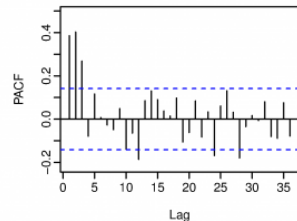
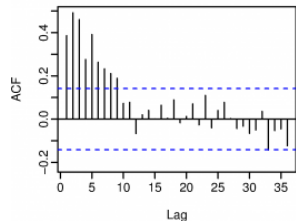
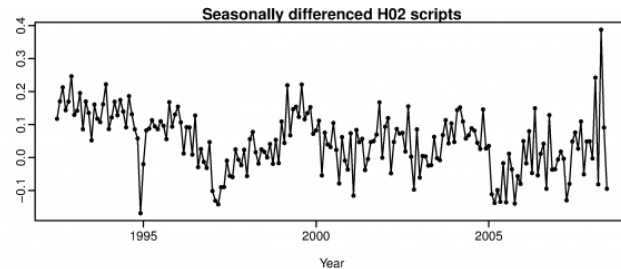
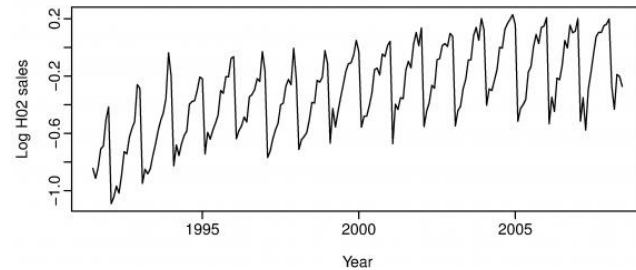
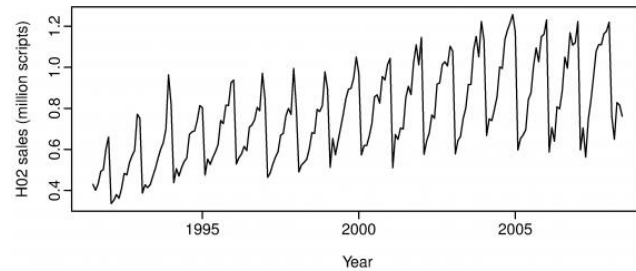


Simple regressions can be **inefficient**, missing out on all the *time-varying* information

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.

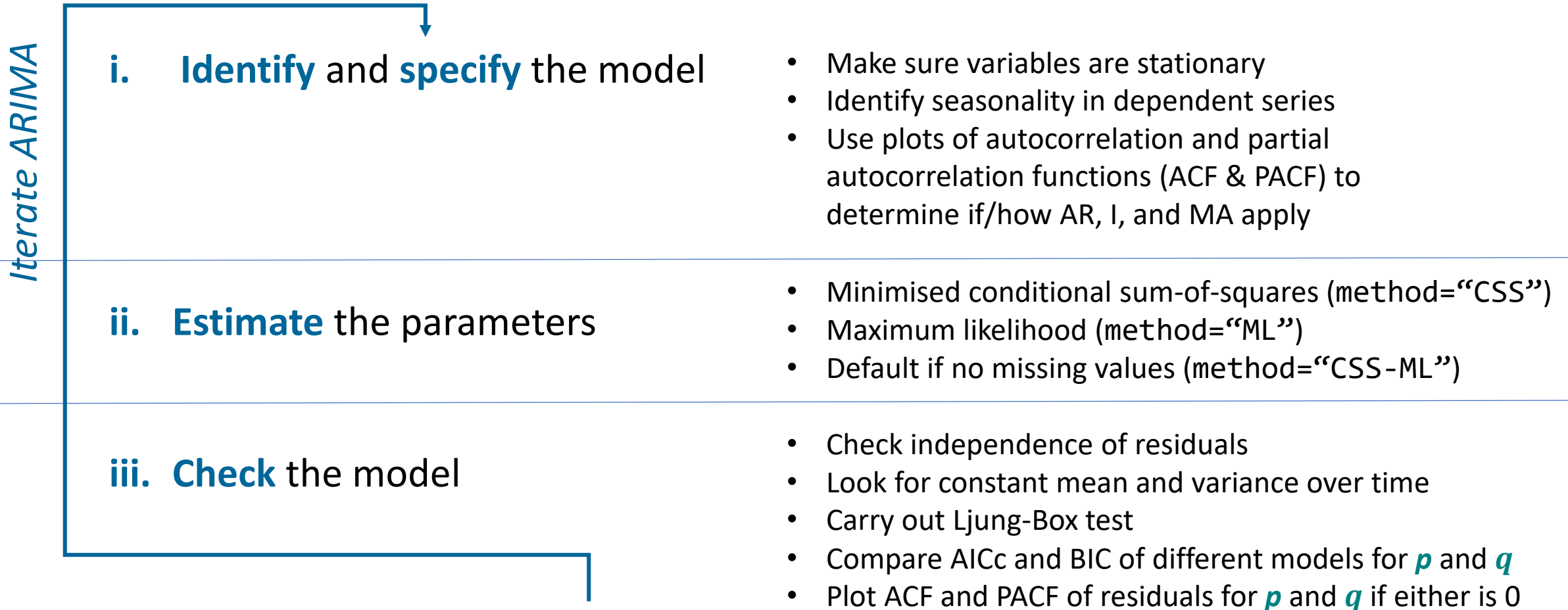
Statistical analysis steps

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()**



Fitting procedures in this practice case

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

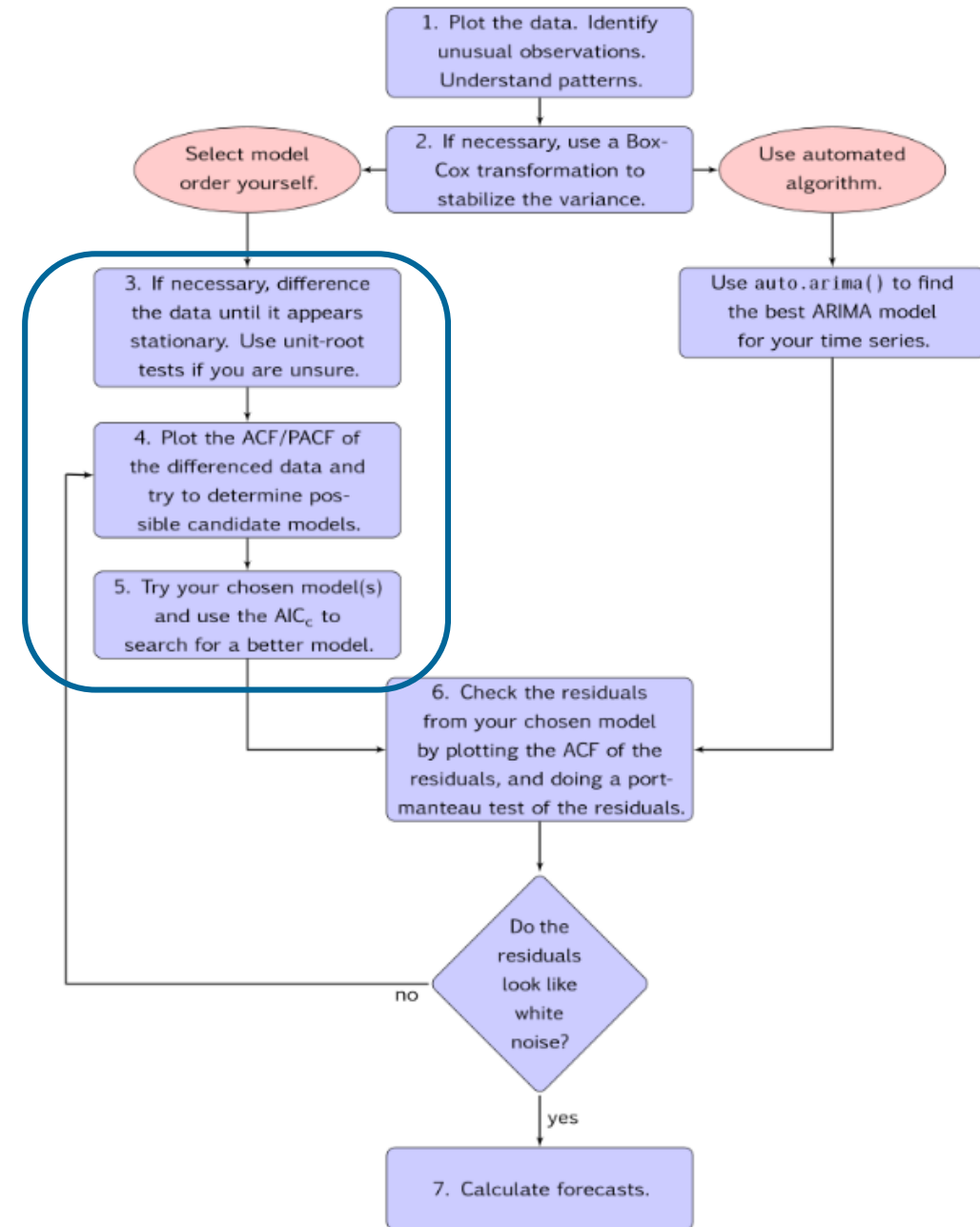


Hyndman-Khandakar algorithm for automatic ARIMA modelling

1. The number of differences $0 \leq d \leq 2$ 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:
 - $\text{ARIMA}(0, d, 0)$,
 - $\text{ARIMA}(2, d, 2)$,
 - $\text{ARIMA}(1, d, 0)$,
 - $\text{ARIMA}(0, d, 1)$.A constant is included unless $d = 2$. If $d \leq 1$, an additional model is also fitted:
 - $\text{ARIMA}(0, d, 0)$ 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.

Fitting procedures in this practice case

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



Practice case

Start by loading libraries and packages

```
12 library(tidyverse)
13 library(forecast)
14 library(here)

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




ONS_pop_est	out_deaths_total	exp_days_dto	exp_days_acute	exp_days_nonacute	exp_patients_dto	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

```
23 str(data_orig)
24 levels(data_orig$i..month)

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

37 data_use <- data_use %>%
38   select(-i..month, -year)
```



	i..month	year	ONS_pop_est	on
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

```
45 data_use <- data_use %>%
46   mutate(annual_mort = out_deaths_total * 1000 / ONS_pop_est,
47          dtoc_days_rate = exp_days_dtoc * 1000 / ONS_pop_est,
48          dtoc_pts_rate = exp_patients_dtoc * 100000 / ONS_pop_est,
49          dtoc_days_percapita = exp_days_dtoc / exp_patients_dtoc,
50          prop_acute_days = exp_days_acute / exp_days_dtoc,
51          prop_acute_patients = exp_patients_acute / exp_patients_dtoc)
52
53 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
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					

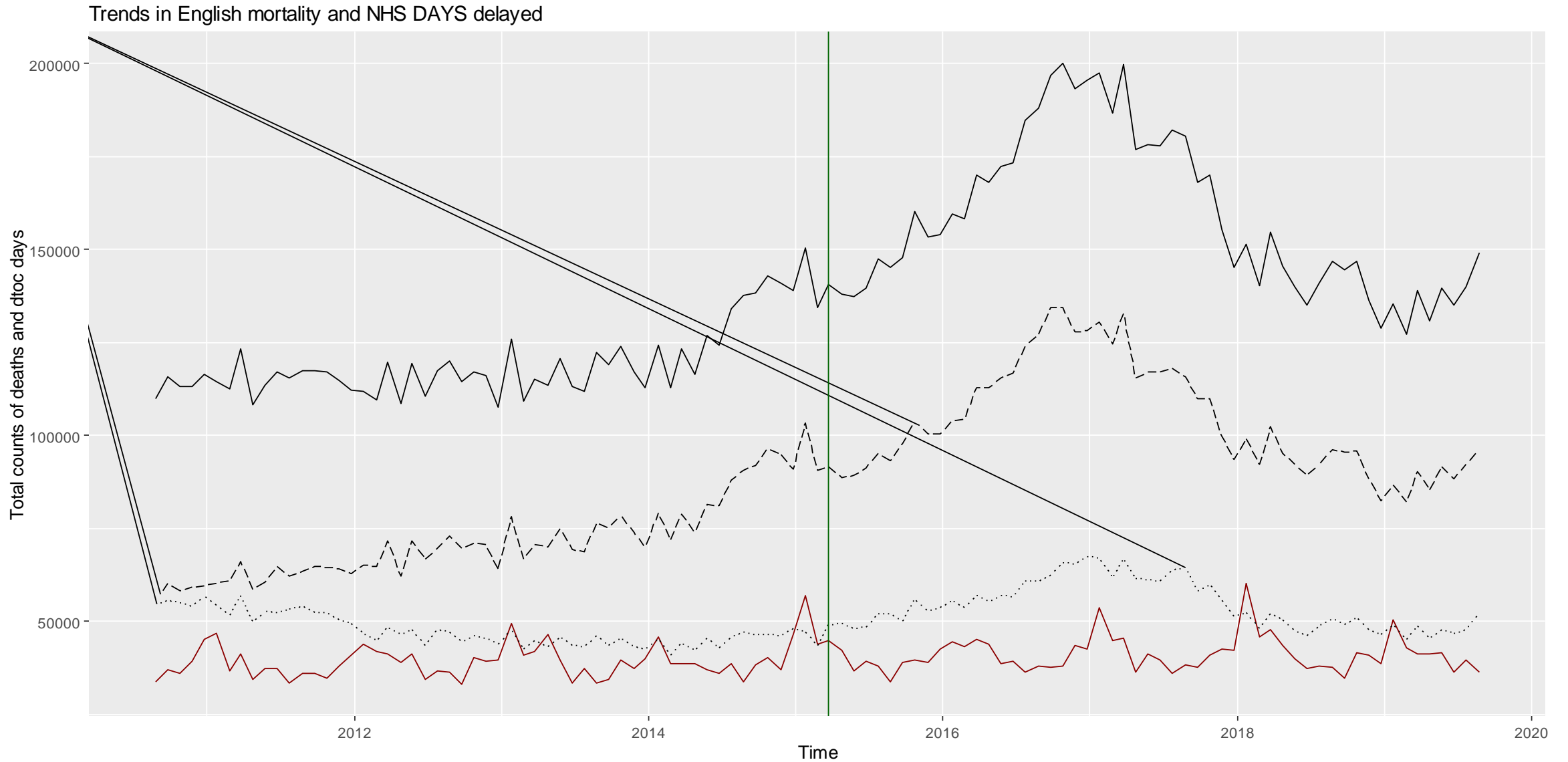
exp_patients_nonacute	date	annual_mort	dtoc_days_rate	dtoc_pts_rate	dtoc_days_pe
rcapita					
Min. :1371	Min. :2010-08-24	Min. :0.5822	Min. :1.903	Min. : 6.140	Min. :28.0
0					
1st Qu.:1522	1st Qu.:2012-11-24	1st Qu.:0.6441	1st Qu.:2.057	1st Qu.: 6.804	1st Qu.:30.0
0					
Median :1635	Median :2015-02-24	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)

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

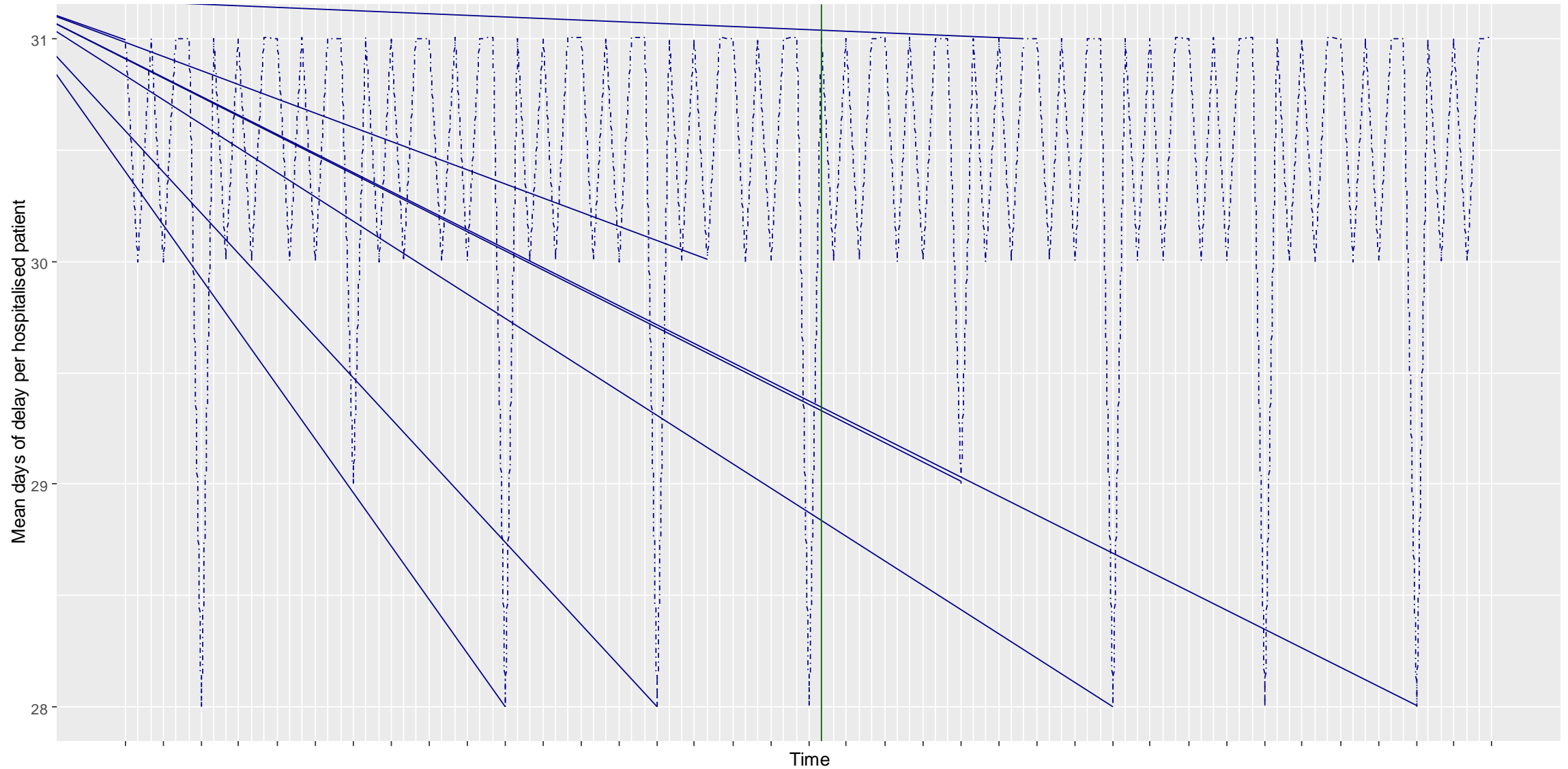
3. Explore visual plots of the raw data across overlapping exposure(s) and outcome(s)



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

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

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



4. Create time series objects and examine for seasonality

```
114 data_green <- data_use %>%  
115   filter(date<"2016-04-24")
```

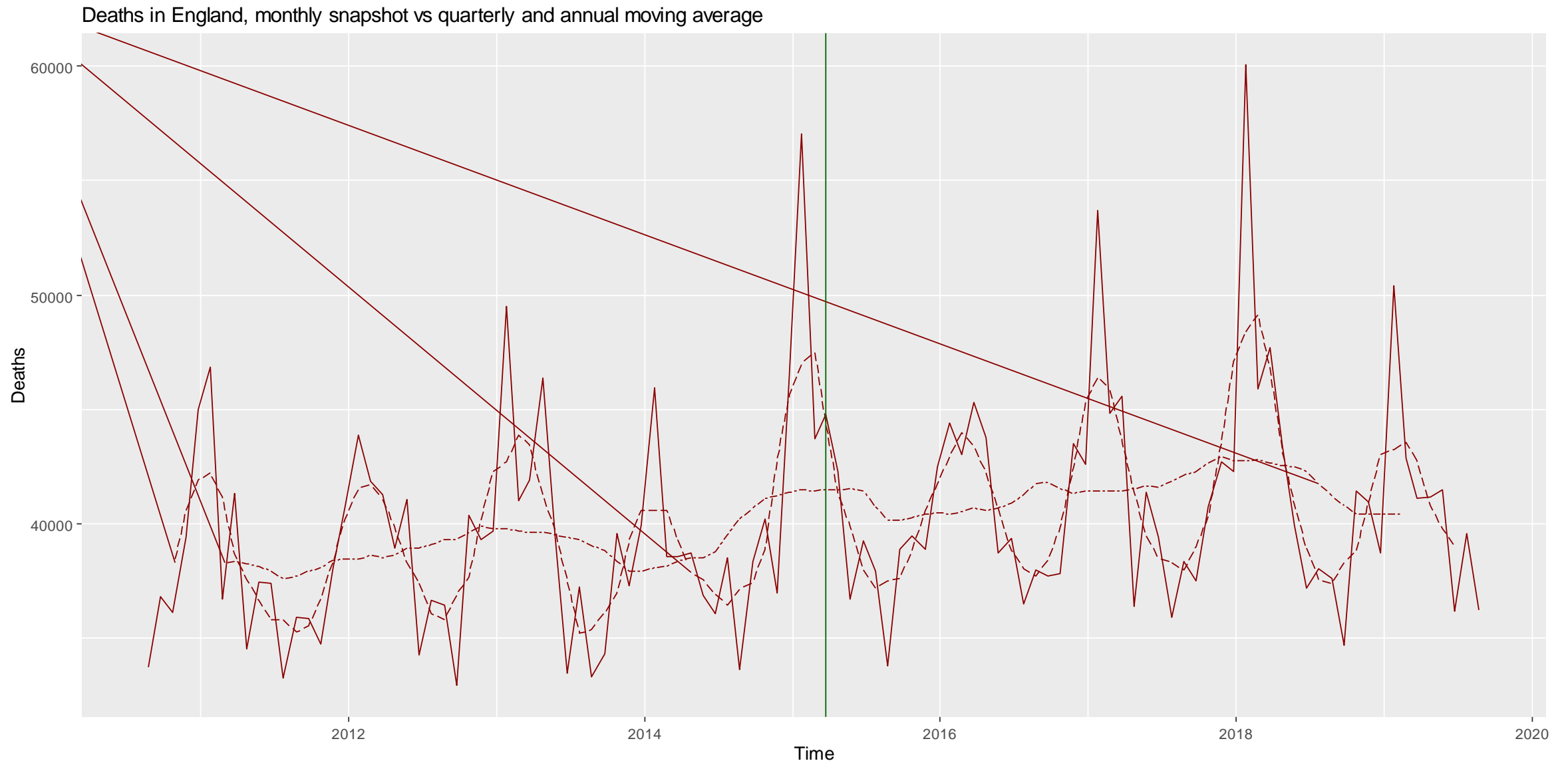
```
119 ts_dtocdays_all <- ts(data_use$exp_days_dtoc, frequency=12, start=c(2010, 8))  
120 ts_deaths_all <- ts(data_use$out_deaths_total, frequency=12, start=c(2010, 8))  
  
122 ts_dtocdays_green <- ts(data_green$exp_days_dtoc, frequency=12, start=c(2010, 8))  
123 ts_deaths_green <- ts(data_green$out_deaths_total, frequency=12, start=c(2010, 8))
```


4. Create time series objects and examine for seasonality

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

133 ggplot(data=data_use, aes(x=date)) +
134 ggtitle("Deaths in England, monthly snapshot vs quarterly and annual moving
average") +
135 geom_line(aes(y=out_deaths_total), color="darkred") +
136 geom_line(aes(y=deaths_ma_q), color="darkred", linetype="longdash") +
137 geom_line(aes(y=deaths_ma_y), color="darkred", linetype="twodash") +
138 xlab("Time") + ylab("Deaths") +
139 geom_vline(aes(xintercept = as.numeric(as.Date("2015/03/24"))), color="darkgreen")
```

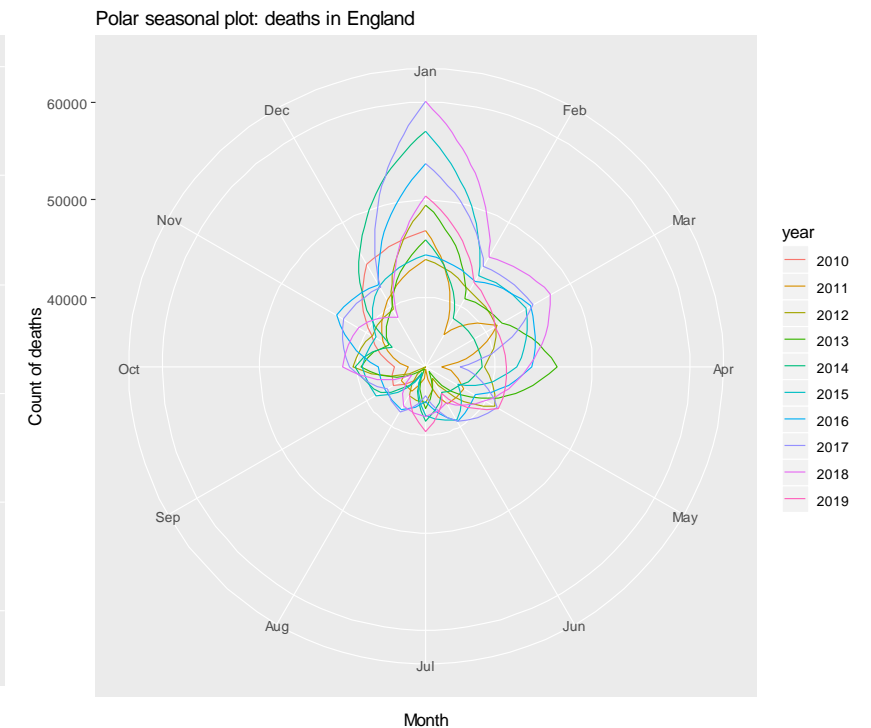
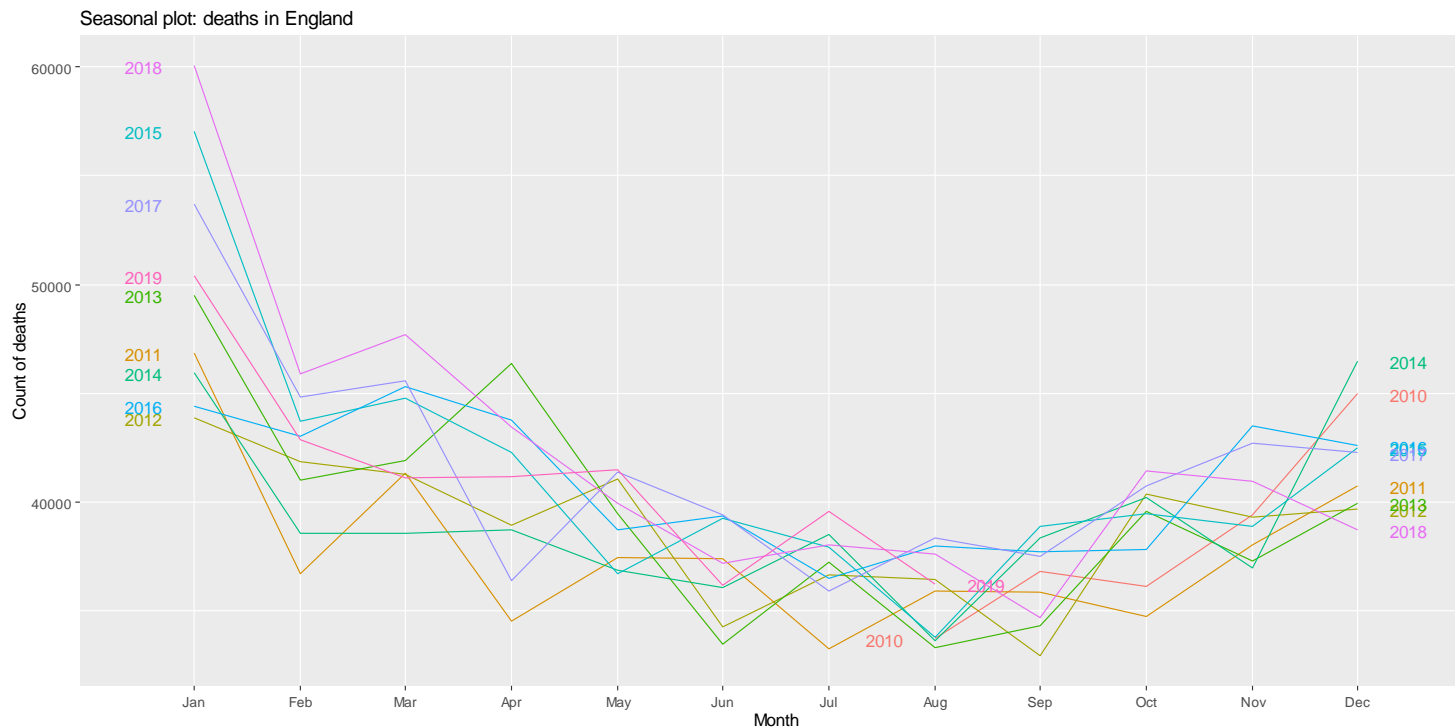
4. Create time series objects and examine for seasonality



4. Create time series objects and examine for seasonality

Deaths in England

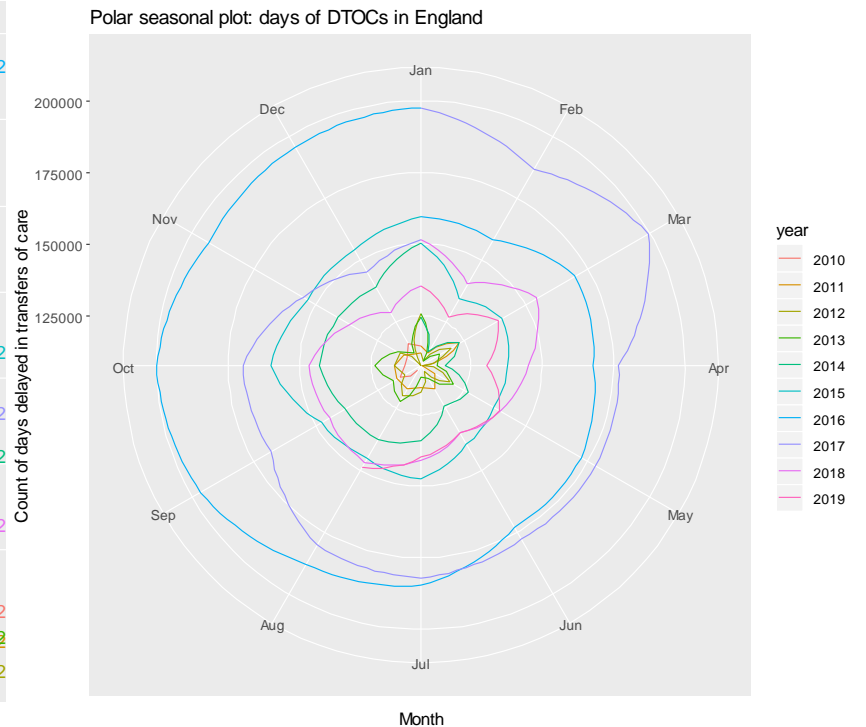
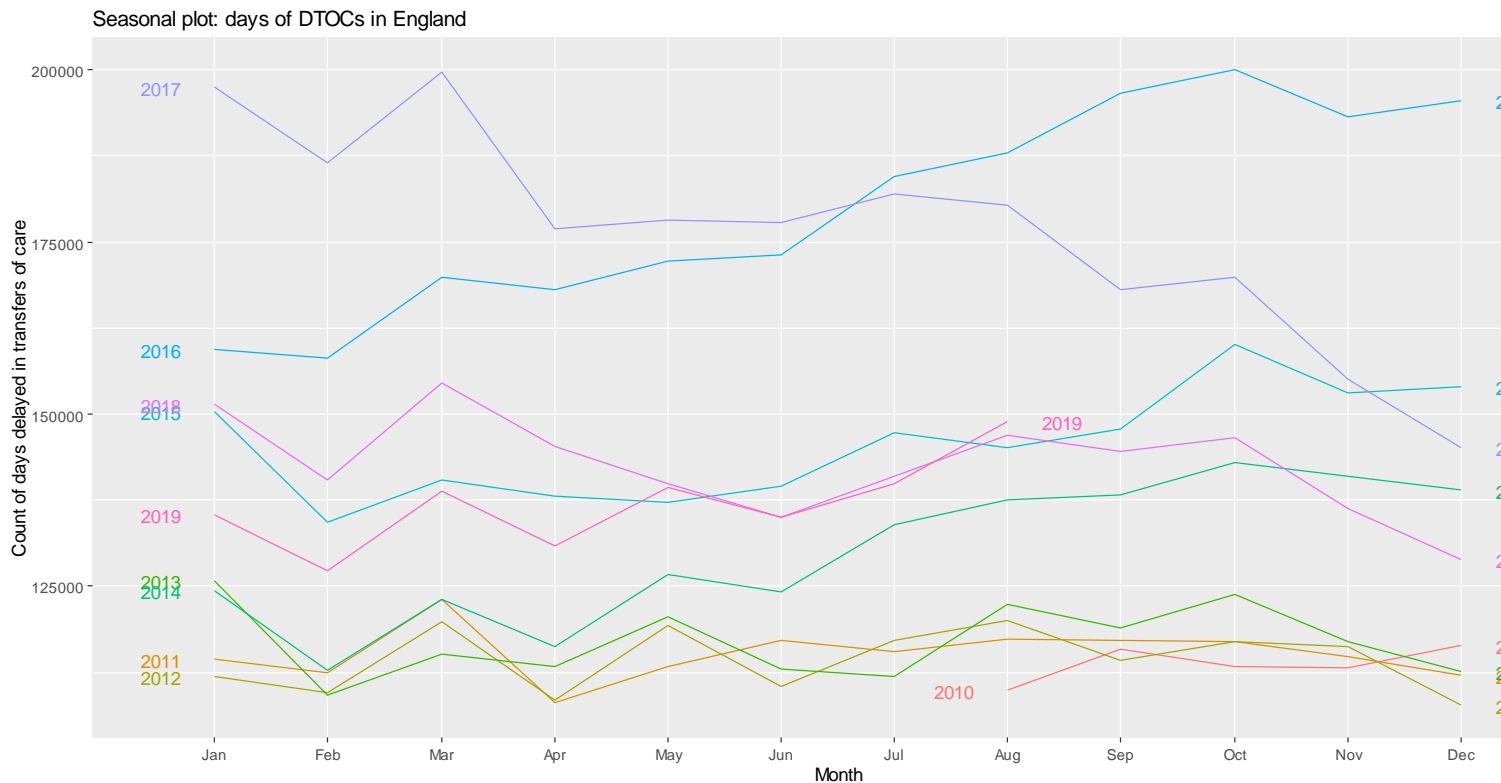
```
155 ggseasonplot(ts_deaths_all, year.labels=TRUE, year.labels.left=TRUE) +  
156   ylab("Count of deaths") + ggtitle("Seasonal plot: deaths in England")  
  
162 ggseasonplot(ts_death_all, polar=TRUE) +  
163   ylab("Count of deaths") + ggtitle("Polar seasonal plot: deaths in England")
```



4. Create time series objects and examine for seasonality

Days of delayed transfers of care in NHS England

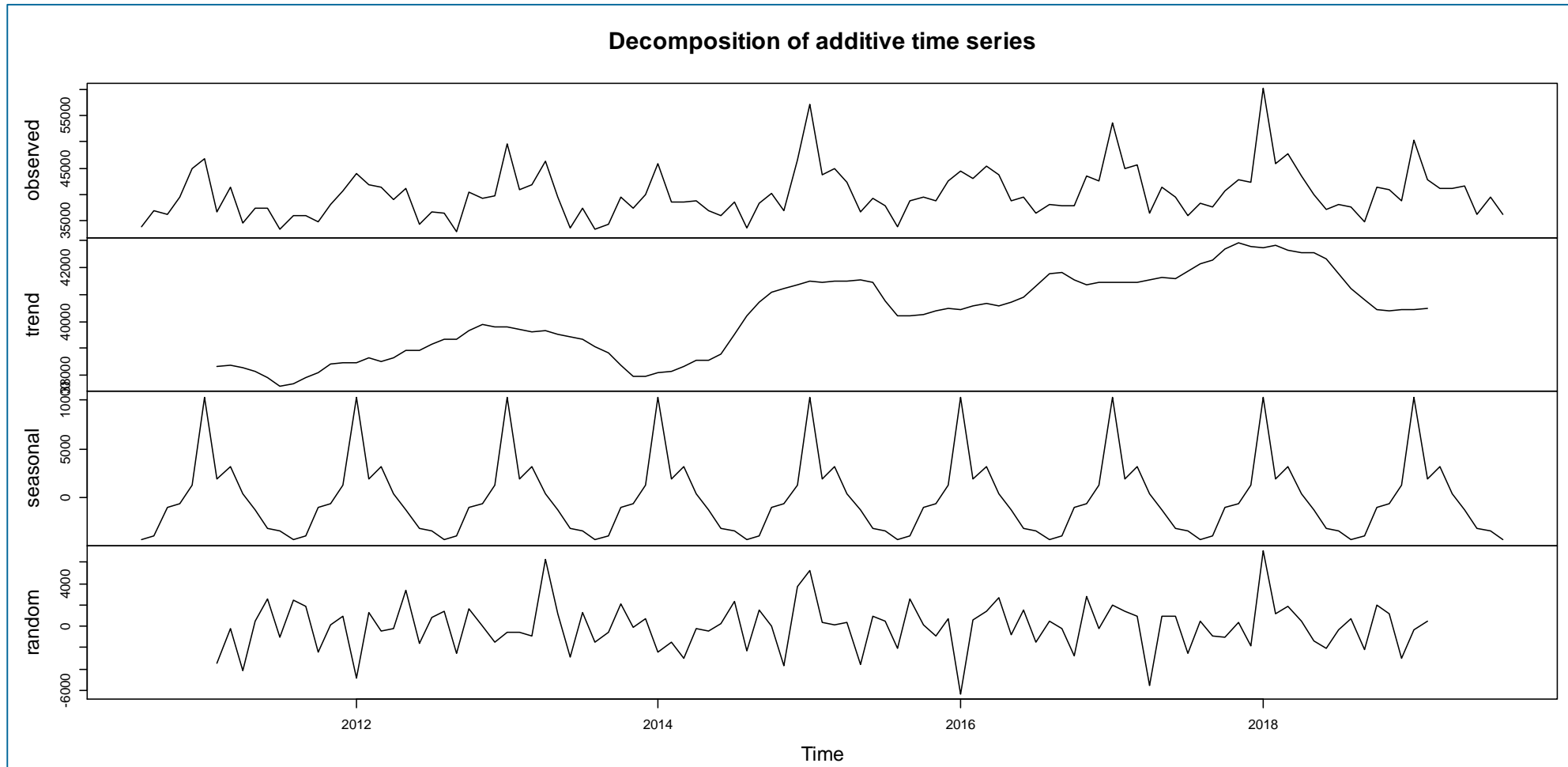
```
158 ggseasonplot(ts_dtocdays_all, year.labels=TRUE, year.labels.left=TRUE) +  
159   ylab("Count of days delayed in transfers of care") + ggtitle("Seasonal plot: days  
of DTOCs in England")  
165 ggseasonplot(ts_dtocdays_all, polar=TRUE) +  
166   ylab("Count of days delayed in transfers of care") + ggtitle("Polar seasonal  
plot: days of DTOCs in England")
```



5. Decompose the time series of interest

Deaths in England

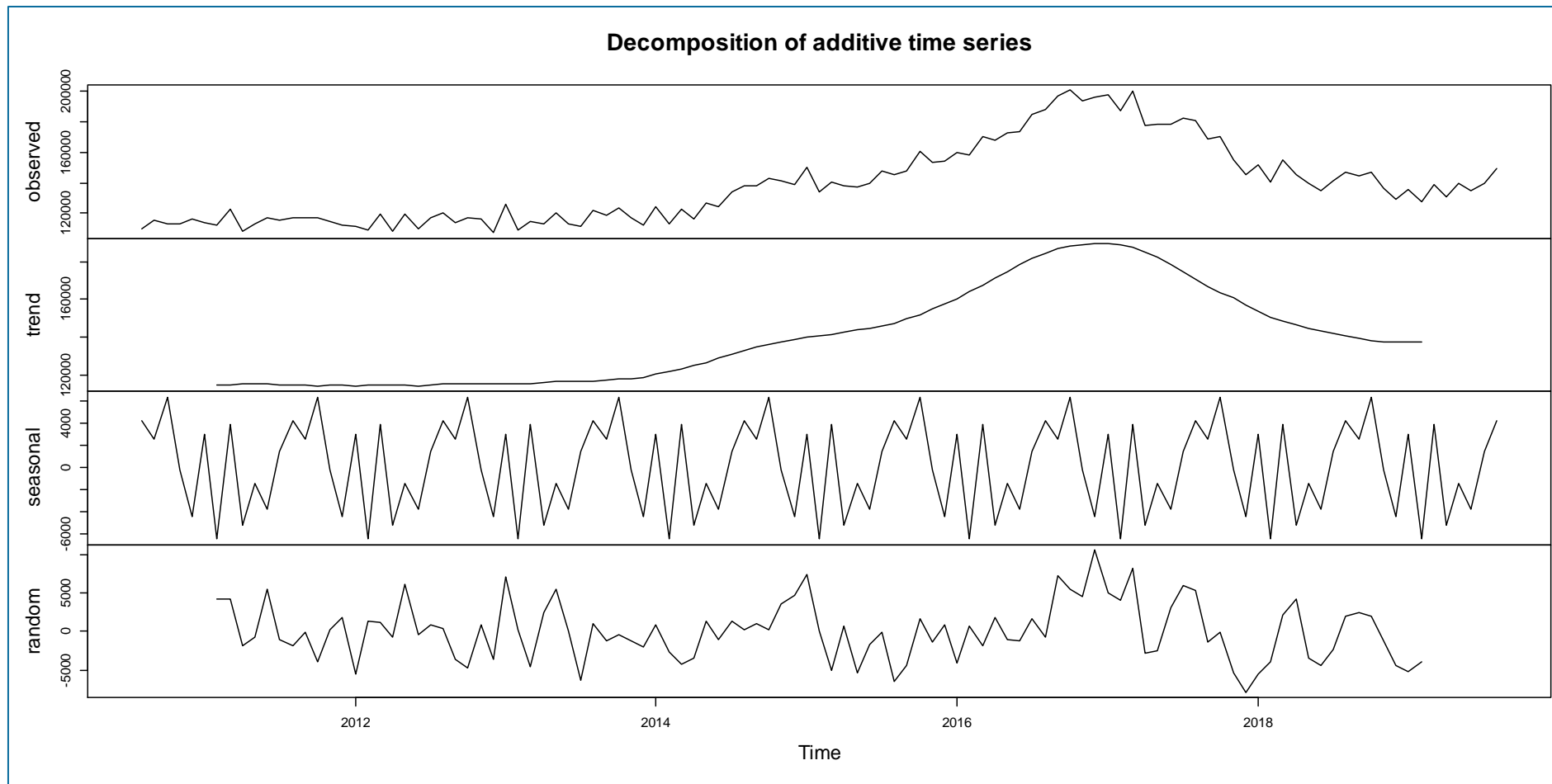
```
179 decomp_deaths_add <-  
180   decompose(ts_death_counts, type="additive")  
181 plot(decomp_deaths_add)
```



5. Decompose the time series of interest

Days of delayed transfers of care in NHS England

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



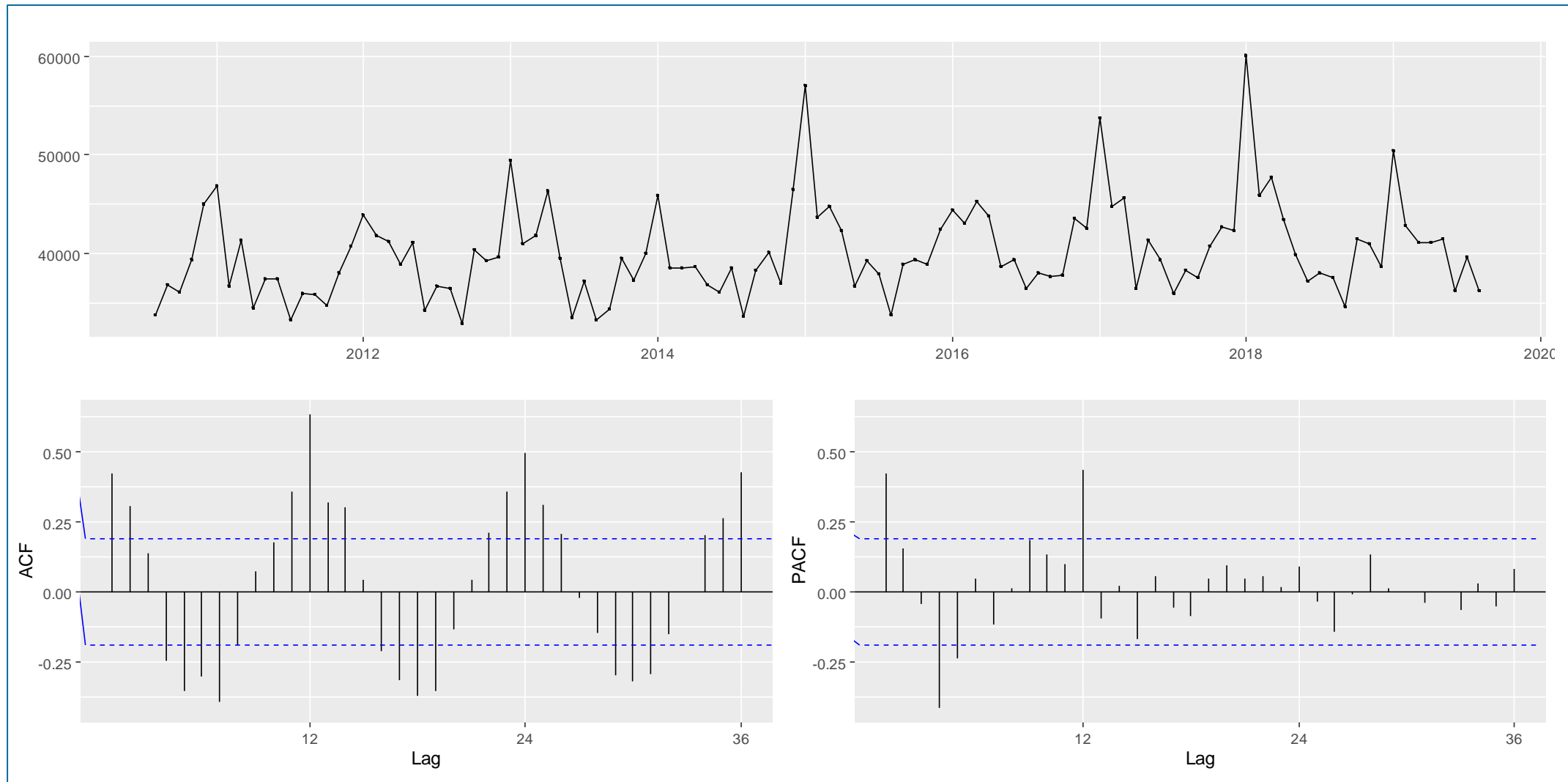
6. Evaluate stationarity assumption of decompositions

Deaths in England

```
203  Acf(ts_deaths_all, lag.max = NULL, type = c("correlation", "covariance",  
                                                "partial"), plot = TRUE, na.action = na.contiguous,  
demean = TRUE)  
  
205  Pacf(ts_deaths_all, lag.max = NULL, plot = TRUE, na.action = na.contiguous, demean  
= TRUE)  
  
206  ts_deaths_all %>% ggtsdisplay() # summary view across data, Acf, and Pacf  
207  Box.test(diff(ts_deaths_all), lag=12, type="Ljung-Box")  
  
214  ts_deaths_all %>% diff(lag=12) %>% ggtsdisplay()  
215  ts_deaths_all %>% diff(lag=12) %>% diff(lag=12) %>% ggtsdisplay()
```

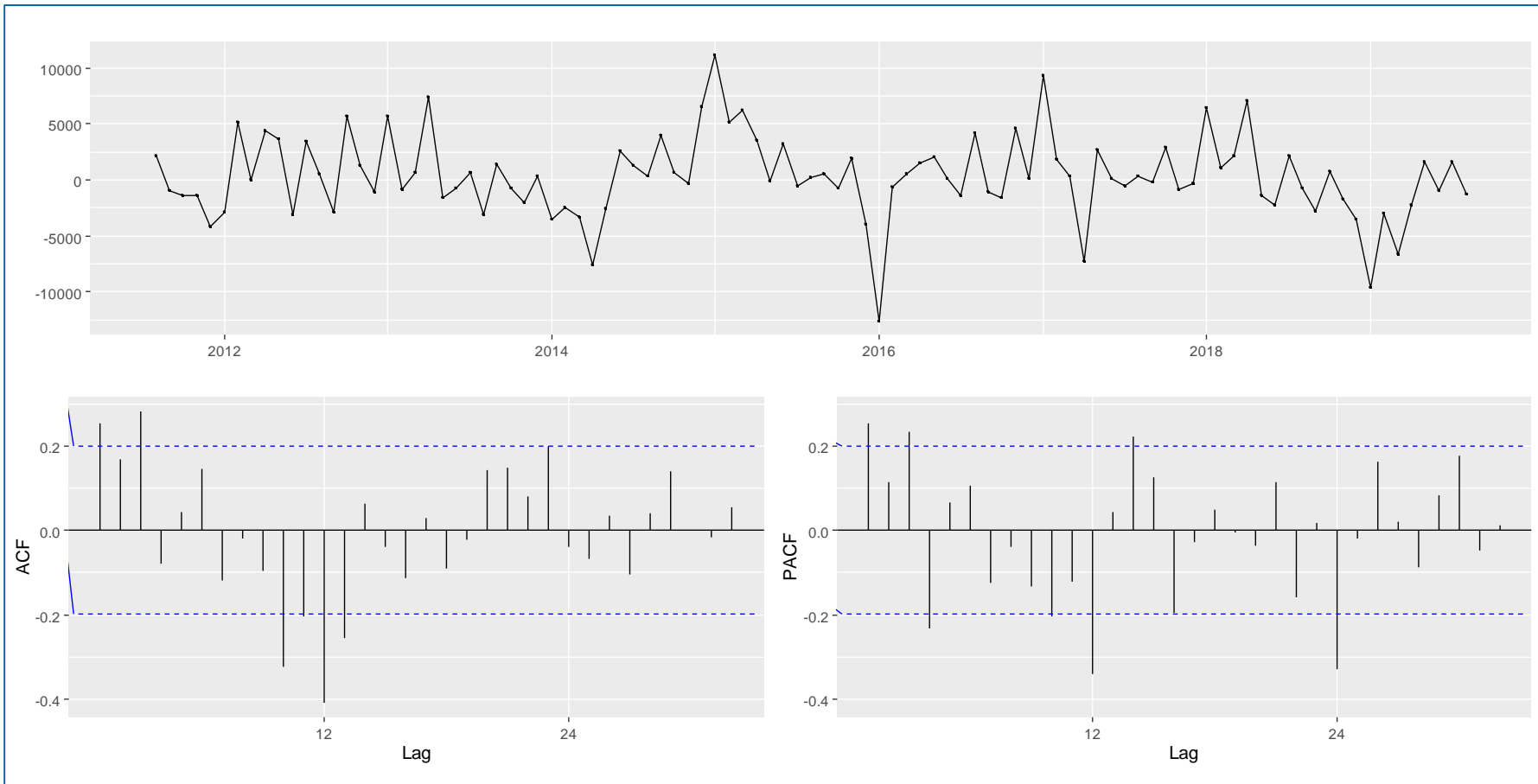
6. Evaluate stationarity assumption of decompositions

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

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

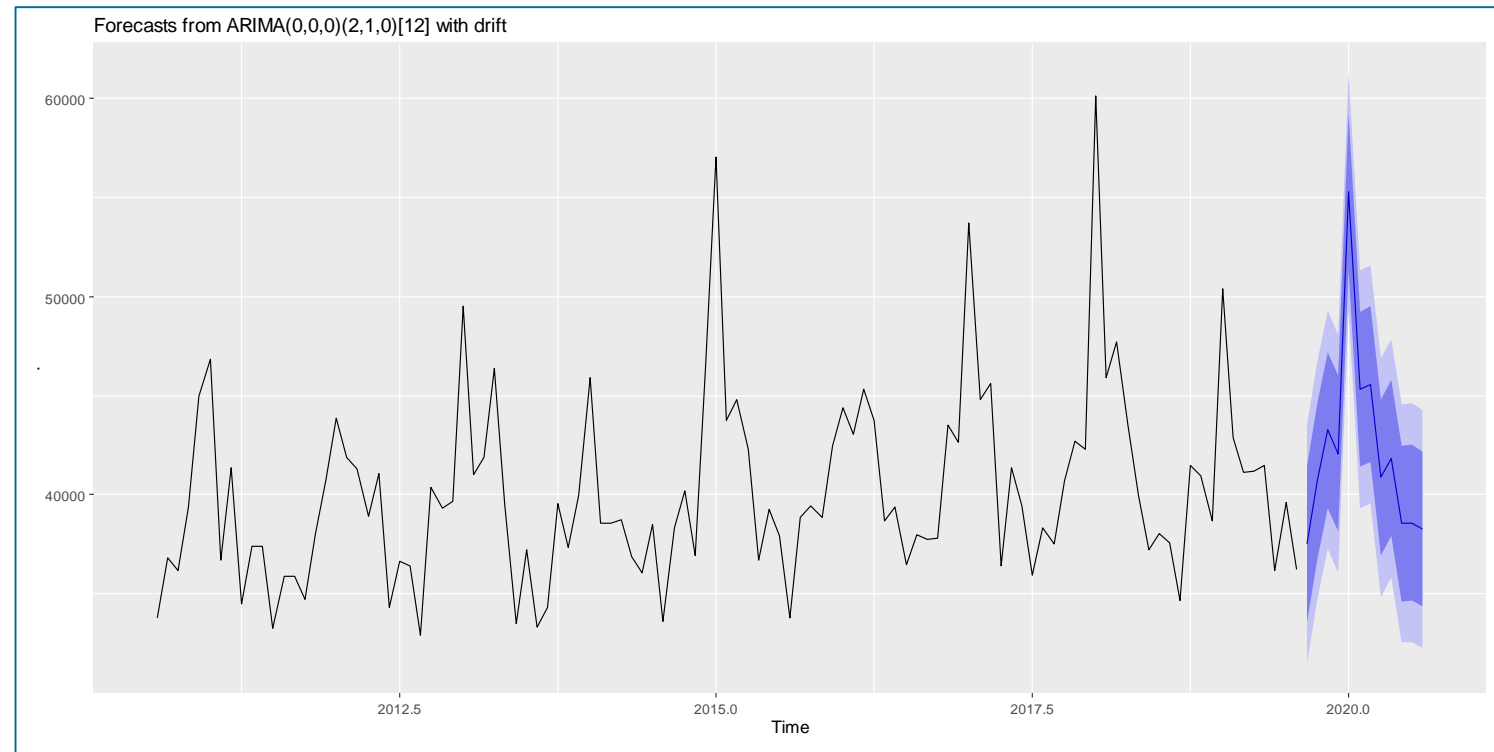
Series: .
ARIMA(0,0,0)(2,1,0)[12] with drift

Coefficients:

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


sigma² estimated as 9354336: log likelihood=-918.08
AIC=1844.16 AICc=1844.6 BIC=1854.46

```
225 fit_deaths_all %>% forecast(h=12) %>% autoplot()
```



7. Fit univariate ARIMA models, with forecast using `auto.arima()` *Deaths in England – Green et al replication, through March 2016*

```
227 fit_deaths_green <- ts_deaths_green %>%  
228   auto.arima()  
229 fit_deaths_green
```

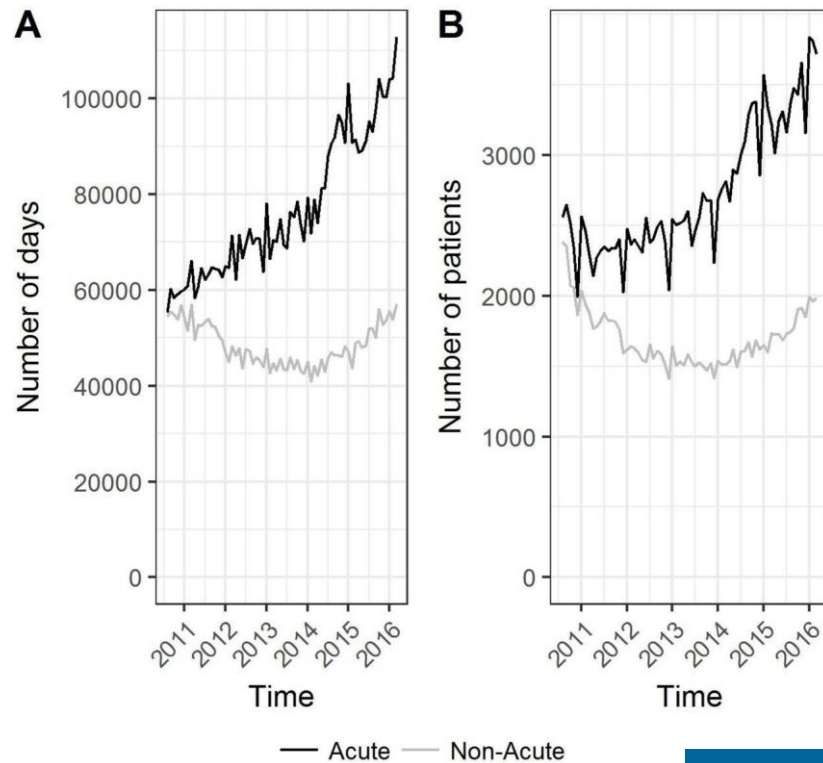


```
> fit_deaths_green  
Series: .  
ARIMA(0,0,0)(1,1,0)[12] with drift  
  
Coefficients:  
          sar1      drift  
      -0.5258   44.0512  
s.e.    0.1176   25.5866  
  
sigma^2 estimated as 10826652:  log likelihood=-533.91  
AIC=1073.83   AICc=1074.29   BIC=1079.9
```

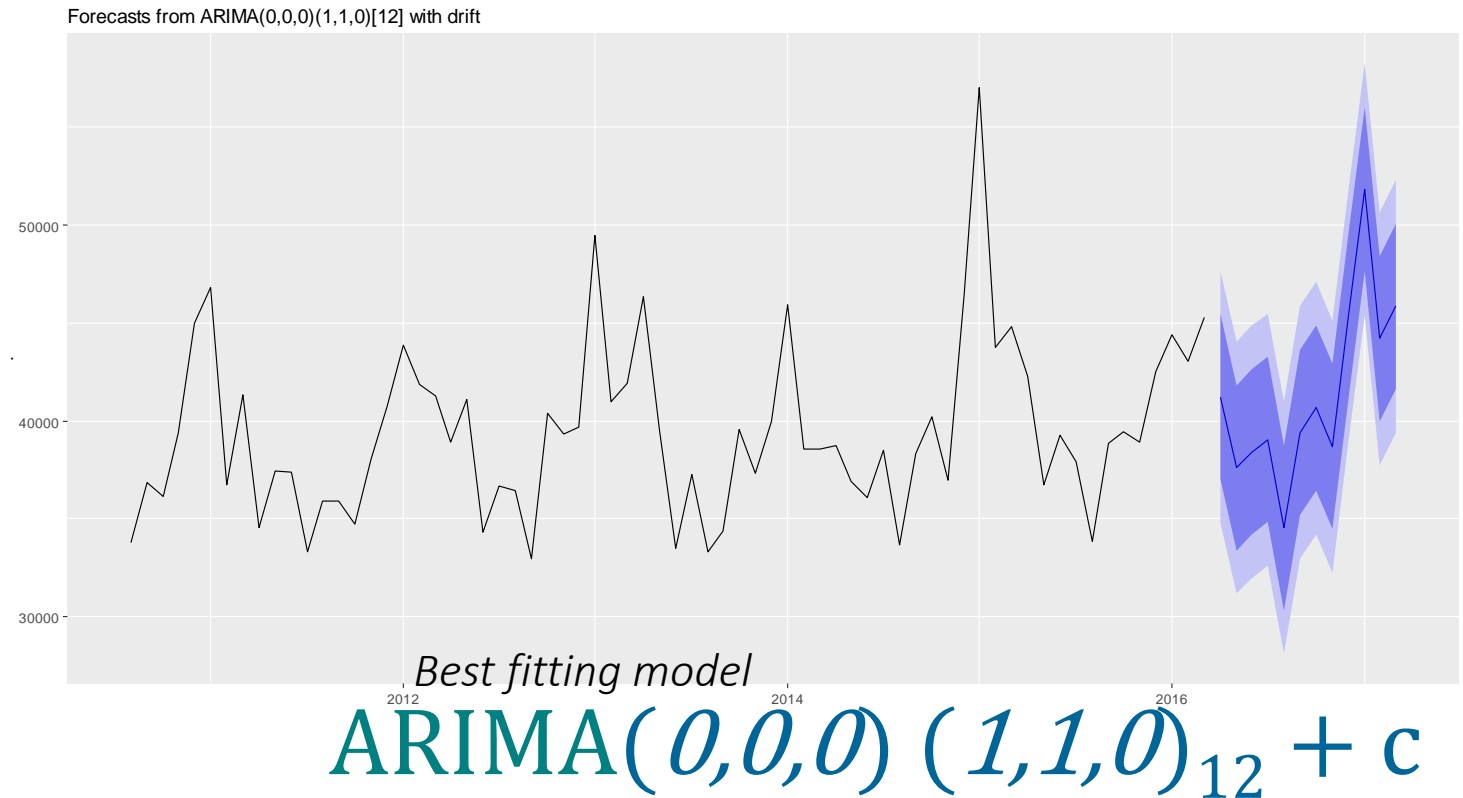
```
231 fit_deaths_green %>% forecast(h=12) %>% autoplot()
```

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



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

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

```
237 checkresiduals(fit_deaths_all, lag=12)
238 checkresiduals(fit_deaths_green)
```

```
> checkresiduals(fit_deaths_all)
```

Ljung-Box test

data: Residuals from ARIMA(0,0,0)(2,1,0)[12] with drift
Q* = 124.64, df = 19, p-value < 2.2e-16

Model df: 3. Total lags used: 22

```
> checkresiduals(fit_deaths_green)
```

Ljung-Box test

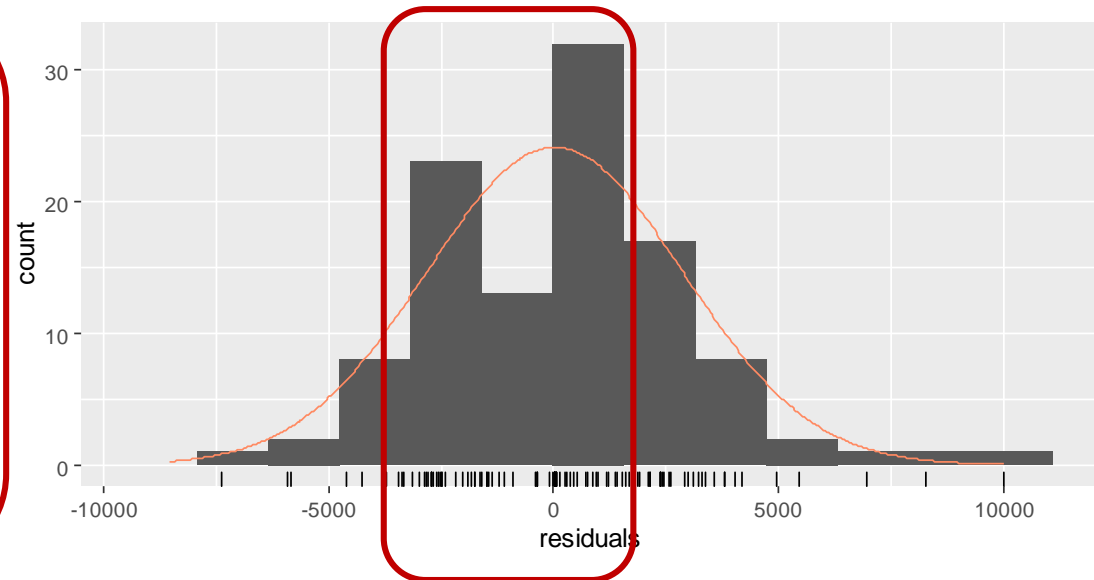
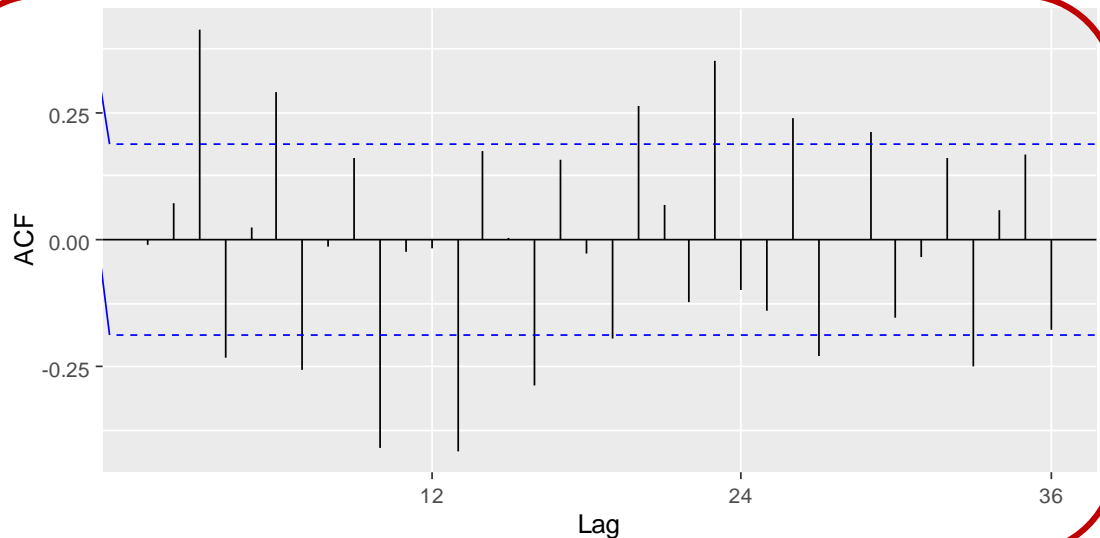
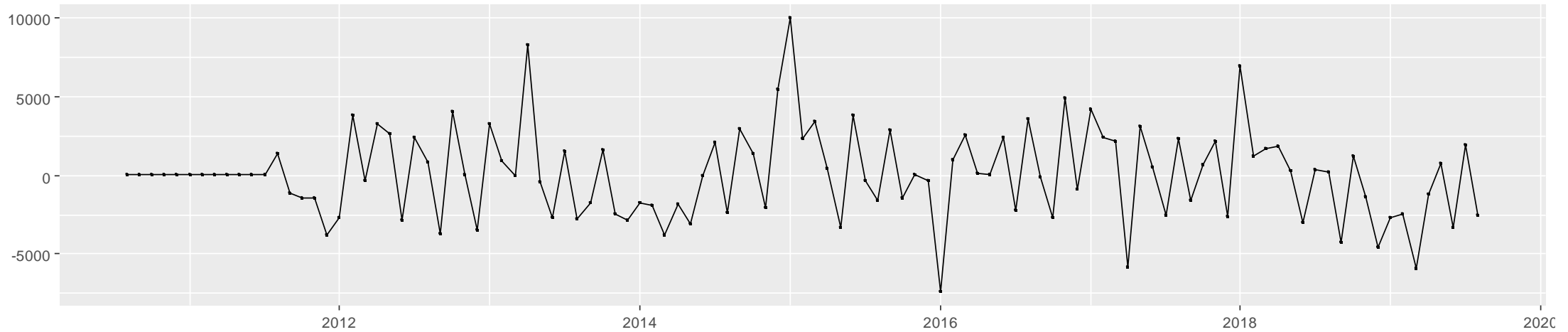
data: Residuals from ARIMA(0,0,0)(1,1,0)[12] with drift
Q* = 68.616, df = 12, p-value = 5.809e-10

Model df: 2. Total lags used: 14

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

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

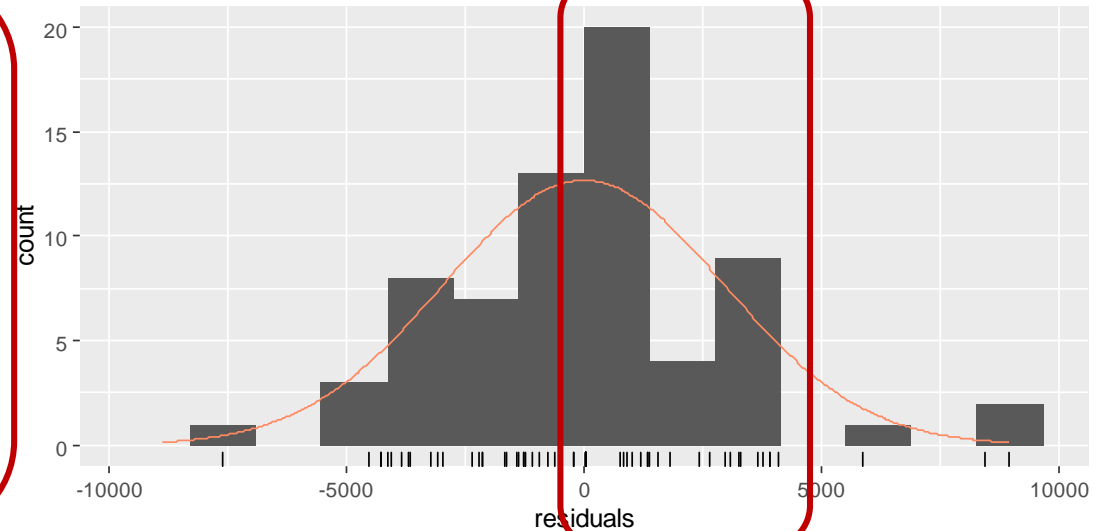
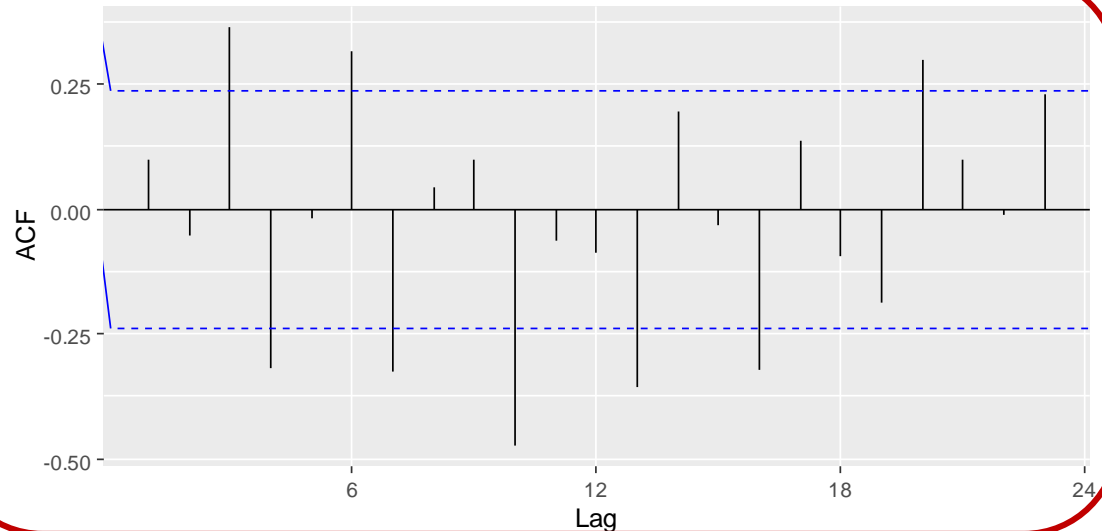
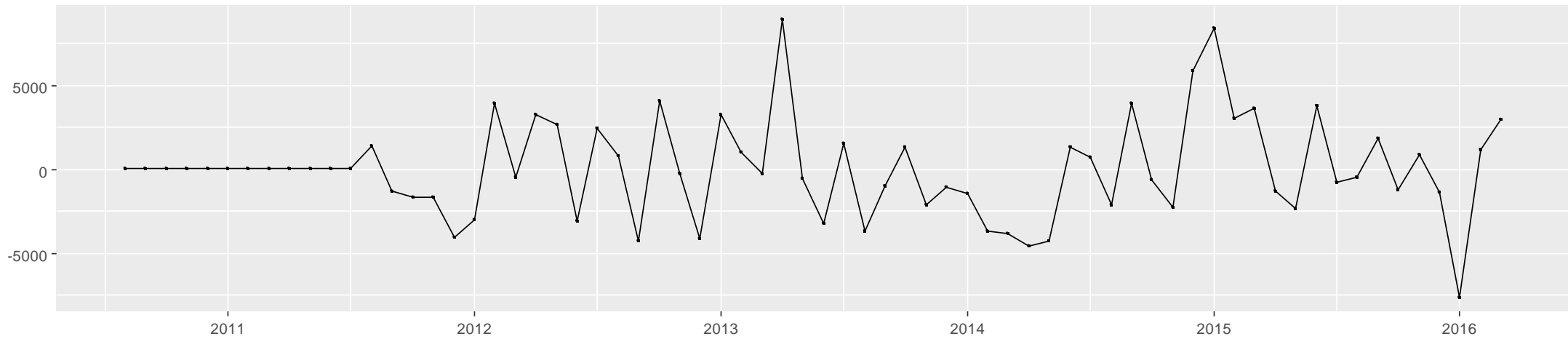
Residuals from ARIMA(0,0,0)(2,1,0)[12] with drift



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

Deaths in England – Green et al replication, through March 2016

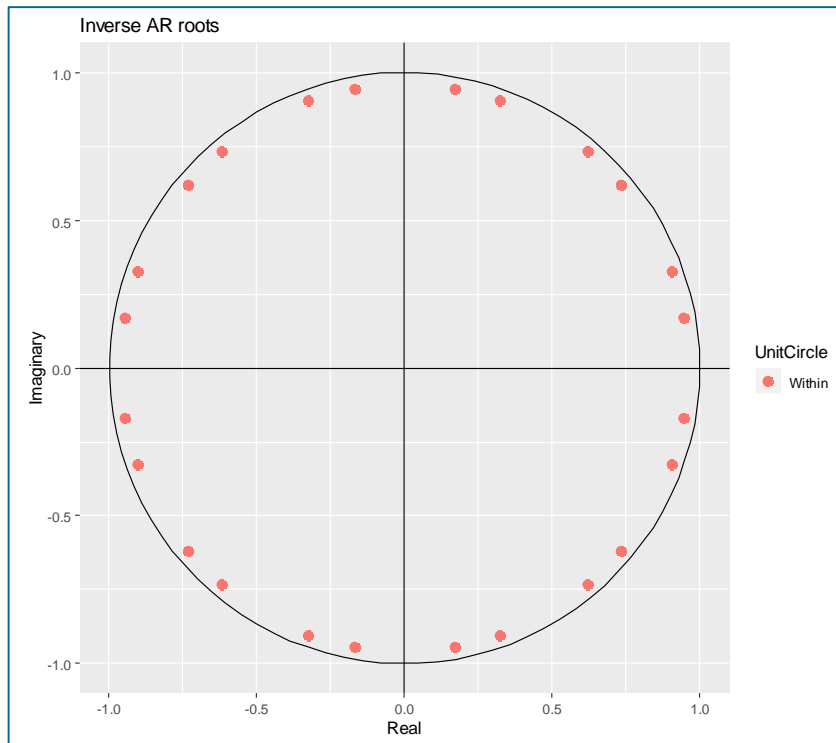
Residuals from ARIMA(0,0,0)(1,1,0)[12] with drift



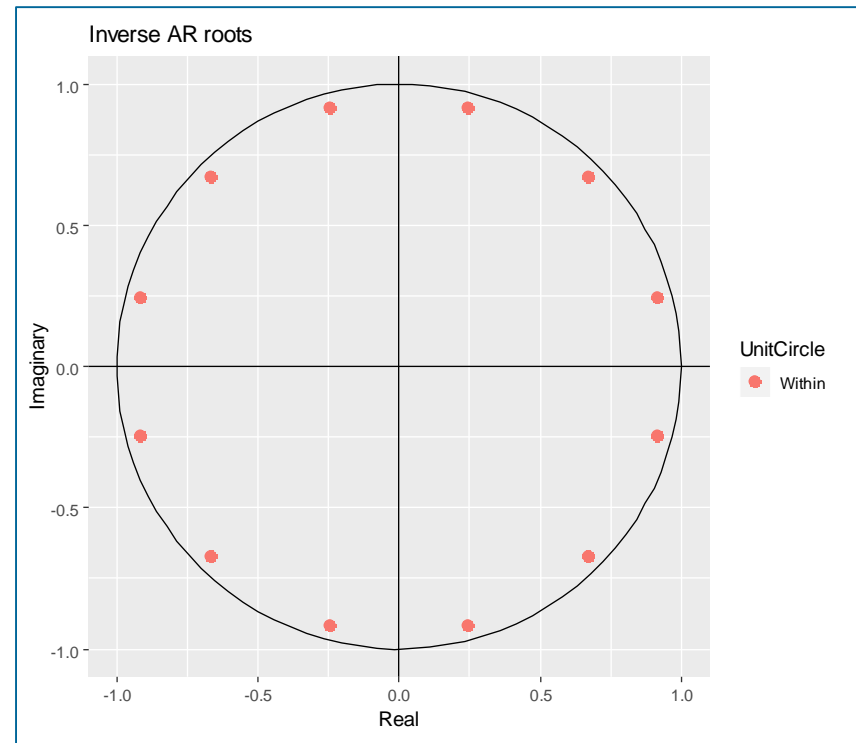
8. Check forecasts for fit and predictiveness, refining and updating as needed

```
248 autoplot(refit_deaths_all)  
249 autoplot(refit_deaths_green)
```

All current data through August 2019



Green et al replication through March 2016




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

```
256 ts_acutedays_all <- ts(data_use$exp_days_acute, frequency=12, start=c(2010, 8))
257 ts_acutedays_green <- ts(data_green$exp_days_acute, frequency=12, start=c(2010, 8))
```

Firstly, can we replicate the same estimates as [the Green et al paper](#) through March 2016?

```
260 regress_green_auto <- auto.arima(ts_deaths_green, xreg=ts_acutedays_green,
allowdrift=TRUE)
261 regress_green_auto
262 regress_green_auto %>% forecast(xreg=ts_acutedays_green, h=12) %>% autoplot()
```

Er, nope.



```
> regress_green_auto
Series: ts_deaths_green
Regression with ARIMA(1,0,1)(1,1,0)[12] errors


Coefficients:
          ar1          ma1          sar1          drift          xreg
      -0.5583    0.8870   -0.4167   -104.6961    0.2148
s.e.    0.2320    0.1938    0.1466    63.8297    0.0816

sigma^2 estimated as 8463055:  log likelihood=-524.99
AIC=1061.98  AICc=1063.69  BIC=1074.13
```

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

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

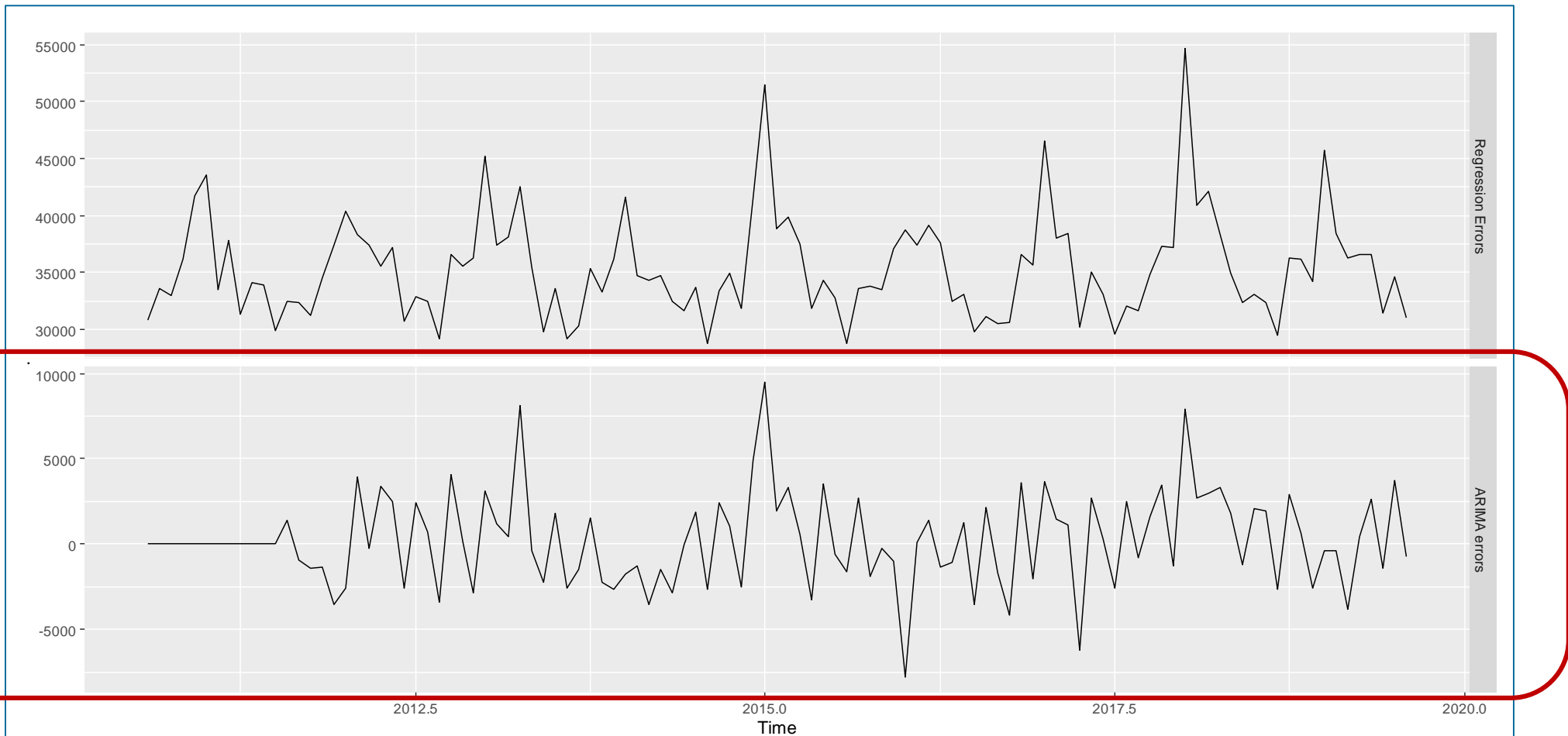
Next, what happens when we include the additional 41 further months of data now available (through August 2019)?



```
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
```

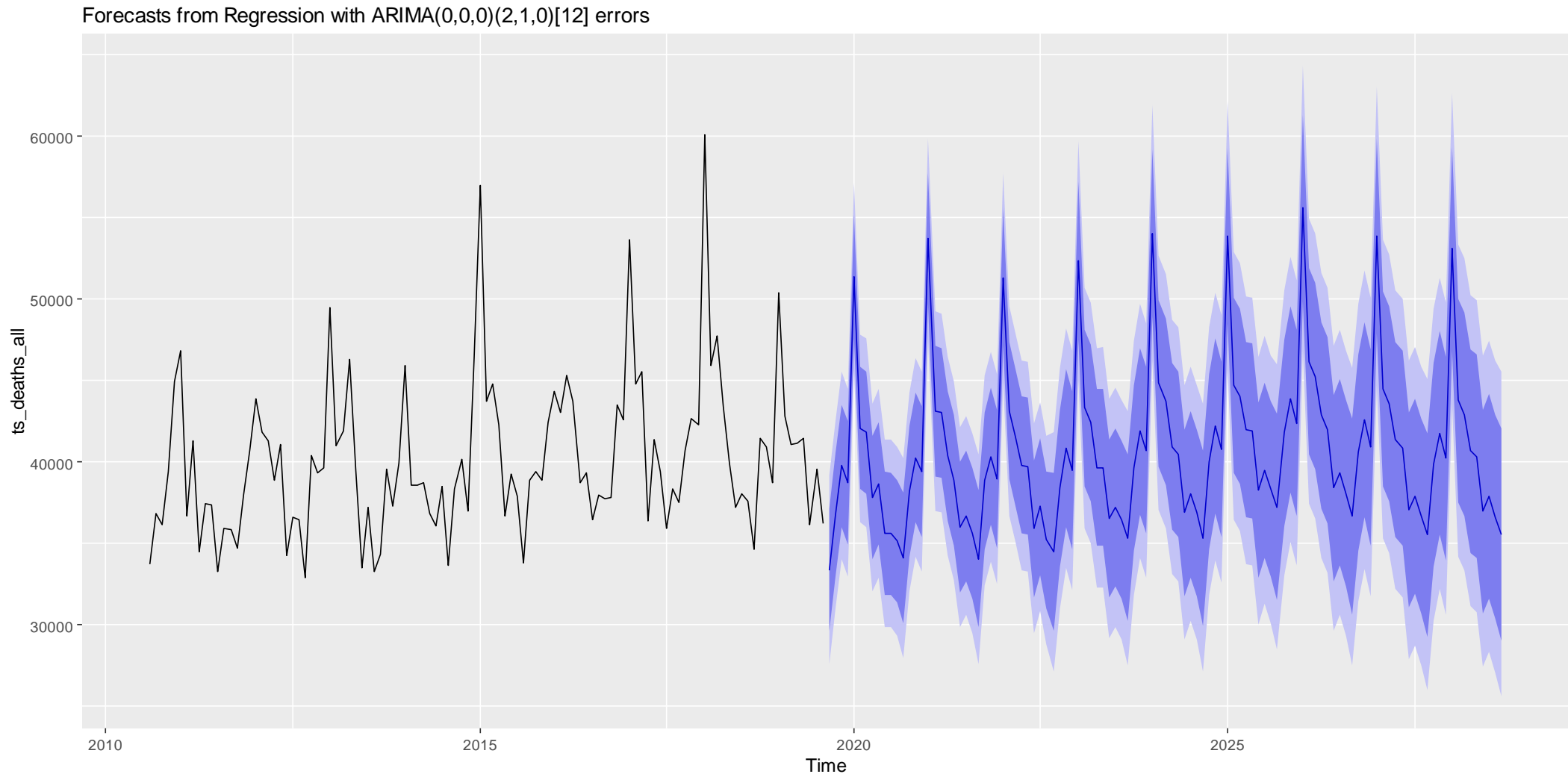
9. Extend to ARIMA-based regression between exposures and outcomes

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



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

```
279 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

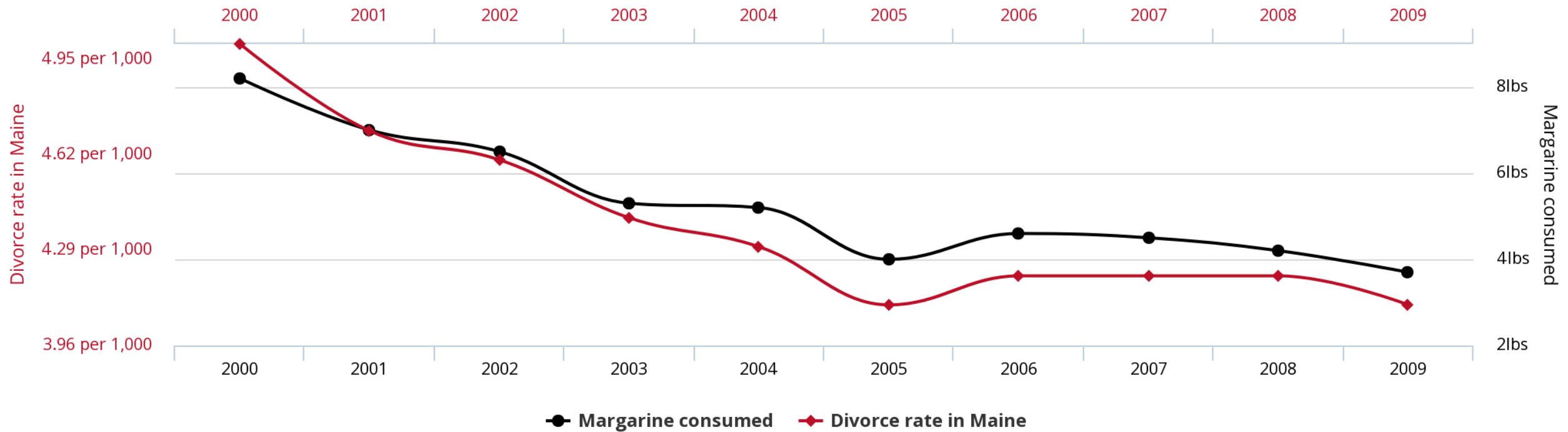
- Very limited causal strength (cf. Granger causality vs. more strict Pearl/Rubin sense of counterfactual causality)
- Cannot capture exogenous shocks or information
- Risk of model over-specification
- Data hungry, requiring constant updates
- Limited generalisability

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

Divorce rate in Maine correlates with Per capita consumption of margarine



tylervigen.com

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