Causal impact study

GB

February 5, 2017

getting data file and setting experiment start, end and rollout times

```
myfile<-file.choose()
startime<-as.Date('2016-04-01')
endtime<-as.Date('2016-06-01')
rolout<-as.Date('2016-09-01')
```

```
mydata<-read.csv(myfile, header = TRUE)
mydata$ga.date<-as.Date(mydata$ga.date)
sprintf('start date in file: %s', min(mydata$ga.date))
```

```
## [1] "start date in file: 2015-04-01"
```

```
sprintf('end date in file: %s', max(mydata$ga.date))
```

```
## [1] "end date in file: 2016-11-21"
```

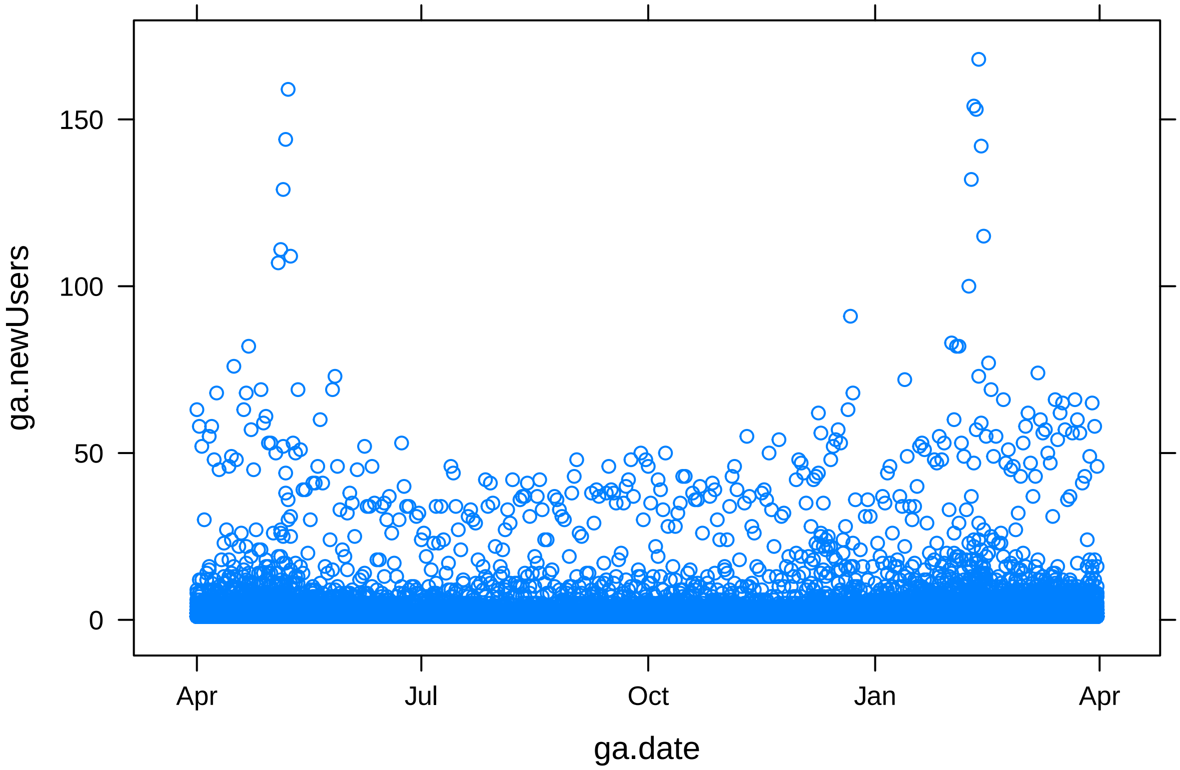
```
if (min(mydata$ga.date) >= startime) { print('Warning: no data prior to experiment start!')
  data_for_strata<-FALSE} else {data_for_strata<-TRUE}
data_for_strata
```

```
## [1] TRUE
```

```
library(CausalImpact)
library(dplyr)
library(lattice)
```

preparing data for stratas. taking only data from time period before the experiment start as described in etsy experiment

```
if (data_for_strata==TRUE) {
  prior_to_exp<-mydata[mydata$ga.date < startime,]
  xyplot(ga.newUsers~ga.date, prior_to_exp)
}
```



```
paths<-unique(mydata$ga.landingPagePath)    # selects unique landingPagePath's for the entire data file
sprintf('total different landing pages in the entire data file: %s', length(paths))
```

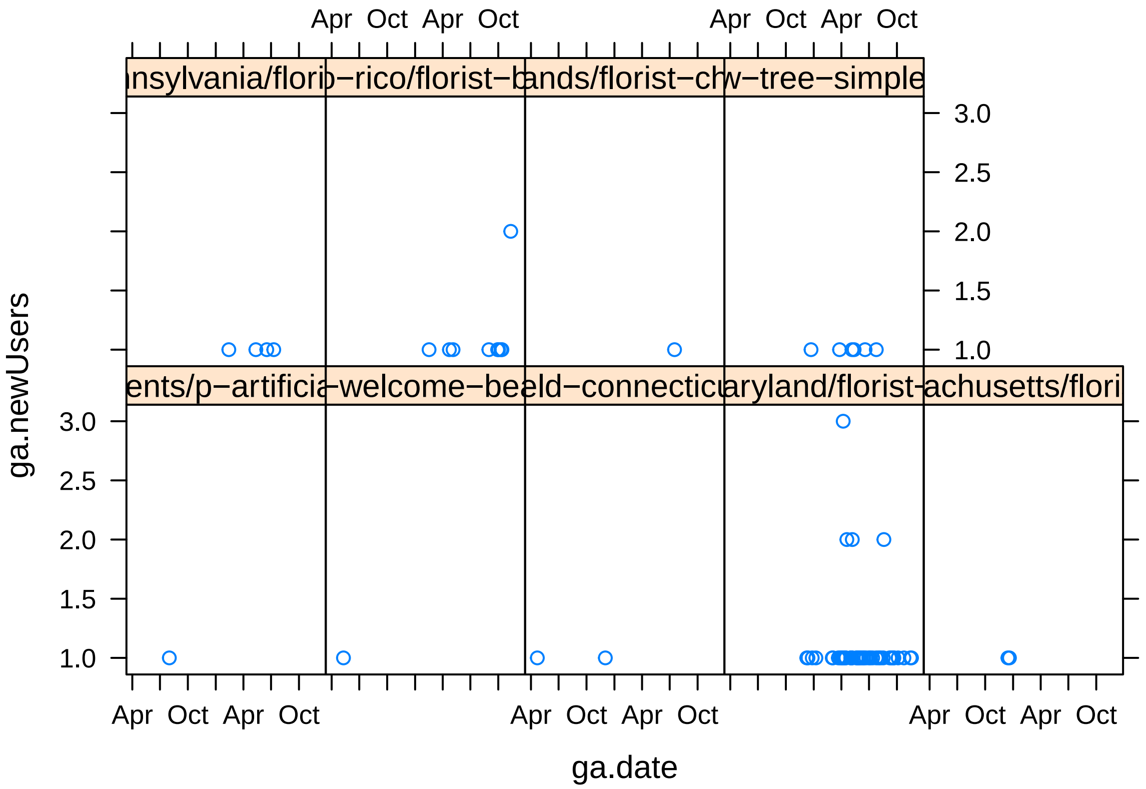
```
## [1] "total different landing pages in the entire data file: 7226"
```

```
paths2<-unique(prior_to_exp$ga.landingPagePath) #selects unique landingPagePath's before the experiment
sprintf('total different landing pages prior to experiment: %s', length(paths2))
```

```
## [1] "total different landing pages prior to experiment: 6103"
```

ploting some times series for the random NP=9 landing Pages, the number can be changed

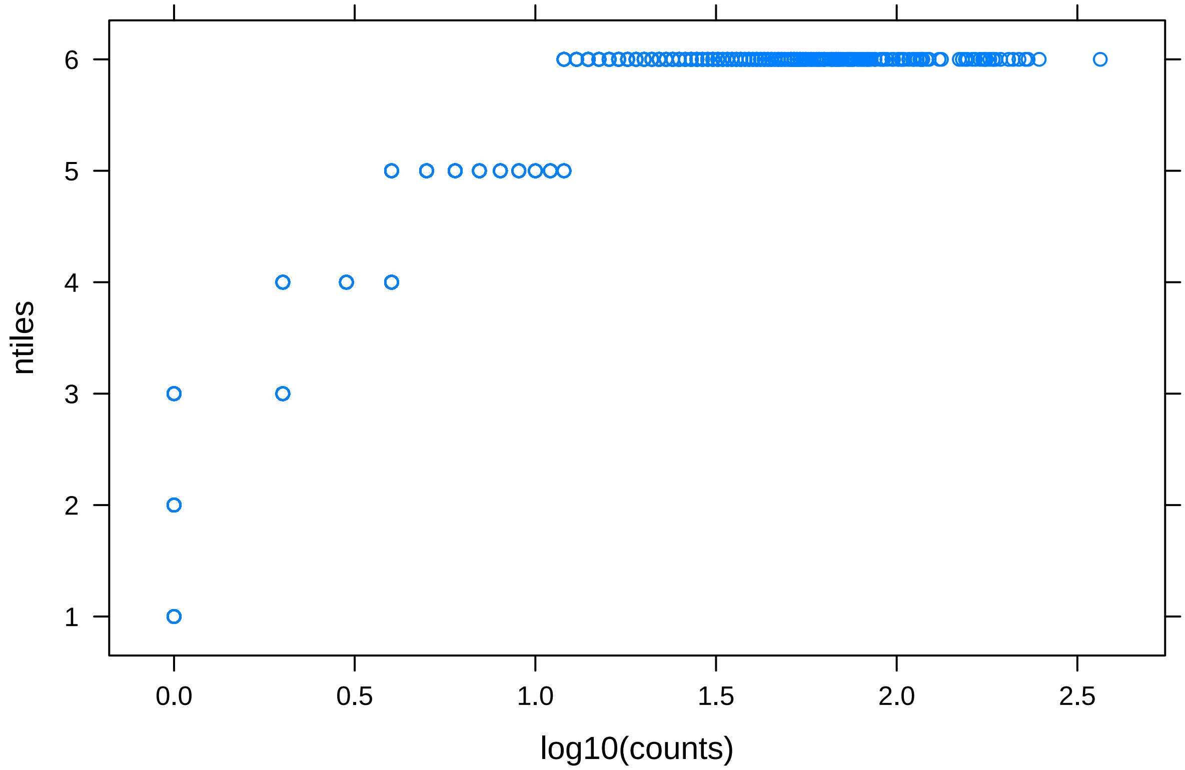
```
NP<-9
testpaths<-sample(length(paths), NP)
data2<-mydata[mydata$ga.landingPagePath %in% paths[testpaths], ]
xyplot(ga.newUsers~ga.date|ga.landingPagePath, data2)
```



```
rm(data2)
```

generating ntiles and plotting ntiles versus counts. counts here are total NewUsers per page. Number of ntiles is 6. can be changed.

```
kn=6 # numer of ntiles, can be changed
sumdata<-summarize(group_by(prior_to_exp, ga.landingPagePath), counts=n()) # getting total counts per l
sumdata<-sumdata[order(-sumdata$counts), ]   # ordering in descreasing order
sumdata$ntiles<-ntile(sumdata$counts, kn)  # creating ntiles column and assinging ntile number to each
xyplot(ntiles~log10(counts), sumdata)
```



repackaging data into different kg=7 groups. Number kg can be changed. number of unique landingPagePath's per ntile is printed out

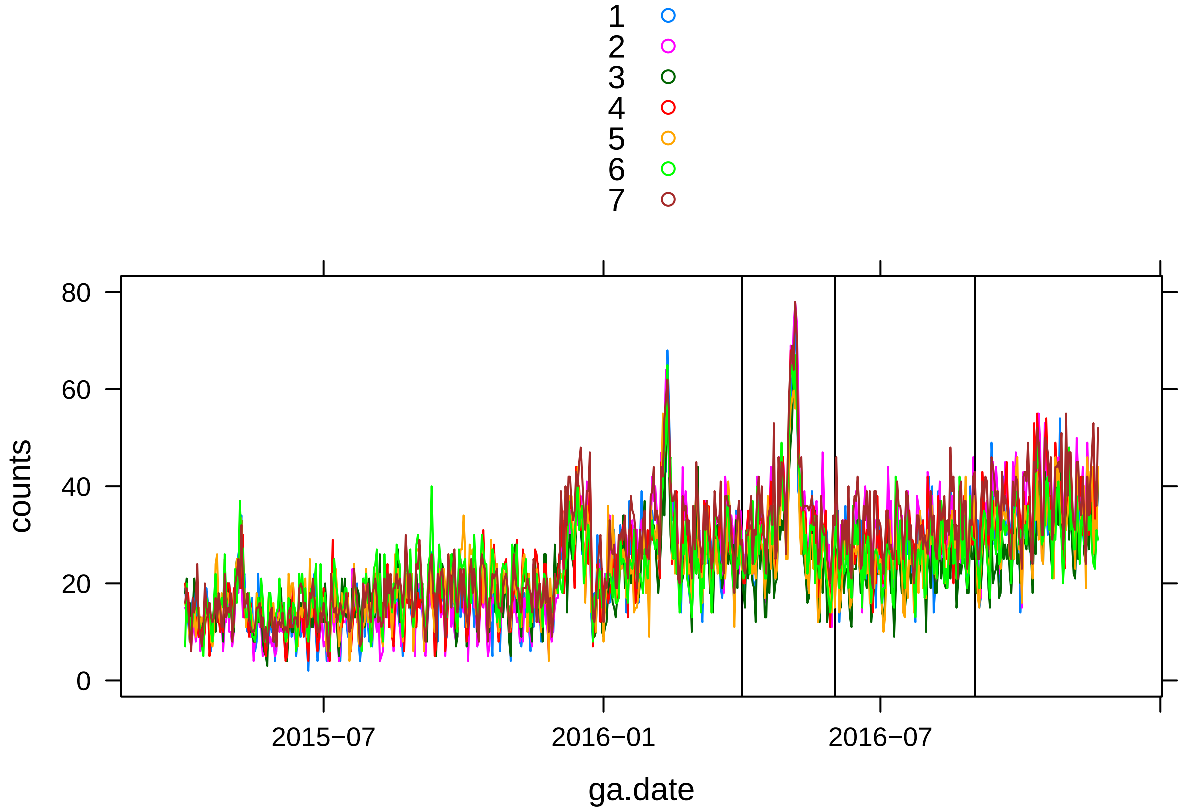
```
kg=7   # kg = to number of test groups
sumdata$group<-0   #creating group marker column
groupeddata<-data.frame()  # new data frame where results will be
for (i in 1:kn) {   # looping through ntiles
  sampledsumdata<-filter(sumdata, ntiles==i)   # selecting data for specific ntile only
  set.seed(12345)      #
  sampledsumdata2<-sampledsumdata[sample(nrow(sampledsumdata)),]     # reshufling rows randomly in that
  print(length(sampledsumdata$ga.landingPagePath))  # prints the number of paths per ntile
  sampledsumdata2$group<-rep_len(1:kg, length(sampledsumdata2$counts))     # assign group to randomized
  groupeddata<-rbind(groupeddata, sampledsumdata2)  # combine into final table
  rm(sampledsumdata, sampledsumdata2)  # remove extra variables
  }
```

```
## [1] 1018
## [1] 1017
## [1] 1017
## [1] 1017
## [1] 1017
## [1] 1017
```

here groupeddata contains landingPagePath, total counts per that path, ntile and group for that landingPagePath. this table is used to map landingPagePath to group in the original data file

```
x<-groupeddata[match(mydata$ga.landingPagePath,groupeddata$ga.landingPagePath),4,drop=F]
mydata$group<-as.factor(x$group)
rm(x)
```

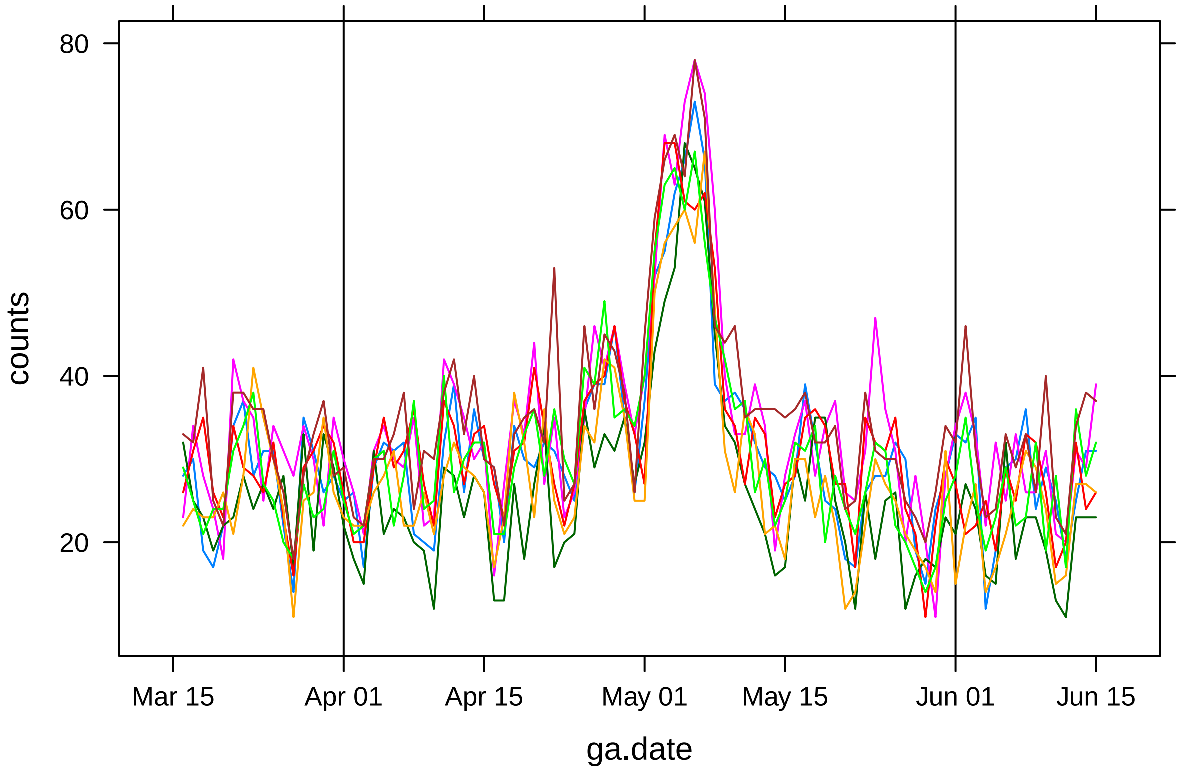
```
mydata_na<-mydata[!is.na(mydata$group),]    #removes rows with NA in group column, puts result into myda
my.lines<-c(startime, endtime, rolout)
groupmydata<-summarize(group_by(mydata_na, group, ga.date), counts=n()) # agregates per group sum withi
xyplot(counts~ga.date, groups=group, groupmydata, type='l',auto.key=T, panel = panel.superpose,
       panel.groups = function(..., group.number) {
           panel.abline(v = my.lines[group.number])
           panel.xyplot(...)})
```



plotting for specific time window, adjust values below:

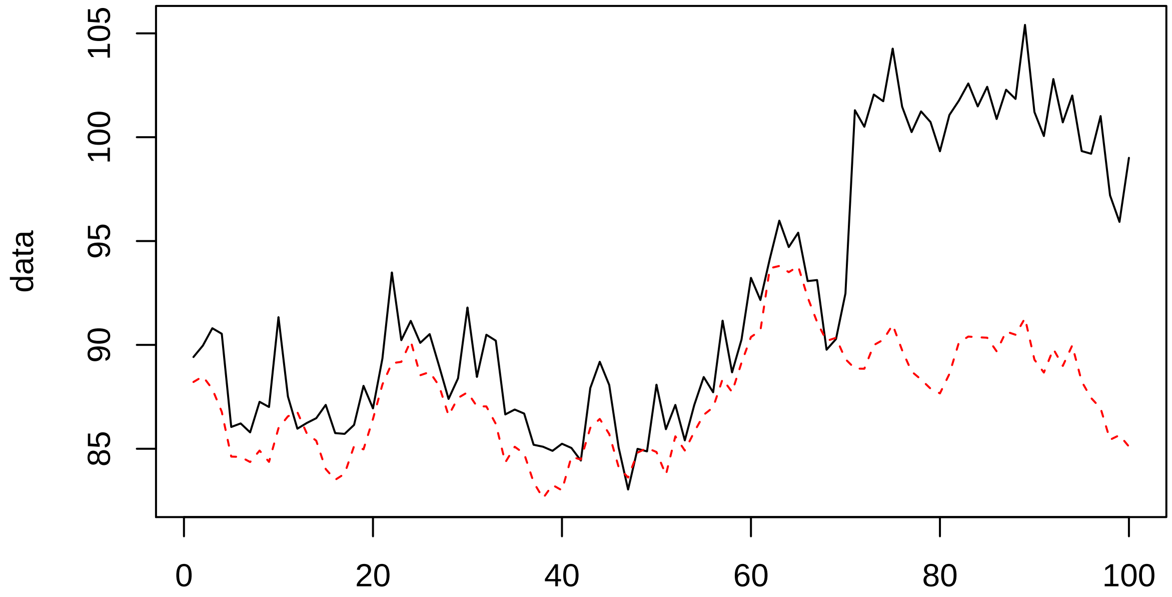
```
st<-as.Date('2016-03-15')
en<-as.Date('2016-06-15')
groupmydata2<-groupmydata[groupmydata$ga.date<=en & groupmydata$ga.date>st, ]
xyplot(counts~ga.date, groups=group, groupmydata2, type='l', panel = panel.superpose,
       panel.groups = function(..., group.number) {
           panel.abline(v = my.lines[group.number])
```

```
            panel.xyplot(...) })
```



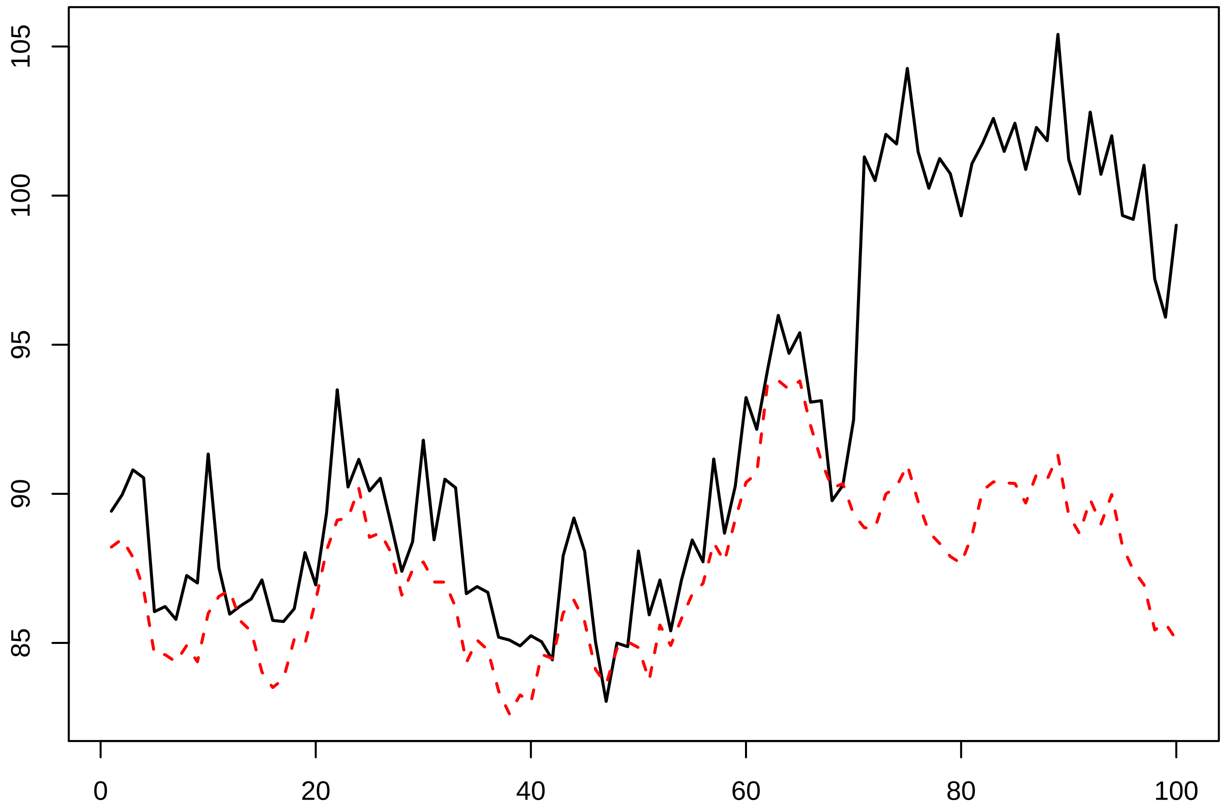
below is testing the package with synthetic data from the tutorial. this was done to assure that all dependences (supporting sub-packages) are in working condition

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



```
#par(cex = 0.85, oma = c(0, 0, 0, 0), mar = c(3, 2, 1, 1))
#matplot(data, type = "l", lwd = 1.5)
```

```
par(cex = 0.85, oma = c(0, 0, 0, 0), mar = c(3, 2, 1, 1))
matplot(data, type = "l", lwd = 1.5)
```



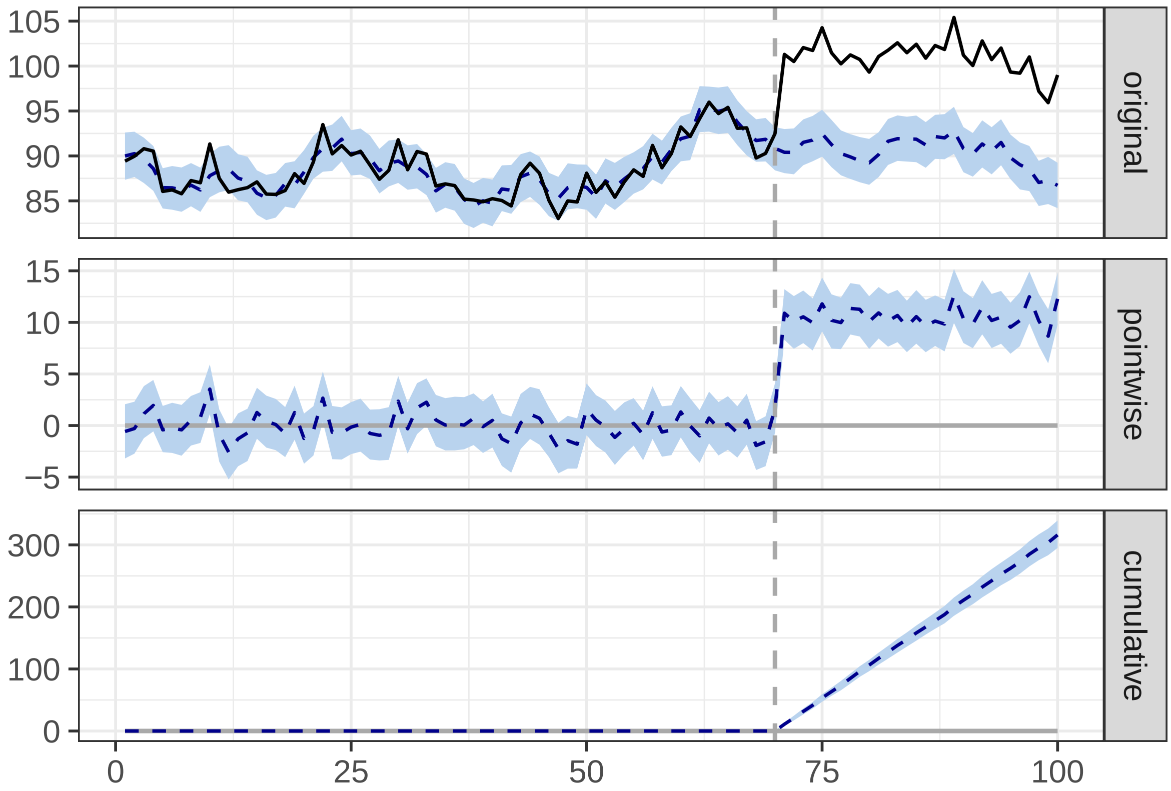
```
pre.period <- c(1, 70)
post.period <- c(71, 100)
```

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

```
plot(impact)
```

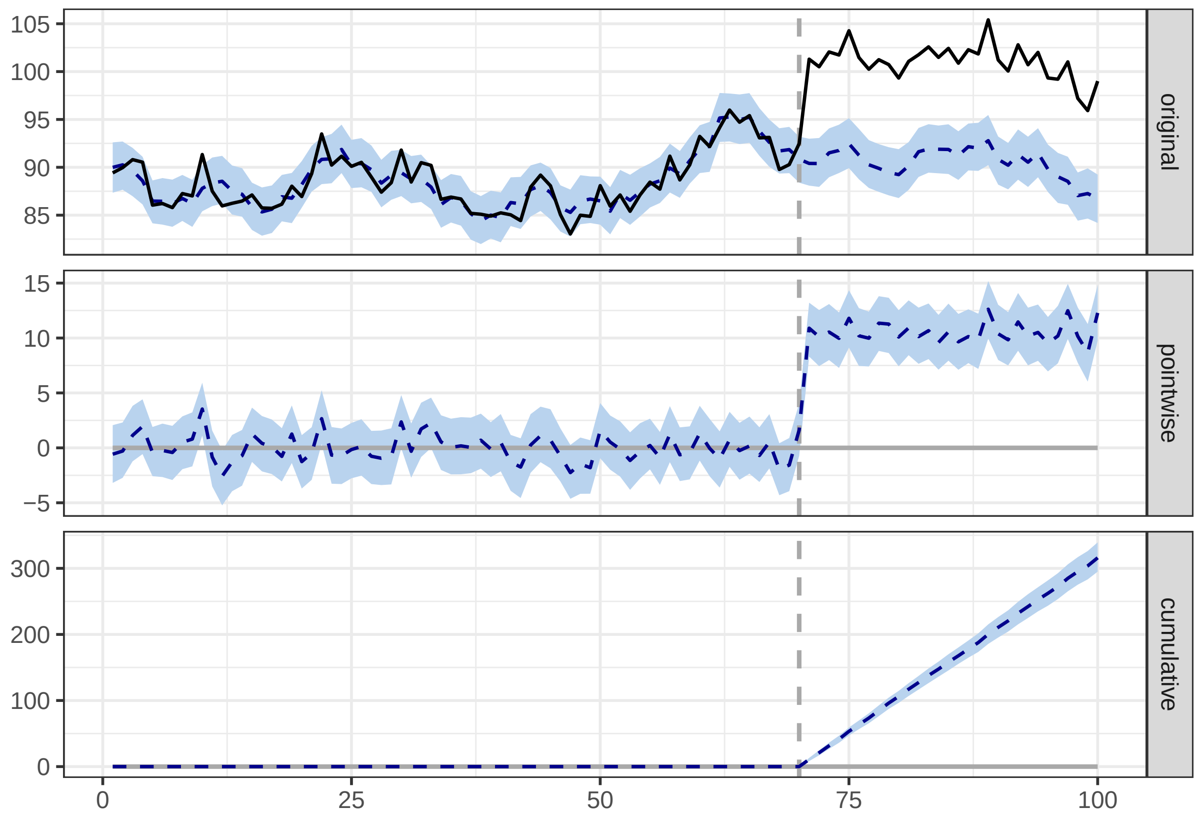
```
## Warning: Removed 100 rows containing missing values (geom_path).

## Warning: Removed 200 rows containing missing values (geom_path).
```



```
library(ggplot2)
```

```
q <- plot(impact) + theme_bw(base_size = 11)
suppressWarnings(plot(q))
```



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

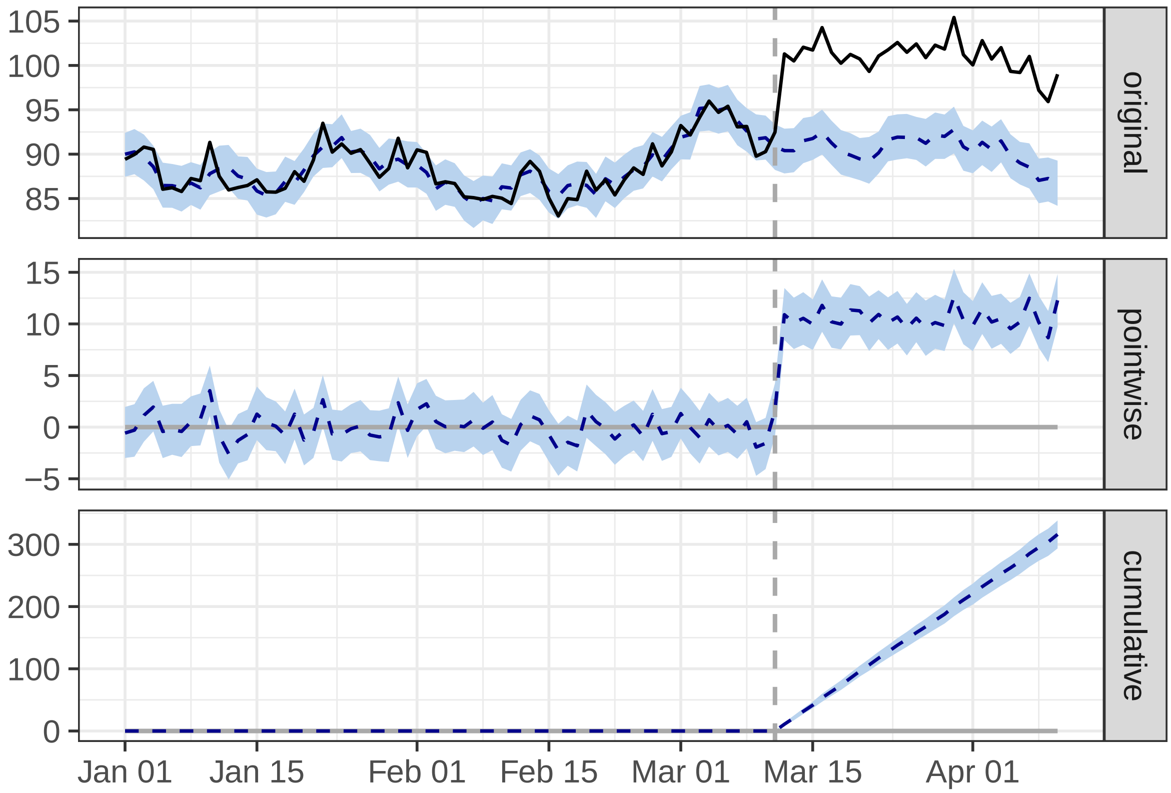
```
##                     y       x1
## 2014-01-01 89.41626 88.21513
## 2014-01-02 89.96716 88.48415
## 2014-01-03 90.80304 87.87684
## 2014-01-04 90.53689 86.77954
## 2014-01-05 86.04914 84.62243
## 2014-01-06 86.21926 84.60650
```

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

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

```
## Warning: Removed 100 rows containing missing values (geom_path).
```

```
## Warning: Removed 200 rows containing missing values (geom_path).
```


105
100
95
90
85
original
15
10
5
0
−5
pointwise
300
200
100
0
cumulative
Jan 01
Jan 15
Feb 01
Feb 15
Mar 01
Mar 15
Apr 01