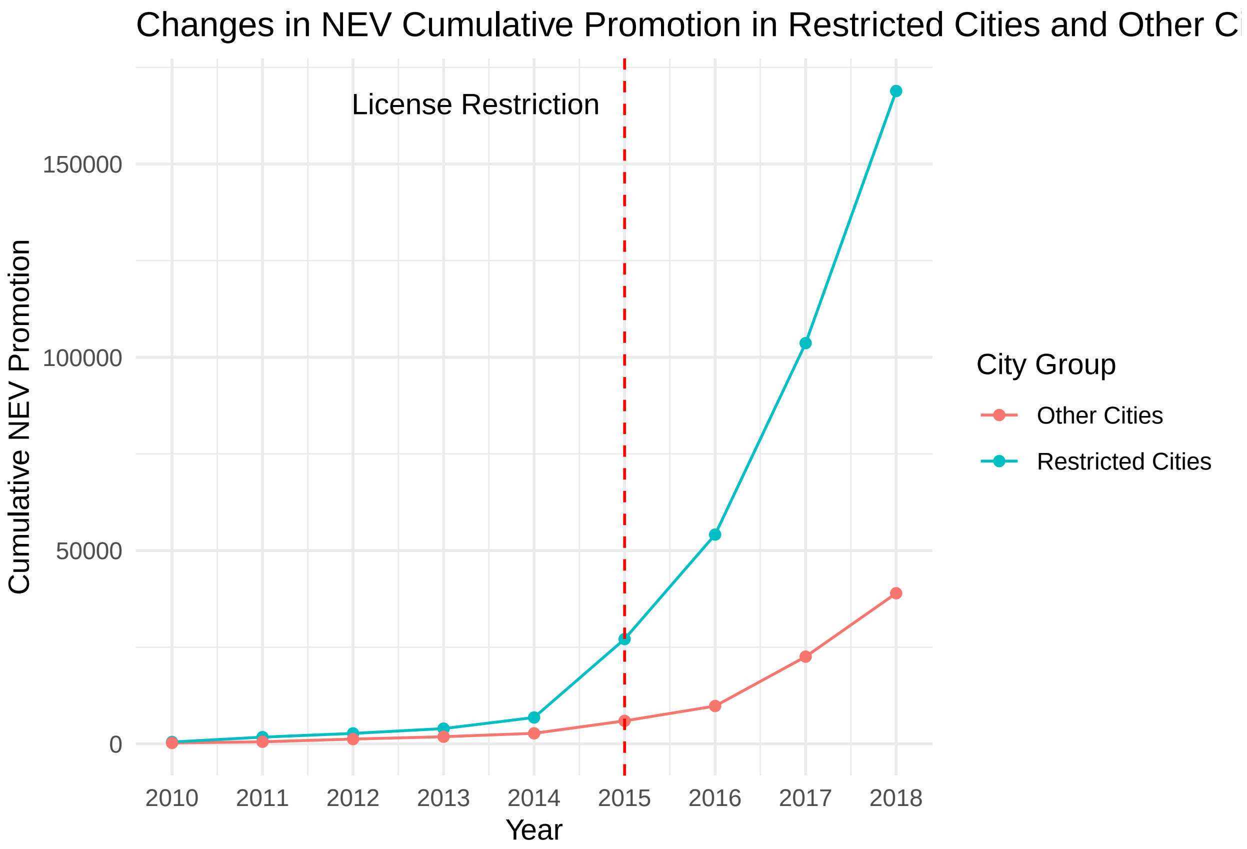
# under_thesis_SCM_code

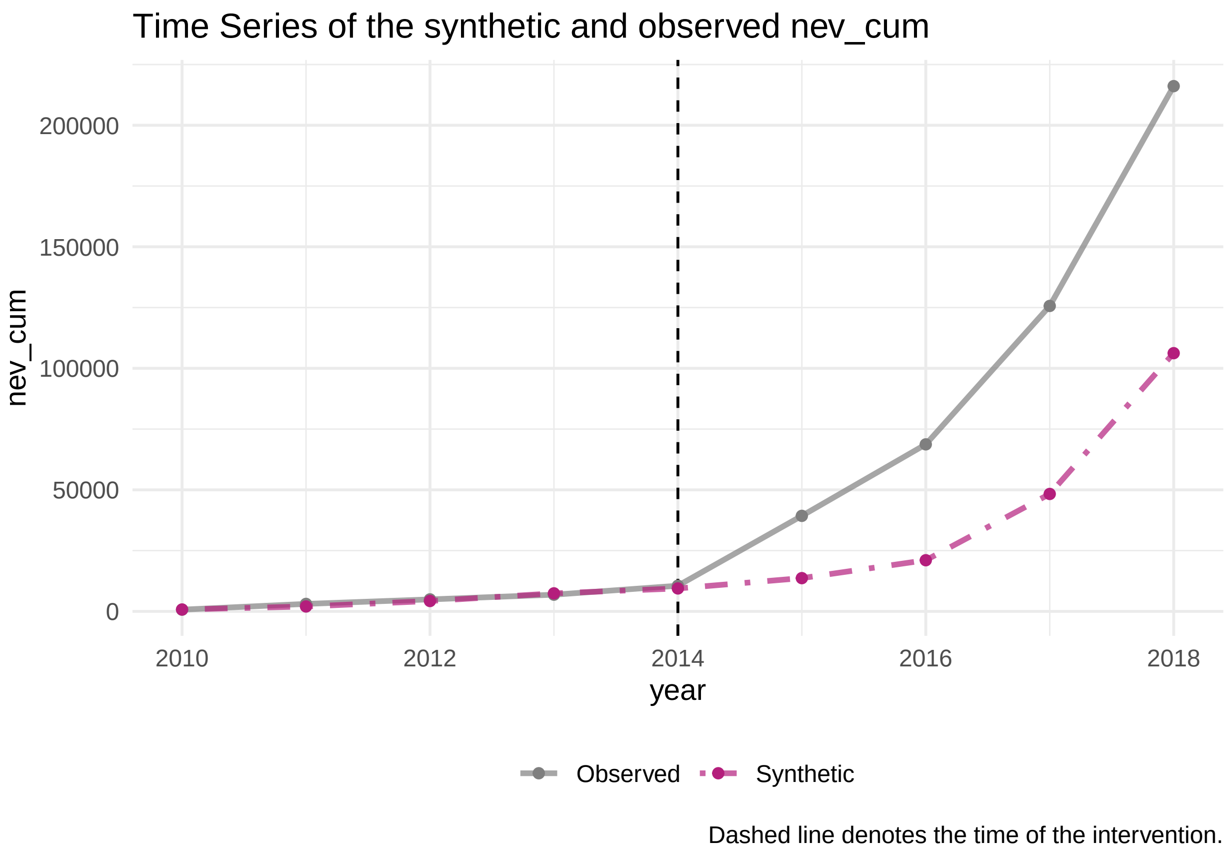
2024-10-08

```r
# Read new energy vehicle data
nev <- read.csv("E:\\qjy\\ecnu\\  \\    \\  SCM- - \\  SCM-   .csv", header = TRUE)

# Perform log transformation on GDP and income columns
nev$logGDP_per <- log10(nev$GDP_per)
nev$logincome_per <- log10(nev$income_per)
```

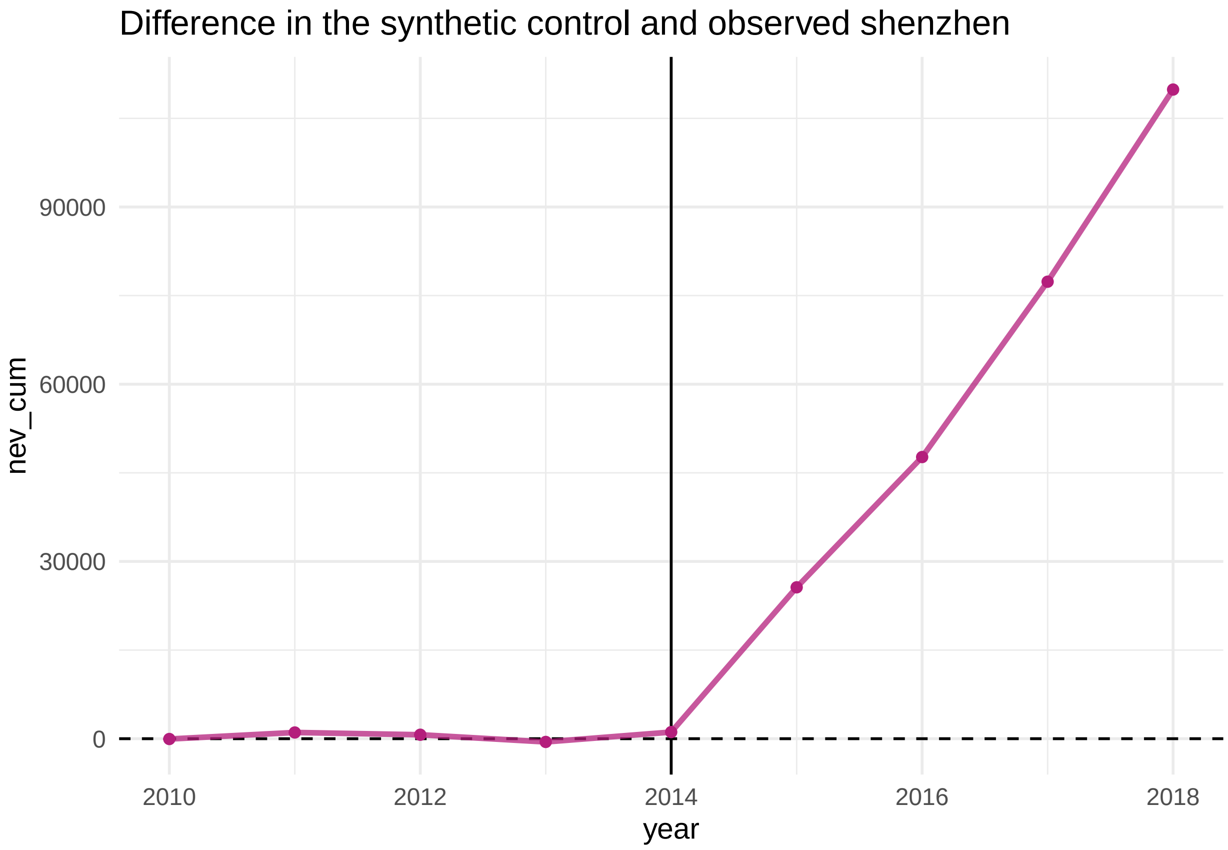
```r
# Calculate the annual average cumulative promotion of new energy vehicles in other cities
other_cities_mean <- nev %>%
  filter(city %in% c("chongqing", "changchun", "wuhan", "nantong", "shenyang", "zhengzhou", "hefei", "cl
  filter(year >= 2010 & year <= 2018) %>%
  group_by(year) %>%
  summarise(mean_nev_cum = mean(nev_cum, na.rm = TRUE))

# Create data frame for Shenzhen and Tianjin
xiangou_data <- nev %>%
  filter(city %in% c("shenzhen", "tianjin")) %>%
  filter(year >= 2010 & year <= 2018) %>%
  group_by(year) %>%
  summarise(mean_nev_cum = mean(nev_cum, na.rm = TRUE))

# Combine annual average data of restricted cities and other cities, preparing for plotting
xiangou_data$group <- "Restricted Cities"
other_cities_mean$group <- "Other Cities"
combined_data <- rbind(xiangou_data, other_cities_mean)

# Plot the cumulative promotion of new energy vehicles in restricted cities and other cities over the y
ggplot(combined_data, aes(x = year, y = mean_nev_cum, group = group, color = group)) +
  geom_line() +
  geom_point() +
  geom_vline(xintercept = 2015, linetype = "dashed", color = "red") + # Add vertical line for 2015
  annotate("text", x = 2015, y = max(combined_data$mean_nev_cum, na.rm = TRUE), label = "License Restri
  theme_minimal() +
  labs(title = "Changes in NEV Cumulative Promotion in Restricted Cities and Other Cities",
       x = "Year",
       y = "Cumulative NEV Promotion",
       color = "City Group") +
  scale_x_continuous(breaks = seq(2010, 2018, 1))
```



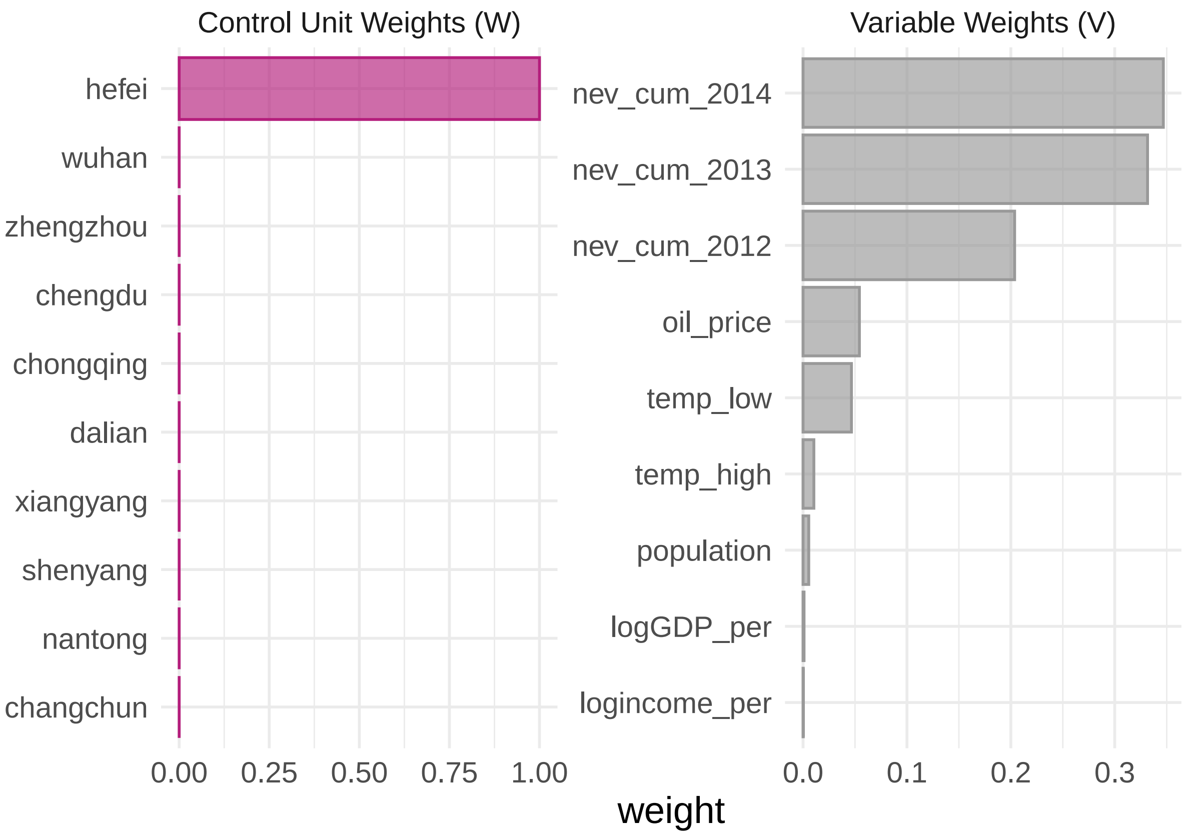
```r
# Shenzhen license restriction policy, selecting treatment and control groups
nev_shenzhen <- nev %>%
  filter(city %in% c("chongqing", "changchun", "wuhan", "nantong", "shenyang", "zhengzhou", "hefei", "cl
  filter(year >= 2010 & year <= 2018)

nev_out <- nev_shenzhen %>%
  synthetic_control(outcome = nev_cum,
                    unit = city,
                    time = year,
                    i_unit = "shenzhen",
                    i_time = 2014,
                    generate_placebos = TRUE) %>%
  generate_predictor(time_window = 2010:2014,
                     logGDP_per = mean(logGDP_per, na.rm = TRUE),
                     logincome_per = mean(logincome_per, na.rm = TRUE),
                     population = mean(population, na.rm = TRUE)) %>%
  generate_predictor(time_window = 2011:2014,
                     temp_low = mean(temp_low, na.rm = TRUE),
                     temp_high = mean(temp_high, na.rm = TRUE)) %>%
  generate_predictor(time_window = 2013:2014,
                     oil_price = mean(oil_price92, na.rm = TRUE)) %>%
  generate_predictor(time_window = 2014,
                     nev_cum_2014 = nev_cum) %>%
  generate_predictor(time_window = 2013,
                     nev_cum_2013 = nev_cum) %>%
  generate_predictor(time_window = 2012,
                     nev_cum_2012 = nev_cum) %>%
  generate_weights(optimization_window = 2010:2014,
                   margin_ipop = 0.02, sigf_ipop = 7, bound_ipop = 6) %>%
  generate_control()

# Plot the effect of Shenzhen's license restriction policy
nev_out %>% plot_trends()
```



```
nev_out %>% plot_differences()
```



```
nev_out %>% plot_weights()
```



```
nev_out %>% grab_unit_weights()
```

```
## # A tibble: 10 x 2
##    unit             weight
##    <chr>             <dbl>
##  1 changchun 0.00000000222
##  2 chengdu   0.00000000705
##  3 chongqing 0.00000000680
##  4 dalian    0.00000000503
##  5 hefei     1.00
##  6 nantong   0.00000000349
##  7 shenyang  0.00000000422
##  8 wuhan     0.00000000992
##  9 xiangyang 0.00000000429
## 10 zhengzhou 0.00000000922
```

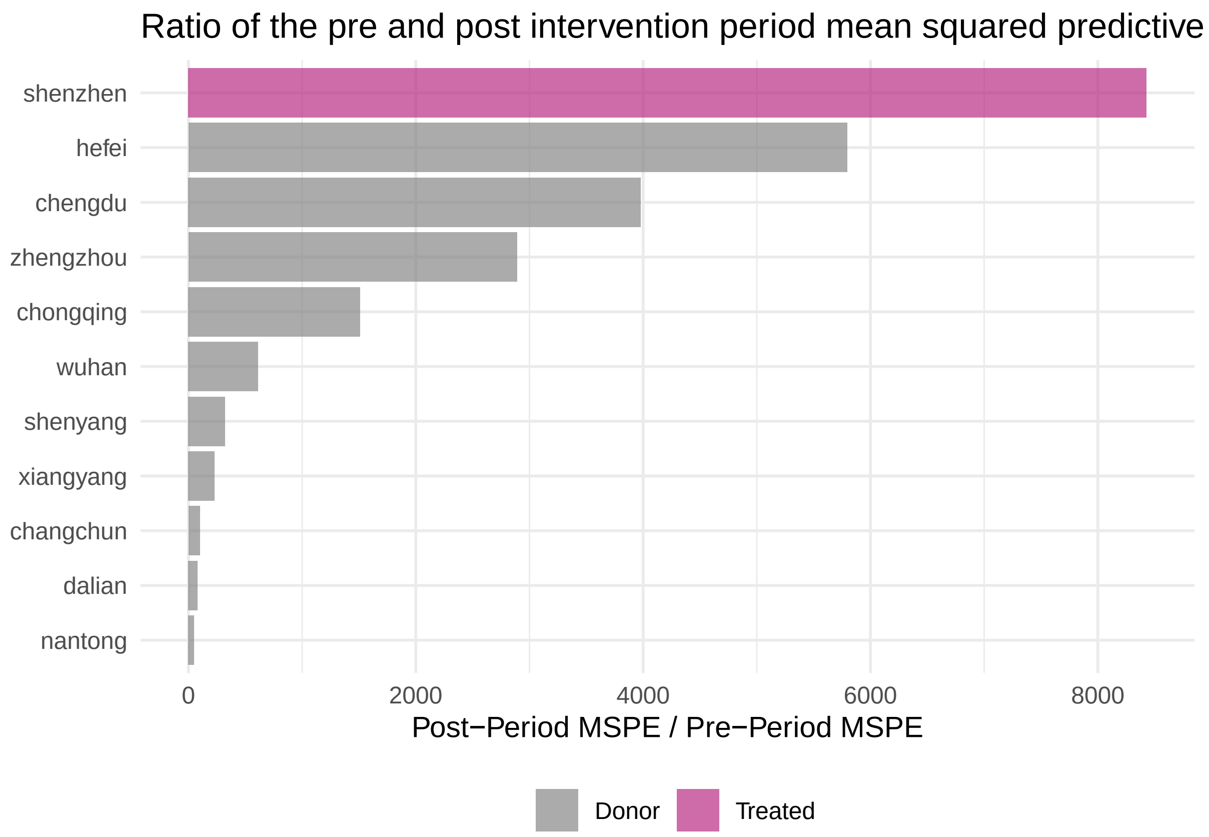
```
nev_out %>% grab_synthetic_control()
```

```
## # A tibble: 9 x 3
##   time_unit real_y synth_y
##       <int>  <dbl>   <dbl>
## 1      2010    720    774.
## 2      2011   3076   2026.
## 3      2012   4939   4270.
## 4      2013   6872   7422.
## 5      2014  10559   9438.
## 6      2015  39305  13683.
## 7      2016  68720  21048.
## 8      2017 125668  48316.
## 9      2018 216112 106239.
```

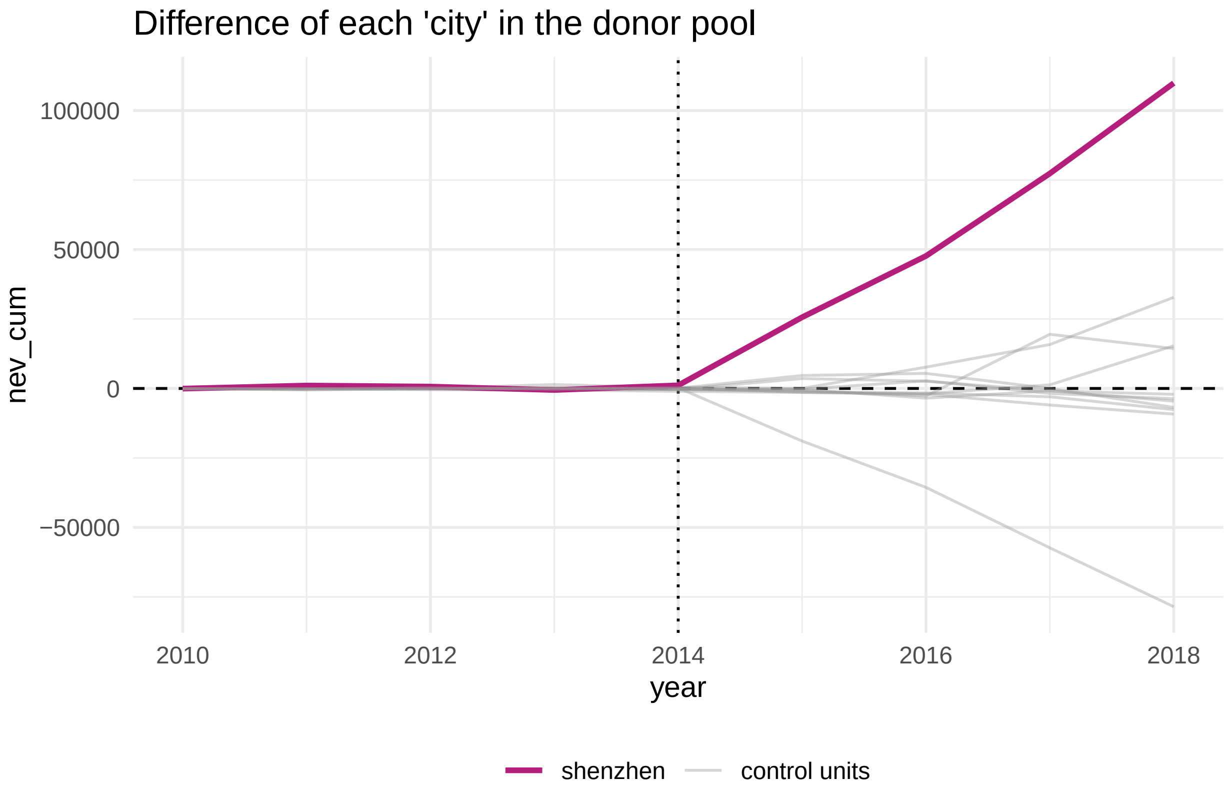
```
nev_out %>% grab_balance_table()
```

```
## # A tibble: 9 x 4
##   variable      shenzhen synthetic_shenzhen donor_sample
##   <chr>            <dbl>              <dbl>        <dbl>
## 1 logGDP_per        5.06               4.75         4.78
## 2 logincome_per     4.59               4.39         4.38
## 3 population     1186.                722.         1056.
## 4 temp_high        35                 37.5         36.3
## 5 temp_low          5.75              -5.50        -9.4
## 6 oil_price         5.55               6.86         7.34
## 7 nev_cum_2014  10559               9438.        2713.
## 8 nev_cum_2013   6872               7422.        1839.
## 9 nev_cum_2012   4939               4270.        1230.
```

```
nev_out %>% plot_mspe_ratio()
```

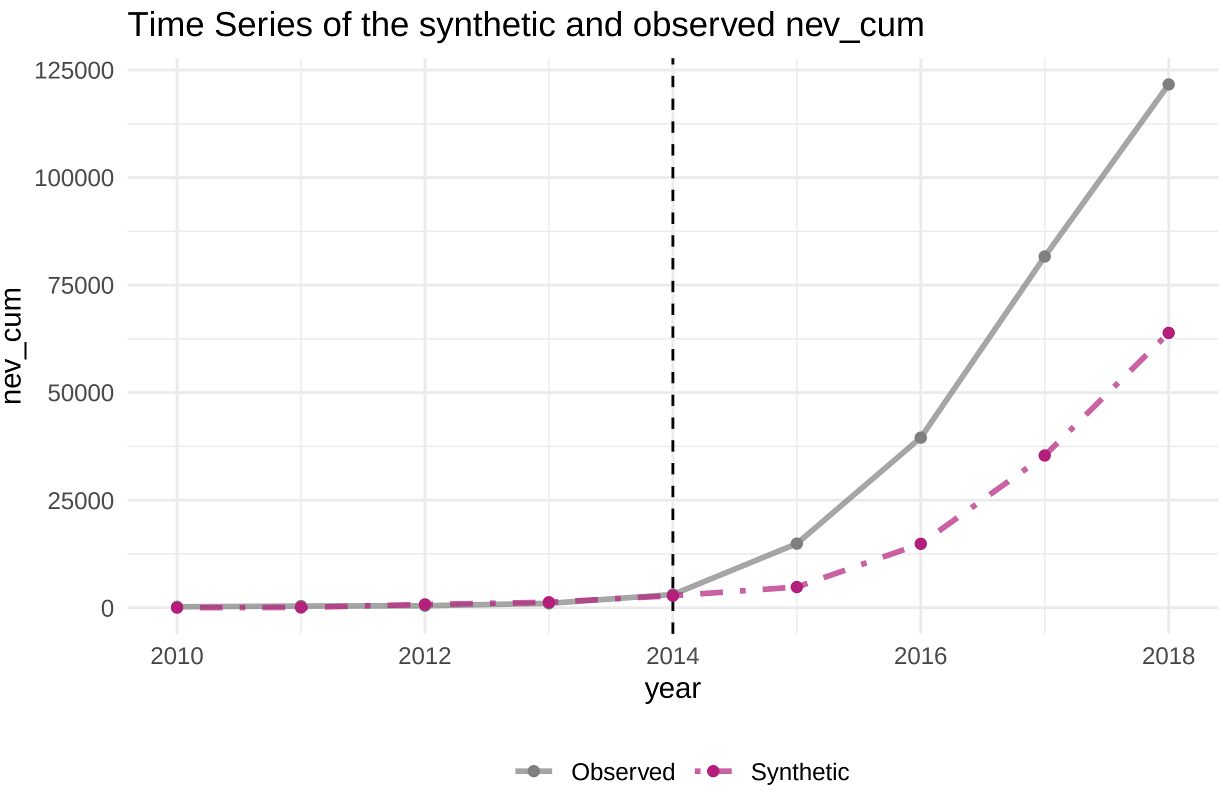


```
nev_out %>% plot_placebos()
```



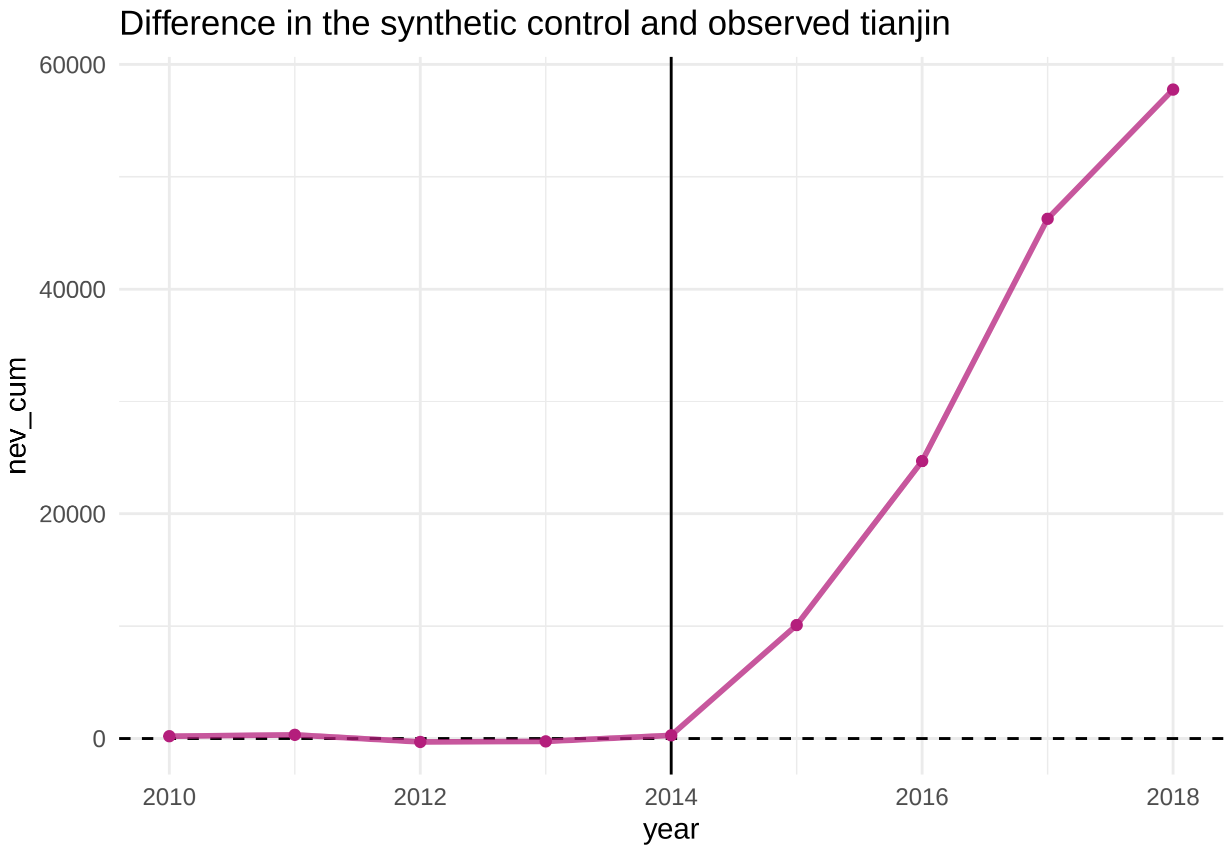
Pruned all placebo cases with a pre−period RMSPE exceeding two times the treated unit's pre−period RMSPE.

```r
# Tianjin license restriction policy
nev_tianjin <- nev %>%
  filter(city %in% c("chongqing", "changchun", "wuhan", "nantong", "shenyang", "zhengzhou", "hefei", "cl
  filter(year >= 2010 & year <= 2018)

nev_out <- nev_tianjin %>%
  synthetic_control(outcome = nev_cum,
                    unit = city,
                    time = year,
                    i_unit = "tianjin",
                    i_time = 2014,
                    generate_placebos = TRUE) %>%
  generate_predictor(time_window = 2010:2014,
                     logGDP_per = mean(logGDP_per, na.rm = TRUE),
                     logincome_per = mean(logincome_per, na.rm = TRUE),
                     population = mean(population, na.rm = TRUE)) %>%
  generate_predictor(time_window = 2011:2014,
                     temp_low = mean(temp_low, na.rm = TRUE),
                     temp_high = mean(temp_high, na.rm = TRUE)) %>%
  generate_predictor(time_window = 2013:2014,
                     oil_price = mean(oil_price92, na.rm = TRUE)) %>%
  generate_predictor(time_window = 2014,
                     nev_cum_2014 = nev_cum) %>%
  generate_predictor(time_window = 2013,
                     nev_cum_2013 = nev_cum) %>%
  generate_predictor(time_window = 2012,
                     nev_cum_2012 = nev_cum) %>%
  generate_weights(optimization_window = 2010:2014,
                   margin_ipop = 0.02, sigf_ipop = 7, bound_ipop = 6) %>%
  generate_control()

# Plot the effect of Tianjin's license restriction policy
nev_out %>% plot_trends()
```

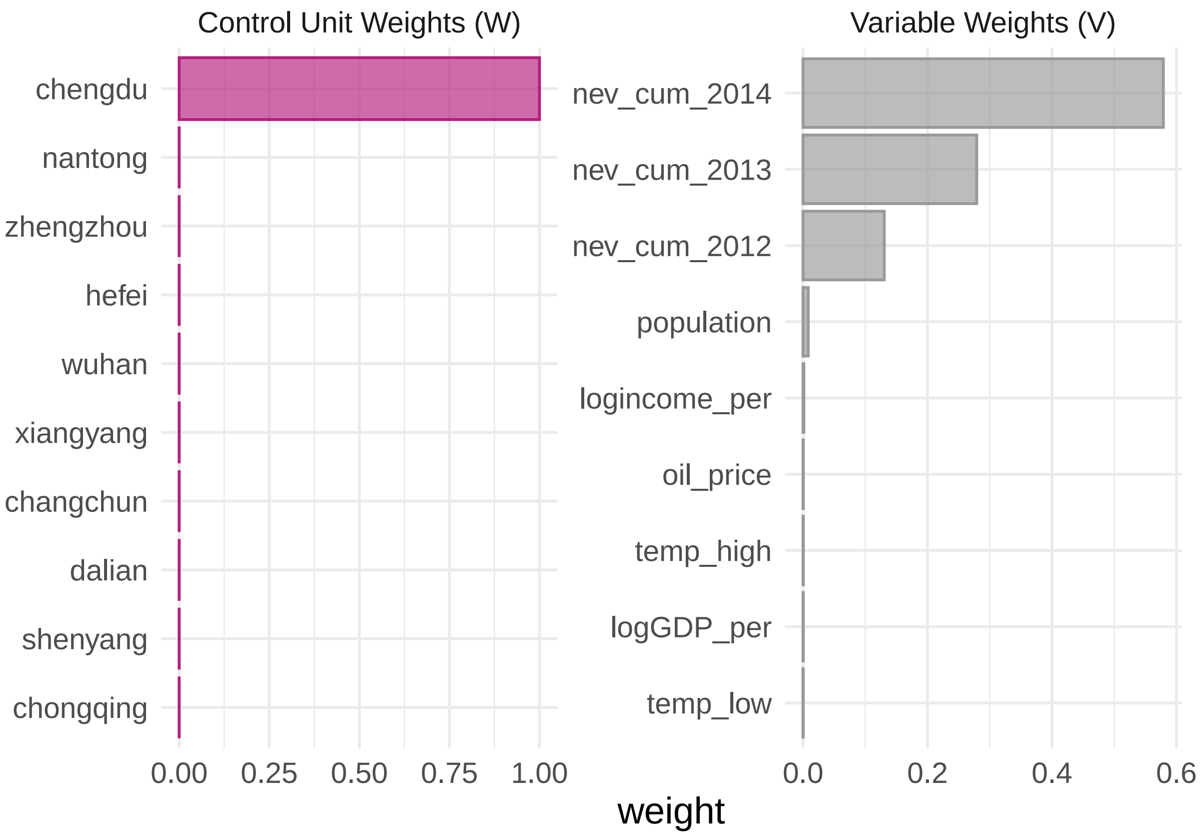


Dashed line denotes the time of the intervention.

```
nev_out %>% plot_differences()
```



```
nev_out %>% plot_weights()
```



```
nev_out %>% grab_unit_weights()
```

```
## # A tibble: 10 x 2
##    unit             weight
##    <chr>             <dbl>
##  1 changchun 0.0000000304
##  2 chengdu   1.00
##  3 chongqing 0.0000000157
##  4 dalian    0.0000000273
##  5 hefei     0.0000000581
##  6 nantong   0.000000247
##  7 shenyang  0.0000000253
##  8 wuhan     0.0000000548
##  9 xiangyang 0.0000000479
## 10 zhengzhou 0.0000000999
```

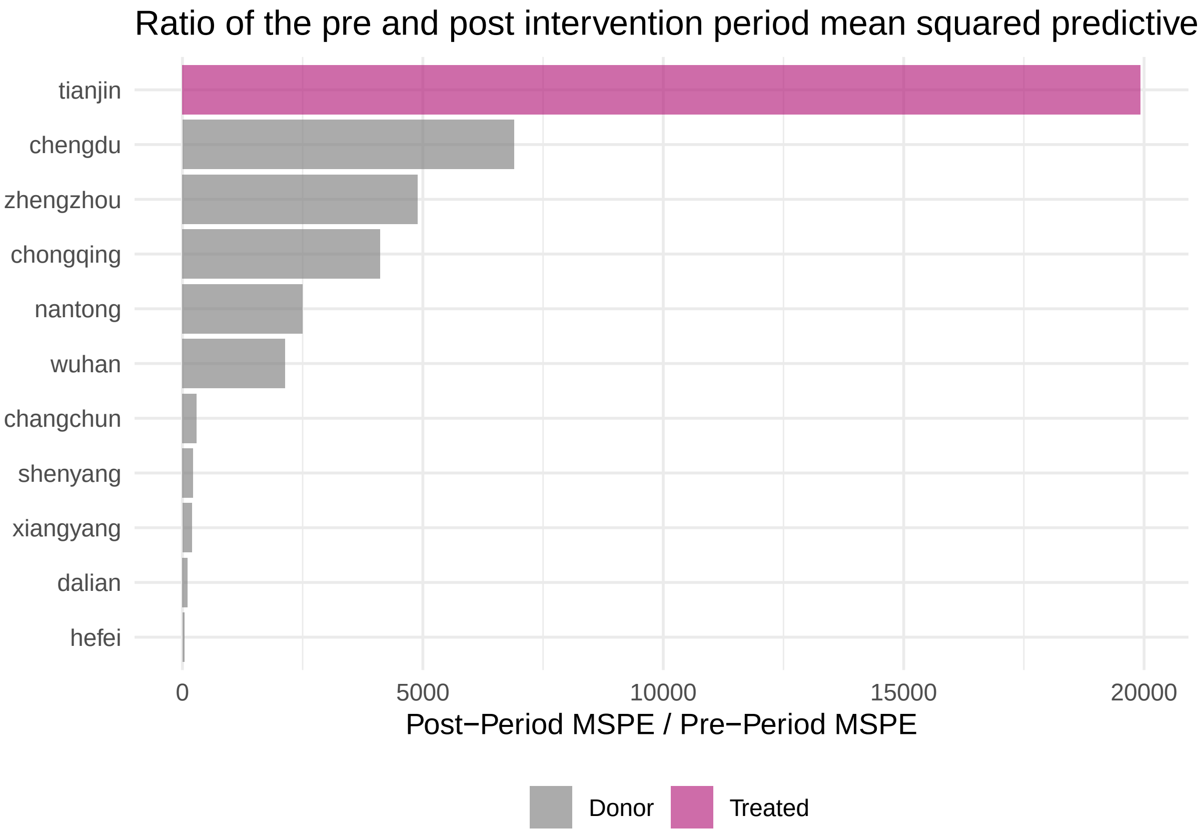
```
nev_out %>% grab_synthetic_control()
```

```
## # A tibble: 9 x 3
##   time_unit real_y synth_y
##       <int>  <dbl>   <dbl>
## 1      2010    214    10.0
## 2      2011    382    56.0
## 3      2012    458   765.
## 4      2013   1011  1270.
## 5      2014   3066  2785.
## 6      2015  14898  4802.
## 7      2016  39537 14848.
## 8      2017  81649 35389.
## 9      2018 121649 63887.
```

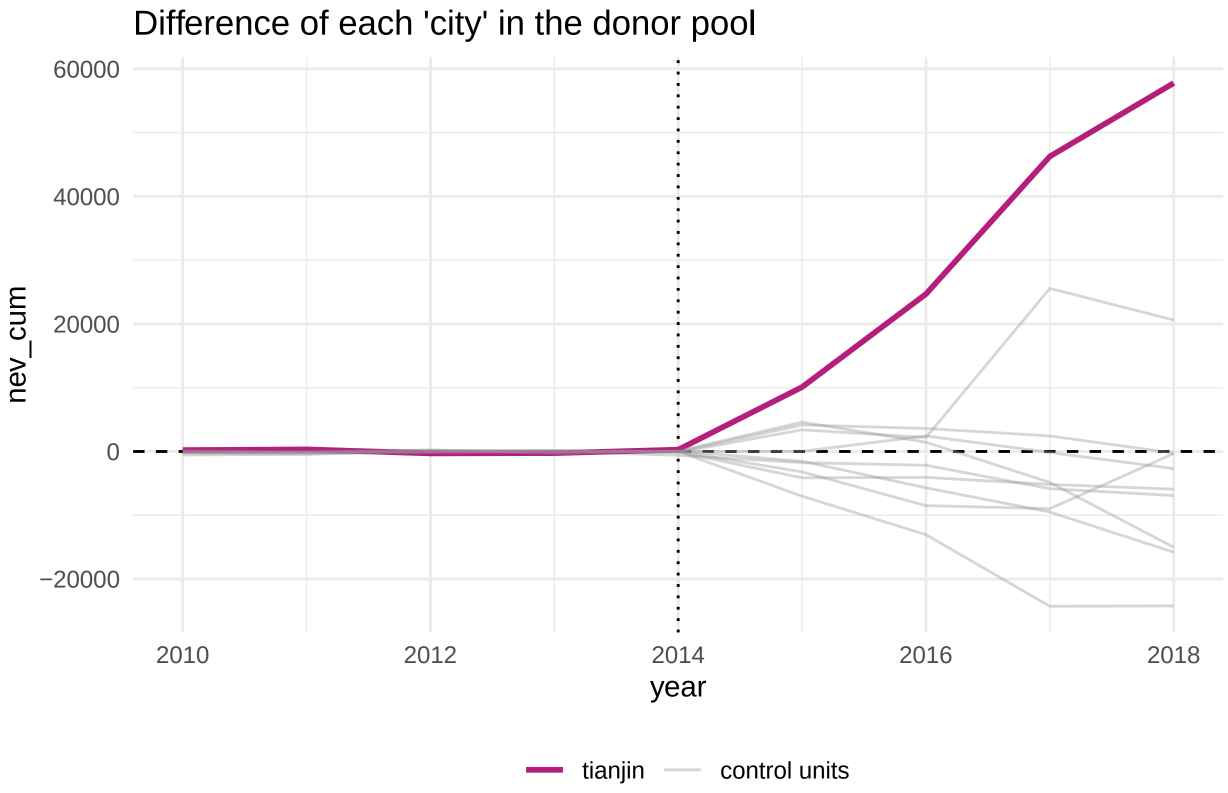
```
nev_out %>% grab_balance_table()
```

```
## # A tibble: 9 x 4
##   variable      tianjin synthetic_tianjin donor_sample
##   <chr>           <dbl>             <dbl>        <dbl>
## 1 logGDP_per       4.94              4.76         4.78
## 2 logincome_per    4.45              4.41         4.38
## 3 population    1371.             1420.        1056.
## 4 temp_high       37.9              35.0         36.3
## 5 temp_low       -10                -2.75        -9.4
## 6 oil_price        7.45              7.49         7.34
## 7 nev_cum_2014  3066              2785.        2713.
## 8 nev_cum_2013  1011              1270.        1839.
## 9 nev_cum_2012   458               765.        1230.
```

```
nev_out %>% plot_mspe_ratio()
```



```
nev_out %>% plot_placebos()
```



Pruned all placebo cases with a pre−period RMSPE exceeding two times the treated unit's pre−period RMSPE.