GeoLift MMM Calibration

I. Introduction

While Marketing Mix Modeling (MMM) is highly useful in helping to attribute incremental sales to marketing and nonmarketing activities, its regression-based methodology does have limitations. That's why an **important aspect of modern modeling is to undertake calibration with experiments**.

Experiments such as conversion Lift—randomized control trials and Geo-experiments measure the true value of a channel's adspend and can create a foundation for marketing mix models by establishing ground-truth measurements. The combination of these approaches creates a powerful analytic toolset for advertisers looking for unified and consistent measurement.

GeoLift is an end-to-end solution from Meta Open Source that measures geographic incrementality in a resilient, transparent, and reproducible way, using the latest advances in synthetic control methods. Through GeoLift it is possible to measure a channel's total (online/offline/omni-channel) incremental effect which can be used to accurately calibrate the MMM model.

II. MMM Calibration

Jasper's Market is a Fruit and Vegetable store with presence across all of the US that has a successful e-commerce which complements their historically robust in-store sales. In an effort to address dwindling sales growth during the past couple of years, the client is looking to optimize their media spend based on MMM results. Taking into consideration the hybrid (in-store and online) nature of their total sales, Jasper's Market BI team will calibrate Meta's effectiveness with a GeoLift.

The following data is available for this calibration exercise: * Historical daily sales data from January 1st 2021. * Data from 03/06/2021 will be used to build the GeoLift models. * A GeoLift test was conducted in September 2021 to calibrate Meta's Lift. * Meta's historical iROAS is 1.6

Calibrating with GeoLift

MMM calibration with standard GeoLift tests is straightforward. In this type of test, the treatment is delivered to the test regions while the control ones do not see the ads. Since the incrementality estimate provided by GeoLift measures **all** the investment at Meta during the test period, this value can be directly provided into the MMM model for calibration. The steps to calibrate with Standard GeoLift tests are:

1. Plan a GeoLift test using historical data.
2. Execute the GeoLift test.
3. Plugin the incrementality estimate into the MMM model.

```
library(GeoLift)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

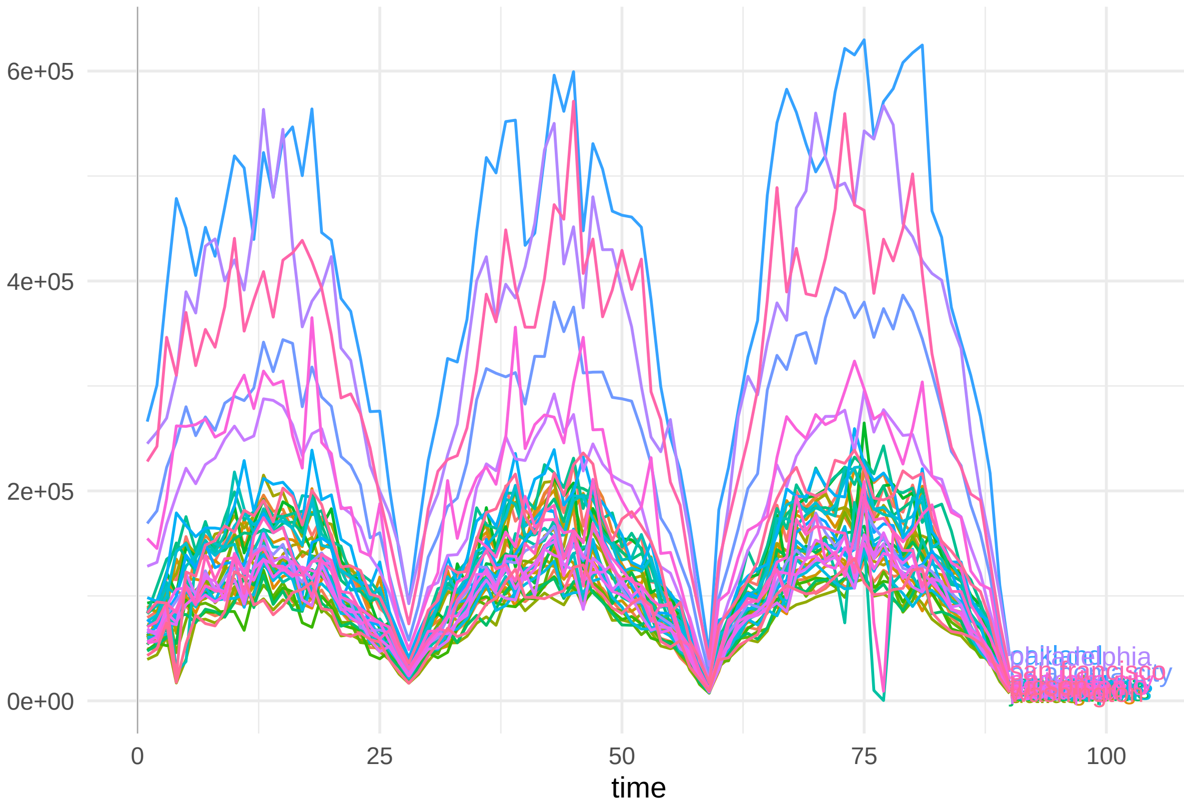
```
data <- read.csv("data_positiveGLv1.csv")
head(data)
```

```
##   location         Y     date
## 1  atlanta  79360.01 03/06/21
## 2  atlanta  84903.39 04/06/21
## 3  atlanta 116965.60 05/06/21
## 4  atlanta 115300.81 06/06/21
## 5  atlanta 156516.03 07/06/21
## 6  atlanta 130905.70 08/06/21
```

1. Designing a GeoLift Test

The first step to calibrate an MMM with a GeoLift is to design the test. This can be easily accomplished with the `GeoLiftMarketSelection()` function. Given that the KPI being modeled in this experiment is Sales ($), instead of providing the CPIC estimate the cost of each incremental dollar in Sales is used. This value can be obtained as the inverse of the historical iROAS, that is: $1/iROAS$. Given that the historical iROAS=1.6, the value used is $1/1.6 = 0.625$.

```
data$date <- lubridate::dmy(data$date)

#Keep only pre-treatment data for power analysis
data <- data %>% dplyr::filter(date <  lubridate::dmy("01/09/2021"))

GeoTestData_Test <- GeoDataRead(data = data,
                                   date_id = "date",
                                   location_id = "location",
                                   Y_id = "Y",
                                   format = "yyyy-mm-dd")

GeoPlot(GeoTestData_Test,
        Y_id = "Y",
        time_id = "time",
        location_id = "location")
```



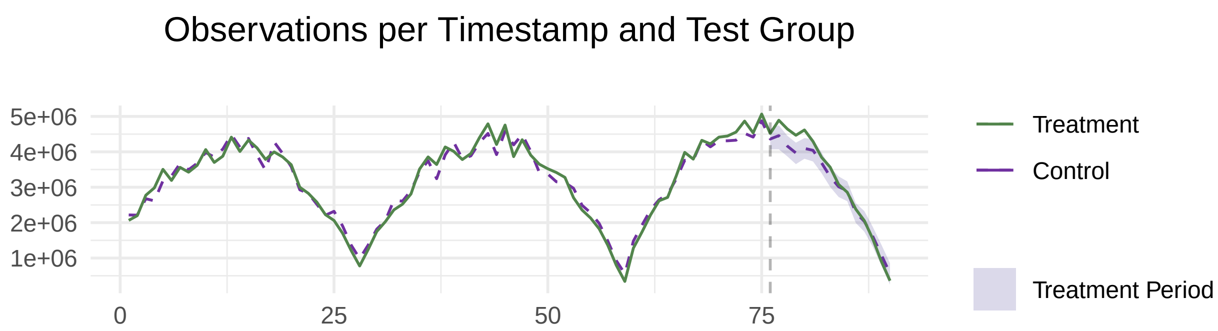

```
MarketSelections <- GeoLiftMarketSelection(data = GeoTestData_Test,
                                           treatment_periods = c(15),
                                           N = c(15,20,25),
                                           Y_id = "Y",
                                           location_id = "location",
                                           time_id = "time",
                                           effect_size = seq(0, 0.15, 0.01),
                                           lookback_window = 1,
                                           holdout = c(0, 0.55),
                                           cpic = 1/1.6,
                                           alpha = 0.1,
                                           fixed_effects = TRUE)
```

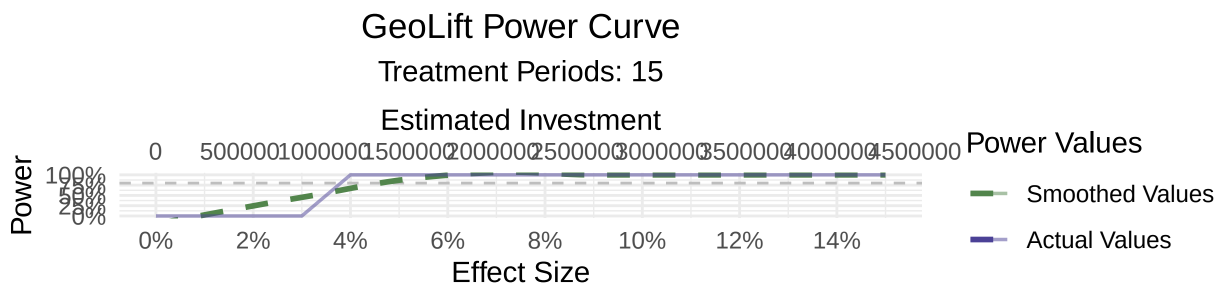
```
## Setting up cluster.

## Importing functions into cluster.

##
## Deterministic setup with 15 locations in treatment.

##
## Deterministic setup with 20 locations in treatment.

##
## Deterministic setup with 25 locations in treatment.

## ID
## 1 1
## 2 2
## 3 3
## 4 4
```

```
## 5 5
## 6 6
## location
## 1 atlanta, austin, chicago, columbus, dallas, denver,
honolulu, houston, indianapolis, kansas city, las vegas,
milwaukee, nashville, oakland, oklahoma city, orlando,
phoenix, reno, saint paul, san francisco
## 2 atlanta, austin, baltimore, baton rouge, chicago,
cincinnati, columbus, dallas, denver, honolulu, houston,
indianapolis, las vegas, minneapolis, nashville, new york,
oklahoma city, orlando, philadelphia, phoenix, portland,
reno, saint paul, san diego, san francisco
## 3 atlanta, austin, baltimore, baton rouge, chicago,
cincinnati, columbus, honolulu, houston, las vegas, miami,
minneapolis, nashville, new york, orlando, philadelphia,
phoenix, portland, reno, san diego
## 4 atlanta, chicago, cleveland, columbus, dallas, denver,
indianapolis, jacksonville, kansas city, las vegas,
memphis, milwaukee, oklahoma city, orlando, phoenix, reno,
saint paul, san francisco, tucson, washington
## 5 atlanta, austin, baltimore, baton rouge, chicago,
cincinnati, denver, honolulu, houston, indianapolis, las
vegas, nashville, oklahoma city, orlando, philadelphia,
phoenix, portland, reno, san diego, san francisco
## 6 atlanta, austin, boston, cleveland, dallas, denver,
detroit, indianapolis, jacksonville, kansas city, las
vegas, milwaukee, new orleans, oakland, oklahoma city,
orlando, phoenix, saint paul, san francisco, tucson
## duration EffectSize Power AvgScaledL2Imbalance
Investment AvgATT
## 1 15 0.06 1 0.5194424 1587141.1 8425.547
## 2 15 0.05 1 0.4662100 1499188.1 6919.021
## 3 15 0.06 1 0.5766735 1345781.4 7765.217
## 4 15 0.08 1 0.6115481 1907117.6 9686.985
## 5 15 0.03 1 0.4966899 767318.4 5637.949
## 6 15 0.07 1 0.4863569 1879298.1 10851.153
## Average_MDE ProportionTotal_Y abs_lift_in_zero Holdout
rank
## 1 0.05970551 0.5407168 0.000 0.4592832 1
## 2 0.05430592 0.6123432 0.004 0.3876568 2
## 3 0.06523348 0.4554923 0.005 0.5445077 3
## 4 0.07590165 0.4915345 0.004 0.5084655 4
## 5 0.04180390 0.5198237 0.012 0.4801763 5
## 6 0.07622526 0.5508474 0.006 0.4491526 5
```

Analyzing the different feasible market selections obtained from `GeoLiftMarketSelection()`, we select Atlanta, Austin, Boston, Chicago, Columbus, Denver, Detroit, Indianapolis, Jacksonville, Kansas City, Las Vegas, Milwaukee, New Orleans, Oakland, Oklahoma City, Orlando, Phoenix, Reno, Saint Paul, and San Francisco based on the model fit metrics and holdout size.

```
plot(MarketSelections, market_ID = 11, print_summary = FALSE)
```





Rank: 10

onolulu, houston, indianapolis, kansas city, las vegas, milwaukee, oakland, oklahoma city

Treatment Periods: 15

Effect Size: 0.04

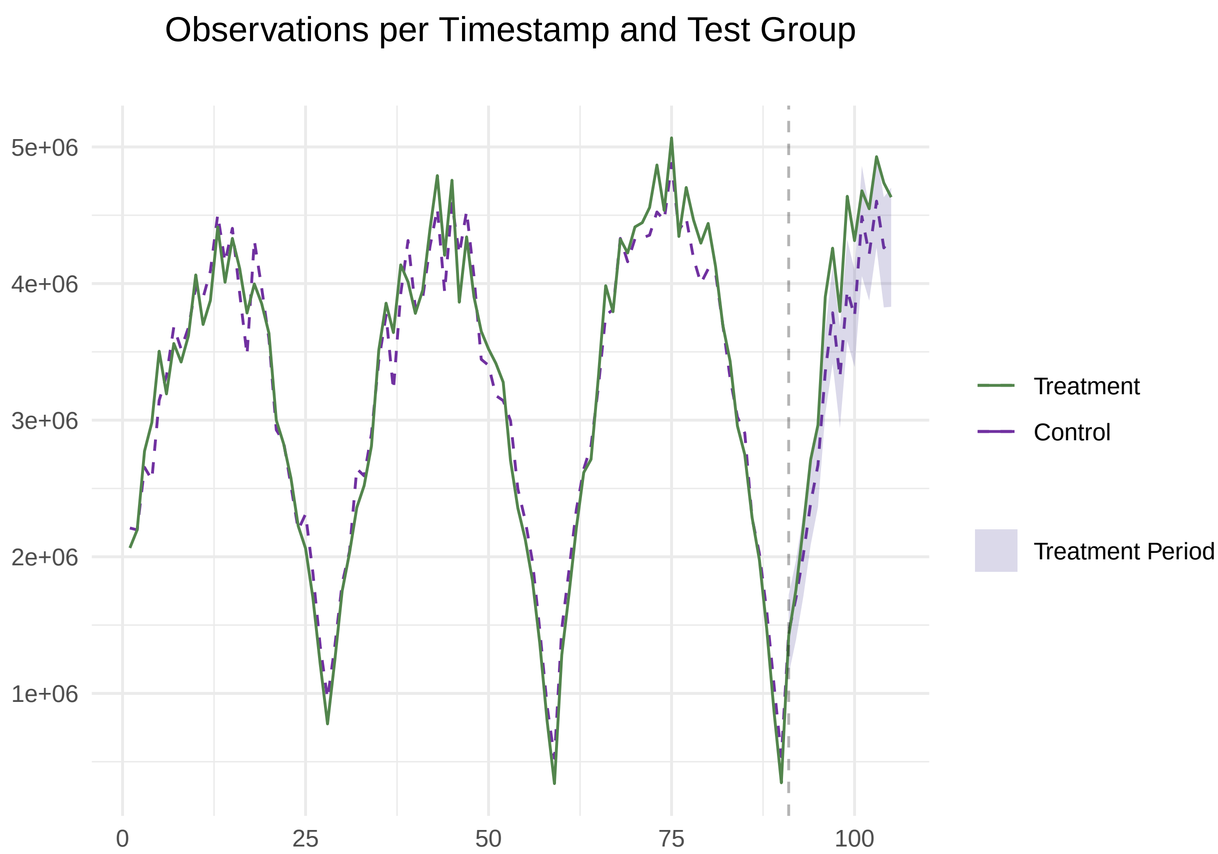
```
locs <- unlist(strsplit(stringr::str_replace_all(
  MarketSelections$BestMarkets$location[11], ", ", ","),split = ","))
```

2. Execute the GeoLift test

Once the GeoLift test has concluded, we can use the `GeoLift()` function to identify the total incremental effect of the channel for the duration of the experiment.

```
data <- read.csv("data_positiveGLv1.csv")

GeoTestData_Test <- GeoDataRead(data = data,
                                date_id = "date",
                                location_id = "location",
                                Y_id = "Y",
                                format = "dd/mm/yyyy")

GeoTestDataResults <- GeoLift(Y_id = "Y",
                              data = GeoTestData_Test,
                              locations = locs,
                              treatment_start_time = 91,
                              treatment_end_time = 105,
                              model = "None",
                              fixed_effects = TRUE,
                              stat_test = "Total")

plot(GeoTestDataResults, type = "Lift")
```

```
## You can include dates in your chart if you supply the end date of the treatment. Just specify the tr
```



3. Calibrate the MMM model

GeoLift's incrementality estimate: 7159530 can now be used to calibrate the MMM.

Calibrating with Negative GeoLift

Standard GeoLift experiments provides the user with a simple, accurate, and unbiased way to calibrate MMM experiments. However, this type of tests have the disadvantage that they typically require large holdouts. Whenever a large holdout isn't desirable, Inverse or Negative GeoLifts offer a great alternative at the cost of a little bit of additional complexity. Given that the effect measured in Inverse GeoLift experiments is the cost of opportunity of not running a campaign (how many sales/incremental KPI units would have happened if the campaigns were served to the holdout regions), the incremental effect must be scaled to the locations that were served ads. Namely, the steps required to calibrate MMM results with Negative GeoLifts are:

1. Plan an Inverse/Negative GeoLift test using historical data.
2. Execute the Inverse/Negative GeoLift test.
3. Obtain the CPIC or iROAS estimate.
4. Find the incrementality estimate based on the investment during the test.
5. Plugin the incrementality estimate into the MMM model.

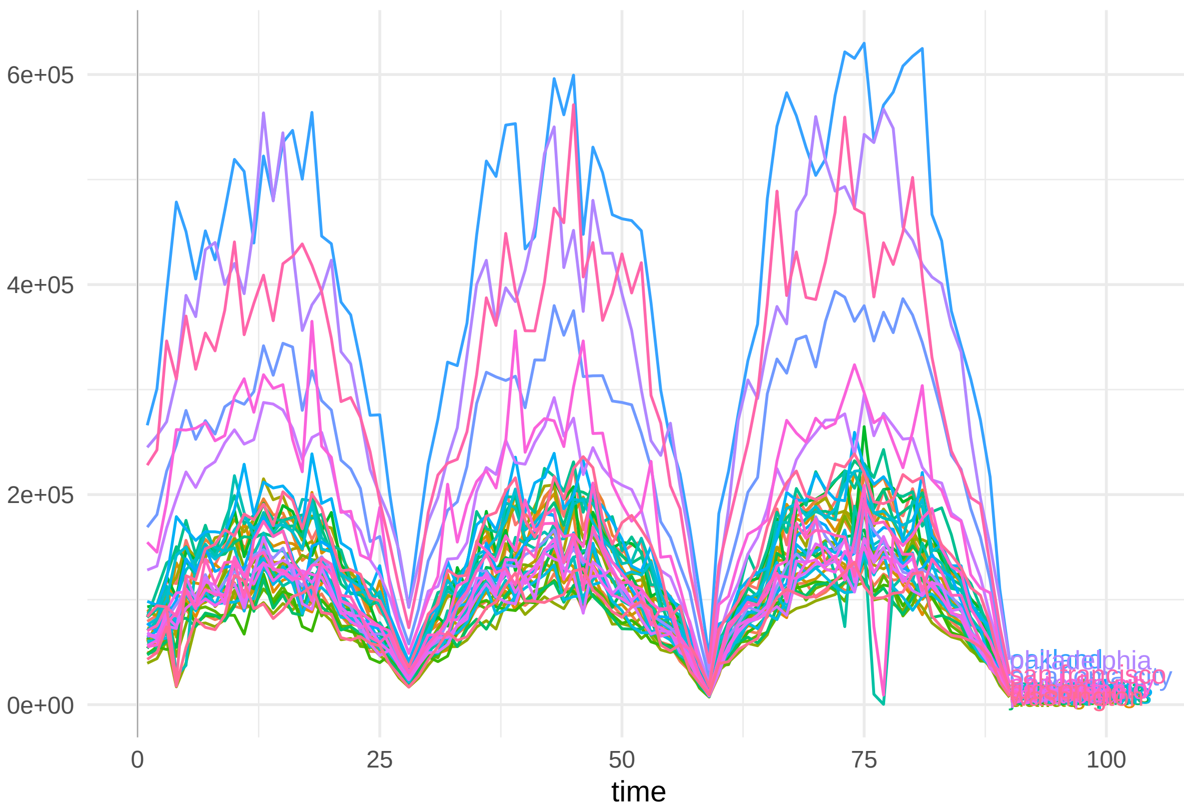
1. Plan the Geo-Experiment

The first step to calibrate an MMM with a Negative or Inverse GeoLift is to design the test with `GeoLiftMarketSelection()`. Calibration with Negative or Inverse GeoLifts requires the projection of the incrementality estimate found in the test regions to the locations that did receive an adspend. Therefore, it is important to make sure that there is a close relationship between them. We can analyze how similar/dissimilar the test and control regions are by adding the `Correlations = TRUE` parameter to `GeoLiftMarketSelection()`. It is recommended to ensure that this correlation coefficient is above 0.90.

```r
data <- read.csv("data_negativeGLv2.csv")
head(data)
```

```
##   X location     date         Y
## 1 1  atlanta 03/06/21  79360.01
## 2 2  atlanta 04/06/21  84903.39
## 3 3  atlanta 05/06/21 116965.60
## 4 4  atlanta 06/06/21 115300.81
## 5 5  atlanta 07/06/21 156516.03
## 6 6  atlanta 08/06/21 130905.70
```

```r
data$date <- lubridate::dmy(data$date)

#Keep only pre-treatment data for power analysis
data <- data %>% dplyr::filter(date <  lubridate::dmy("01/09/2021"))

GeoTestData_Test <- GeoDataRead(data = data,
                                date_id = "date",
                                location_id = "location",
                                Y_id = "Y",
                                format = "yyyy-mm-dd")

GeoPlot(GeoTestData_Test,
        Y_id = "Y",
        time_id = "time",
        location_id = "location")
```



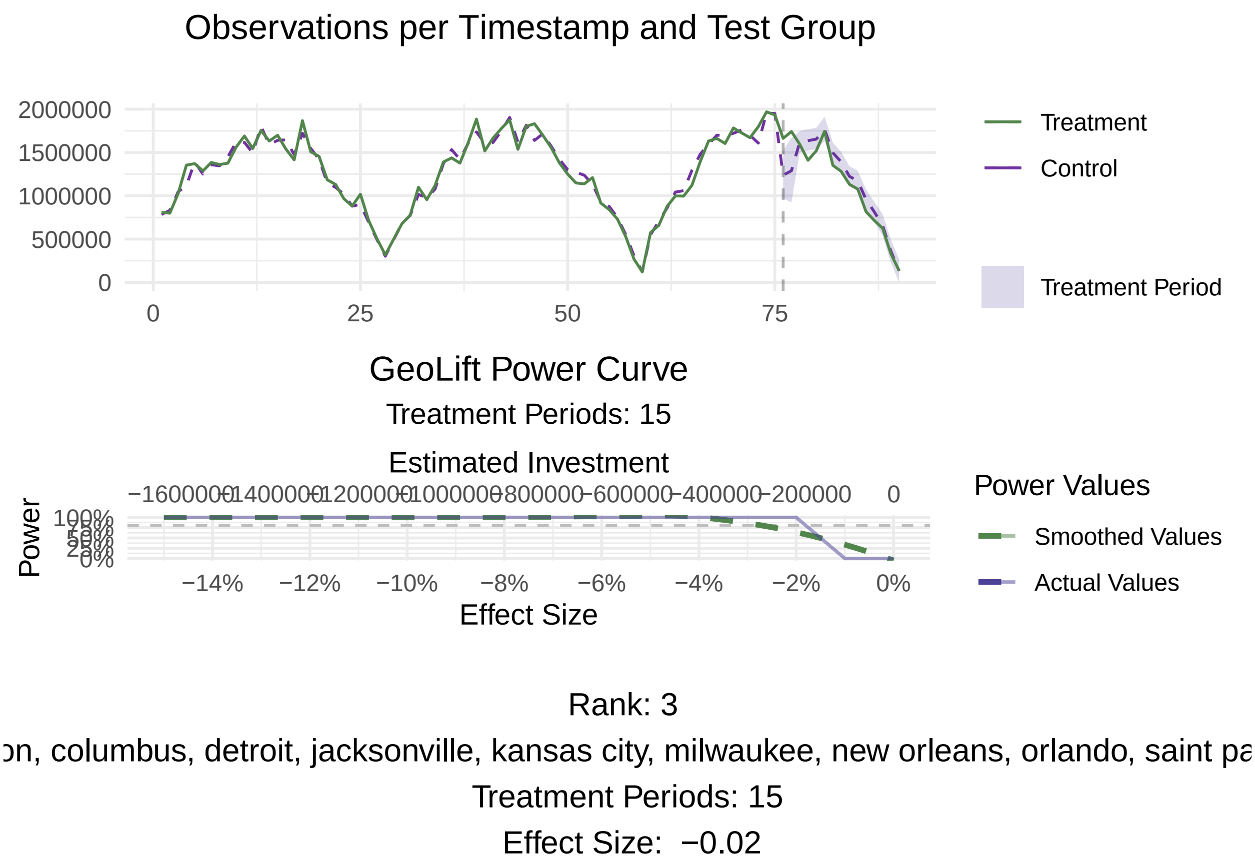

```r
MarketSelections <- GeoLiftMarketSelection(data = GeoTestData_Test,
                                           treatment_periods = c(15),
```

```r
                                            N = c(5,10,15,20,25),
                                            Y_id = "Y",
                                            location_id = "location",
                                            time_id = "time",
                                            effect_size = -seq(0, 0.15, 0.01),
                                            lookback_window = 1,
                                            #holdout = c(0, 0.55),
                                            cpic = 1/1.6,
                                            alpha = 0.1,
                                            fixed_effects = TRUE,
                                            Correlations = TRUE)
```

```
## Setting up cluster.

## Importing functions into cluster.

##
## Deterministic setup with 5 locations in treatment.

##
## Deterministic setup with 10 locations in treatment.

##
## Deterministic setup with 15 locations in treatment.

##
## Deterministic setup with 20 locations in treatment.

##
## Deterministic setup with 25 locations in treatment.

## ID
## 1 1
## 2 2
## 3 3
## 4 4
## 5 5
## 6 6
## location
## 1 atlanta, austin, chicago, cincinnati, columbus,
denver, detroit, honolulu, indianapolis, jacksonville, las
vegas, los angeles, milwaukee, nashville, new york,
oklahoma city, orlando, saint paul, san antonio, tucson
## 2 atlanta, austin, chicago, columbus, dallas, denver,
honolulu, houston, indianapolis, kansas city, las vegas,
milwaukee, nashville, oakland, oklahoma city, orlando,
phoenix, reno, saint paul, san francisco
## 3 boston, columbus, detroit, jacksonville, kansas city,
milwaukee, new orleans, orlando, saint paul, salt lake city
## 4 boston, columbus, detroit, jacksonville, kansas city,
milwaukee, new orleans, phoenix, saint paul, salt lake city
## 5 atlanta, austin, denver, detroit, honolulu,
indianapolis, las vegas, los angeles, milwaukee, nashville,
new york, orlando, saint paul, san antonio, tucson
## 6 chicago, columbus, minneapolis, orlando, phoenix
## duration EffectSize Power AvgScaledL2Imbalance
Investment AvgATT
```

```
## 1 15 -0.01 1 0.1672144 207180.5 -1322.769
## 2 15 -0.06 1 0.5194424 1587141.1 -8503.958
## 3 15 -0.02 1 0.5489390 218584.0 -1763.592
## 4 15 -0.05 1 0.6131421 585503.5 -6513.151
## 5 15 -0.01 1 0.1994970 148057.2 -1916.595
## 6 15 -0.06 1 0.3401327 321063.1 -6572.733
## Average_MDE ProportionTotal_Y abs_lift_in_zero Holdout
rank correlation
## 1 -0.01194762 0.4240233 0.002 0.4240233 1 0.9917626
## 2 -0.06026115 0.5407168 0.000 0.5407168 2 0.9862834
## 3 -0.01520205 0.2307305 0.005 0.2307305 3 0.9672900
## 4 -0.05203228 0.2465187 0.002 0.2465187 3 0.9708067
## 5 -0.01805573 0.3037626 0.008 0.3037626 5 0.9879541
## 6 -0.05771673 0.1114977 0.002 0.1114977 5 0.9862221
```

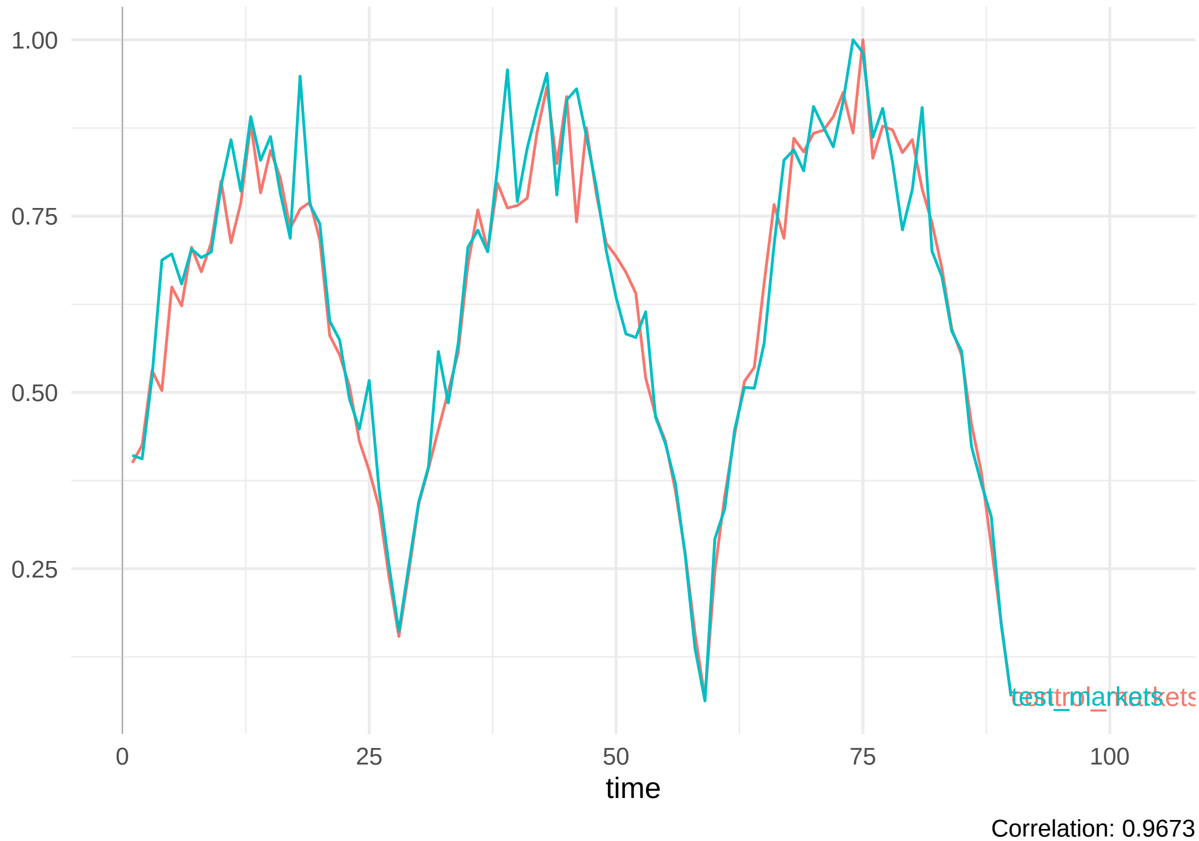
```
plot(MarketSelections, market_ID = 3, print_summary = FALSE)
```




```
locs <- unlist(strsplit(stringr::str_replace_all(
  MarketSelections$BestMarkets$location[3], ", ", ","),split = ","))
```

The correlation between the test and control time-series can be visualized with the `plotCorrels()` function.

```
plotCorrels(data = GeoTestData_Test,
            locs = locs,
            scaled = TRUE)
```



[Optional] Check the Synthetic Control Weights It is highly recommended in Negative/Inverse GeoLift studies to double-check that the regions that compose the Synthetic Control Model will in fact have the necessary investment suggested by the Power Analysis. This can be accomplished by looking into the model weights with the `GetWeights()` function and making sure that highly weighted regions do have the required investment to observe the Lift.

```
weights <- GetWeights(data = GeoTestData_Test,
                      locations = locs,
                      pretreatment_end_time = 90)
```

```
## One outcome and one treatment time found. Running single_augsynth.
```

```
weights %>% dplyr::arrange(desc(weight))
```

```
##         location       weight
## 1    minneapolis  5.032143e-01
## 2      cleveland  1.221685e-01
## 3    los angeles  1.205580e-01
## 4        memphis  1.046417e-01
## 5        oakland  8.314780e-02
## 6        phoenix  3.417960e-02
## 7      las vegas  1.970593e-02
## 8       new york  1.238402e-02
## 9        houston  9.809525e-11
## 10     baltimore  7.549392e-11
## 11         miami  5.073400e-11
## 12        austin  3.414344e-11
## 13          reno  3.264449e-11
```

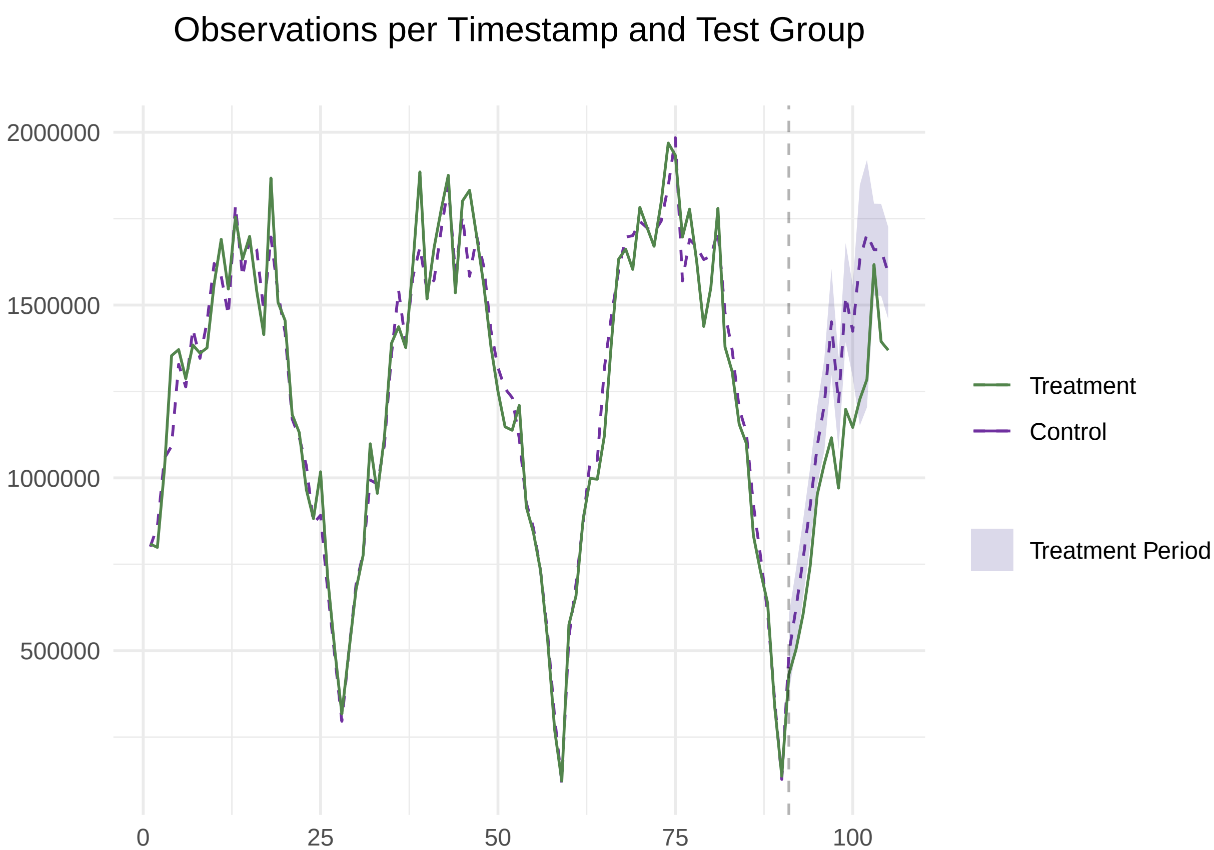
```
## 14      portland  3.173860e-11
## 15    cincinnati  2.928515e-11
## 16   baton rouge  2.595999e-11
## 17       chicago  2.573438e-11
## 18     san diego  2.357071e-11
## 19      honolulu  2.144327e-11
## 20     nashville  1.967272e-11
## 21        dallas  1.391890e-11
## 22    washington  1.264350e-12
## 23  indianapolis -1.777786e-12
## 24   san antonio -2.218565e-12
## 25        denver -3.240882e-12
## 26 oklahoma city -4.718438e-12
## 27       atlanta -7.879408e-12
## 28  philadelphia -1.462958e-11
## 29 san francisco -3.300819e-11
## 30        tucson -8.935607e-11
```

2. Execute the Inverse/Negative GeoLift test

By analyzing the test period with `GeoLift()` it is possible to obtain the incrementality estimate. The negative sign of this estimate points out that the test regions had a smaller KPI total than the synthetic control which did receive an adspend.

```r
data <- read.csv("data_negativeGLv2.csv")

GeoTestData_Test <- GeoDataRead(data = data,
                                   date_id = "date",
                                   location_id = "location",
                                   Y_id = "Y",
                                   format = "dd/mm/yyyy")

GeoTestDataResults <- GeoLift(Y_id = "Y",
                              data = GeoTestData_Test,
                              locations = locs,
                              treatment_start_time = 91,
                              treatment_end_time = 105,
                              model = "None",
                              fixed_effects = TRUE,
                              stat_test = "Total")

plot(GeoTestDataResults, type = "Lift")
```

```
## You can include dates in your chart if you supply the end date of the treatment. Just specify the tr
```



The results indicate that the client sacrificed $3366433 in sales by not running ads in our test regions.

3. Obtain the CPIC or iROAS estimate.

The CPIC or iROAS estimates can be obtained by dividing the incremental KPI of the synthetic control by the total investment on the synthetic control regions. More specifically:

$$iROAS = \frac{Incremental\ Sales}{Synthetic\ Investment} = \frac{\left|\sum_{t=t_0+1}^{T} ATT_t\right|}{\sum_i^N w_i \cdot Investment_i}$$

Or:

$$CPIC = \frac{Synthetic\ Investment}{Incremental\ Conversions} = \frac{\sum_i^N w_i \cdot Investment_i}{\left|\sum_{t=t_0+1}^{T} ATT_t\right|}$$

Where t_0 represents the last pre-treatment time-stamp, T represents the treatment end, and N represents the number of units in the pool of controls. For simplicity, the Average Estimated Treatment Effect ($\overline{ATT}$) provided in the GeoLift output, which represents the average ATT over the entire treatment, can also be used to calculate the iROAS and CPIC.

$$iROAS = \frac{\left|\overline{ATT}\right| \cdot (T - t_0 + 1)}{\sum_i^N w_i \cdot Investment_i}$$

$$CPIC = \frac{\sum_i^N w_i \cdot Investment_i}{\left|\overline{ATT}\right| \cdot (T - t_0 + 1)}$$

```r
investment <- read.csv("test_investment_by_city_v2.csv")

# Calculating the Investment in the Synthetic Control
#total investment per city
investment <- investment %>%
              group_by(location) %>%
              summarize(total_meta_investment = sum(investment))
# Add weight from GeoLift Results
weights <- GetWeights(data = GeoTestData_Test,
                      locations = locs,
                      pretreatment_end_time = 90)
```

```
## One outcome and one treatment time found. Running single_augsynth.
```

```r
investment <- investment %>%
              left_join(weights, by="location") %>%
              mutate(synth_investment = total_meta_investment * weight)


# Calculating the iROAS estimate
iROAS <- round(abs(GeoTestDataResults$inference$ATT *
                     (GeoTestDataResults$TreatmentEnd-GeoTestDataResults$TreatmentStart+1)) /
                 (sum(investment$synth_investment)),2)
iROAS
```

```
## [1] 1.6
```

```r
# Alternatively:
#iROAS <- round(abs(sum(GeoTestDataResults$summary$att$Estimate
#                          [GeoTestDataResults$TreatmentStart:GeoTestDataResults$TreatmentEnd])) /
#                 (sum(investment$synth_investment)),2)
```

4. Find the incrementality estimate

By using the CPIC or iROAS estimate from our GeoLift test, we can find the total incremental effect generated by Meta in order to calibrate our MMM model.

```r
sum(investment$total_meta_investment)*iROAS
```

```
## [1] 11015655
```