# Causal Impact Helper Example

## Dataset Case

This example goes through the `registration_data` dataset included in the `CausalImpactHelper` package. The dataset is made of daily sampled observations from a mobile platform, with two series: `New_Users` and `Registered_Users`. There was a product intervention at the 81st observation aimed at increasing the rate of registered users on the platform.

```
head(registration_data)
```

```
##   Observation_Number New_Users Registered_Users
## 1                  1      5031             3018
## 2                  2      5596             3552
## 3                  3      5013             3081
## 4                  4      5813             3677
## 5                  5      5444             3441
## 6                  6      4952             3039
```

## Causal Impact Analysis: Full Period

### A/A Test

The first step of validating the approach is to run an A/A test. Using the pre-period data we should be able to run a CausalImpact (CI) analysis and not detect any changes. If changes are detected, it is indicative that the relationships between the treatment and control series are not stable during the pre-period, in which case they would be unsuitable to generate post-period predictions.

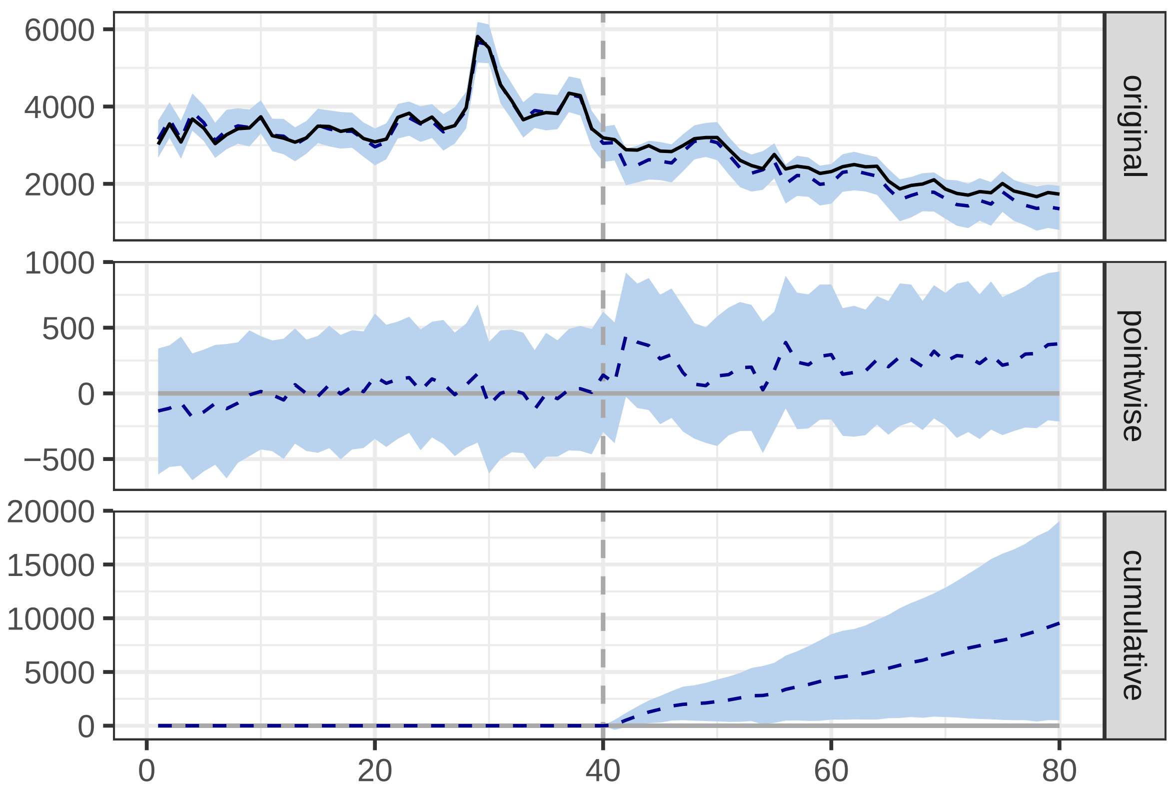
In this example we see that with the more likely `prior.level.sd` value of `0.01` (as opposed to `0.1`), a significant change is detected in the pre-period.

```
causalimpact_data = registration_data %>%
  dplyr::select(Registered_Users, New_Users)

pre_period = c(1, 80)
post_period = c(81, 105)
aa_args = list(niter = 1000, nseasons = 7, prior.level.sd = 0.01)

ci_model = RunAATest(causalimpact_data, pre_period, aa_args)
```

```
## [1] "Pre Period 1:40"
## [1] "Post Period 41:80"
## Posterior inference {CausalImpact}
##
```

```
##                            Average        Cumulative
## Actual                     2364           94580
## Prediction (s.d.)          2126 (116)     85036 (4635)
## 95% CI                     [1888, 2352]   [75537, 94082]
##
## Absolute effect (s.d.)     239 (116)      9544 (4635)
## 95% CI                     [12, 476]      [498, 19043]
##
## Relative effect (s.d.)     12% (6.2%)     12% (6.2%)
## 95% CI                     [0.53%, 25%]   [0.53%, 25%]
##
## Posterior tail-area probability p:   0.01913
## Posterior prob. of a causal effect:  98.087%
##
## For more details, type: summary(impact, "report")
##
## NULL
```

```
plot(ci_model$aa_causalimpact_model)
```




### Time Series Inspection

Assessing the time series being used can help us determine if or where problems might be occurring.

Several assessments are made through the `RunCointegrationTest` method:

- Stationarity of the treatment series in the pre-period. If the treatment series is stationary, we should expect CI to be a suitable methodology (this is not a requirement though).

- Stationarity of the control series in the pre-period. If the control series are stationary, we should expect CI to be a suitable methodology (this is not a requirement though).
- Cointegration of the treatment and control series in the pre-period. Residual stationarity indicates a legitimate cointegration relationship between the noted control series and the treatment. This indicates that the control series should be a suitable choice for CI. Non-stationary residuals suggests that there is an unstable relationship with the treatment during the pre-period, in which case the control may not be suitable or CI might not be feasible (if there aren't other suitable control series which could be used instead).

In this case, we see the control and treatment series are both non-stationary, which is just informative. However, the cointegration between the two is also non-stationary, indicating that the pre-period data isn't suitable to the post-period forecasting. This explains the false intervention detected in the A/A test.

```
cointegration_test = RunCointegrationTest(causalimpact_data, pre_period, "Registered_Users", run_statio
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

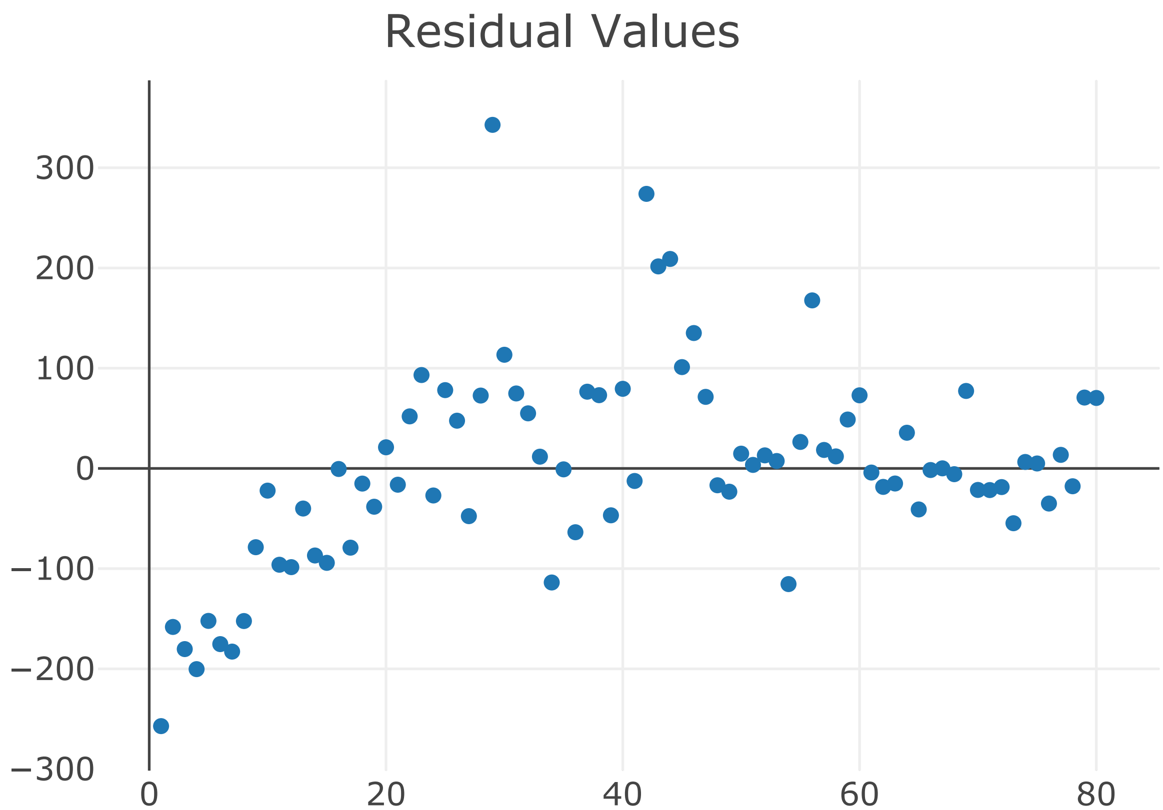
```
kable(cointegration_test$test_results)
```

| Series | Test | Result | Confidence.Level |
|---|---|---|---|
| Registered_Users | ADF - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| Registered_Users | ERS - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| Registered_Users | KPSS - Mean | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 1pct |
| Registered_Users | ERS - Trend | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| Registered_Users | KPSS - Trend | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 1pct |
| New_Users | ADF - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| New_Users | ERS - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| New_Users | KPSS - Mean | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 1pct |
| New_Users | ERS - Trend | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| New_Users | KPSS - Trend | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 1pct |
| New_Users_cointegration | ADF - Mean | Reject Non-Stationary (Potential Stationarity) | 1pct |
| New_Users_cointegration | ERS - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| New_Users_cointegration | KPSS - Mean | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 5pct |
| New_Users_cointegration | ERS - Trend | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| New_Users_cointegration | KPSS - Trend | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 1pct |
| New_Users_cointegration | BG - AutoCorrelation | Reject Non-Autocorrelated (Autocorrelation, Possible Non-Stationarity) | 1pct |

Visually inspecting the cointegration residuals may provide some insight into the dynamics occurring (though there is no guarantee of this).

In this case, we can see a structural change occur around the 20th observation, where there was a trend in previous residuals, but residuals after the 20th point look like they could be stationary. Notably, this can also be seen in the CausalImpact plots above.

```
subplot(cointegration_test$cointegration_residual_graphs)
```




## Causal Impact Analysis: Reduced Period

### A/A Tests

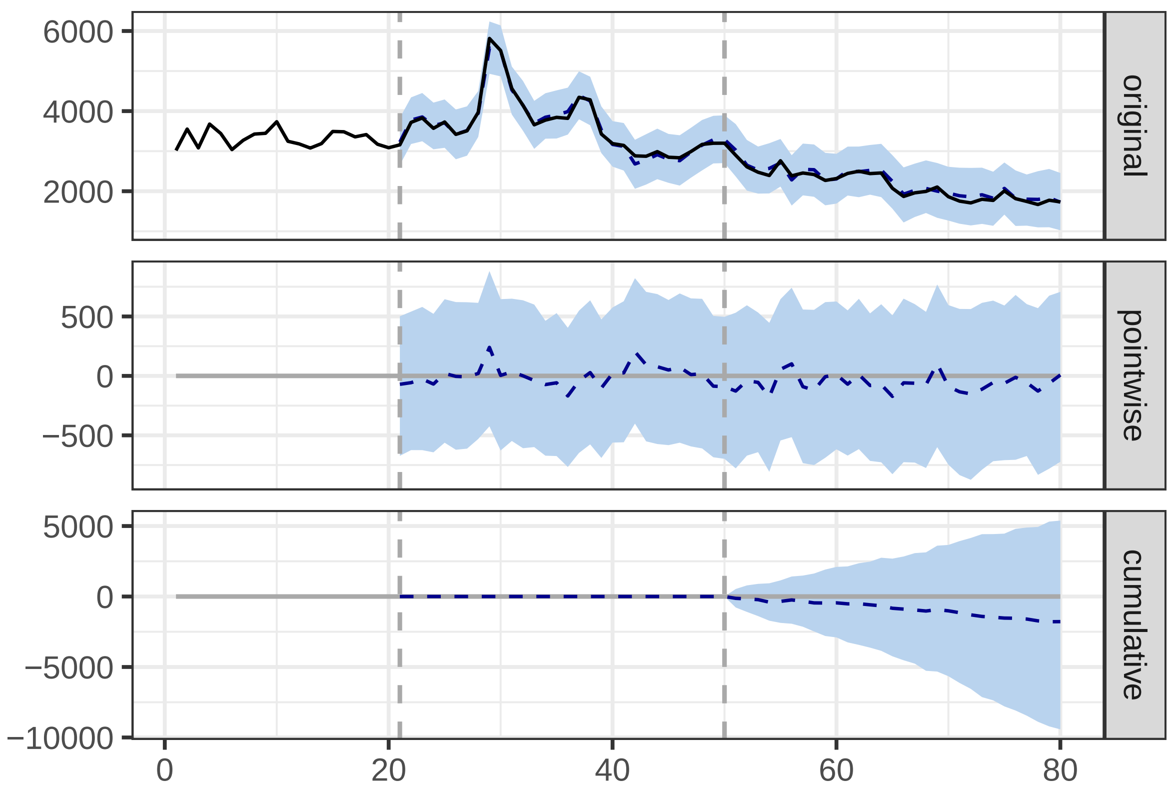
Using the information above, we can remove some of the pre-period data from our analysis. Truncating the beginning of the pre-period and repeating the A/A test, we get a better result, with a lower indication of any intervention. The 95% CI bounds provide an rough estimate of the effect size we might need to have from the intervention in order to have a conclusive detection.

```
pre_period = c(21, 80)
post_period = c(81, 105)
aa_args = list(niter = 1000, nseasons = 7, prior.level.sd = 0.01)

ci_model = RunAATest(causalimpact_data, pre_period, aa_args)
```

```
## [1] "Pre Period 21:50"
## [1] "Post Period 51:80"
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   2148           64452
## Prediction (s.d.)        2208 (130)     66233 (3906)
## 95% CI                   [1969, 2462]   [59065, 73874]
```

```
##
## Absolute effect (s.d.)   -59 (130)      -1781 (3906)
## 95% CI                   [-314, 180]    [-9422, 5387]
##
## Relative effect (s.d.)   -2.3% (5.8%)   -2.3% (5.8%)
## 95% CI                   [-13%, 9.1%]   [-13%, 9.1%]
##
## Posterior tail-area probability p:   0.32931
## Posterior prob. of a causal effect:  67%
##
## For more details, type: summary(impact, "report")
##
## NULL
```

```
plot(ci_model$aa_causalimpact_model)
```




We can inspect the cointegration series again to further assess if we have made a constructive adjustment. Here we see that there may still be some autocorrelation present in the cointegration residuals, however this is not necessarily indicative of non-stationarity (there may be constant autocorrelation due to trends or seasonality which CI will account for). All other results indicate stationarity, which is sufficient to continue with the analysis.

```
cointegration_test = RunCointegrationTest(causalimpact_data, pre_period, "Registered_Users", run_statio
kable(cointegration_test$test_results)
```

| Series | Test | Result | Confidence.Level |
|---|---|---|---|
| Registered_Users | ADF - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| Registered_Users | ERS - Mean | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| Registered_Users | KPSS - Mean | Reject Stationarity, Accept Unit Root (Non-Stationarity) | 1pct |
| Registered_Users | ERS - Trend | Fail to Reject Non-Stationary (Non-Stationary) | NA |
| Registered_Users | KPSS - Trend | Fail to Reject Stationary (Stationary) | NA |
| New_Users_cointegration | ADF - Mean | Reject Non-Stationary (Potential Stationarity) | 1pct |
| New_Users_cointegration | ERS - Mean | Reject Non-Stationary (Potential Stationarity) | 1pct |
| New_Users_cointegration | KPSS - Mean | Fail to Reject Stationary (Stationary) | NA |
| New_Users_cointegration | ERS - Trend | Reject Non-Stationary (Potential Stationarity) | 5pct |
| New_Users_cointegration | KPSS - Trend | Fail to Reject Stationary (Stationary) | NA |
| New_Users_cointegration | BG - AutoCorrelation | Reject Non-Autocorrelated (Autocorrelation, Possible Non-Stationarity) | 5pct |

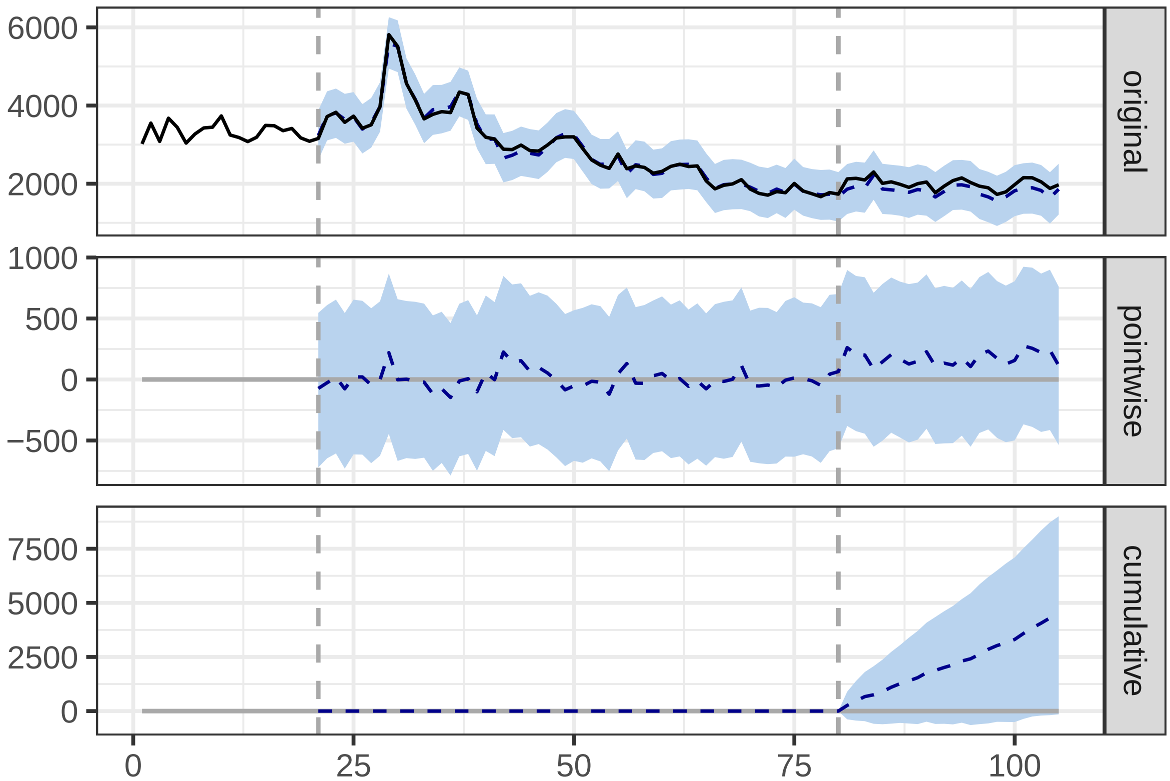
### Actual Pre Post Assessment

Here we run the actual CI analysis, with our reduced pre_period, and more MCMC iterations (indicated by `niter`) so as to help narrow the credible interval if possible.

```
pre_period = c(21, 80)
post_period = c(81, 105)

impact = CausalImpact(as.matrix(causalimpact_data), pre_period, post_period, model.args = list(niter =
summary(impact)
```

```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   2006           50160
## Prediction (s.d.)        1830 (93)      45744 (2334)
## 95% CI                   [1646, 2012]   [41160, 50310]
##
## Absolute effect (s.d.)   177 (93)       4416 (2334)
## 95% CI                   [-6, 360]      [-150, 9000]
##
## Relative effect (s.d.)   9.9% (5.6%)    9.9% (5.6%)
## 95% CI                   [-0.3%, 22%]   [-0.3%, 22%]
##
## Posterior tail-area probability p:   0.02922
## Posterior prob. of a causal effect:  97.078%
##
## For more details, type: summary(impact, "report")
```

```
plot(impact)
```



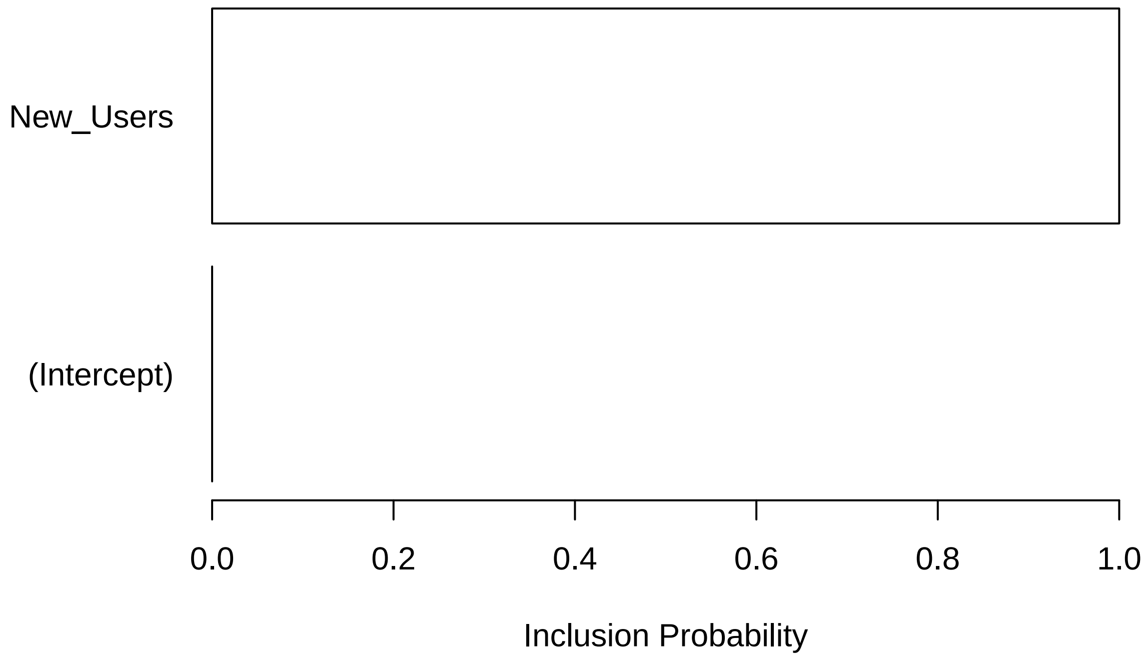

The analysis yields strong results, with a 97% probability of a causal effect from the intervention. The credible interval for the intervention is [-0.38%, 20%], with a 9.7% midpoint (the 95% CI indicates that this interval is generated from the central 95% of prediction iterations).

## Further Analysis

While the above is sufficient for reporting, we may wish to dig a little deeper into our CI results.

The first graph shows the inclusion likelihood of our control series in one of the final posterior models. In this case we only have 1 control series with 100% probability, though in a more sophisticated example this may provide an indication of how much value the different control series are providing.

```
parameter_summary = GenerateParameterSummary(impact)
```




The statistics below offer some further insight regarding the final models:

- Regression Coefficients: similarly to above, this may offer some insight into the impact of different series (assuming there is no multicollinearity).
- Noise Observation Standard Deviation: This provides some insight into the degree of unexplained noise in the posterior models.
- Local Level Standard Deviation: Should converge to the `prior.level.sd` parameter.

```
kable(parameter_summary$statistics_dataframe)
```

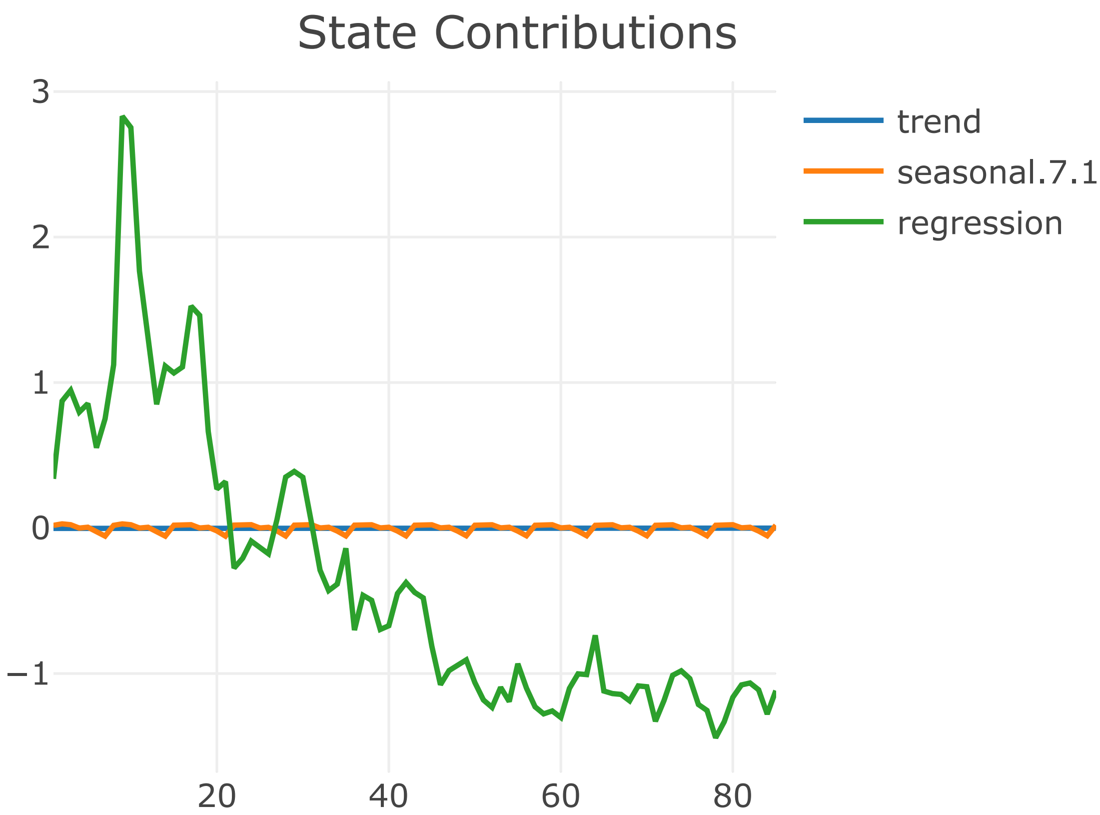
| Statistic | Values |
|---|---|
| Mean Value of Regression Coefficients* (Intercept), New_Users | 0, 0.9954 |
| Mean Value of Noise Observation Std | 0.319 |
| Mean Value of Level Std (should approx. prior.level.sd) | 0.0102 |

* Multicollinearity can lead to unstable coefficients and invalidate causative interpretations.

The state contributions graph shows a summary of which components in the BSTS models contributed to the final states. This provides some insight into how much of the posterior predictions were based on trends, seasonality and the control regression series.

In this case, it's clear the model is primarily based off of the control series, with some additional input provided by seasonality.

```
(parameter_summary$state_contributions_plot)
```


State Contributions
3
2
1
0
−1
20
40
60
80
trend
seasonal.7.1
regression