In [7]

```r
# Load necessary libraries
library(Synth)
library(tidyverse)

if (!requireNamespace("remotes", quietly = TRUE)) {
  install.packages("remotes")
}
# Install the Synth package from the CRAN archive
remotes::install_version("Synth", version = "1.1-5")
```

Run Code

Out [7]

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

✔ ggplot2 3.3.2     ✔ purrr   0.3.4
✔ tibble  3.0.3     ✔ dplyr   1.0.2
✔ tidyr   1.1.1     ✔ stringr 1.4.0
✔ readr   1.3.1     ✔ forcats 0.5.0

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()

Downloading package from url: https://cran.r-project.org/src/contrib/Archive/Synth/Synth_1.1-5.tar.gz

optimx  (2020-2.2 -> 2022-4.30) [CRAN]
kernlab (0.9-29   -> 0.9-32   ) [CRAN]
Installing 2 packages: optimx, kernlab

Updating HTML index of packages in '.Library'

Making 'packages.html' ...
 done
```

+ New Code   + New Text

In [8]

```r
# Read the data
data_url <- "https://docs.google.com/spreadsheets/d/1bnM0C8HCAf1J85onn9LGLgIXD5eGT_Z1ep-IYVKxdS4/export?format=csv"
```

```r

# Convert the Time variable to a numeric variable
dataset$Time <- as.numeric(as.character(dataset$Year))

# Convert the City variable to a numeric variable
dataset$CityID <- as.numeric(factor(dataset$City, levels = unique(dataset$City)))

```

Run Code   + New Code   + New Text

In [9]

```r
dataprep_out <- dataprep(
  foo = dataset,
  predictors = c("GDP", "LabourProductivity", "ForeignBornShare"),
  predictors.op = "mean",
  dependent = "EmploymentRate",
  unit.variable = "CityID",
  time.variable = "Year",
  treatment.identifier = 1,
  controls.identifier = 2:5,
  time.predictors.prior = c(2007:2011),
  time.optimize.ssr = c(2007:2011),
  time.plot = c(2007:2017),
)

# Run the synthetic control estimation
synth_out <- synth(dataprep_out)

# Show the weights of the donor pool in tabular form
synth.tables <- synth.tab(dataprep.res = dataprep_out,
                          synth.res = synth_out)
synth.tables$tab.w[1:4, ]
```

Run Code

Out [9]

```
X1, X0, Z1, Z0 all come directly from dataprep object.


****************
 searching for synthetic control unit
```

```
****************
****************
****************


MSPE (LOSS V): 1.477781

solution.v:
 0.4568457 0.05463308 0.4885212

solution.w:
 4.44e-07 1.0077e-06 0.8225722 0.1774263
```

A data.frame: 4 × 3

	w.weights	unit.names	unit.numbers
	<dbl>	<dbl>	<dbl>
2	0.000	2	2
3	0.000	3	3
4	0.823	4	4
5	0.177	5	5

+ New Code

+ New Text

In [10]

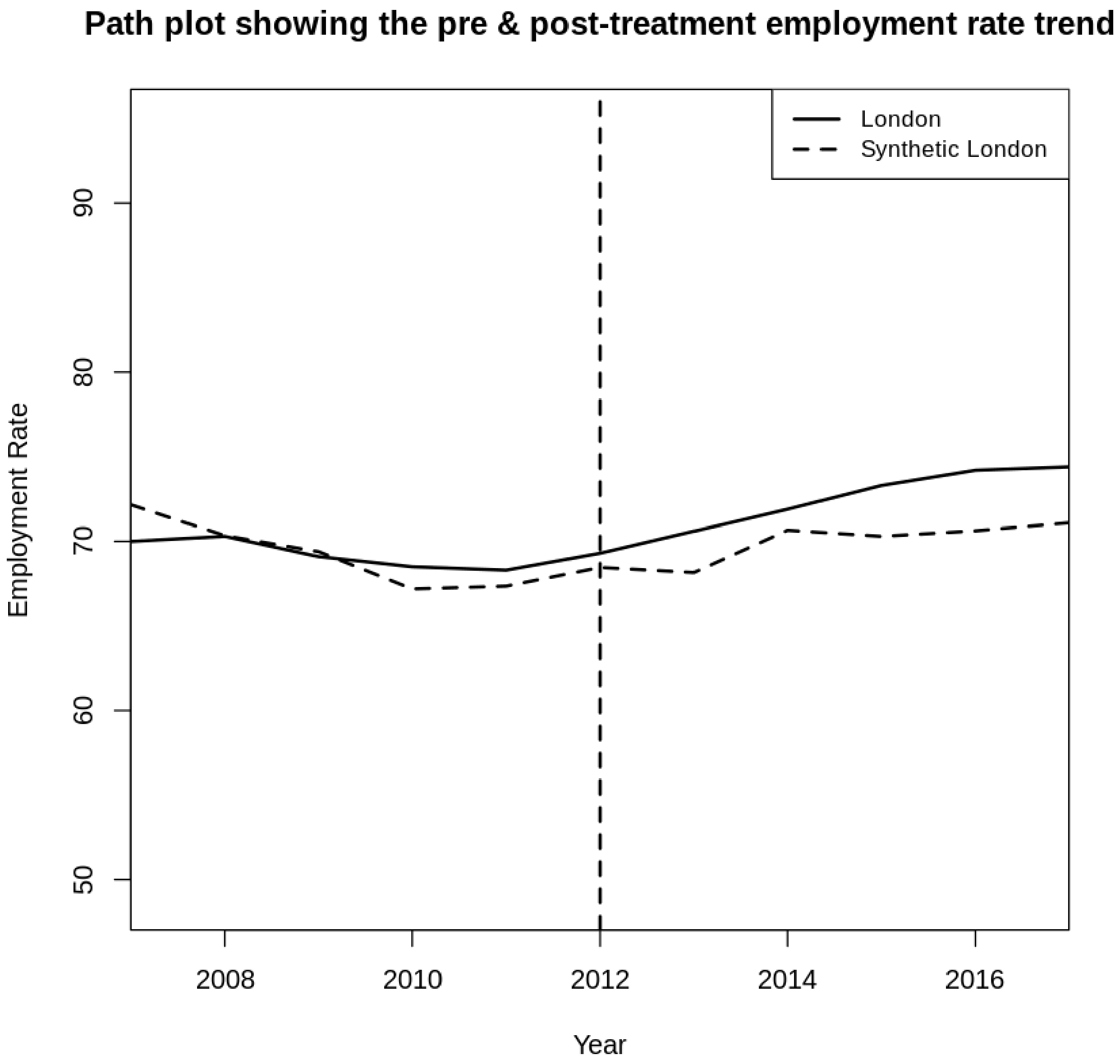
```r
# Generate the results and path plot to show the pre- and post-treatment trend for the outcome variable for both the treated unit and the synthetic control unit
path.plot(synth_out, dataprep_out,
          Ylab = "Employment Rate",
          Xlab = "Year",
          Main = "Path plot showing the pre & post-treatment employment rate trend",
          Legend = c("London","Synthetic London"))
abline(v = 2012, lty = 2, lwd = 2)

# Store the output in a file
sink("results.txt")
summary(synth_out)
sink()
```

Run Code

Out [10]

```
                    Length Class      Mode
```

```
solution.w 4      -none-     numeric
loss.v     1      -none-     numeric
loss.w     1      -none-     numeric
custom.v   0      -none-     NULL
rgV.optim  3      -none-     list
```



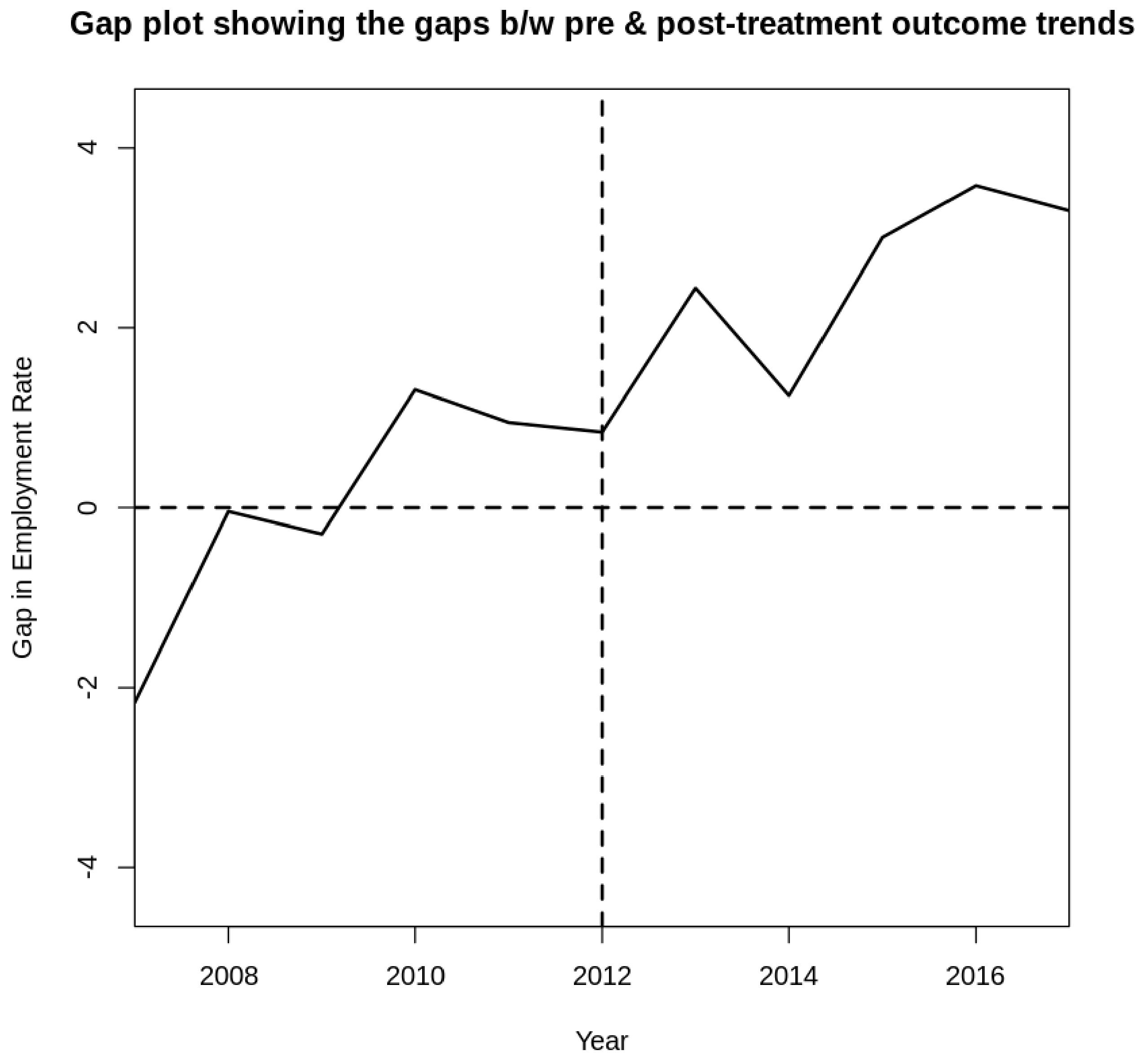
+ New Code   + New Text

In [11]

```
# Generate the gap plot to visualize the divergence between the potential outcome of the treated unit and the synthetic unit
gaps.plot(synth_out, dataprep_out,
          Ylab = "Gap in Employment Rate",
          Xlab = "Year",
          Main = "Gap plot showing the gaps b/w pre & post-treatment outcome trends")
abline(v = 2012, lty = 2, lwd = 2)
```

Run Code

Out [11]



+ New Code

+ New Text

In [12]

```r
# Calculate the gaps between the treated unit and its synthetic control unit (i.e. the treatment effect)
gaps <- dataprep_out$Y1plot - (dataprep_out$Y0plot %*% synth_out$solution.w)
gaps

# Calculate MSPE for the pre-treatment and post-treatment period
mspepre <- mean((gaps[1:5, 1])^2)
mspepost <- mean((gaps[6:11, 1])^2)
msperatio = mspepost/mspepre

#printing out the values
```

R (1GB RAM) | Edit   Run All Cells   Kernel Ready |

```
mspepre
mspepost
```

Run Code

Out [12]

A matrix: 11 × 1 of type dbl

	1
2007	-2.16455310
2008	-0.04198044
2009	-0.29358280
2010	1.31124418
2011	0.94673047
2012	0.84027498
2013	2.43864882
2014	1.24510743
2015	3.00316362
2016	3.57896472
2017	3.30158150

```
4.61635756325339
1.47778064806323
6.82196387151621
```

+ New Code + New Text

In [13]

```
# In-Time Placebo Test
dataprep_out1 <- dataprep(
  foo = dataset,
  predictors = c("GDP", "LabourProductivity", "ForeignBornShare"),
  predictors.op = "mean",
  dependent = "EmploymentRate",
  unit.variable = "CityID",
  time.variable = "Year",
  treatment.identifier = 1,
  controls.identifier = 2:5,
  time.predictors.prior = c(2007:2011),
```

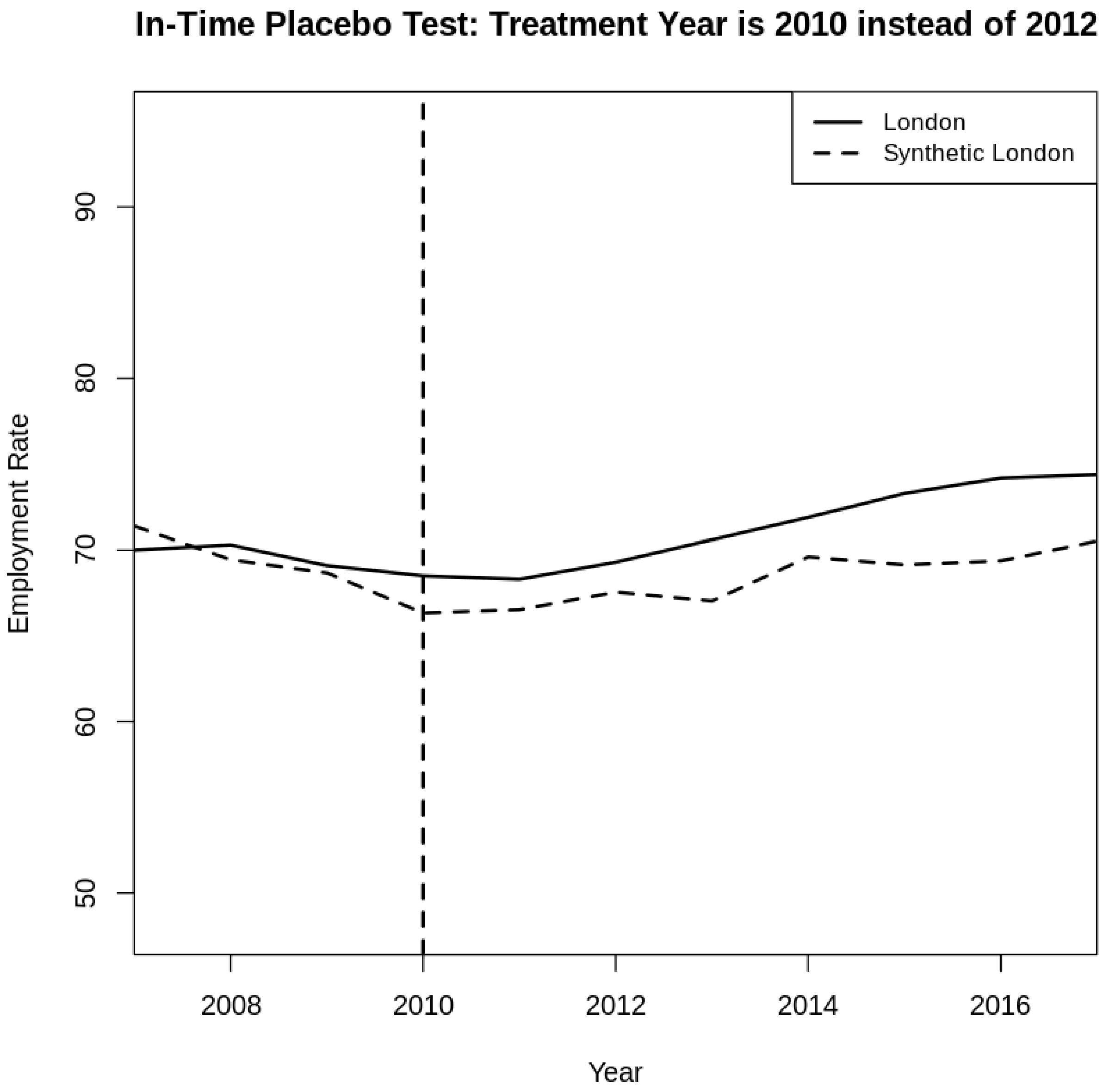
```r
  time.optimize.ssr = c(2007:2009), # pretending that the treatment year is 2010
  time.plot = c(2007:2017)
)

# Run the synthetic control estimation
synth_out1 <- synth(dataprep_out1)

# Generate the results and path plot
path.plot(synth_out1, dataprep_out1,
          Ylab = "Employment Rate",
          Xlab = "Year",
          Main = "In-Time Placebo Test: Treatment Year is 2010 instead of 2012",
          Legend = c("London","Synthetic London"))
abline(v = 2010, lty = 2, lwd = 2)

# Store the output in a file
sink("results.txt")
summary(synth_out1)
sink()


```

Run Code

Out [13]

```
X1, X0, Z1, Z0 all come directly from dataprep object.


****************
 searching for synthetic control unit


****************
****************
****************

MSPE (LOSS V): 0.9529391

solution.v:
 0.1768387 0.2563905 0.5667708

solution.w:
 1.73e-08 6.75e-08 0.9508421 0.0491578


           Length Class      Mode
```

```
solution.w 4      -none-     numeric
loss.v     1      -none-     numeric
loss.w     1      -none-     numeric
custom.v   0      -none-     NULL
rgV.optim  3      -none-     list
```



+ New Code

+ New Text

In [14]

```
#In-Space Placebo Test
dataprep_out2 <- dataprep(
  foo = dataset,
  predictors = c("GDP", "LabourProductivity", "ForeignBornShare"),
  predictors.op = "mean",
  dependent = "EmploymentRate",
  unit.variable = "CityID",
  time.variable = "Year",
  treatment.identifier = 2, # pretending that the treated unit is unit 2
(Liverpool)
```

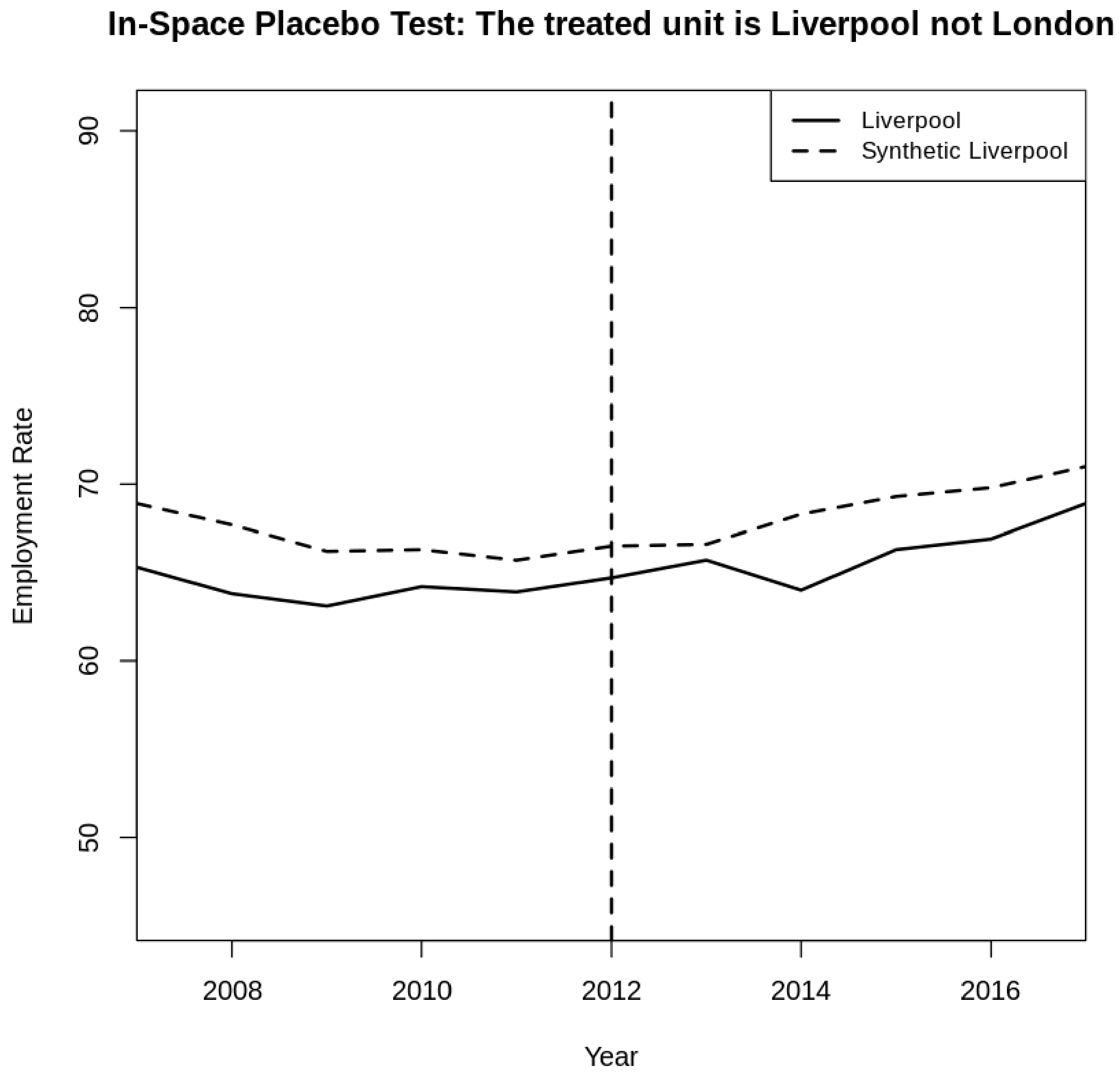
```
  controls.identifier = c(1, 3:5),
  time.predictors.prior = c(2007:2011),
  time.optimize.ssr = c(2007:2011),
  time.plot = c(2007:2017),
)

# Run the synthetic control estimation
synth_out2 <- synth(dataprep_out2)

# Generate the results and path plot
path.plot(synth_out2, dataprep_out2,
         Ylab = "Employment Rate",
         Xlab = "Year",
         Main = "In-Space Placebo Test: The treated unit is Liverpool not London",
         Legend = c("Liverpool","Synthetic Liverpool"))
abline(v = 2012, lty = 2, lwd = 2)

# Store the output in a file
sink("results.txt")
summary(synth_out2)
sink()
```

Run Code

Out [14]

```
X1, X0, Z1, Z0 all come directly from dataprep object.


****************
 searching for synthetic control unit


****************
****************
****************

MSPE (LOSS V): 9.086007

solution.v:
 0.5252717 0.3112842 0.1634441

solution.w:
 6e-10 0.9999991 8.77e-07 0


           Length Class      Mode
```

```
solution.w 4      -none-     numeric
loss.v     1      -none-     numeric
loss.w     1      -none-     numeric
custom.v   0      -none-     NULL
rgV.optim  3      -none-     list
```

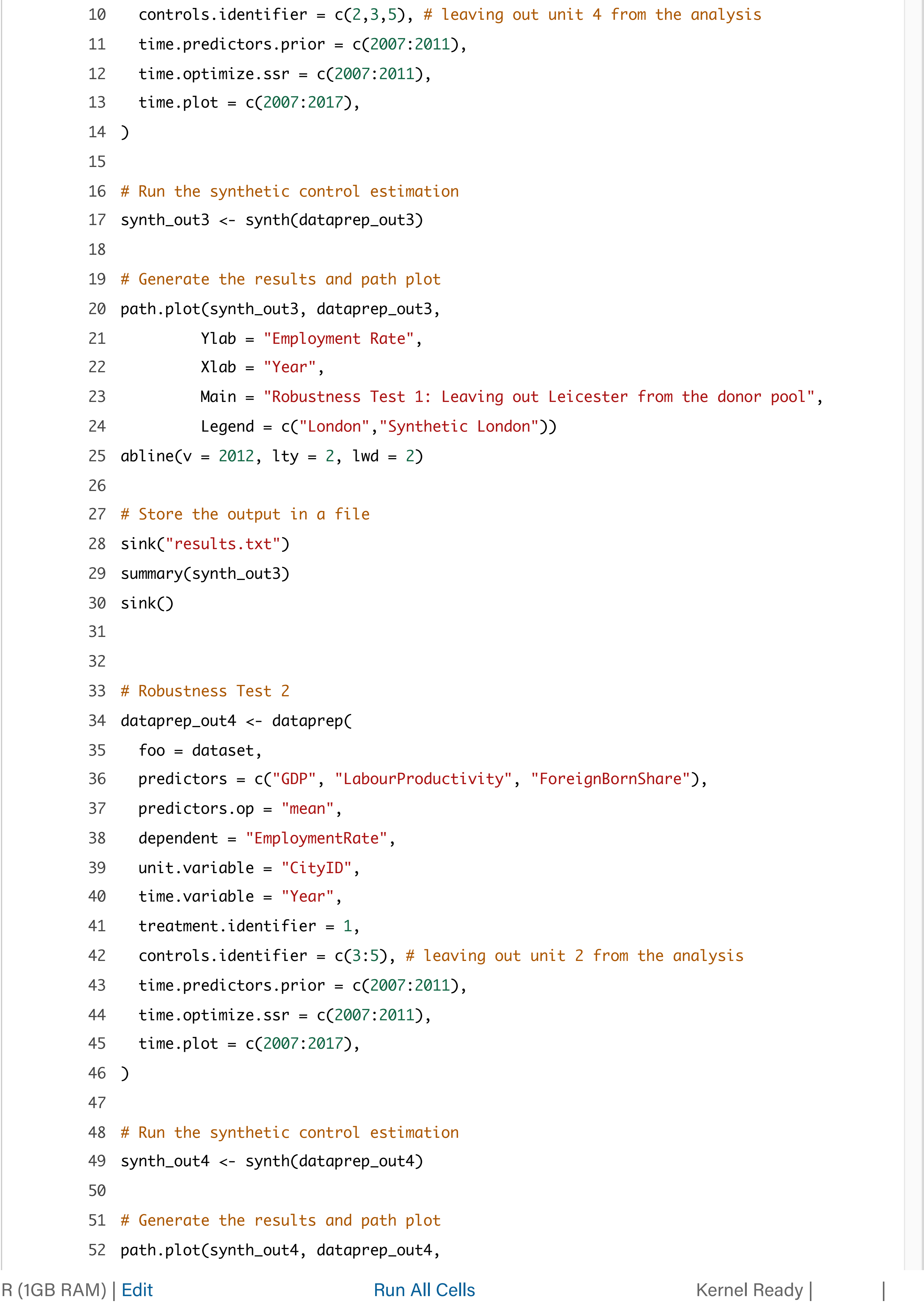


+ New Code

+ New Text

In [15]

```
# Robustness Test 1 (to evaluate the extent to which results are driven by any specific control unit)
dataprep_out3 <- dataprep(
  foo = dataset,
  predictors = c("GDP", "LabourProductivity", "ForeignBornShare"),
  predictors.op = "mean",
  dependent = "EmploymentRate",
  unit.variable = "CityID",
  time.variable = "Year",
  treatment.identifier = 1,
```

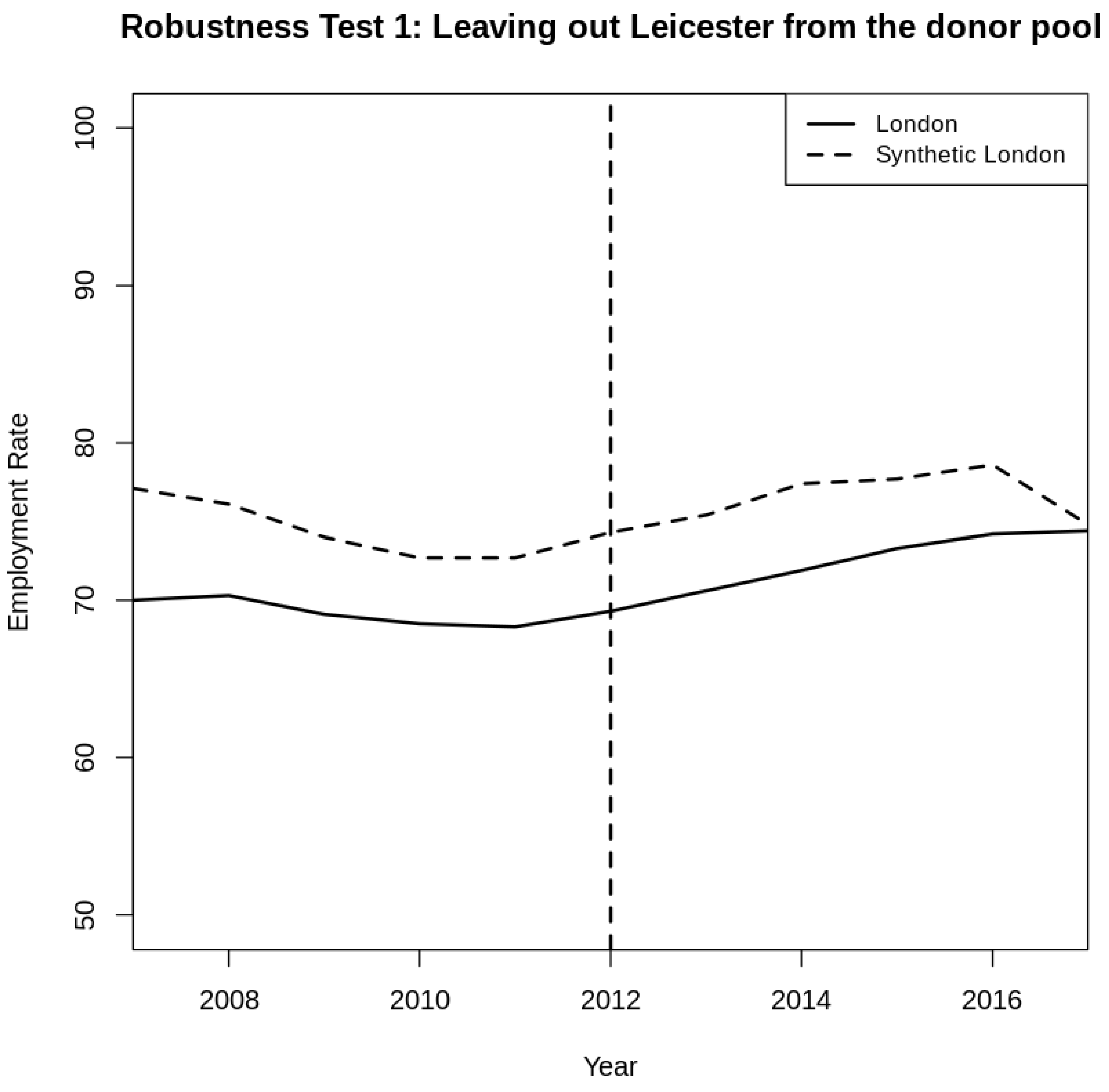


```
  controls.identifier = c(2,3,5), # leaving out unit 4 from the analysis
  time.predictors.prior = c(2007:2011),
  time.optimize.ssr = c(2007:2011),
  time.plot = c(2007:2017),
)

# Run the synthetic control estimation
synth_out3 <- synth(dataprep_out3)

# Generate the results and path plot
path.plot(synth_out3, dataprep_out3,
          Ylab = "Employment Rate",
          Xlab = "Year",
          Main = "Robustness Test 1: Leaving out Leicester from the donor pool",
          Legend = c("London","Synthetic London"))
abline(v = 2012, lty = 2, lwd = 2)

# Store the output in a file
sink("results.txt")
summary(synth_out3)
sink()


# Robustness Test 2
dataprep_out4 <- dataprep(
  foo = dataset,
  predictors = c("GDP", "LabourProductivity", "ForeignBornShare"),
  predictors.op = "mean",
  dependent = "EmploymentRate",
  unit.variable = "CityID",
  time.variable = "Year",
  treatment.identifier = 1,
  controls.identifier = c(3:5), # leaving out unit 2 from the analysis
  time.predictors.prior = c(2007:2011),
  time.optimize.ssr = c(2007:2011),
  time.plot = c(2007:2017),
)

# Run the synthetic control estimation
synth_out4 <- synth(dataprep_out4)

# Generate the results and path plot
path.plot(synth_out4, dataprep_out4,
          Xlab = "Year",
          Main = "Robustness Test 2: Leaving out Liverpool from the donor pool",
          Legend = c("London","Synthetic London"))
abline(v = 2012, lty = 2, lwd = 2)

# Store the output in a file
sink("results.txt")
summary(synth_out4)
sink()
```

Out [15]

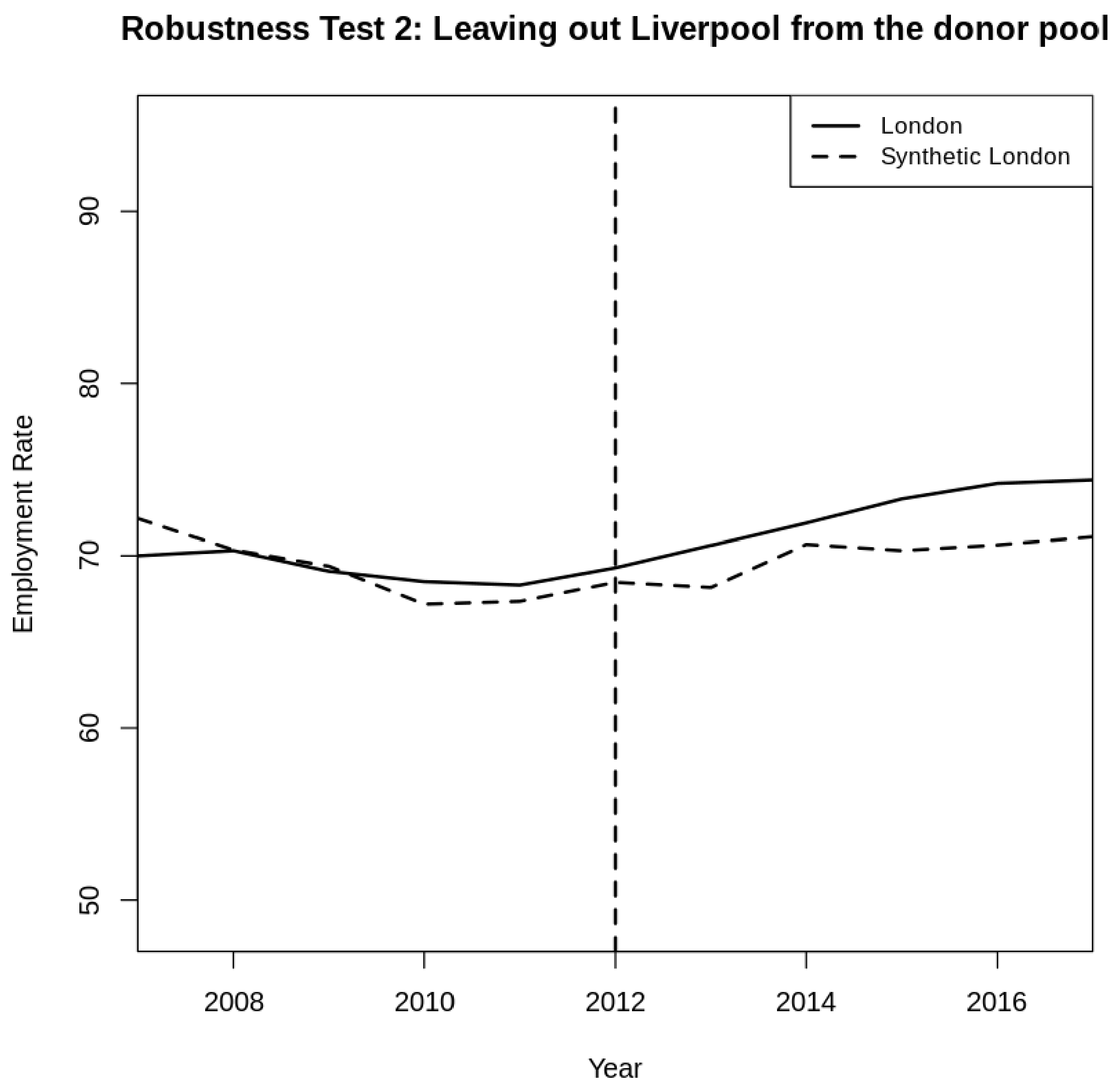
```
X1, X0, Z1, Z0 all come directly from dataprep object.


****************
 searching for synthetic control unit


****************
****************
****************

MSPE (LOSS V): 29.00441

solution.v:
 1.26331e-05 0.2802387 0.7197487

solution.w:
 3.6017e-05 4.335e-05 0.9999206


           Length Class      Mode
solution.v 3      data.frame list
solution.w 3      -none-     numeric
loss.v     1      -none-     numeric
loss.w     1      -none-     numeric
custom.v   0      -none-     NULL
rgV.optim  3      -none-     list


X1, X0, Z1, Z0 all come directly from dataprep object.


****************
 searching for synthetic control unit


****************
```

```
****************

MSPE (LOSS V): 1.477783

solution.v:
 0.23296 0.300553 0.466487

solution.w:
 8.051e-07 0.8225059 0.1774933
```



Robustness Test 1: Leaving out Leicester from the donor pool

```
            Length Class      Mode
solution.v 3      data.frame list
solution.w 3      -none-     numeric
loss.v     1      -none-     numeric
loss.w     1      -none-     numeric
custom.v   0      -none-     NULL
rgV.optim  3      -none-     list
```

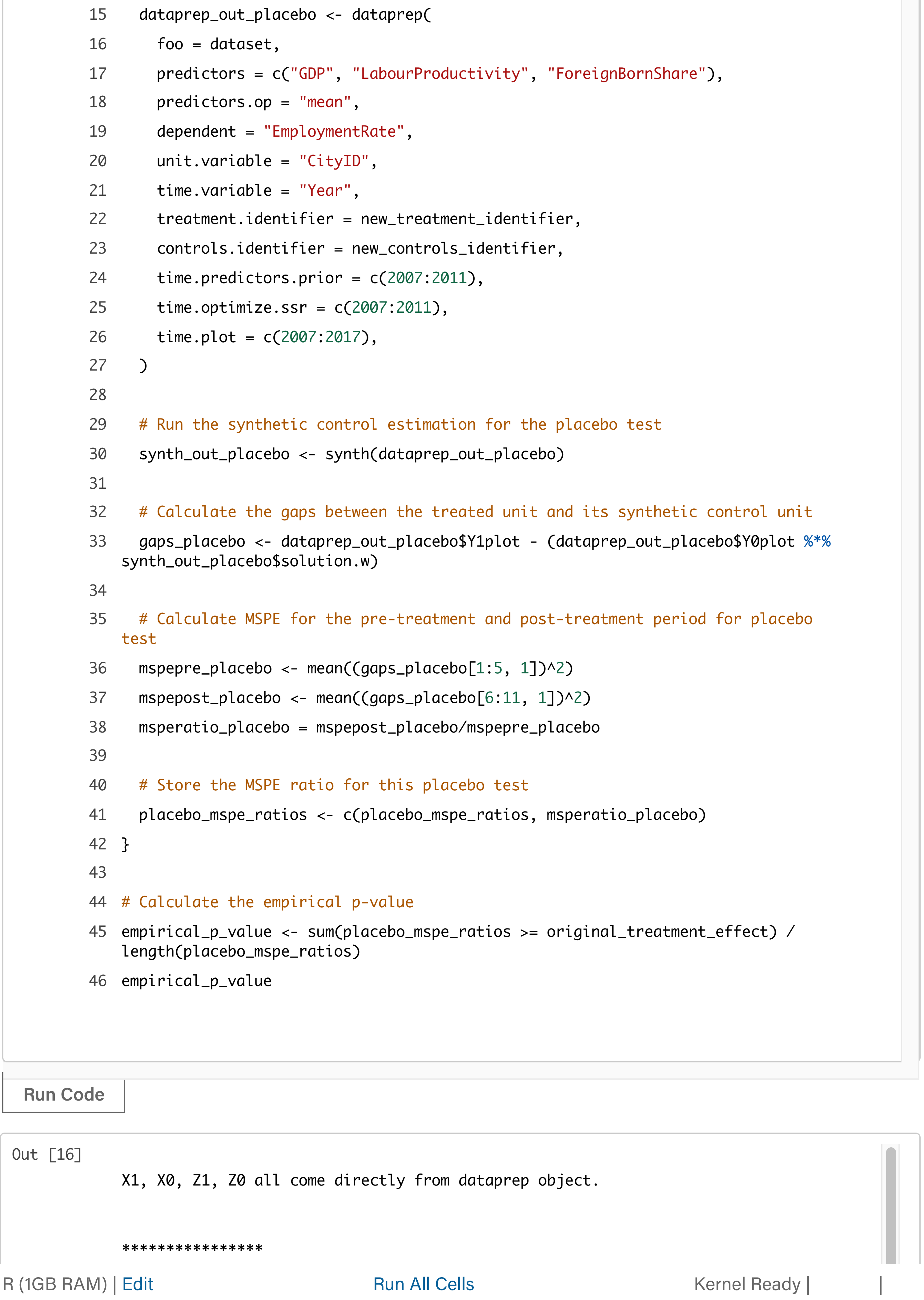


+ New Code

+ New Text

In [16]

```r
library(Synth)
# Store original treatment effect
original_treatment_effect <- msperatio

# Create a vector to store MSPE ratios for placebo tests
placebo_mspe_ratios <- numeric()

# Loop through control units
for (i in 2:5) {
  # Reassign treatment and control identifiers
  new_treatment_identifier <- i
  new_controls_identifier <- c(2:5)[-which(2:5 == i)]

```



```
  dataprep_out_placebo <- dataprep(
    foo = dataset,
    predictors = c("GDP", "LabourProductivity", "ForeignBornShare"),
    predictors.op = "mean",
    dependent = "EmploymentRate",
    unit.variable = "CityID",
    time.variable = "Year",
    treatment.identifier = new_treatment_identifier,
    controls.identifier = new_controls_identifier,
    time.predictors.prior = c(2007:2011),
    time.optimize.ssr = c(2007:2011),
    time.plot = c(2007:2017),
  )

  # Run the synthetic control estimation for the placebo test
  synth_out_placebo <- synth(dataprep_out_placebo)

  # Calculate the gaps between the treated unit and its synthetic control unit
  gaps_placebo <- dataprep_out_placebo$Y1plot - (dataprep_out_placebo$Y0plot %*% synth_out_placebo$solution.w)

  # Calculate MSPE for the pre-treatment and post-treatment period for placebo test
  mspepre_placebo <- mean((gaps_placebo[1:5, 1])^2)
  mspepost_placebo <- mean((gaps_placebo[6:11, 1])^2)
  msperatio_placebo = mspepost_placebo/mspepre_placebo

  # Store the MSPE ratio for this placebo test
  placebo_mspe_ratios <- c(placebo_mspe_ratios, msperatio_placebo)
}

# Calculate the empirical p-value
empirical_p_value <- sum(placebo_mspe_ratios >= original_treatment_effect) / length(placebo_mspe_ratios)
empirical_p_value
```

Out [16]

```
X1, X0, Z1, Z0 all come directly from dataprep object.


****************
```

```
****************
****************
****************


MSPE (LOSS V): 9.086002

solution.v:
 0.4547147 0.04640664 0.4988787

solution.w:
 0.9999997 2.857e-07 1e-10


X1, X0, Z1, Z0 all come directly from dataprep object.


****************
 searching for synthetic control unit


****************
****************
****************
0
```

+ New Code + New Text

```
In [20]
```

Run Code + New Code + New Text