Tutorial for the ‘its2es’ R package

Based on the paper Effect Size Quantification for Interrupted Time Series Analysis: Implementation in R for Covid-19 Research

Load library and data

We start by loading the ‘its2es’ package and examining the Israeli unemployment data. This is the same data-set analyzed in the paper.

```
library(its2es)
data <- unemployed
summary(data)
```

```
##       year          month          unemployed         labour       
##  Min.   :2013   Min.   : 1.000   Min.   :126535   Min.   :3647758  
##  1st Qu.:2015   1st Qu.: 3.000   1st Qu.:165443   1st Qu.:3809574  
##  Median :2017   Median : 6.000   Median :190716   Median :3962857  
##  Mean   :2017   Mean   : 6.327   Mean   :188631   Mean   :3943976  
##  3rd Qu.:2019   3rd Qu.: 9.000   3rd Qu.:210913   3rd Qu.:4076160  
##  Max.   :2021   Max.   :12.000   Max.   :249666   Max.   :4152842  
##     percent            dt                  time    
##  Min.   :3.081   Min.   :2013-01-01   Min.   :  1  
##  1st Qu.:4.126   1st Qu.:2015-02-01   1st Qu.: 26  
##  Median :4.724   Median :2017-03-01   Median : 51  
##  Mean   :4.812   Mean   :2017-03-01   Mean   : 51  
##  3rd Qu.:5.463   3rd Qu.:2019-04-01   3rd Qu.: 76  
##  Max.   :6.735   Max.   :2021-05-01   Max.   :101  
```

1. Fit an ITS linear regression model to continuous outcomes

First we show how to fit an ITS linear regression model to the continuous outcome unemployment percent.

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.

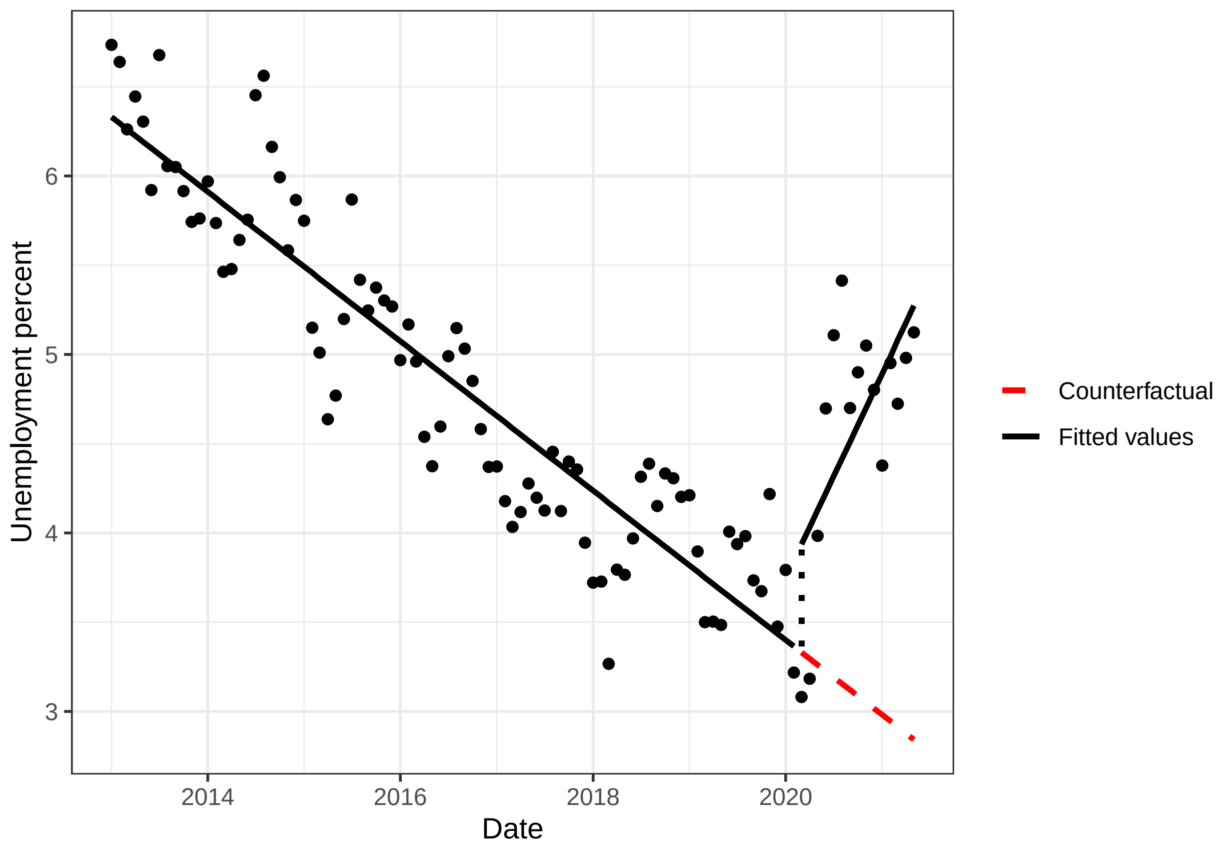
```r
fit <- its_lm(data=data,form=form,time_name = "time",intervention_start_ind=intervention_start_ind,
              freq=12,seasonality= "none", impact_model = "full",counterfactual = TRUE)
```

```
##
## Call:
## lm(formula = form_update, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90067 -0.28348  0.00614  0.28684  0.99957
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.365092   0.084581  75.254  < 2e-16 ***
## time                   -0.034883   0.001689 -20.656  < 2e-16 ***
## indicator               0.606417   0.208999   2.902   0.0046 **
## indicator:shifted_time  0.130377   0.023295   5.597 2.03e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3888 on 97 degrees of freedom
## Multiple R-squared:  0.8222, Adjusted R-squared:  0.8167
## F-statistic: 149.5 on 3 and 97 DF,  p-value: < 2.2e-16
##
## Mean difference         2.5% CI        97.5% CI         P-value
##        1.519055        1.248129        1.789981        0.000000
## Cohen's d   2.5% CI  97.5% CI   P-value
##  4.724988  3.146811  8.176221  0.000000
```

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.

```r
p <- plot_its_lm(data=fit$data,intervention_start_ind=intervention_start_ind,
                 y_lab="Unemployment percent", response="percent", date_name= "dt")
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. The only difference is that this time we call the its_lm() function with seasonality= "full".

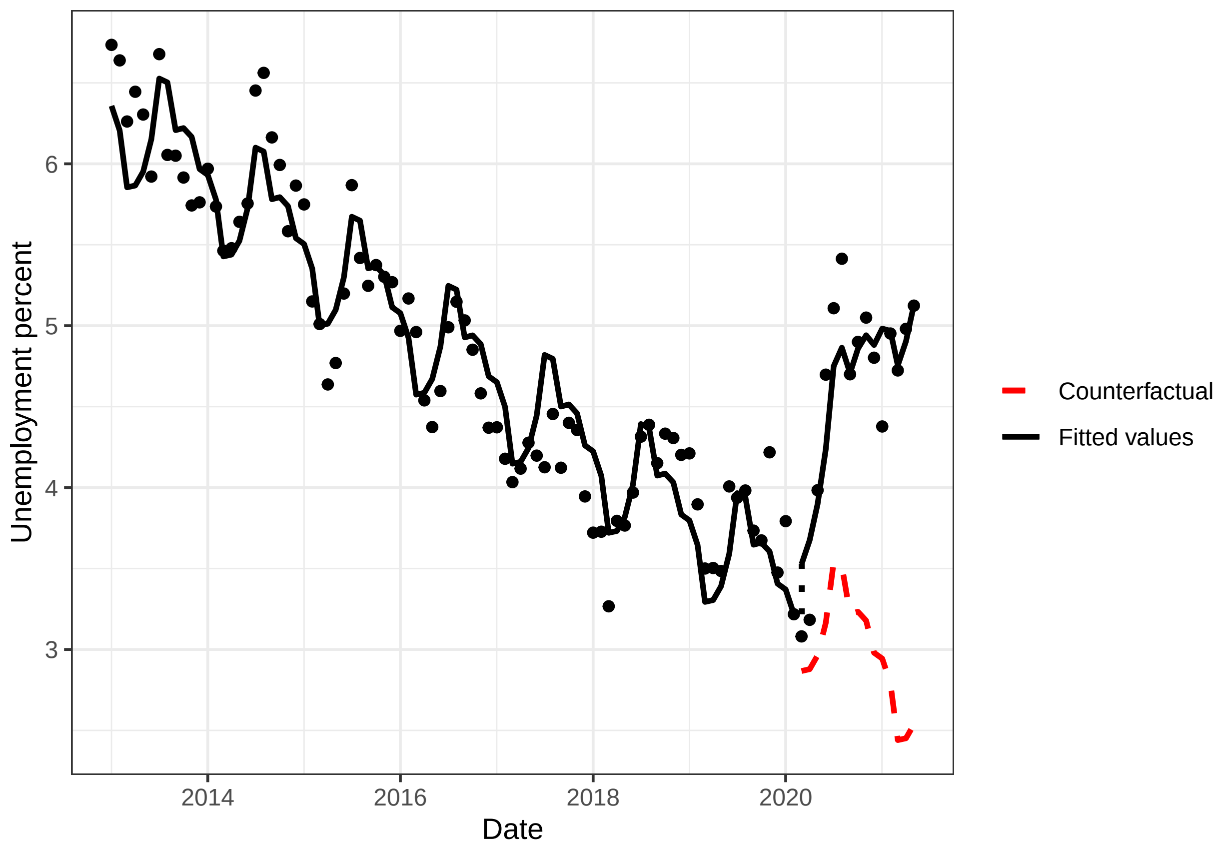
```
fit <- its_lm(data=data,form=form,time_name = "time",intervention_start_ind=intervention_start_ind,
              freq=12,seasonality= "full", impact_model = "full",counterfactual = TRUE)
```

```
##
## Call:
## lm(formula = form_update_full, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69381 -0.20561 -0.00052  0.19816  0.61270
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.3962162  0.0664519  96.253  < 2e-16 ***
## time                   -0.0355664  0.0013279 -26.784  < 2e-16 ***
## indicator               0.6593104  0.1660928   3.970 0.000149 ***
## S1.12                  -0.3176263  0.0429572  -7.394 8.75e-11 ***
## C1.12                   0.0162128  0.0435278   0.372 0.710459
```

```
## S2.12                    0.1389014  0.0427749   3.247 0.001662 **
## C2.12                    0.0623942  0.0431377   1.446 0.151701
## S3.12                   -0.0405147  0.0427737  -0.947 0.346198
## C3.12                   -0.0048997  0.0431291  -0.114 0.909815
## S4.12                    0.0004498  0.0427164   0.011 0.991623
## C4.12                   -0.0838035  0.0431786  -1.941 0.055551 .
## S5.12                   -0.0062410  0.0427090  -0.146 0.884163
## C5.12                    0.0029662  0.0431452   0.069 0.945349
## C6.12                    0.0053295  0.0303420   0.176 0.860982
## indicator:shifted_time  0.1379238  0.0184930   7.458 6.51e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3043 on 86 degrees of freedom
## Multiple R-squared:  0.9034, Adjusted R-squared:  0.8877
## F-statistic: 57.47 on 14 and 86 DF,  p-value: < 2.2e-16
##
## Mean difference         2.5% CI        97.5% CI         P-value
##        1.624777        1.410380        1.839174        0.000000
## Cohen's d   2.5% CI  97.5% CI   P-value
##  3.737341  2.875479  4.578553  0.000000
```

Plot predicted values (fitted values and counterfactual values)

```
p <- plot_its_lm(data=fit$data,intervention_start_ind=intervention_start_ind,
                 y_lab="Unemployment percent", response="percent", date_name= "dt")
p
```



3. Fit an ITS Poisson regression model to count outcomes

Third we show how to fit an ITS Poisson regression model to the number of unemployed.

Define formula

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("unemployed ~ 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 labour force as our offset term (as we are interested in the unemployment 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. As before, 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.

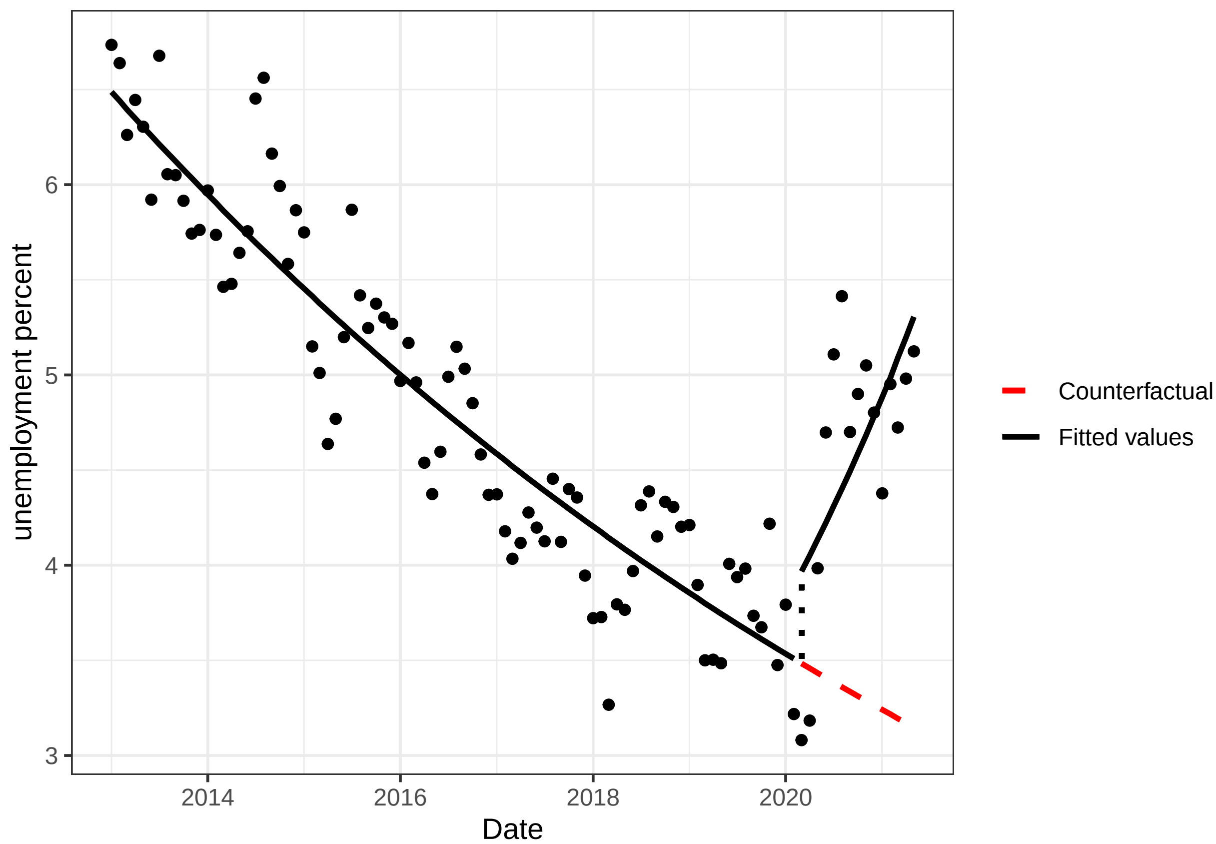
```
fit <- its_poisson(data=data,form=form,offset_name = "labour", time_name = "time",
                   intervention_start_ind=intervention_start_ind, over_dispersion=TRUE,
                   freq=12,seasonality= "none", impact_model = "full",counterfactual = TRUE)
```

```
##
## Call:
## glm(formula = form_update, family = quasipoisson, data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -93.92  -25.61    0.15   24.45   94.58
##
## Coefficients:
##                         Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           -2.7280744  0.0166921 -163.435  < 2e-16 ***
## time                  -0.0072301  0.0003551  -20.360  < 2e-16 ***
## indicator              0.1300257  0.0457173    2.844  0.00543 **
## indicator:shifted_time  0.0279903  0.0048861    5.729 1.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1243.289)
##
##     Null deviance: 667331  on 100  degrees of freedom
## Residual deviance: 121020  on  97  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
##
##       RR  2.5% CI 97.5% CI  P-value
## 1.385357 1.306718 1.468728 0.000000
```

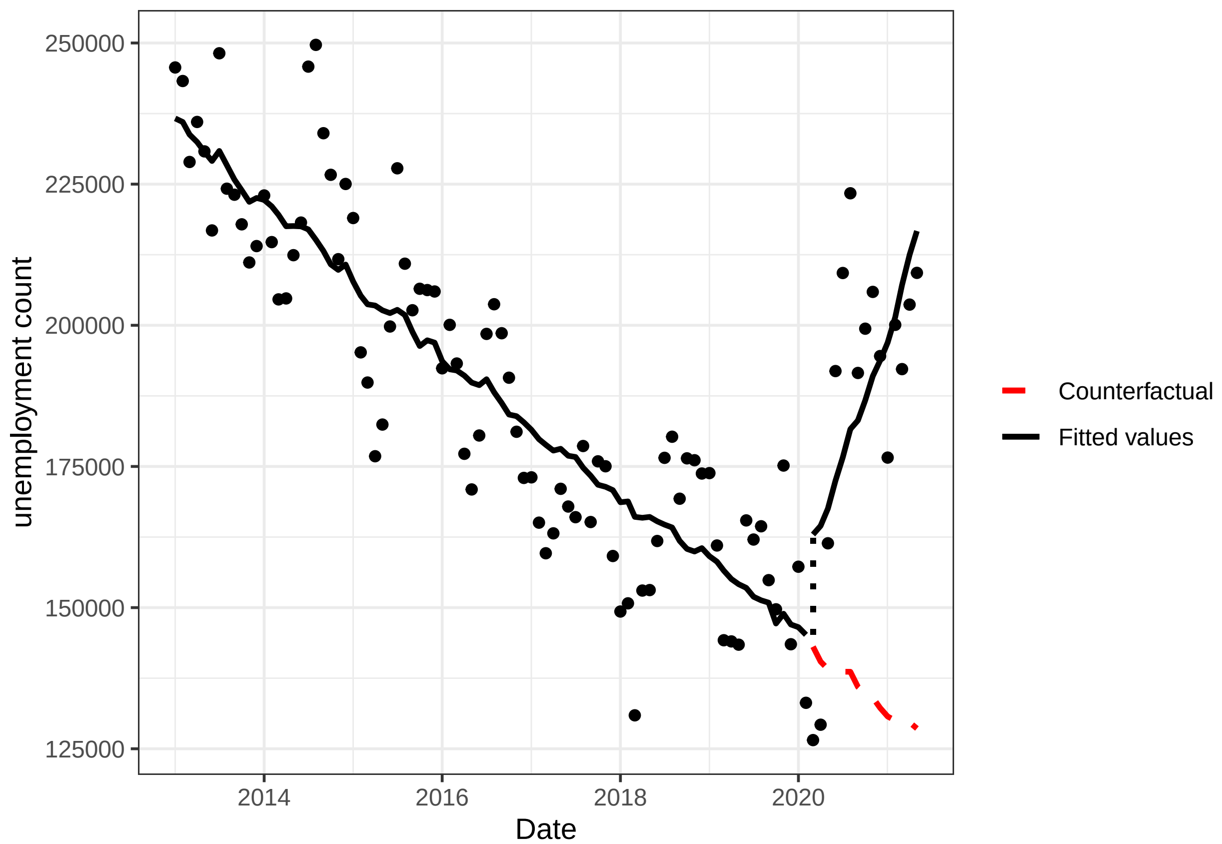
Plot predicted values (fitted values and counterfactual values)

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. We also must supply 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 original outcomes are plotted on their original count scale, or set to the column name of the offset term (if exists), in which case the predictions and the outcome will be divided by the offset and multiplied by 100.

```r
p <- plot_its_poisson(data=fit$data,intervention_start_ind=intervention_start_ind,
                      y_lab="unemployment percent",response="unemployed", offset_name = "labour",
                      date_name= "dt")
p
```




```
p2 <- plot_its_poisson(data=fit$data,intervention_start_ind=intervention_start_ind,
                       y_lab="unemployment count",response="unemployed", offset_name = NULL,
                       date_name="dt")
p2
```



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

Fourth we show how to fit an ITS Poisson regression model with seasonal adjustments. We use the same formula and intervention start index as in 3. The only difference is that this time we call the its_poisson() function with seasonality= “full”.

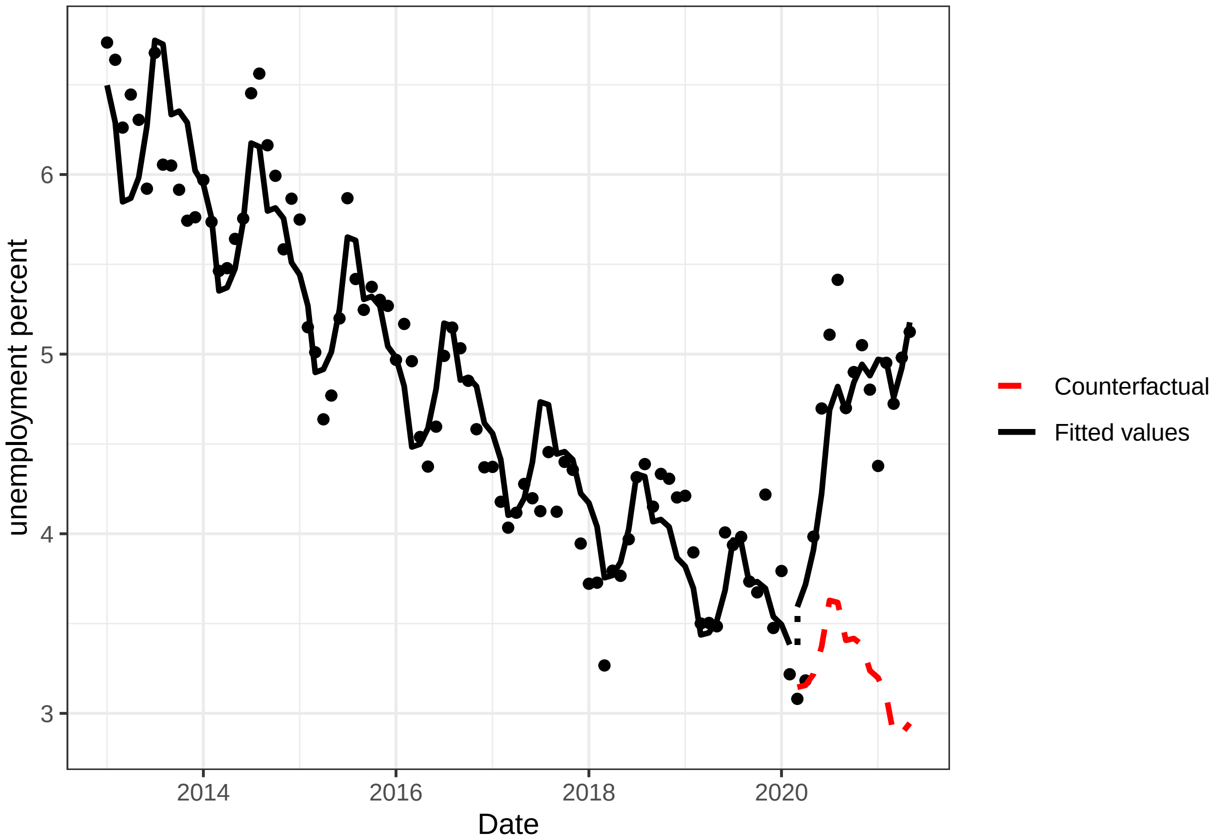
```
fit <- its_poisson(data=data,form=form,offset_name = "labour", time_name = "time",
                   intervention_start_ind=intervention_start_ind, over_dispersion=TRUE,
                   freq=12,seasonality= "full", impact_model = "full",counterfactual = TRUE)
```

```
##
## Call:
## glm(formula = form_update_full, family = quasipoisson, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -57.336  -17.981   -0.681   15.884   54.103
##
## Coefficients:
##                          Estimate Std. Error  t value Pr(>|t|)
## (Intercept)            -2.723e+00  1.286e-02 -211.768  < 2e-16 ***
## time                   -7.383e-03  2.739e-04  -26.953  < 2e-16 ***
## indicator               1.330e-01  3.597e-02    3.699 0.000381 ***
## S1.12                  -6.925e-02  8.835e-03   -7.839 1.12e-11 ***
```

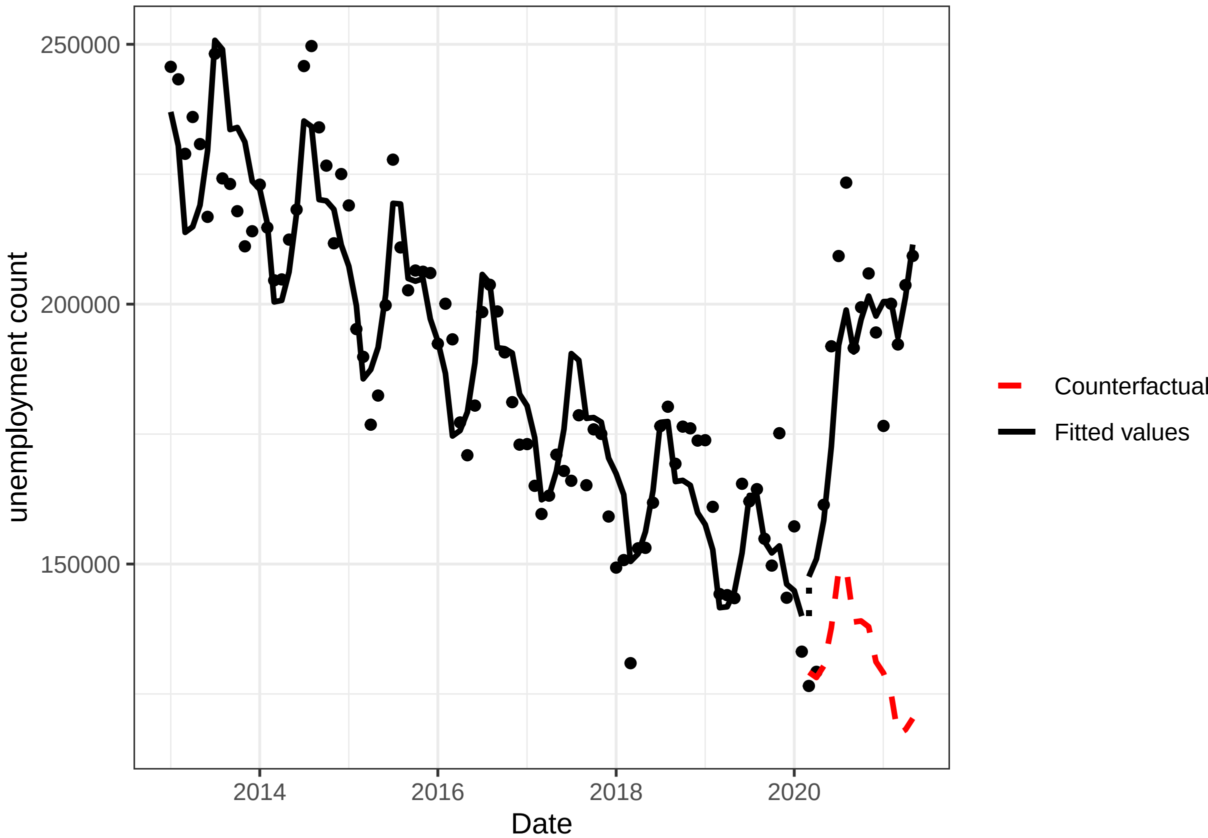
```
## C1.12                  3.422e-03  8.856e-03    0.386 0.700176
## S2.12                  2.750e-02  8.707e-03    3.158 0.002192 **
## C2.12                  1.427e-02  8.865e-03    1.610 0.111018
## S3.12                 -8.679e-03  8.742e-03   -0.993 0.323564
## C3.12                 -1.419e-03  8.827e-03   -0.161 0.872618
## S4.12                 -4.685e-04  8.691e-03   -0.054 0.957134
## C4.12                 -1.644e-02  8.875e-03   -1.852 0.067470 .
## S5.12                 -1.475e-03  8.755e-03   -0.168 0.866592
## C5.12                 -6.092e-05  8.802e-03   -0.007 0.994494
## C6.12                  1.518e-03  6.205e-03    0.245 0.807266
## indicator:shifted_time  3.079e-02  3.892e-03    7.912 7.98e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 731.5963)
##
##     Null deviance: 667331  on 100  degrees of freedom
## Residual deviance:  63181  on  86  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 3
##
##       RR  2.5% CI 97.5% CI  P-value
## 1.417029 1.354297 1.482667 0.000000
```

Plot predicted values (fitted values and counterfactual values)

```r
p <- plot_its_poisson(data=fit$data,intervention_start_ind=intervention_start_ind,
                      y_lab="unemployment percent",response="unemployed", offset_name = "labour",
                      date_name= "dt")
p
```




```r
p2 <- plot_its_poisson(data=fit$data,intervention_start_ind=intervention_start_ind,
                       y_lab="unemployment count",response="unemployed", offset_name = NULL,
                       date_name="dt")
p2
```



250000
200000
150000
unemployment count
2014
2016
2018
2020
Date
Counterfactual
Fitted values