Quantium Task 2: Causal Analysis

Yihan Yang

2023-11-27

```r
data <- fread(paste0("/Users/yangyihan/Downloads/","QVI_data.csv"))
#### Set themes for plots
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

```r
data[, YEARMONTH := format(DATE, "%Y%m")]
head(data,2)
```

```
##    LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
## 1:           1000 2018-10-17         1      1        5
## 2:           1002 2018-09-16         1      2       58
##                                PROD_NAME PROD_QTY TOT_SALES PACK_SIZE   BRAND
## 1: Natural Chip        Compny SeaSalt175g        2       6.0       175 NATURAL
## 2:  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150     RRD
##                LIFESTAGE PREMIUM_CUSTOMER YEARMONTH
## 1: YOUNG SINGLES/COUPLES          Premium    201810
## 2: YOUNG SINGLES/COUPLES       Mainstream    201809
```

```r
measureOverTime <- data[, .(
  totSales = sum(TOT_SALES), # Sum of sales
  nCustomers = uniqueN(LYLTY_CARD_NBR), # Count of unique customers
  nTxn = .N,
  nTxnPerCust = .N / uniqueN(LYLTY_CARD_NBR), # Number of transactions per customer
  nChipsPerTxn = sum(PROD_QTY) / .N, # Chips (assuming 'Quantity') per transaction
  avgPricePerUnit = sum(TOT_SALES) / sum(PROD_QTY) # Average price per unit
), by = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]
```

```r
#### Filter to the pre-trial period and stores with full observation periods
storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])
fullobsdata <- measureOverTime[STORE_NBR %in% storesWithFullObs]
```

```r
# Use dcast to reshape the data
salesdata <- dcast(fullobsdata,YEARMONTH ~ STORE_NBR, value.var = "totSales")
presalesData <- salesdata[YEARMONTH < 201902]

# View the reshaped dataset
head(presalesData,2)
```

```
##    YEARMONTH     1     2       3      4     5     6      7     8     9     10
## 1:    201807 206.9 150.8 1205.70 1399.9 812.0 260.0 1024.7 381.6 289.7 892.00
## 2:    201808 176.1 193.8 1079.75 1259.5 745.1 203.2 1119.9 383.0 327.5 878.65
##       12    13   14    15    16    17    18    19    20    21    22    23    24
## 1: 429.6 811.8 46.9 742.6 113.8 485.7 326.6 729.1 269.2 409.4 309.2 890.8 719.2
## 2: 360.8 756.9 46.8 840.2 149.2 353.3 273.6 737.8 195.7 510.1 201.3 952.1 792.7
```

```
##       25     26    27     28    29    30    32     33    34    35      36
## 1: 414.6 1245.0 470.1 754.00 309.2 879.8 761.4 1045.6 335.8 124.2 1014.20
## 2: 340.3 1092.8 526.9 753.95 382.3 935.8 751.0  934.8 343.7 147.6 1078.05
##        37    38    39   40    41   42     43    45    46     47    48     49
## 1: 471.60 301.6 866.0 1332 216.4 27.3 1003.8 981.2 253.0 290.30 929.4 1099.7
## 2: 547.75 283.9 919.2 1208 209.8 44.9 1005.7 669.1 240.7 407.85 851.1  840.7
##       50    51   52    53    54    55    56    57      58     59     60   61
## 1: 314.4 116.2 28.3 229.8 480.8 889.6 674.6 839.6 1627.20 1267.6 1106.6 38.4
## 2: 292.4 208.3 40.3 255.1 384.2 910.3 634.5 915.4 1043.75  932.5 1064.6 27.9
##       62     63    64     65    66    67    68     69    70     71     72    73
## 1: 983.6 1053.2 515.4 1013.4 371.6 859.4 313.1 956.60 920.2 1098.6 1323.6 361.5
## 2: 792.4  986.6 374.3  965.0 484.8 902.9 303.7 983.85 860.8  972.1 1243.7 310.6
##       74     75    77    78      79     80     81    82    83    84     86
## 1: 206.0 1092.5 296.8 810.8 1080.10 1029.9 1235.9 348.5 792.4 511.4 892.20
## 2: 135.9 1070.4 255.5 786.8  924.05  952.8 1073.3 326.7 799.8 449.1 764.05
##       87     88    89    90    91     93      94      95    96     97    98
## 1: 315.0 1310.0 219.8 235.4 827.7 1080.4  940.00 1053.40 357.4 848.20 128.7
## 2: 301.2 1323.8 185.5 224.5 916.1  998.1 1162.25 1277.45 394.0 917.35  95.0
##      99    100   101   102   103   104   105     106   107   108   109   110
## 1: 18.5 1027.0 876.6 782.4 196.3 817.0 928.9 1042.80 805.4 242.7 884.0 698.1
## 2: 14.8  925.8 866.0 986.4 255.1 881.9 923.7  799.85 868.7 401.4 828.3 761.4
##      111   112    113    114   115   116    118    119   120   121   122    123
## 1: 224.9 998.8  918.4 945.00 358.9 835.0 799.40 1074.4  98.7 391.7 864.4 1105.4
## 2: 272.6 974.8 1009.9 870.95 322.2 832.4 755.95  991.1 248.0 294.1 892.5 1353.0
##      124     125   126  127    128   129     130   131  132    133    134  135
## 1: 104.9 1039.80 331.6 43.9 1036.8 738.6 1273.80 267.4 52.3 1021.8 419.20 24.4
## 2: 200.9 1114.75 314.3 39.7  910.1 898.7 1233.95 233.4 24.2  916.1 432.65 33.3
##      136     137   138  139  140   141   142    143   144   145 146    147
## 1: 118.8 1034.40 822.4 36.5  8.5 272.8 457.0 344.30 716.0 388.0  12 755.60
## 2: 173.2  888.95 707.4 34.1 24.3 225.3 506.6 421.85 703.2 309.9  48 674.45
##      148   149   150   151    152    153     154   155    156    157  158  159
## 1: 692.3 418.0 428.0 155.0 1067.6 1092.0  929.80 924.6 1024.6 1045.4 28.0 26.8
## 2: 763.5 469.9 413.1 135.5  971.1 1019.7 1193.35 782.7 1017.8  904.1 29.8 14.7
##      160  161   162   163   164    165    166   167     168   169   170   171
## 1: 894.8 35.5 889.8 188.6 853.2 1457.0 1143.3 236.3 1075.80 217.9 523.8 302.2
## 2: 756.2 32.5 887.1 183.1 920.2 1206.6  876.6 206.6  922.35 238.2 332.6 394.2
##      172    173   174   175   176  177   178   179   180     181   182   183
## 1: 820.8 451.00 337.6 964.6 287.2  9.6 952.0 934.0 816.6 1379.90 388.4 870.2
## 2: 758.0 565.75 483.0 994.6 147.1 23.7 915.5 961.9 788.5 1040.75 461.9 707.4
##      184   185   186   187   188   189   190   191  192     194    195   196
## 1: 983.8 225.6 141.3 253.9 234.4 137.5 829.4 826.2 34.3 1111.10 227.50 876.2
## 2: 874.4 276.3 144.7 214.3 210.3 151.8 817.8 861.4 39.0  941.65 332.25 848.7
##      197  198    199   200    201   202    203  204   205   207    208   209
## 1: 363.2 20.3 1299.6 497.2 1107.2 628.2 1266.8 40.4 320.6 934.2 680.60 723.6
## 2: 294.4 10.7 1194.8 681.3 1118.9 478.3 1216.8 39.9 283.6 812.1 649.65 769.5
##       210   212    213   214   215    216    217   219   220   221   222   223
## 1: 1210.4 698.2 1098.4 182.4 375.2 1130.4 1329.8 897.8 244.1 956.3 944.6 930.4
## 2:  994.4 659.6  873.2 249.4 392.8 1042.6 1073.1 790.1 275.0 921.5 758.0 898.4
##     224   225     226   227   228   229    230    231    232   233   234    235
## 1: 19.2 865.0 1470.00 885.8 326.8 876.0  976.8 1102.5 1026.7 290.7 459.7 533.40
## 2: 44.5 833.4 1210.05 862.3 291.7 757.8 1010.7  959.7  727.9 285.9 318.1 422.25
##      236    237    238   239   240   241   242   243  244   245   246   247
## 1: 952.0 1448.4 1086.3 367.9 401.7 838.8 450.0 344.4 15.3 375.2 403.4 852.4
## 2: 970.8 1367.8 1241.7 332.6 345.2 727.0 446.6 410.3 32.8 341.0 453.6 781.0
```

```
##      248   249    250    251   253   254   255   256    257  258   259   260
## 1: 386.6 250.4 1161.5 439.40 391.0 156.4 254.1 256.2 1062.8 16.2 979.4 453.0
## 2: 373.4 298.5 1322.9 348.55 390.3 199.9 171.9 356.6  963.2 21.0 879.7 324.2
##       261   262  263   264   265   266  267    268   269     270   271    272
## 1: 1150.3 747.9 38.7 232.6 247.8 127.3  6.2 224.00 982.0  962.80 956.6 433.10
## 2: 1131.1 780.1 28.0 203.3 227.1 154.5 24.9 322.65 835.1 1003.75 683.9 372.85
```

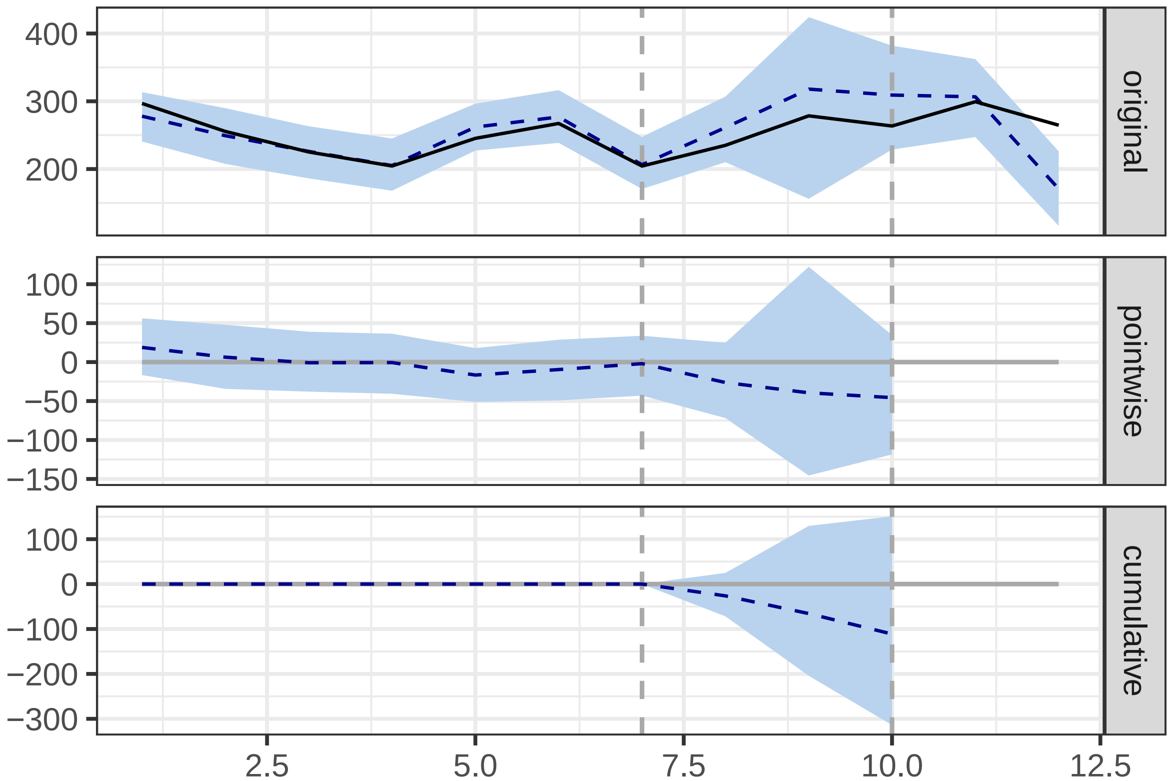
```r
start= 1
treatment= 8
end= 10
```

```r
#Sales of Store 77
# Compute the correlation matrix excluding the YEARMONTH column
correlation_matrix <- cor(presalesData[, -1, with = FALSE])

# Extract the correlations for store 77
store_77_correlations <- correlation_matrix[77, ]

# Sort the correlations and exclude the correlation of store 77 with itself
sorted_correlations <- sort(store_77_correlations[store_77_correlations != 1], decreasing = TRUE)

# Get the top 10 highest correlations
top_corr_with_77 <- head(sorted_correlations, 10)

# View the store numbers and their correlations
top_corr_with_77
```

```
##       248        93         5       229       234        26       151        83
## 0.9191805 0.8816236 0.8772682 0.8310234 0.8227086 0.7588167 0.7319340 0.7200666
##       116        10
## 0.7155475 0.7107457
```

```r
salesdata <- as.data.frame(salesdata)
# Assuming 'salesdata' is your dataframe
# Loop over each column name in the dataframe
names(salesdata) <- sapply(names(salesdata), function(name) {
  # Check if the name is numeric by removing all non-digit characters and seeing if the name is still t
  if (grepl("^[0-9]+$", name)) {
    return(paste0("store.", name))  # Add prefix if the name is numeric
  } else {
    return(name)  # Return the name as is if it's not numeric
  }
})
```

```r
final_sales_77 <- salesdata[,c("store.77","store.248", "store.93", "store.5", "store.229","store.234")]
final_sales_77 <- as.data.frame(final_sales_77)

impact <- CausalImpact(data = final_sales_77,
                       pre.period = c(start, treatment-1),
                       post.period = c(treatment, end))
plot(impact)
```




```r
summary(impact)
```

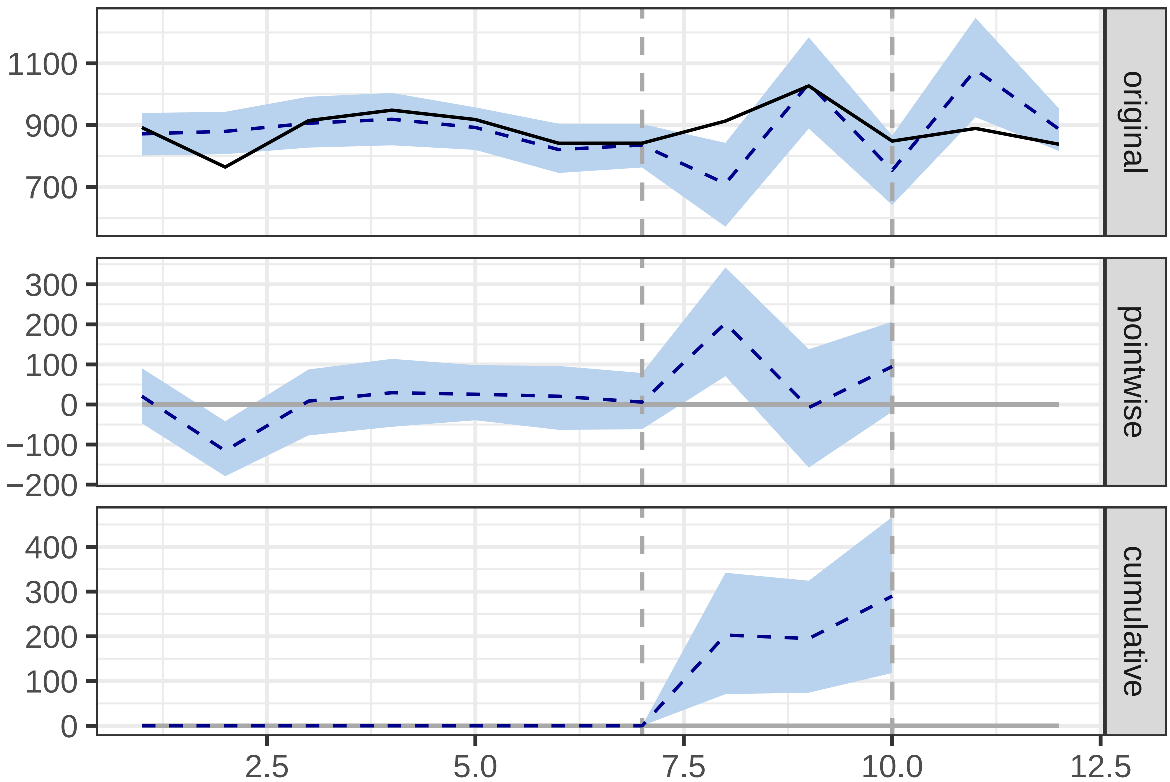
```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   259            777
## Prediction (s.d.)        296 (43)       888 (129)
## 95% CI                   [209, 364]     [626, 1091]
##
## Absolute effect (s.d.)   -37 (43)       -111 (129)
## 95% CI                   [-105, 50]     [-314, 151]
##
## Relative effect (s.d.)   -11% (15%)     -11% (15%)
## 95% CI                   [-29%, 24%]    [-29%, 24%]
##
## Posterior tail-area probability p:   0.20441
## Posterior prob. of a causal effect:  80%
##
## For more details, type: summary(impact, "report")
```

```r
#Sales of Store 86
# Compute the correlation matrix excluding the YEARMONTH column
correlation_matrix <- cor(presalesData[, -1, with = FALSE])

# Extract the correlations for store 86
store_86_correlations <- correlation_matrix[86, ]

# Sort the correlations and exclude the correlation of store 86 with itself
sorted_correlations <- sort(store_86_correlations[store_86_correlations != 1], decreasing = TRUE)
```

```r
# Get the top 10 highest correlations
top_corr_with_86 <- head(sorted_correlations, 10)

# View the store numbers and their correlations
top_corr_with_86
```

```
##       204       161        88       102       159       163         1        61
## 0.8343126 0.8054585 0.7766884 0.7566111 0.7302802 0.7116636 0.7111100 0.7051572
##         7       125
## 0.6856054 0.6515087
```

```r
final_sales_86 <- salesdata[,c("store.86","store.204", "store.161")]
final_sales_86 <- as.data.frame(final_sales_86)
set.seed(111)
impact <- CausalImpact(data = final_sales_86,
                       pre.period = c(start, treatment-1),
                       post.period = c(treatment, end))
plot(impact)
```




```r
summary(impact)
```

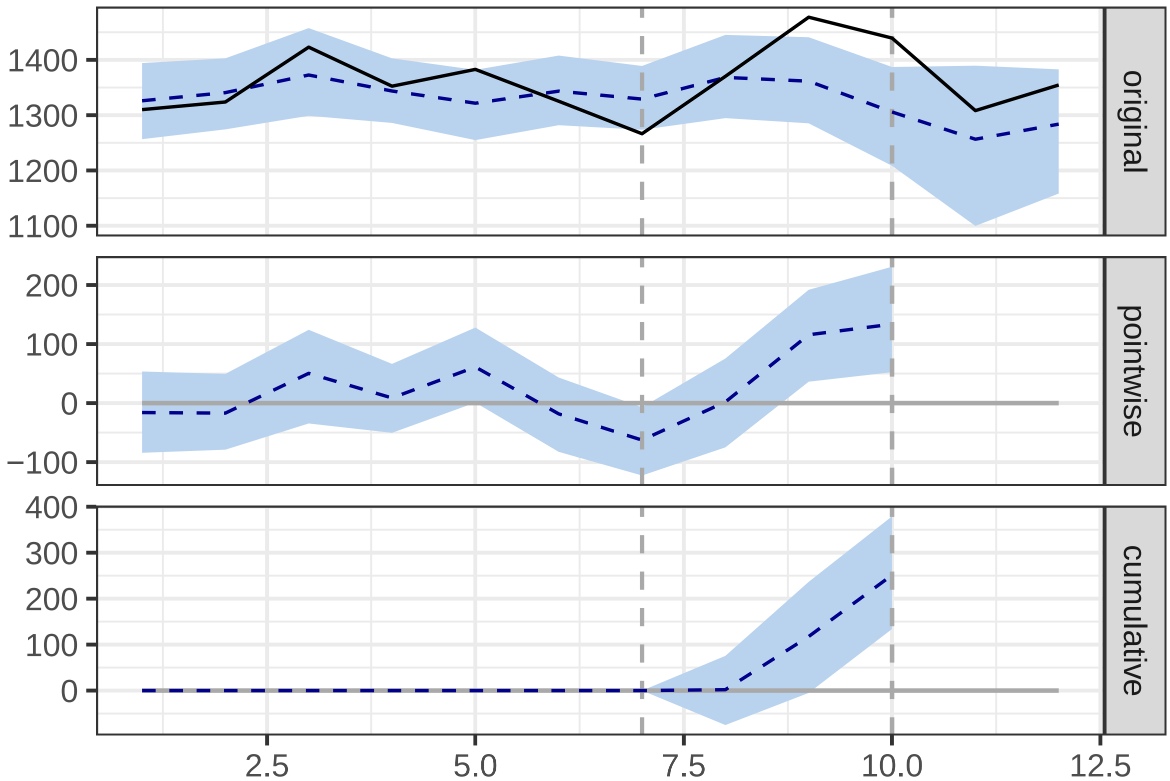
```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   929            2788
## Prediction (s.d.)        833 (29)       2498 (88)
## 95% CI                   [774, 890]     [2321, 2670]
##
## Absolute effect (s.d.)   97 (29)        290 (88)
## 95% CI                   [39, 156]      [118, 467]
```

```
##
## Relative effect (s.d.)   12% (4%)      12% (4%)
## 95% CI                   [4.4%, 20%]   [4.4%, 20%]
##
## Posterior tail-area probability p:   0.001
## Posterior prob. of a causal effect:  99.8996%
##
## For more details, type: summary(impact, "report")
```

```r
#Sales of Store 88
# Compute the correlation matrix excluding the YEARMONTH column
correlation_matrix <- cor(presalesData[, -1, with = FALSE])

# Extract the correlations for store 88
store_88_correlations <- correlation_matrix[88, ]

# Sort the correlations and exclude the correlation of store 88 with itself
sorted_correlations <- sort(store_88_correlations[store_88_correlations != 1], decreasing = TRUE)

# Get the top 10 highest correlations
top_corr_with_88 <- head(sorted_correlations, 10)

# View the store numbers and their correlations
top_corr_with_88
```

```
##         2        95       224       123       189       174       268       146
## 0.9933426 0.8956014 0.8155792 0.7527943 0.7453760 0.7345330 0.7276591 0.7162016
##       200       149
## 0.7128330 0.7122518
```

```r
final_sales_88 <- salesdata[,c("store.88","store.2", "store.95","store.224")]
final_sales_88 <- as.data.frame(final_sales_88)
set.seed(111)
impact <- CausalImpact(data = final_sales_88,
                       pre.period = c(start, treatment-1),
                       post.period = c(treatment, end))
plot(impact)
```




```
summary(impact)
```

```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   1429           4287
## Prediction (s.d.)        1345 (21)      4036 (62)
## 95% CI                   [1303, 1384]   [3908, 4153]
##
## Absolute effect (s.d.)   84 (21)        251 (62)
## 95% CI                   [45, 126]      [134, 379]
##
## Relative effect (s.d.)   6.3% (1.6%)    6.3% (1.6%)
## 95% CI                   [3.2%, 9.7%]   [3.2%, 9.7%]
##
## Posterior tail-area probability p:   0.001
## Posterior prob. of a causal effect:  99.9%
##
## For more details, type: summary(impact, "report")
```

```
# Get number of customers data
# Use dcast to reshape the data
ncustData <- dcast(fullobsdata,YEARMONTH ~ STORE_NBR, value.var = "nCustomers")
prencustData <- ncustData[YEARMONTH < 201902]
# View the reshaped dataset
head(ncustData,2)
```

```
##    YEARMONTH  1  2   3   4  5  6   7  8  9  10 12 13 14  15 16 17 18 19 20 21
## 1:    201807 49 39 112 128 93 48 100 40 46 100 47 97  8  84 32 56 52 84 56 42
## 2:    201808 42 39 112 123 97 44 109 46 52  96 44 96  8 105 35 40 48 86 40 56
```

```
##    22  23 24 25  26 27 28 29  30 32  33 34 35  36 37 38  39  40 41 42  43  45
## 1: 49  96 90 43 127 51 90 37  94 90 116 38 36 102 48 46  99 128 49  7 118 115
## 2: 36 105 90 34 113 56 97 41 111 93 109 44 40 111 64 50 100 123 43  8 109  84
##    46 47  48  49 50 51 52 53 54  55 56  57  58  59  60 61  62  63 64  65 66  67
## 1: 45 45 100 104 53 29  5 42 47 100 84 103 146 122 106  6 108 113 54 103 43  98
## 2: 44 55  98  87 53 41  7 48 42 105 81 102 106 100 100  6  96 111 42 100 51 105
##    68  69  70  71  72 73 74  75 77 78  79  80  81 82  83 84 86 87  88 89 90 91
## 1: 38 116 111 114 118 53 42  99 51 91 110 104 117 46  95 55 99 39 129 49 41 85
## 2: 38 118  95 107 120 46 31 110 47 91 101 103 110 43 100 54 94 37 131 45 38 93
##     93  94  95 96  97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112
## 1: 114 108 100 41  87 34  5 116  99  92  37  97  97 111  90  30 108  89  51 106
## 2: 112 118 129 40 116 25  3 109 103 104  42 104 112 102 100  44  89  89  44 109
##    113 114 115 116 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132
## 1: 103 103  49  91  92 107  28  57  99 107  28  98  35   9 116  83 121  56  11
## 2: 106 105  47 100 103 104  55  54 106 133  38 111  38   7 102  97 127  46   4
##    133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151
## 1: 110  47   5  27 121  93   9   2  43  50  38  84  47   3  92  94  45  48  34
## 2: 111  48   8  38 103  86   6   7  31  60  51  88  39   8  84  98  55  46  30
##    152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
## 1: 105 109  92 101 110 124   5   6 102   5 101  31  97 133 122  53 112  41  52
## 2: 115 107 124  91 122 103   8   2  94   8 108  31 112 109  95  37 102  37  41
##    171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
## 1:  33  89  49  39 100  51   3 107  99  93 130  44  96 105  41  31  39  35  35
## 2:  45  90  66  50 107  33   7 108 106 108 112  51  88 100  54  30  36  37  28
##    190 191 192 194 195 196 197 198 199 200 201 202 203 204 205 207 208 209 210
## 1:  98 100   8 109  44  93  48   4 122  56 119  65 115   6  49 104  83  88 114
## 2: 103 100   7 105  62  96  47   2 121  71 112  46 119   7  46  96  82  85 103
##    212 213 214 215 216 217 219 220 221 222 223 224 225 226 227 228 229 230 231
## 1:  84 113  38  40 119 117 100  47 106 107  97   3 100 135 102  37 100 113 111
## 2:  79 101  43  44 109  99  95  42 110  86 122   7  93 127  94  33  95 115  96
##    232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
## 1: 106  51  50  56 101 128 110  42  43  92  50  48   3  41  40  97  56  43 120
## 2:  91  48  36  46 108 135 124  37  39  92  48  55   6  41  46  96  60  54 135
##    251 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
## 1:  48  41  45  47  40 118   2 109  49 116  84   9  53  51  31   2  48 103  96
## 2:  37  42  44  33  54 112   3 106  40 119  80   6  40  51  37   5  50 109 122
##    271 272
## 1:  96  48
## 2:  84  44
```

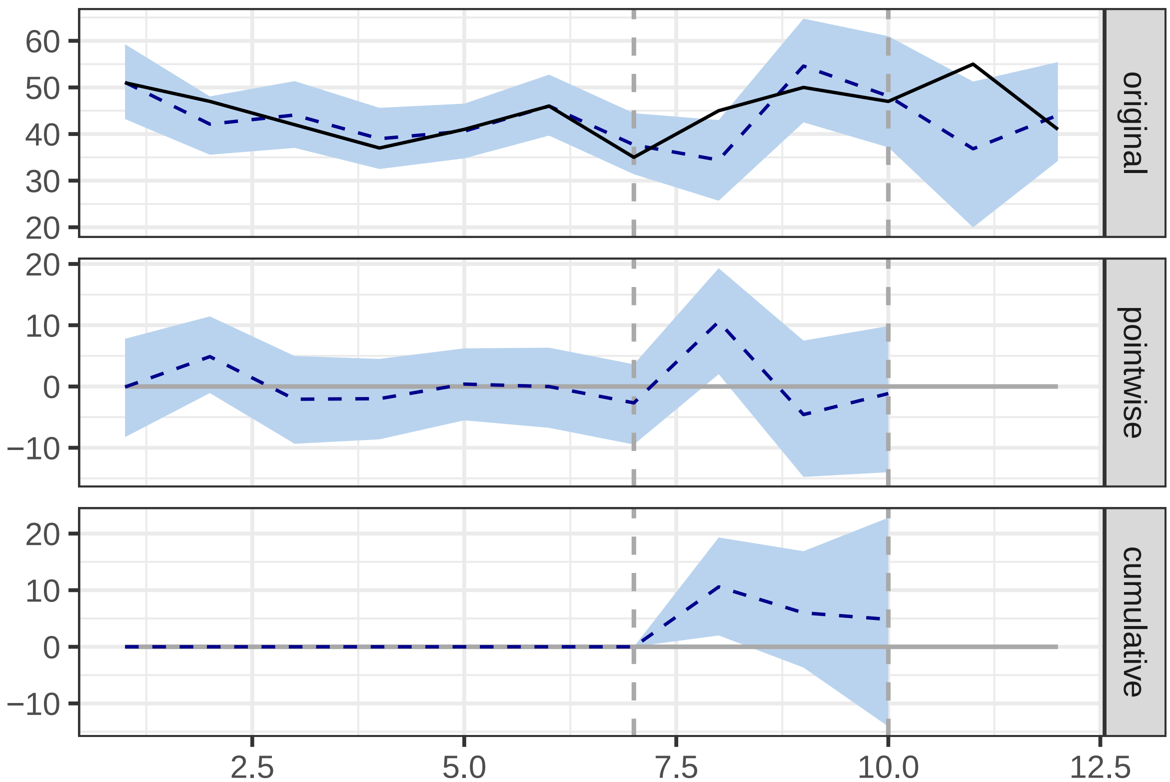
```r
# number of customers for store 77
# Compute the correlation matrix excluding the YEARMONTH column
correlation_matrix <- cor(prencustData[, -1, with = FALSE])

# Extract the correlations for store 77
store_77_correlations <- correlation_matrix[77, ]

# Sort the correlations and exclude the correlation of store 77 with itself
sorted_correlations <- sort(store_77_correlations[store_77_correlations != 1], decreasing = TRUE)

# Get the top 10 highest correlations
top_corr_with_77 <- head(sorted_correlations, 10)

# View the store numbers and their correlations
```

```
top_corr_with_77
```

```
##        40        93        10       234         5        26       151       264
## 0.9207690 0.9123574 0.8465923 0.8446967 0.8228706 0.8033945 0.7739576 0.7325663
##       248        62
## 0.7169252 0.7117749
```

```
ncustData <- as.data.frame(ncustData)
# Loop over each column name in the dataframe
names(ncustData) <- sapply(names(ncustData), function(name) {
  # Check if the name is numeric by removing all non-digit characters and seeing if the name is still t
  if (grepl("^[0-9]+$", name)) {
    return(paste0("store.", name))  # Add prefix if the name is numeric
  } else {
    return(name)  # Return the name as is if it's not numeric
  }
})
```

```
final_ncust_77 <- ncustData[,c("store.77","store.40", "store.93", "store.10", "store.234","store.5","sto
final_ncust_77 <- as.data.frame(final_ncust_77)
set.seed(111)
impact <- CausalImpact(data = final_ncust_77,
                       pre.period = c(start, treatment-1),
                       post.period = c(treatment, end))
plot(impact)
```




```
summary(impact)
```

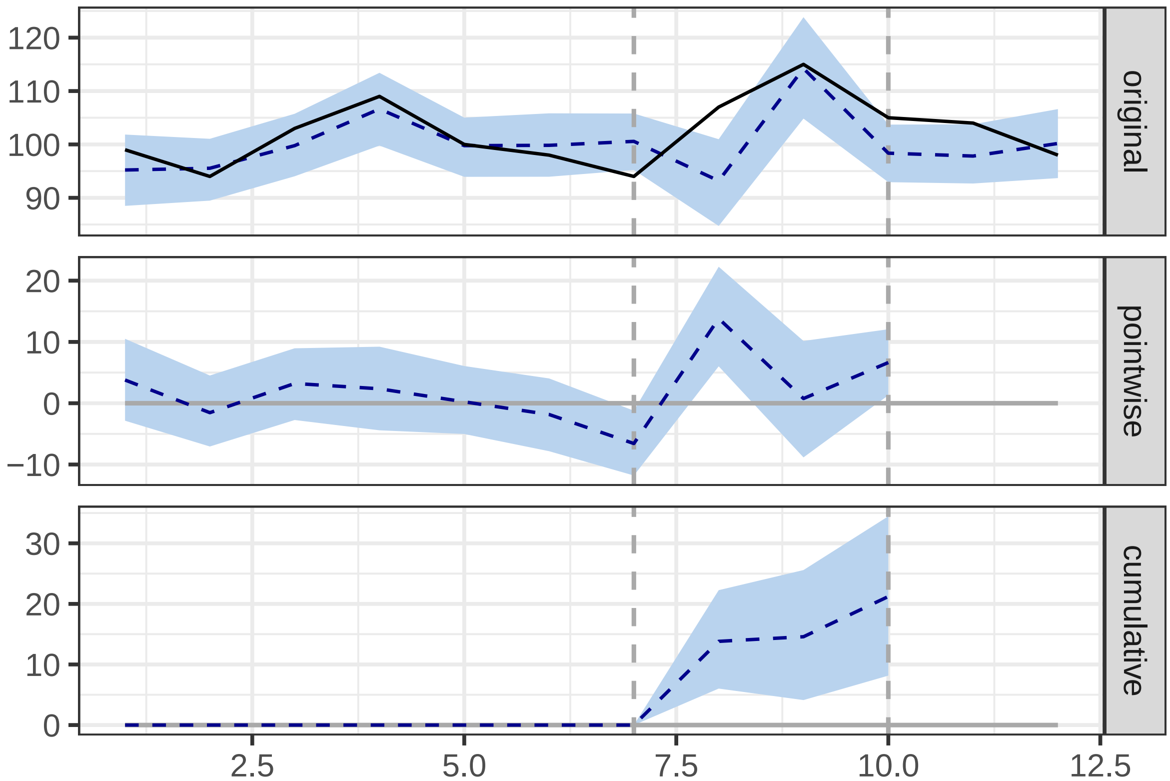
```
## Posterior inference {CausalImpact}
##
```

```
##                          Average       Cumulative
## Actual                   47            142
## Prediction (s.d.)        46 (3.4)      137 (10.1)
## 95% CI                   [40, 52]      [119, 156]
##
## Absolute effect (s.d.)   1.6 (3.4)     4.8 (10.1)
## 95% CI                   [-4.7, 7.6]   [-14.1, 22.8]
##
## Relative effect (s.d.)   4.2% (7.7%)   4.2% (7.7%)
## 95% CI                   [-9%, 19%]    [-9%, 19%]
##
## Posterior tail-area probability p:   0.32139
## Posterior prob. of a causal effect:  68%
##
## For more details, type: summary(impact, "report")
```

```r
# number of customers for store 86
# Compute the correlation matrix excluding the YEARMONTH column
correlation_matrix <- cor(prencustData[, -1, with = FALSE])

# Extract the correlations for store 86
store_86_correlations <- correlation_matrix[86, ]

# Sort the correlations and exclude the correlation of store 86 with itself
sorted_correlations <- sort(store_86_correlations[store_86_correlations != 1], decreasing = TRUE)

# Get the top 10 highest correlations
top_corr_with_86 <- head(sorted_correlations, 10)

# View the store numbers and their correlations
top_corr_with_86
```

```
##       125       161       163        61         7       238       253       152
## 0.8055618 0.8055053 0.7405134 0.7069515 0.6904089 0.6854470 0.6609828 0.6570667
##       171       188
## 0.6564792 0.6504383
```

```r
final_ncust_86 <- ncustData[,c("store.86","store.125", "store.161")]

final_ncust_86 <- as.data.frame(final_ncust_86)

set.seed(111)
impact <- CausalImpact(data = final_ncust_86,
                       pre.period = c(start, treatment-1),
                       post.period = c(treatment, end))
plot(impact)
```




```r
summary(impact)
```

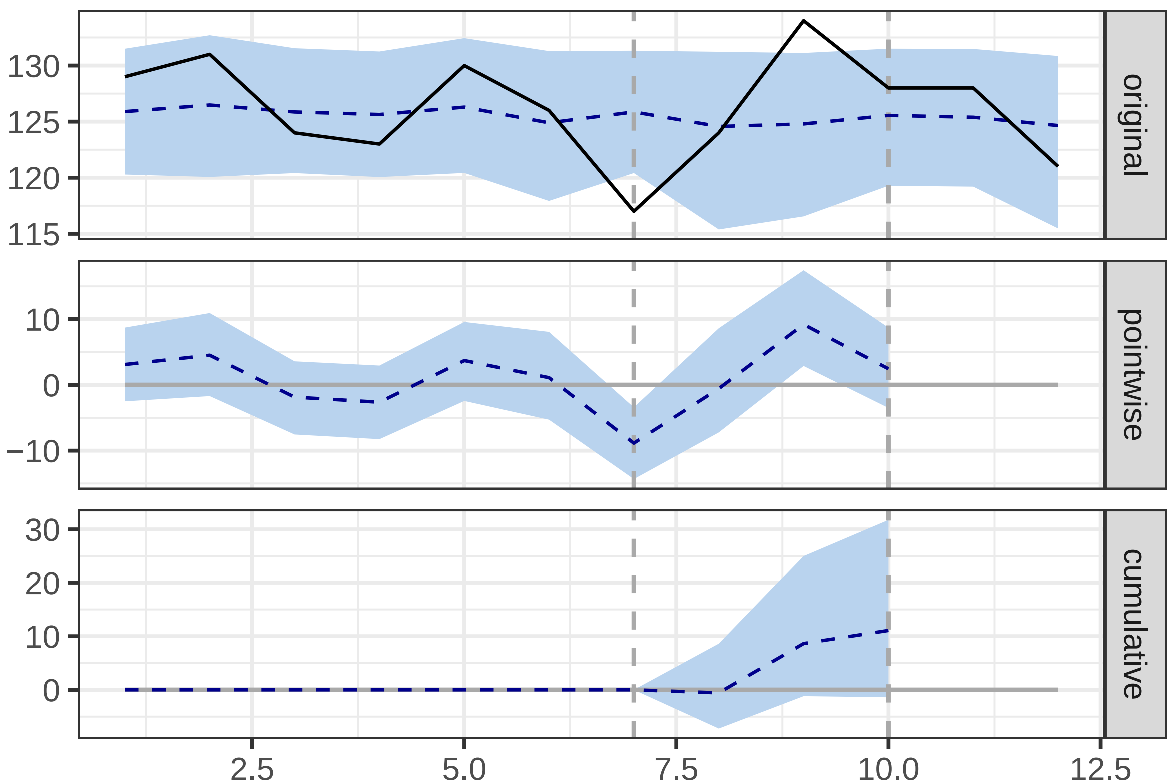
```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   109            327
## Prediction (s.d.)        102 (2.2)      306 (6.6)
## 95% CI                   [98, 106]      [293, 319]
##
## Absolute effect (s.d.)   7.1 (2.2)      21.2 (6.6)
## 95% CI                   [2.7, 11]      [8.2, 34]
##
## Relative effect (s.d.)   7% (2.3%)      7% (2.3%)
## 95% CI                   [2.6%, 12%]    [2.6%, 12%]
##
## Posterior tail-area probability p:   0.001
## Posterior prob. of a causal effect:  99.8996%
##
## For more details, type: summary(impact, "report")
```

```r
# number of customers for store 88
# Compute the correlation matrix excluding the YEARMONTH column
correlation_matrix <- cor(prencustData[, -1, with = FALSE])

# Extract the correlations for store 88
store_88_correlations <- correlation_matrix[88, ]

# Sort the correlations and exclude the correlation of store 88 with itself
sorted_correlations <- sort(store_88_correlations[store_88_correlations != 1], decreasing = TRUE)
```

```r
# Get the top 10 highest correlations
top_corr_with_88 <- head(sorted_correlations, 10)

# View the store numbers and their correlations
top_corr_with_88
```

```
##       200       182       185       142       146       149       270       101
## 0.8416134 0.8269231 0.8211540 0.8042994 0.7992266 0.7866581 0.7294448 0.7255287
##       108       107
## 0.6926709 0.6830894
```

```r
final_ncust_88 <- ncustData[,c("store.88","store.200", "store.182","store.185","store.142")]

final_ncust_88 <- as.data.frame(final_ncust_88)

set.seed(111)
impact <- CausalImpact(data = final_ncust_88,
                       pre.period = c(start, treatment-1),
                       post.period = c(treatment, end))
plot(impact)
```




```r
summary(impact)
```

```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   129            386
## Prediction (s.d.)        125 (2.6)      375 (7.9)
## 95% CI                   [118, 129]     [354, 387]
##
```

```
## Absolute effect (s.d.)   3.7 (2.6)      11.1 (7.9)
## 95% CI                   [-0.46, 11]    [-1.39, 32]
##
## Relative effect (s.d.)   3% (2.2%)      3% (2.2%)
## 95% CI                   [-0.36%, 9%]   [-0.36%, 9%]
##
## Posterior tail-area probability p:   0.049
## Posterior prob. of a causal effect:  95.1%
##
## For more details, type: summary(impact, "report")
```