When A/B Experimentation is not possible.

Alternative Methodologies for causality estimation.

Jennifer Vlasiu | 29 June 2022

WOMEN IN DATA SCIENCE
@STANFORD UNIVERSITY

Welcome:

widsconference.org/jennifer-vlasiu.html

HOME ABOUT CONFERENCE DATATHON PODCAST EDUCATION

Jennifer Vlasiu

Data Science & Big Data Instructor
York University

Jennifer is a Data Scientist & Advance Analytics Strategy professional, currently focused on decision intelligence, quasi-experimentation, and model/feature/cloud solution deployments. Jennifer has over 10 years of experience in a data science, analytics and operations capacity. Her passion for business, and emerging technologies has been refined over the years via strategic and data science roles at leading global organizations. Her past engagements include Bell, ABInBev, Uber, PepsiCo, JP Morgan and the U.S. Department of Commerce.

Jennifer is passionate about digital literacy, serving as a Chapter Lead for a national non-profit, Canada Learning Code, and as a course developer and instructor for numerous colleges and universities.

When not immersed in an IDE, or exploring the latest python package, Jennifer can be found sailing or going for a run along a nearby waterfront. She can be reached at jennvlasiu.com

Session Objectives:

- Introduction to the concept of A/B Experimentation, and why it matters
- Challenges and situations when A/B Experimentation may not be possible
- Basic understanding of what is meant by estimating a causal effect
- Introduction to a few alternative Methodologies for causality estimation
- Demo: Walkthrough Jupyter and R on how to implement causal impact

Session Objectives:

- Introduction to the concept of A/B Experimentation, and why it matters
- Challenges and situations when A/B Experimentation may not be possible
- Basic understanding of what is meant by estimating a causal effect
- Introduction to a few alternative Methodologies for causality estimation
- Demo: Walkthrough Jupyter and R on how to implement causal impact

What is A/B Experimentation

A/B Experimentation is the gold standard for estimating causal effects is a truly randomized experiment, evenly distributed treatment and control cohorts.

A/B Experimentation is where each participant is randomly assigned a group.

- **Treatment** – all the participants in the group that **do** receive an intervention

- **Control** – all the participants in the group that **do NOT** receive an intervention

Session Objectives:

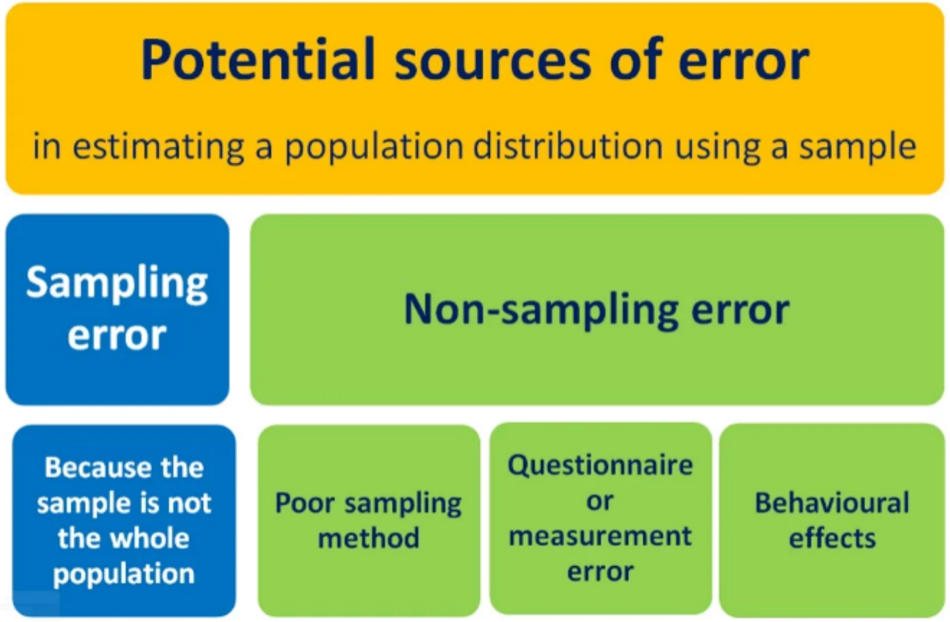
- Introduction to the concept of A/B Experimentation, and why it matters
- Challenges and situations when A/B Experimentation may not be possible
- Basic understanding of what is meant by estimating a causal effect
- Introduction to a few alternative Methodologies for causality estimation
- Demo: Walkthrough Jupyter and R on how to implement causal impact

When might the “Gold Standard” A/B Experimentation not be possible?

A/B Experimentation is the gold standard for estimating causal effects is a truly randomized experiment, evenly distributed treatment and control cohorts.

- Too Expensive
- Too Difficult
- Unethical
- Assembly/Instrumentation not done

In these situations we want to obtain a causal effect, in the absence of experimentation



Source: https://learnandteachstatistics.wordpress.com/2014/09/04/sampling-and-non-sampling-error/

The Challenge of Inference in the Social Sciences (slideshare.net)

Session Objectives:

- Introduction to the concept of A/B Experimentation, and why it matters
- Challenges and situations when A/B Experimentation may not be possible
- Basic understanding of what is meant by estimating a causal effect
- Introduction to a few alternative Methodologies for causality estimation
- Demo: Walkthrough Jupyter and R on how to implement causal impact

What is Causal Inference?

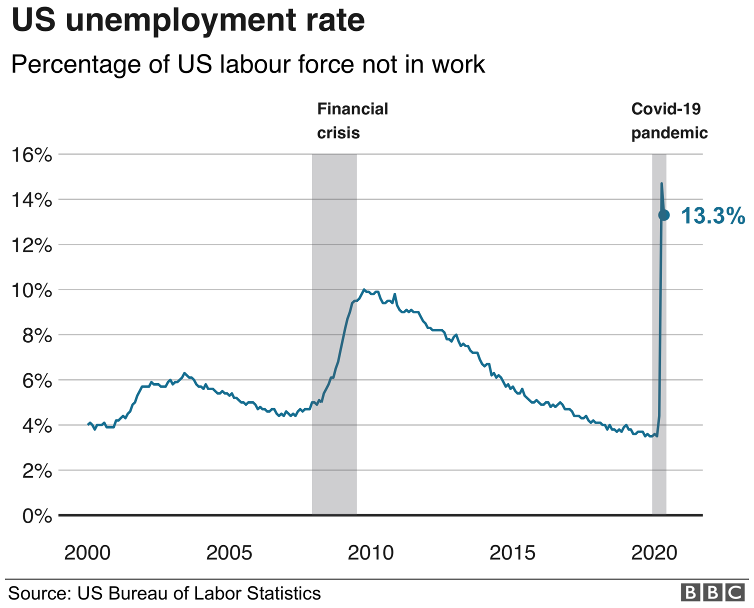
Causal inference is the branch of statistics that is concerned about effects; the consequences of our actions.

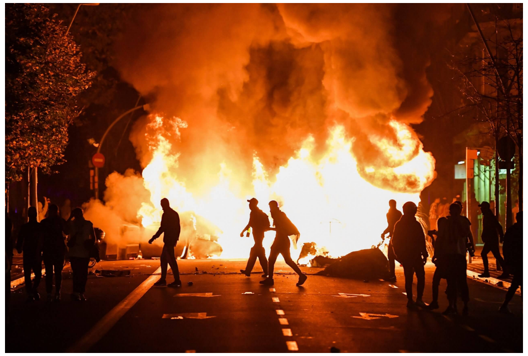
Identifying one causal law can often be **more important** than identifying multiple correlation patterns that you might find

Understanding causal effects is the **bedrock** of big data and **of modern data science.**

What is Causal Inference?

Causal inference seeks to understand the effect of some variable(s) on some other variables(s)





Example: The causal effect of **unemployment** on the **probability that a riot** will occur

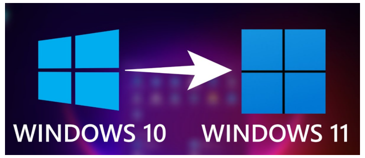
Example: The causal effect of **a riot** on **government spending**

US unemployment sees surprise improvement in May - BBC News
State Unemployment Map, January 2016 (businessinsider.com)

What is “Encouragement” Design?

Encouragement Design is a term used when we are **unable to force** all experimental subjects to take the (randomly) assigned treatment or control groups. It includes:

1. Can’t force users in treatment control to take the treatment



Example: **Software Upgrades**. One can only encourage the users to take the treatment by making “Download Upgrade” as an option

Example: **Marketing Campaign via Notification**. User may have turned off notifications, hence they won’t see the campaign

2. Can’t force users in the Control group to NOT take the treatment

Example: **Software Upgrades**. User could download the upgrade directly from the internet even when “Download Upgrade” option is not available

turn off notifications chrome - Bing images

Session Objectives:

- Introduction to the concept of A/B Experimentation, and why it matters
- Challenges and situations when A/B Experimentation may not be possible
- Basic understanding of what is meant by estimating a causal effect
- Introduction to a few alternative Methodologies for causality estimation
- Demo: Walkthrough Jupyter and R on how to implement causal impact

Alternative Methodologies for Causality Estimation

As with all non-experimental approaches to causal inference, valid conclusions require strong assumptions. Try at least two methods, with different assumptions to give confidence

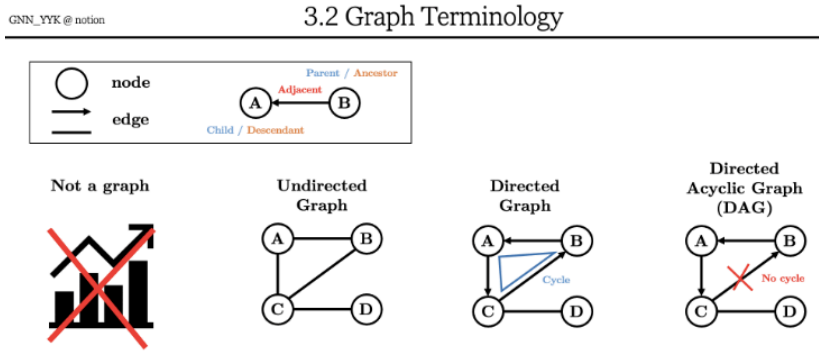
	Google Causal Impact Bayesian Time Series	Uber	Microsoft EconML	Microsoft Microsoft "Do Why" Package	IBM Causal Library
About	• **Used to estimate the causal effect of a designed intervention** on a time series (i.e. daily clicks generated from ad campaign	• Uplift modeling and causal inference methods using machine learning algorithms based on recent research • Estimates causal impact of intervention T on outcome Y for users with observed feature X • Treatment effects on observational and experimental data	• Estimating **heterogeneous treatment effects from observational and experimental data** via machine learning • Machine Learning Techniques and latest econometrics techniques • Designed to measure causal effect of some treatment variable T on outcome variable Y, controlling for set of features X, W	• Use people who think about prediction problems, to also consider Causation problems • End-to-end library for causal inference, wrapper for different libraries (that hold some estimation code) that can be called from Do Why • Outer layer, common API that helps you reason.	Used to estimate the causal effect of an intervention on **some outcome from real-world, non-experimental observational data**
Underlying Statistical Technique	• **Bayesian Time-Series Model (in R)** to estimate how response metric may have evolved after the intervention and if the intervention had not occurred. The delta is then deemed the net impact. Note: there is also an MLE version (in Python)	• **Conditional Average Treatment Effect (CATE)** or Individual Treatment Effect (ITE) from experimental or observational data	• Orthogonal/Double Machine Learning • Forest Based Estimators • Doubly Robust Learning • Estimation Methods with Instruments (deep instrumental variables) • CATE estimators	• Wrapper for causal-inference estimation approaches	• Wrapper for causal-inference estimation approaches
Model Assumptions	• Assumes a graph, not explicitly stated	• No strong assumptions on the model form	• Standard econometrics literature	• Need to explicitly understand causal effect graph (can use DAGitty.net • Like an API	• Package encompasses suite of causal methods using a sklearn-like fit-predict API
When to Use	• **When randomized experiment is not available**. • When seeking to understand causal effects; what would have happened in the absence of the action whose effects we're interested in	• **Campaign targeting optimization** – ads campaign where target of ad is set of customers that will have a favorable response in a given KPI like engagement or sales. CATE identifies customers by estimating effect of the KPI from ad exposure	• **Recommendation A/B Testing**; Interpret experiments with imperfect compliance • **Customer segmentation**; estimate individualized responses to incentives • **Multi-investment Attribution;** distinguish effects of multiple outreach effort	• Breaks down causality into 4 steps, to evaluate causal models	• If looking for flexible causal modelling, with high tweaking availability.

Alternative Methodologies for Causality Estimation

As with all non-experimental approaches to causal inference, valid conclusions require strong assumptions. Try at least two methods, with different assumptions to give confidence

	Google Causal Impact Bayesian Time Series	CausalML	Microsoft EconML	Microsoft "Do Why" Package	IBM Causal Library
About	• **Used to estimate the causal effect of a designed intervention** on a time series (i.e. daily clicks generated from ad campaign	• Uplift modeling and causal inference methods using machine learning algorithms based on recent research • Estimates causal impact of intervention T on outcome Y for users with observed feature X • Treatment effects on observational and experimental data	• Estimating **heterogeneous treatment effects from observational and experimental data** via machine learning • Machine Learning Techniques and latest econometrics techniques • Designed to measure causal effect of some treatment variable T on outcome variable Y, controlling for set of features X, W	• Use people who think about prediction problems, to also consider Causation problems • End-to-end library for causal inference, wrapper for different libraries (that hold some estimation code) that can be called from Do Why • Outer layer, common API that helps you reason.	Used to estimate the causal effect of an intervention on **some outcome from real-world, non-experimental observational data**
Links	CausalImpact (google.github.io) This package in R implements a Bayesian approach to causal impact estimation in time series, as described in Brodersen et. Al (2015)	GitHub - uber/causalml: Uplift modeling and causal inference with machine learning algorithms	**https://econml.azurewebsites.net/** **https://github.com/Microsoft/EconML**	Causality and Machine Learning – Microsoft Research GitHub - py-why/dowhy: DoWhy is a Python library for causal inference that supports explicit modeling and testing of causal assumptions. DoWhy is based on a unified language for causal inference, combining causal graphical models and potential outcomes frameworks.	GitHub - IBM/causallib: A Python package for modular causal inference analysis and model evaluations

High Level Implementation Steps



Step 1:

Setup Knowledge Graph

[(DAGitty - drawing and analyzing causal diagrams (DAGs)](#)

Many kinds of identification methods

Graphical constraint-based methods

- Randomized and natural experiments
- Adjustment Sets
  - Backdoor, “towards necessity”
- Front-door criterion
- Mediation formula

Identification under additional non-graphical constraints

- Instrumental variables
- Regression discontinuity
- Difference-in-differences

Step 2:

Validate the assumptions via refute tests if knowledge graph holds

Google Microsoft

Uber

Step 3:

Select and implement causal impact packages from Google, Microsoft, Uber

```
model = CausalModel(data, graph,
    treatment, outcome)
estimand = model.identify_effect()
estimate = model.estimate_effect(estimand,
 method_name="propensity_score_weighting")
refute = model.refute_estimate(estimand,
    estimate,
    method_name="placebo_treatment_refuter")
```

Step 4:

Implement and validate via wrapper, such as Microsoft’s Do Why Package to understand causal assumptions and their validation Microsoft

Session Objectives:

- Introduction to the concept of A/B Experimentation, and why it matters
- Challenges and situations when A/B Experimentation may not be possible
- Basic understanding of what is meant by estimating a causal effect
- Introduction to a few alternative Methodologies for causality estimation
- Demo: Walkthrough a Jupyter and R sample on how to implement causal impact

Inferring effect of an event using Causal Impact – DEMO

Disclaimer: For educational purposes only. All information and logos are publicly available. Information is not to be misconstrued as official stance or position of any company, individual or party referenced

Sample Walkthrough – Causal Impact

1. Installing the package

`CausalImpact` is available on CRAN and can be installed as follows in an R session:

```
install.packages("CausalImpact")
```

Once installed, the package can be loaded in a given R session using:

```
library(CausalImpact)
```

2. Creating an example dataset

To illustrate how the package works, we create a simple toy dataset. It consists of a response variable `y` and a predictor `x1`. Note that in practice, we'd strive for including many more predictor variables and let the model choose an appropriate subset. The example data has 100 observations. We create an *intervention effect* by lifting the response variable by 10 units after timepoint 71.

```
set.seed(1)
x1 <- 100 + arima.sim(model = list(ar = 0.999), n = 100)
y <- 1.2 * x1 + rnorm(100)
y[71:100] <- y[71:100] + 10
data <- cbind(y, x1)
```

We now have a simple matrix with 100 rows and two columns:

We now have a simple matrix with 100 rows and two columns:

```
dim(data)
```

```
## [1] 100   2
```

```
head(data)
```

```
##             y       x1
## [1,] 105.2950 88.21513
## [2,] 105.8943 88.48415
## [3,] 106.6209 87.87684
## [4,] 106.1572 86.77954
## [5,] 101.2812 84.62243
## [6,] 101.4484 84.60650
```

We can visualize the generated data using:

```
matplot(data, type = "l")
```


https://google.github.io/CausalImpact/CausalImpact.html

Sample Walkthrough – Causal Impact

1. Installing the package

`CausalImpact` is available on CRAN and can be installed as follows in an R session:

```
install.packages("CausalImpact")
```

Once installed, the package can be loaded in a given R session using:

```
library(CausalImpact)
```

2. Creating an example dataset

To illustrate how the package works, we create a simple toy dataset. It consists of a response variable `y` and a predictor `x1`. Note that in practice, we'd strive for including many more predictor variables and let the model choose an appropriate subset. The example data has 100 observations. We create an *intervention effect* by lifting the response variable by 10 units after timepoint 71.

```
set.seed(1)
x1 <- 100 + arima.sim(model = list(ar = 0.999), n = 100)
y <- 1.2 * x1 + rnorm(100)
y[71:100] <- y[71:100] + 10
data <- cbind(y, x1)
```

We now have a simple matrix with 100 rows and two columns:

We now have a simple matrix with 100 rows and two columns:

```
dim(data)
```

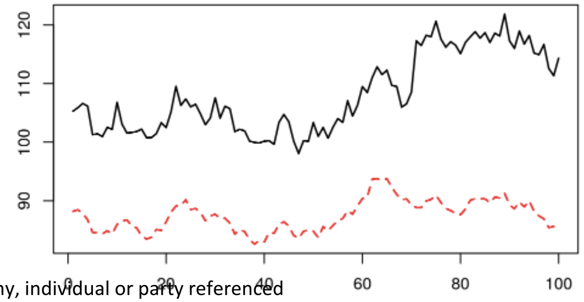
```
## [1] 100   2
```

```
head(data)
```

```
##              y       x1
## [1,] 105.2950 88.21513
## [2,] 105.8943 88.48415
## [3,] 106.6209 87.87684
## [4,] 106.1572 86.77954
## [5,] 101.2812 84.62243
## [6,] 101.4484 84.60650
```

We can visualize the generated data using:

```
matplot(data, type = "l")
```



Sample Walkthrough – Causal Impact

3. Running an analysis

To estimate a causal effect, we begin by specifying which period in the data should be used for training the model (*pre-intervention period*) and which period for computing a counterfactual prediction (*post-intervention period*).

```
pre.period <- c(1, 70)
post.period <- c(71, 100)
```

This says that time points 1 … 70 will be used for training, and time points 71 … 100 will be used for computing predictions. Alternatively, we could specify the periods in terms of dates or time points; see Section 5 for an example.

To perform inference, we run the analysis using:

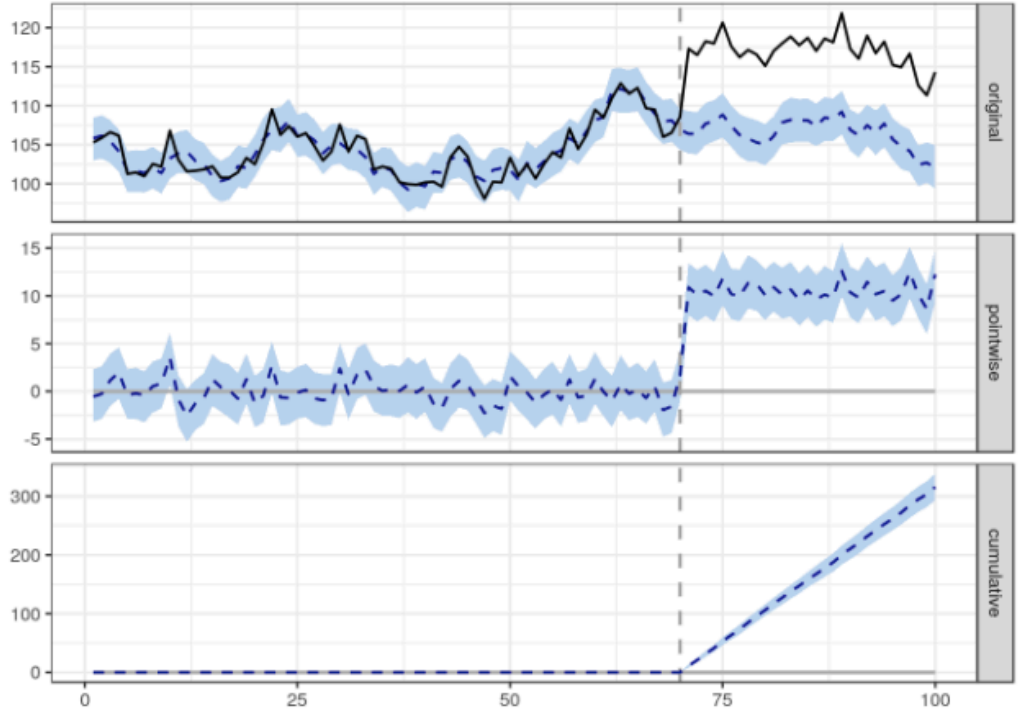
```
impact <- CausalImpact(data, pre.period, post.period)
```

This instructs the package to assemble a structural time-series model, perform posterior inference, and compute estimates of the causal effect. The return value is a `CausalImpact` object.

4. Plotting the results

The easiest way of visualizing the results is to use the `plot()` function that is part of the package:

```
plot(impact)
```



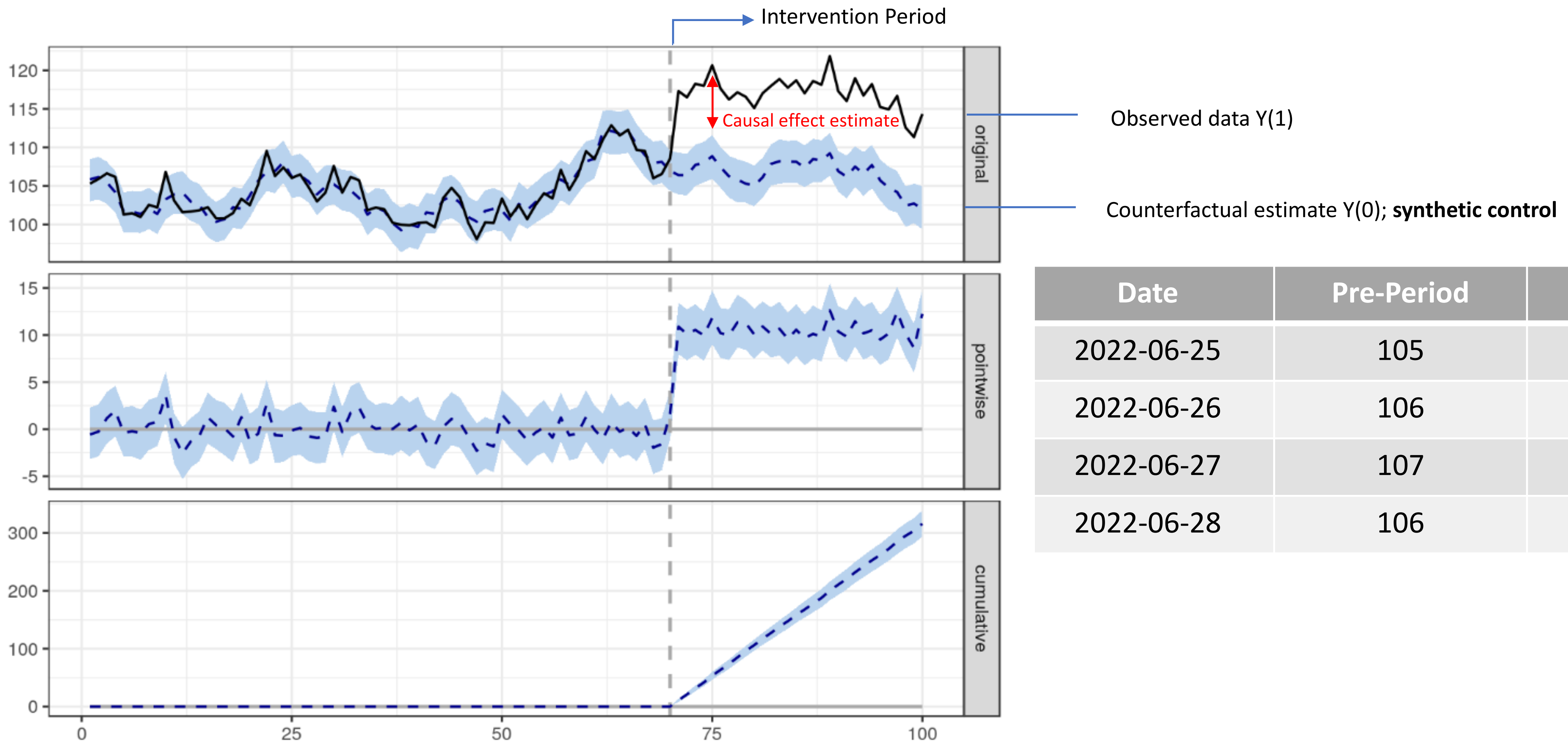
By default, the plot contains three panels. The first panel shows the data and a counterfactual prediction for the post-treatment period. The second panel shows the difference between observed data and counterfactual predictions. This is the *pointwise* causal effect, as estimated by the model. The third panel adds up the pointwise contributions from the second panel, resulting in a plot of the *cumulative* effect of the intervention.

Remember, once again, that all of the above inferences depend critically on the assumption that the covariates were not themselves affected by the intervention. The model also assumes that the relationship between covariates and treated time series, as established during the pre-period, remains stable throughout the post-period.

https://google.github.io/CausalImpact/CausalImpact.html

Sample Walkthrough – Causal Impact

- Uses Bayesian structural time-series; statistical technique that intakes 3 variables
- Looks at observed data y1, and to understand causal effects, we estimate the counterfactual; what would have happened in the absence of the action whose effects we're interested in
- Since don't really have an experiment, there isn't a true "control" in experiment sense. If want to be able to estimate something that looks like a control, can do so via **synthetic control**
- **Difference between counterfactual and the actual observed data is the causal effect**



Date	Pre-Period	Post-Period
2022-06-25	105	115
2022-06-26	106	120
2022-06-27	107	115
2022-06-28	106	113

https://google.github.io/CausalImpact/CausalImpact.html

Sample Walkthrough – Causal Impact

5. Working with dates and times

It is often more natural to feed a time-series object into `CausalImpact()` rather than a data frame. For example, we might create a `data` variable as follows:

```
time.points <- seq.Date(as.Date("2014-01-01"), by = 1, length.out = 100)
data <- zoo(cbind(y, x1), time.points)
head(data)
```

```
##                   y       x1
## 2014-01-01 105.2950 88.21513
## 2014-01-02 105.8943 88.48415
## 2014-01-03 106.6209 87.87684
## 2014-01-04 106.1572 86.77954
## 2014-01-05 101.2812 84.62243
## 2014-01-06 101.4484 84.60650
```

We can now specify the pre-period and the post-period in terms of time points rather than indices:

```
pre.period <- as.Date(c("2014-01-01", "2014-03-11"))
post.period <- as.Date(c("2014-03-12", "2014-04-10"))
```

As a result, the x-axis of the plot shows time points instead of indices:

```
impact <- CausalImpact(data, pre.period, post.period)
plot(impact)
```

6. Printing a summary table

To obtain a numerical summary of the analysis, we use:

```
summary(impact)
```

```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   117            3511
## Prediction (s.d.)        107 (0.37)     3196 (11.03)
## 95% CI                   [106, 107]     [3174, 3217]
##
## Absolute effect (s.d.)   11 (0.37)      316 (11.03)
## 95% CI                   [9.8, 11]      [294.9, 337]
##
## Relative effect (s.d.)   9.9% (0.35%)   9.9% (0.35%)
## 95% CI                   [9.2%, 11%]    [9.2%, 11%]
##
## Posterior tail-area probability p:   0.001
## Posterior prob. of a causal effect:  99.9%
##
## For more details, type: summary(impact, "report")
```

The **Average** column talks about the average (across time) during the post-intervention period (in the example: time points 71 through 100). The **Cumulative** column sums up individual time points, which is a useful perspective if the response variable represents a flow quantity (such as queries, clicks, visits, installs, sales, or revenue) rather than a stock quantity (such as number of users or stock price).

Sample Walkthrough – Causal Impact

```
summary(impact, "report")
```

The individual numbers in the table, at full precision, can be accessed using:

```
impact$summary
```

See below for tips on how to use these commands with *knitr / R Markdown*.

7. Adjusting the model

So far, we've simply let the package decide how to construct a time-series model for the available data. However, there are several options that allow us to gain a little more control over this process. These options are passed into `model.args` as individual list elements, for example:

```
impact <- CausalImpact(..., model.args = list(niter = 5000, nseasons = 7))
```

Available options

- `niter` Number of MCMC samples to draw. More samples lead to more accurate inferences. Defaults to **1000**.
- `standardize.data` Whether to standardize all columns of the data before fitting the model. This is equivalent to an empirical Bayes approach to setting the priors. It ensures that results are invariant to linear transformations of the data. Defaults to **TRUE**.
- `prior.level.sd` Prior standard deviation of the Gaussian random walk of the local level. Expressed in terms of data standard deviations. Defaults to **0.01**, a typical choice for well-behaved and stable datasets with low residual volatility after regressing out known predictors (e.g., web searches or sales in high quantities). When in doubt, a safer option is to use **0.1**, as validated on synthetic data, although this may sometimes give rise to unrealistically wide prediction intervals.
- `nseasons` Period of the seasonal components. In order to include a seasonal component, set this to a whole number greater than 1. For example, if the data represent daily observations, use 7 for a day-of-week component. This interface currently only supports up to one seasonal component. To specify multiple seasonal components, use bsts to specify the model directly, then pass the fitted model in as bsts.model. Defaults to **1**, which means no seasonal component is used.
- `season.duration` Duration of each season, i.e., number of data points each season spans. Defaults to **1**. For example, to add a day-of-week component to data with daily granularity, use `model.args = list(nseasons = 7, season.duration = 1)`. To add a day-of-week component to data with hourly granularity, set `model.args = list(nseasons = 7, season.duration = 24)`.
- `dynamic.regression` Whether to include time-varying regression coefficients. In combination with a time-varying local trend or even a time-varying local level, this often leads to overspecification, in which case a static regression is safer. Defaults to **FALSE**.

Opportunity to also create custom models, using bsts (Bayesian Structural Time Series) package for non-Gaussian error families in the observation equation. Read literature here: https://cran.r-project.org/web/packages/bsts/bsts.pdf

References/Helpful Links

- Inferring the effect of an event using CausalImpact by Kay Brodersen – YouTube
- The Challenge of Inference in the Social Sciences (slideshare.net)
- GitHub - IBM/causallib: A Python package for modular causal inference analysis and model evaluations