Tutorial for the 'its2es' R package

Based on the paper Effect size quantification for interrupted time series analysis: Implementation in R and analysis for Covid-19 research

Load library and data

First we load the 'its2es' package which contains the Israeli all-cause mortality data. This is the same data-set analyzed in the current study.

```
library(its2es)
data <- Israel_mortality
```

1. Fit an ITS linear regression model to continuous outcomes

For illustration purposes only we show how to fit an ITS linear regression model to the continuous outcome mortality percent. For this example only and for simplicity, our first model excludes seasonal adjustment. In reality this is inadvisable as the mortality percent has a clear seasonal pattern that may greatly influence the effect size and our confidence around it. A better linear regression model, which also includes a seasonal adjustment, will be presented in Section 2.

Define formula and intervention start index for the Covid-19 period

We need to define both a formula object, and the intervention start index. The minimal formula must include the response on the left hand side of the ~ operator, and the time covariate on the right. Any additional covariates can also be passed to the right hand side of the formula, separated by + operators.

```
form <- as.formula("percent ~ time")
intervention_start_ind <- which(data$Year==2020 & data$Month>2| data$Year==2021)[1]
```

Fit an ITS linear regression model to continuous outcomes

Next, we need to call the its_lm() function to fit the ITS regression model and to quantify the effect size. Here we use a frequency of 12, corresponding to monthly data, no seasonal adjustments, and a full impact model including both a level change and a slope change following the intervention. Additionally, we set the counterfactual argument to TRUE as we are interested in plotting both the fitted values, and the model-based counterfactual values. For illustration purposes, we print the full model summary by setting the print_summary argument to TRUE. For conciseness, the complete model summary will not be printed hereafter in this tutorial, but only the effect size, together with its corresponding 95% CI and P-value.

```
fit <- its_lm(data=data,form=form,time_name = "time",intervention_start_ind=intervention_start_ind,
              freq=12,seasonality= "none", impact_model = "full",counterfactual = TRUE,print_summary=TRU
```

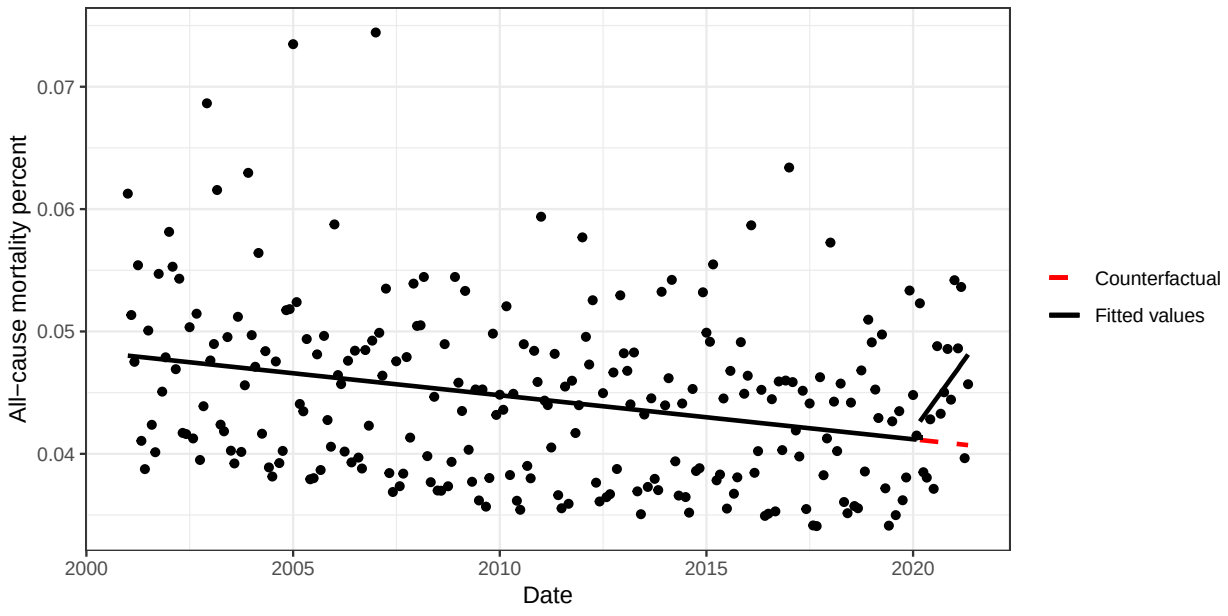
```
##
## Call:
## lm(formula = form_update, data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

```
## -0.0092220 -0.0060100 -0.0001246  0.0037317  0.0285675
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.805e-02  9.104e-04  52.784  < 2e-16 ***
## time                    -2.996e-05  6.833e-06  -4.385 1.73e-05 ***
## indicator                1.514e-03  3.503e-03   0.432    0.666
## indicator:shifted_time   4.207e-04  4.113e-04   1.023    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006881 on 241 degrees of freedom
## Multiple R-squared:  0.07775,    Adjusted R-squared:  0.06627
## F-statistic: 6.772 on 3 and 241 DF,  p-value: 0.0002108
##
## Mean difference         2.5% CI        97.5% CI         P-value
##    0.0044587340    0.0005080167    0.0084094513    0.0300000000
##  Cohen's d    2.5% CI   97.5% CI    P-value
##  3.5984185  0.3322287 37.3103663  0.2465000
```

Plot predicted values (fitted values and counterfactual values)

We use the function plot_its_lm() to plot the predicted values (fitted values and counterfactual values), together with a scatter plot of the original outcome. For the first argument, we use the updated data that includes both the fitted values and the model-based counterfactual values. We also must supply the intervention start index, the ylabel for the figure, the column name of the continuous outcome, and the column name of the date column.

```
p <- plot_its_lm(data=fit$data,intervention_start_ind=intervention_start_ind,
                 y_lab="All-cause mortality percent", response="percent", date_name= "Date")
p
```



2. Fit an ITS linear regression model with seasonal adjustments to continuous outcomes

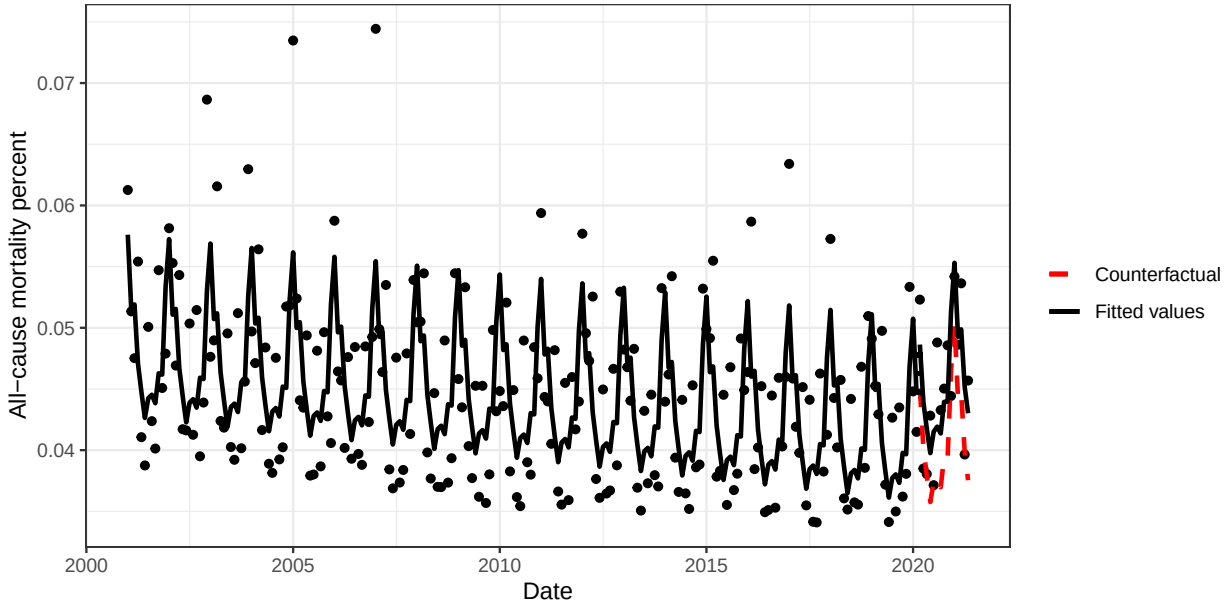
Second, we show how to fit an ITS linear regression model with seasonal adjustments. We use the same formula and intervention start index as before, except that we call the its_lm() function with seasonality= "full".

```
fit <- its_lm(data=data,form=form,time_name = "time",intervention_start_ind=intervention_start_ind,
              freq=12,seasonality= "full", impact_model = "full",counterfactual = TRUE)
```

```
## Cohen's d   2.5% CI  97.5% CI   P-value
##  1.038391  0.332192  1.715101  0.002500
```

Plot predicted values (fitted values and counterfactual values)

```
p <- plot_its_lm(data=fit$data,intervention_start_ind=intervention_start_ind,
                 y_lab="All-cause mortality percent", response="percent", date_name= "Date")
p
```



3. Fit an ITS Poisson regression model with seasonal adjustments to count outcomes

Third, we show how to fit an ITS Poisson regression model to the number of deaths. This is the same model and data used in the primary analysis of our paper.

Define formula and intervention start index for the Covid-19 period

We need to define a new formula object, with the count outcome on the left hand side of the ~ operator, and the time covariate on the right. As before, any additional covariates can also be passed to the right hand side of the formula, separated by + operators.

```
form <- as.formula("monthly_total ~ time")
```

Fit an ITS Poisson regression model to count outcomes

Next, we need to call the its_poisson() function to fit the ITS regression model and to quantify the effect size. We add the total population as our offset term (as we are interested in the mortality rate), and we set the overdispersion argument to TRUE (as the data is over-dispersed) and hence a quasi-Poisson regression model will be used. Here we use a frequency of 12, corresponding to monthly data, seasonal adjustments, and a full impact model including both a level change and a slope change following the intervention. Additionally, we set the counterfactual argument to TRUE as we are interested in plotting both the fitted values, and the model-based counterfactual values.

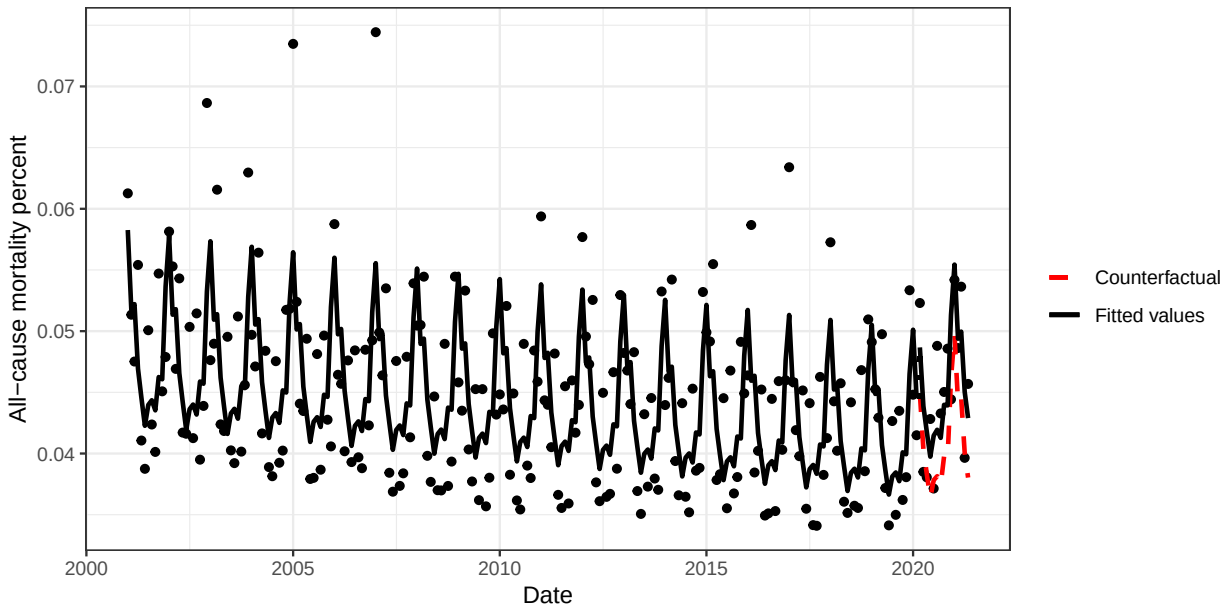
```
fit <- its_poisson(data=data,form=form, offset_name = "monthly_est", time_name = "time",
                   intervention_start_ind=intervention_start_ind, over_dispersion=TRUE,
                   freq=12,seasonality= "full", impact_model = "full",counterfactual = TRUE)
```

```
##       RR  2.5% CI 97.5% CI  P-value
## 1.105594 1.038413 1.177121 0.000000
```

Plot predicted values (fitted values and counterfactual values)

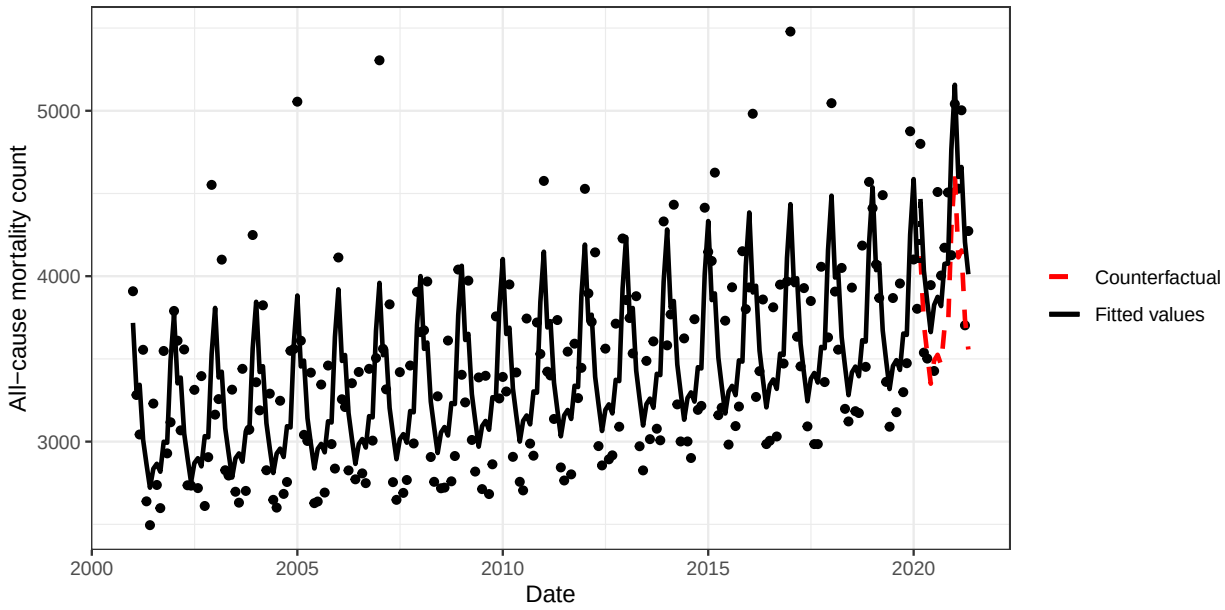
We use the function plot_its_poisson() to plot the predicted values (fitted values and counterfactual values), together with a scatter plot of the outcome. For the first argument, we use the updated data that includes both the fitted values and the model-based counterfactual values. Additionally, the function requires the intervention start index, the ylabel for the figure, the column name of the count outcome, and the column name of the date column. Note that the additional argument offset_name, which is specific to Poisson regression, can either be set to NULL, in which case the predictions and the observed outcomes are plotted on their original count scale, or set to the column name of the offset term (if it exists), in which case the predictions and the outcome will be divided by the offset and multiplied by 100.

```
p <- plot_its_poisson(data=fit$data,intervention_start_ind=intervention_start_ind,
                      y_lab="All-cause mortality percent",response="monthly_total", offset_name = "monthl
                      date_name= "Date")
p
```




```
p2 <- plot_its_poisson(data=fit$data,intervention_start_ind=intervention_start_ind,
                       y_lab="All-cause mortality count",response="monthly_total", offset_name = NULL,
                       date_name="Date")
```

```
p2
```


All−cause mortality count
5000
4000
3000
2000
2005
2010
2015
2020
Date
Counterfactual
Fitted values