Tutorial Week 8 and 9: Matching and Entropy Balancing

Problem Set - Propensity Scores, Matching, and Synthetic Control

Maximilian Birkle Daniel Lehmann Henry Lucas

2025-11-11

Contents

1 Load Packages

```r
library(tidyverse)
library(haven)
library(MatchIt)
library(cobalt)
library(randomForest)
library(lmtest)
library(sandwich)
library(knitr)
library(kableExtra)
library(broom)
library(Synth)
library(stargazer)
library(patchwork)
library(cobalt)
library(Synth)
```

2 Part I: Propensity Scores, Matching, and Robust Post-Matching Inference

2.1 Background

Research Question: Do **United Nations interventions** help shorten the duration of civil wars?

Gilligan and Sergenti (2008) use matching methods to re-evaluate earlier findings suggesting that UN interventions *prolong* conflict. They argue that this conclusion stems from **selection bias** — the UN tends to intervene in the *worst* conflicts.

Dataset: `war_pre_snapshots.dta`

Each row represents a **conflict episode** observed before a potential UN intervention. Our goal is to estimate the causal effect of UN involvement (`UN`) on the length of the conflict (`t1 - t0`), while balancing on key pre-treatment covariates.

2.2 Load Data

```r
# Read the UN intervention dataset
data_un <- read_dta("war_pre_snapshots.dta")

# Display structure and summary
#glimpse(data_un)
#summary(data_un)
```

2.3 Define Treatment and Covariates

```r
# Treatment variable: UN intervention (1 = Yes, 0 = No)
treat <- data_un$UN

# Outcome variable: conflict duration (t1 - t0)
outcome <- data_un$t - data_un$t0

# Covariates for propensity score model
covar <- c(
  "inter", "deaths", "couprev", "sos", "drugs", "t0",
  "ethfrac", "pop", "lmtnest", "milper",
  "eeurop", "lamerica", "asia", "ssafrica"
)

# Create covariate matrix
X <- data_un[, covar]

dta_new <- data.frame(
  treat = treat,
  outcome = outcome,
  X)
```

2.4 Q0. First Check of the Data

2.4.1 Tasks:

1. Why might UN interventions **not** be randomly assigned across conflicts?
2. Which of the listed variables are most likely to confound the relationship between `UN` and conflict duration? Run a quick logistic regression and check.

```
# Logistic regression: UN intervention as function of covariates
logit_selection <- glm(treat ~ inter + deaths + couprev + sos + drugs + t0 +
                         ethfrac + pop + lmtnest + milper +
                         eeurop + lamerica + asia + ssafrica,
                       data = dta_new,
                       family = binomial(link = "logit"))

# Create stargazer table
stargazer(
  logit_selection,
  type = "latex",
  title = "Selection into UN Intervention: Logistic Regression",
  header = FALSE,
  dep.var.labels = "UN Intervention (1 = Yes)",
  covariate.labels = c(
    "Internationalized Conflict",
    "Battle Deaths",
    "Coup/Revolution",
    "Strategic Oil Supply",
    "Drug Activity",
    "Conflict Start Year",
    "Ethnic Fractionalization",
    "Log Population",
    "Log Mountainous Terrain",
    "Military Personnel per Capita",
    "Eastern Europe",
    "Latin America",
    "Asia",
    "Sub-Saharan Africa",
    "Constant"
  ),
  no.space = TRUE,
  omit.stat = c("aic", "ll"),
  notes = "Standard errors in parentheses.",
  notes.align = "l",
  digits = 3
)
```

Discussion:

Why might UN interventions not be randomly assigned?

Cases where the UN intervenes are quite different from where they do not, which is why UN missions are

Table 1: Selection into UN Intervention: Logistic Regression

	Dependent variable:
	UN Intervention (1 = Yes)
Internationalized Conflict	0.929
	(0.846)
Battle Deaths	−0.156
	(0.218)
Coup/Revolution	−0.113
	(1.234)
Strategic Oil Supply	−1.075
	(0.994)
Drug Activity	3.238***
	(1.087)
Conflict Start Year	0.005
	(0.003)
Ethnic Fractionalization	−0.016
	(0.017)
Log Population	−0.271
	(0.471)
Log Mountainous Terrain	0.271
	(0.321)
Military Personnel per Capita	−1.123***
	(0.351)
Eastern Europe	2.186
	(1.485)
Latin America	−3.775**
	(1.731)
Asia	−2.440
	(1.630)
Sub-Saharan Africa	−1.337
	(1.521)
Constant	3.002
	(4.006)
Observations	1,227
Note:	*p<0.1; **p<0.05; ***p<0.01
	Standard errors in parentheses.

not randomly assigned.

The decision to intervene depends on a number of factors, including the UN Security Council's selection process and whether the UN even pays attention to these cases.

Because the "treated" (intervention) and "untreated" (no intervention) cases are so different on so many variables, it becomes almost impossible to distinguish whether a causal effect is due to the treatment itself or some function of these other confounding variables. When you have differences that large between units, you face an extreme problem of constructing the counterfactual—that is, figuring out what would have happened had the UN not intervened.

Which variables show strong associations with UN intervention?

Looking at the results in table 1, we can see a high positive coefficient for **Drug Activity** (coef=3.238, p<0.01), meaning that the UN is more likely to intervene where there is drug-related activity present. This also makes intuitive sense, as interventions are fostered by international drug trafficking concerns. This is a prime example of why it is hard to compare regimes across regions that could potentially receive UN intervention. Since regimes, rebel groups, or terrorist groups in politically unstable systems often build their economic foundation on scarce resources (e.g., diamonds, oil, etc.) and also drugs, their different capabilities (e.g., geographic location, natural resources) offer different incentives for the UN to intervene, especially when their economic activities pose a global threat, as is the case with drugs.

Similarly, in countries with high per capita **military personnel** (coef=-1.123, p<0.01), countries with higher military capacity are less likely to receive UN intervention. This also makes intuitive sense, as weak military states would make it easier and safer to intervene, and they have more need for external support.

Furthermore, countries located in **Latin America** are less likely to receive UN intervention (coef=-3.775, P<0.05). This finding might be slightly confusing at first glance, although it might be explained by a couple of points. As many states in Latin America have faced high levels of political instability and also have a long history with drug trafficking, we might assume that they would be more likely to receive UN interventions. The absence of this finding might be due to the fact that Latin America has always been undisputedly located in the sphere of influence of the United States—historically articulated in the Monroe Doctrine (1823)—which has carried out its own "peacekeeping missions" across the continent, not based on decisions of the UN or other multilateral actors after the Cold War. Looking at recent developments around the coast of Venezuela, this dynamic still dominates today.

Overall, this regression table gives some preliminary hints that UN interventions are indeed not randomly assigned. It demonstrates clear selection bias: the UN systematically intervenes in cases with specific, observable characteristics (like high drug activity or weak militaries) and avoids others (like those in Latin America). While other factors like battle deaths or ethnic fractionalization don't show a significant effect in this model, the strong predictors identified here are exactly why it is extremely useful to use matching to create a valid comparison group.

2.5 Q1. Estimating Propensity Scores

2.5.1 Theoretical Background

Let $T_i \in \{0, 1\}$ be the treatment indicator and $X_i = (X_{i1}, X_{i2}, \dots, X_{ip})$ the vector of pre-treatment covariates.

The **propensity score** is defined as:

$$e(X_i) = P(T_i = 1 \mid X_i)$$

A **propensity score** is the probability that a unit with certain characteristics will be assigned to the treatment group. It ranges between 0 and 1.

2.5.2 Tasks:

1. Define the propensity score
2. Estimate $\hat{e}(X_i)$ in two ways:
   - **(a)** Logistic regression: $\text{logit}(e(X_i)) = X_i'\beta$
   - **(b)** Random forest classifier
3. Report mean, SD, and range of $\hat{e}(X_i)$ for treated and control
4. Create histogram/density plot by treatment status

```r
# (a) Logistic regression propensity score

ps_logistic <- glm(treat ~ inter + deaths + couprev + sos + drugs + t0 +
                     ethfrac + pop + lmtnest + milper +
                     eeurop + lamerica + asia + ssafrica,
                   data = dta_new,
                   family = binomial(link = "logit"),
                   na.action = na.exclude)


dta_new$ps_logistic_probs <- predict(ps_logistic,
                                     newdata = dta_new,
                                     type = "response")
```

```r
# (b) Random forest propensity score

set.seed(68159) #for reproducability

ps_random_forest <- randomForest(
  factor(treat) ~ inter + deaths + couprev + sos + drugs + t0 +
    ethfrac + pop + lmtnest + milper +
    eeurop + lamerica + asia + ssafrica,
  data = dta_new,
  ntree = 1000,
  importance = TRUE
)

# Extract predicted probabilities

dta_new$ps_random_forest <- predict(ps_random_forest, type = "prob")[, "1"]
```

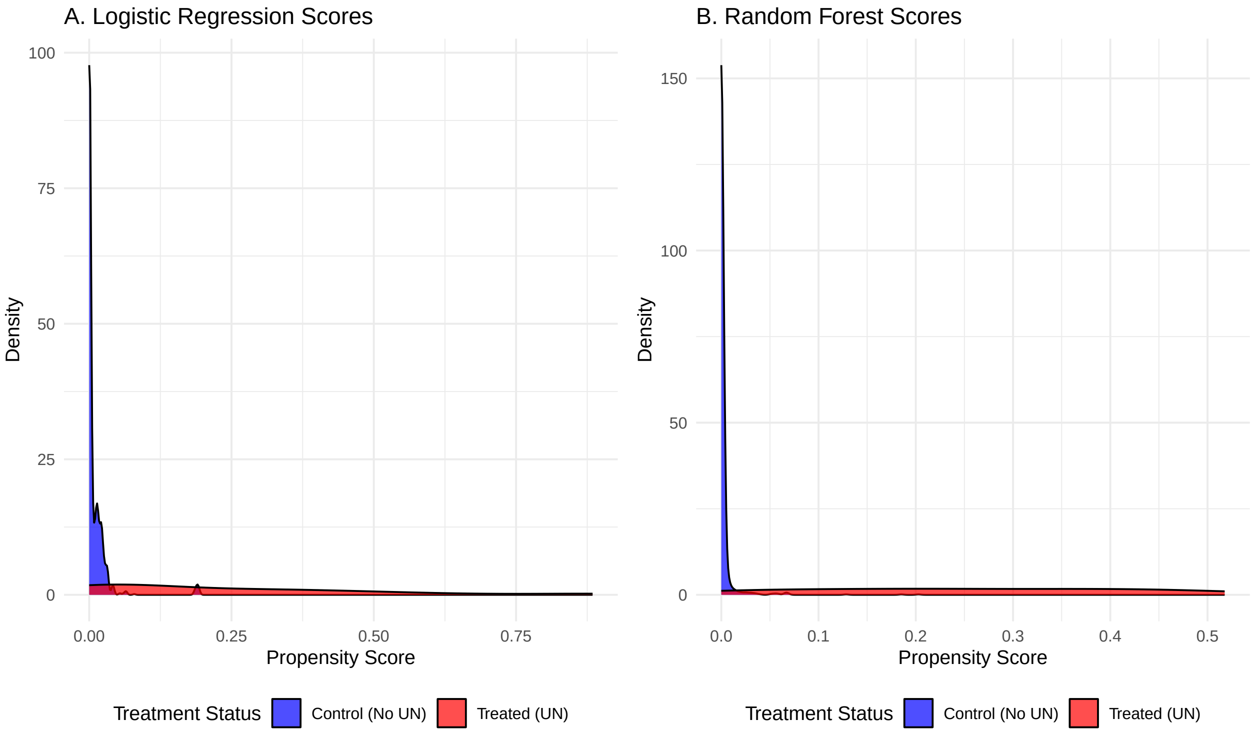
```r
# Summary statistics of propensity scores

# Logistic Regression Stats
logit_control_mean <- mean(dta_new$ps_logistic_probs[dta_new$treat == 0], na.rm = TRUE)
logit_control_sd   <- sd(dta_new$ps_logistic_probs[dta_new$treat == 0], na.rm = TRUE)
logit_control_min  <- min(dta_new$ps_logistic_probs[dta_new$treat == 0], na.rm = TRUE)
logit_control_max  <- max(dta_new$ps_logistic_probs[dta_new$treat == 0], na.rm = TRUE)

logit_treated_mean <- mean(dta_new$ps_logistic_probs[dta_new$treat == 1], na.rm = TRUE)
logit_treated_sd   <- sd(dta_new$ps_logistic_probs[dta_new$treat == 1], na.rm = TRUE)
logit_treated_min  <- min(dta_new$ps_logistic_probs[dta_new$treat == 1], na.rm = TRUE)
logit_treated_max  <- max(dta_new$ps_logistic_probs[dta_new$treat == 1], na.rm = TRUE)

# Random Forest Stats
rf_control_mean <- mean(dta_new$ps_random_forest[dta_new$treat == 0], na.rm = TRUE)
```

```
rf_control_sd   <- sd(dta_new$ps_random_forest[dta_new$treat == 0], na.rm = TRUE)
rf_control_min  <- min(dta_new$ps_random_forest[dta_new$treat == 0], na.rm = TRUE)
rf_control_max  <- max(dta_new$ps_random_forest[dta_new$treat == 0], na.rm = TRUE)

rf_treated_mean <- mean(dta_new$ps_random_forest[dta_new$treat == 1], na.rm = TRUE)
rf_treated_sd   <- sd(dta_new$ps_random_forest[dta_new$treat == 1], na.rm = TRUE)
rf_treated_min  <- min(dta_new$ps_random_forest[dta_new$treat == 1], na.rm = TRUE)
rf_treated_max  <- max(dta_new$ps_random_forest[dta_new$treat == 1], na.rm = TRUE)


# Data Frame with results
summary_stats_q1 <- data.frame(
  Statistic = c("Mean", "Std. Deviation", "Min", "Max"),
  Logit_Control = c(logit_control_mean, logit_control_sd, logit_control_min, logit_control_max),
  Logit_Treated = c(logit_treated_mean, logit_treated_sd, logit_treated_min, logit_treated_max),
  RF_Control = c(rf_control_mean, rf_control_sd, rf_control_min, rf_control_max),
  RF_Treated = c(rf_treated_mean, rf_treated_sd, rf_treated_min, rf_treated_max)
)

tbl <- knitr::kable(
  summary_stats_q1,
  digits = 3,
  col.names = c("Statistic", "Control (T=0)", "Treated (T=1)", "Control (T=0)", "Treated (T=1)"),
  caption = "Table 1: Propensity Score Distribution by Model and Treatment Status"
)

if (knitr::is_latex_output() || knitr::is_html_output()) {
  tbl <- tbl %>%
    kableExtra::add_header_above(c(" " = 1, "Logistic Regression" = 2, "Random Forest" = 2)) %>%
    kableExtra::kable_styling(bootstrap_options = "striped", full_width = FALSE)
}

tbl
```

Table 2: Table 1: Propensity Score Distribution by Model and Treatment Status

	Logistic Regression		Random Forest	
Statistic	Control (T=0)	Treated (T=1)	Control (T=0)	Treated (T=1)
Mean	0.010	0.213	0.002	0.250
Std. Deviation	0.026	0.253	0.011	0.169
Min	0.000	0.000	0.000	0.000
Max	0.194	0.884	0.203	0.518

```
# Density plot of propensity scores by treatment status

# Logistic Regression Plot
p_logit <- ggplot(dta_new, aes(x = ps_logistic_probs, fill = factor(treat))) +
  geom_density(alpha = 0.7) +
  scale_fill_manual(
    name = "Treatment Status",
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "Control (No UN)", "1" = "Treated (UN)")
  ) +
  labs(
    title = "A. Logistic Regression Scores",
    x = "Propensity Score",
    y = "Density"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom") # Move legend to the bottom

# Random Forest Plot
p_rf <- ggplot(dta_new, aes(x = ps_random_forest, fill = factor(treat))) +
  geom_density(alpha = 0.7) +
  scale_fill_manual(
    name = "Treatment Status",
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "Control (No UN)", "1" = "Treated (UN)")
  ) +
  labs(
    title = "B. Random Forest Scores",
    x = "Propensity Score",
    y = "Density"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom") # Move legend to the bottom

# --- 3. Combine the plots ---
p_logit + p_rf
```



Interpretation: The propensity score distributions reveal a severe lack of common support, which will be challenging for the matching analysis. The summary statistics clearly illustrate this imbalance: The control group (N = 1,211) is a highly homogenous population whose propensity scores are tightly clustered at the low end of the distribution. For the logistic model, their scores have a *mean of 0.010* and a *maximum value of only 0.194*. This indicates the model correctly identifies these units as having a very low probability of receiving a UN intervention.

In sharp contrast, the treated group (N = 16) is extremely heterogeneous. Their propensity scores are widely dispersed across the full spectrum, ranging from 0.000 to 0.884. This shows that the treated units, as expected, had a much higher predicted probability of intervention.

The core problem lies in the disconnect between these two distributions. The common support - the region where both treated and control units exist - is effectively confined *to the narrow 0.000 - 0.194 range*. Consequently, any treated unit with a propensity score above 0.194 *has no comparable units available for matching*.

The density plots visually confirm this. We observe a massive, high-density spike at zero for the control group, while the small treated group forms a low, flat distribution that is almost invisible by comparison. The plot's y-axis is scaled to fit the control group, which visually obscures the treated group, and perfectly illustrates the extreme imbalance.

This lack of overlap will probably have direct implications for Q2. When we apply the matching alogrithm with a 0.2 caliper, we must anticipate that a large portion of our 16 treated units will fail to find a match and will be discarded. This finding illustrates that the conflicts the UN intervened in are, based on the covariates in the dataset, fundamentally different from those it did not.

2.6 Q2. Implement 1:1 Nearest-Neighbor Matching

2.6.1 Matching Setup

For each estimated propensity score $\hat{e}(X_i)$, match each treated unit to the nearest control on the **logit of the propensity score**:

$$\ell_i = \log\left(\frac{\hat{e}(X_i)}{1 - \hat{e}(X_i)}\right)$$

Use **replacement** and a **caliper** of $0.2 \times \mathrm{SD}(\ell_i)$ to restrict poor matches.

2.6.2 Tasks:

1. Implement matching using both logit and RF propensity scores
2. Report how many treated units fail to find a match
3. How does this change the estimand?

```r
# Logit-Score Variables
dta_new$logit_ps_logit <- log(dta_new$ps_logistic_probs / (1 - dta_new$ps_logistic_probs))
dta_new$logit_ps_rf    <- log(dta_new$ps_random_forest / (1 - dta_new$ps_random_forest))


# Our RF model likely has p-scores of 0 or 1. log(0) = -Inf. log(1/0) = Inf.
# We must remove these, as they are unmatchable.
dta_new$logit_ps_logit[!is.finite(dta_new$logit_ps_logit)] <- NA
dta_new$logit_ps_rf[!is.finite(dta_new$logit_ps_rf)]       <- NA

# MatchIt requires a dataset with no NAs in the variables used.
dta_complete <- na.omit(dta_new)
```

```r
# Match using Logistic Regression Scores
m_logit <- matchit(
  treat ~ 1,                          # Formula: treatment ~ no covariates
  data = dta_complete,
  distance = dta_complete$logit_ps_logit, # Use our pre-built logit-score
  method = "nearest",               # Use nearest neighbor
  ratio = 1,                          # 1:1 matching
  replace = TRUE,                     # Use replacement
  caliper = 0.2,                      # Caliper width
  std.caliper = TRUE                  # TRUE = 0.2 * SD(distance)
)

# Match using Random Forest Scores
m_rf <- matchit(
  treat ~ 1,
  data = dta_complete,
  distance = dta_complete$logit_ps_rf, # Use our pre-built RF logit-score
  method = "nearest",
  ratio = 1,
  replace = TRUE,
  caliper = 0.2,
  std.caliper = TRUE
)
```

```r
# Summary for Logit Match
summary(m_logit)
```

```
##
## Call:
## matchit(formula = treat ~ 1, data = dta_complete, method = "nearest",
##     distance = dta_complete$logit_ps_logit, replace = TRUE, caliper = 0.2,
##     std.caliper = TRUE, ratio = 1)
##
## Summary of Balance for All Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance       -2.3918       -5.5491          1.1479     1.5583    0.3321
##          eCDF Max
## distance   0.6182
##
## Summary of Balance for Matched Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance       -3.9782       -4.0406          0.0227     1.0314    0.0072
##          eCDF Max Std. Pair Dist.
## distance   0.2222          0.0303
##
## Sample Sizes:
##               Control Treated
## All            139.        15
## Matched (ESS)    7.36       9
## Matched          8.         9
## Unmatched      131.         6
## Discarded        0.         0
```

```r
# Summary for Random Forest Match
summary(m_rf)
```

```
##
## Call:
## matchit(formula = treat ~ 1, data = dta_complete, method = "nearest",
##     distance = dta_complete$logit_ps_rf, replace = TRUE, caliper = 0.2,
##     std.caliper = TRUE, ratio = 1)
##
## Summary of Balance for All Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance       -1.3546       -4.9444          2.6185     1.4654    0.4931
##          eCDF Max
## distance   0.8542
##
## Summary of Balance for Matched Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance       -2.2244       -2.2778           0.039     0.9274    0.0108
##          eCDF Max Std. Pair Dist.
## distance    0.375          0.0555
##
## Sample Sizes:
##               Control Treated
## All            139.        15
## Matched (ESS)    4.57       8
## Matched          6.         8
## Unmatched      133.         7
## Discarded        0.         0
```

Discussion:

Matching on the logit of the propensity score with a 0.2 caliper leads to a loss of treated observations: out of 15 treated conflicts, only 9 (logit) or 8 (RF) find an acceptable match. The remaining ca. 45% of treated units lie outside the region of common support and are discarded to avoid bias.

As a result, the estimand changes. Instead of the ATT for all treated wars, we now estimate the ATT for the subset of treated conflicts whose observed characteristics overlap with those of the controls. This means our causal conclusion applies only to "matchable" conflicts. The matched ATT is therefore internally valid, but less externally generalizable.

2.7 Q3. Standardized Mean Differences (SMDs)

2.7.1 SMD Formulas

For each covariate X^k:

Before matching (ATT version):

$$\text{SMD}_{\text{raw}}(k) = \frac{\bar{X}^k_{T=1} - \bar{X}^k_{T=0}}{\sqrt{s^{2,k}_{T=1}}}$$

After matching:

$$\text{SMD}_{\text{match}}(k) = \frac{\bar{X}^{k,\text{treated}}_{\text{match}} - \bar{X}^{k,\text{control}}_{\text{match}}}{\sqrt{s^{2,k}_{T=1}}}$$

2.7.2 Tasks:

1. Compute SMDs before and after matching for all covariates
2. Create a Love plot showing balance before and after matching (both methods)
3. Add vertical line at 0.1 (acceptable threshold)
4. Comment on which design achieves better covariate balance
5. Create two additional Love plots including interactions and squared terms

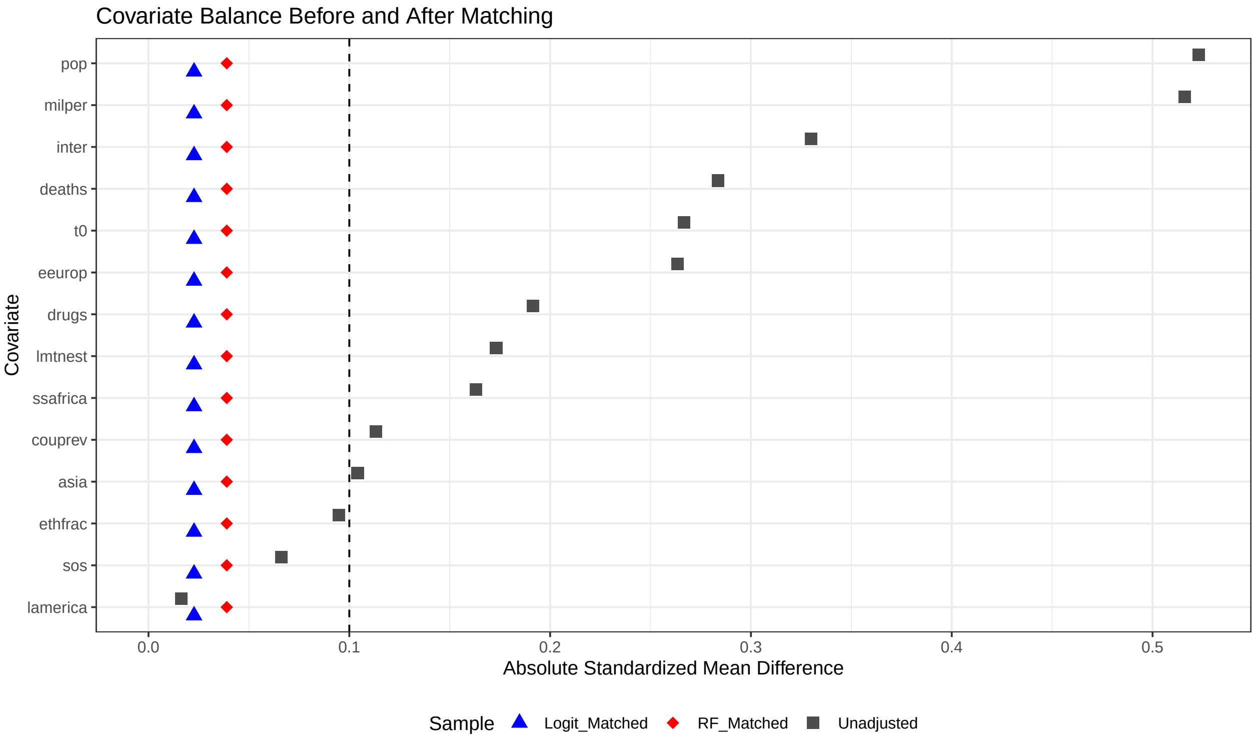
```
# Raw SMDs (before matching)
bal_raw <- bal.tab(
  treat ~ inter + deaths + couprev + sos + drugs + t0 +
    ethfrac + pop + lmtnest + milper +
    eeurop + lamerica + asia + ssafrica,
  data = dta_new,
  estimand = "ATT",
  quick = FALSE
)


# SMDs after logit matching
bal_logit <- bal.tab(m_logit, un = TRUE)

# SMDs after RF matching
bal_rf <- bal.tab(m_rf, un = TRUE)

smd_df <- data.frame(Covariate = rownames(bal_raw$Balance),
                     Raw = bal_raw$Balance$Diff.Un,
                     Logit_Matched = bal_logit$Balance$Diff.Adj,
                     RF_Matched = bal_rf$Balance$Diff.Adj)

knitr::kable(smd_df, digits = 3,
             caption = "Standardized Mean Differences Before and After Matching")
```

Table 3: Standardized Mean Differences Before and After Matching

Covariate	Raw	Logit_Matched	RF_Matched
inter	0.337	0.023	0.039
deaths	0.175	0.023	0.039
couprev	0.014	0.023	0.039
sos	-0.225	0.023	0.039
drugs	0.108	0.023	0.039
t0	-0.511	0.023	0.039
ethfrac	-0.363	0.023	0.039
pop	-0.971	0.023	0.039
lmtnest	-0.071	0.023	0.039
milper	-0.764	0.023	0.039
eeurop	0.318	0.023	0.039
lamerica	-0.049	0.023	0.039
asia	-0.190	0.023	0.039
ssafrica	-0.020	0.023	0.039

```
# Create three SEPARATE balance objects

# Object 1: Raw (Unmatched) Data
# We must specify estimand = "ATT" to get the right SMD denominator
bal_raw <- bal.tab(
  treat ~ inter + deaths + couprev + sos + drugs + t0 +
    ethfrac + pop + lmtnest + milper +
    eeurop + lamerica + asia + ssafrica,
  data = dta_complete,
  estimand = "ATT"
)

# Object 2: Logit-Matched Data
bal_logit <- bal.tab(m_logit)

# Object 3: RF-Matched Data
bal_rf <- bal.tab(m_rf)
```

```
# Create the data frame
smd_df <- data.frame(
  Covariate = rownames(bal_raw$Balance),
  Unadjusted = bal_raw$Balance$Diff.Un,
  Logit_Matched = bal_logit$Balance$Diff.Adj,
  RF_Matched = bal_rf$Balance$Diff.Adj
)

# Reshape data from wide to long format
smd_long <- smd_df %>%
  pivot_longer(
    cols = -Covariate,
    names_to = "Sample",
    values_to = "SMD"     # New column for the SMD value
  ) %>%
  # We want to plot the absolute value
  mutate(Absolute_SMD = abs(SMD))


# Order Covariates by Unadjusted SMD
# We want the y-axis to be sorted from largest imbalance (top)
# to smallest (bottom).

# Get the vector of covariate names in the desired order
covariate_order <- smd_df %>%
  mutate(Absolute_SMD = abs(Unadjusted)) %>%
  arrange(Absolute_SMD) %>%
  pull(Covariate)

# Create the 'long' data frame for plotting
smd_long_ordered <- smd_long %>%
  mutate(Covariate = factor(Covariate, levels = covariate_order))


# Create love plot with ggplot
ggplot(smd_long_ordered, aes(x = Absolute_SMD, y = Covariate, color = Sample, shape = Sample)) +
  geom_point(size = 3, position = position_dodge(width = 0.6)) +
  geom_vline(xintercept = 0.1, linetype = "dashed", color = "black") +
  labs(
    title = "Covariate Balance Before and After Matching",
    x = "Absolute Standardized Mean Difference",
    y = "Covariate"
  ) +
  scale_color_manual(
    name = "Sample",
    values = c("Unadjusted" = "grey30", "Logit_Matched" = "blue", "RF_Matched" = "red")
  ) +
  scale_shape_manual(
    name = "Sample",
    values = c("Unadjusted" = 15, "Logit_Matched" = 17, "RF_Matched" = 18)
  ) +
  theme_bw() +
  theme(legend.position = "bottom") +
  coord_cartesian(xlim = c(0, NA))
```



```
# Love plot: Including interactions
```

```
# Love plot: Including squared terms
```

Interpretation:

[Which method achieves better balance? Are all SMDs below 0.1?]

2.8 Q4. Overlap

2.8.1 Tasks:

1. For each method, report:
   - Min and max of $\hat{e}(X_i)$ for treated and controls
   - Proportion of treated units whose $\hat{e}(X_i)$ lies inside the support of controls (and vice versa)

2. Plot distributions of $\hat{e}(X_i)$ for treated and controls
3. Identify regions of poor overlap or extreme propensities
4. (Optional) Trim observations outside common support and re-compute ATT
5. Examine matched subsets - do matches seem like fair counterfactuals?

```
# Min/max propensity scores by treatment group


# Proportion in common support
```

```
# Plot propensity score distributions
```

```
# Optional: Trim and re-estimate
```

Discussion:

[Is there good overlap? Which observations are on the edge of common support?]

2.9 Q5. Matched-Pair ATT

2.9.1 ATT Estimator

Let each matched pair be denoted by $(i, j(i))$ where i is treated and $j(i)$ is its matched control.

The **average treatment effect on the treated** is:

$$\hat{\tau}_{\text{ATT}} = \frac{1}{N_T^*} \sum_{i \in \mathcal{T}^*} \left(Y_i - Y_{j(i)} \right)$$

where $\mathcal{T}^*$ is the set of treated units with a valid match.

2.9.2 Task:

Compute the ATT for both matching methods.

```
# ATT using logit matching


# ATT using RF matching
```

Interpretation:

[What is the estimated effect of UN intervention on conflict duration?]

2.10 Q5.5. Bias–Variance Tradeoff in Matching Ratios

2.10.1 (a) Conceptual Question

For 1-to-m nearest-neighbor matching without replacement, the ATT estimator is:

$$\hat{\tau}_{\text{ATT}}^{(m)} = \frac{1}{N_T^*} \sum_{i \in \mathcal{T}^*} \left(Y_i - \frac{1}{m} \sum_{j \in \mathcal{J}(i)} Y_j \right)$$

where $\mathcal{J}(i)$ is the set of the m closest control matches for treated unit i.

Tasks:

1. Explain why increasing m tends to:
   - **Decrease variance**
   - **Increase bias**
2. Discuss how this relates to distance in covariate space
3. If overlap is weak, which risk dominates as m grows?

Discussion:

[Your explanation of the bias-variance tradeoff here]

2.10.2 (b) Practical Exercise

Tasks:

1. Re-run matching for 1:1, 2:1, and 3:1 ratios (with replacement and same caliper)
2. Record: number matched, mean distance, ATT estimate
3. Compute cluster-robust standard errors for each design
4. Create results table
5. Plot ATT vs. m with ±1.96 SE error bars

```
# 1:1 matching


# 2:1 matching


# 3:1 matching
```

```
# Create comparison table
```

```
# Plot ATT by matching ratio with error bars
```

Interpretation:

[Do results display expected bias-variance pattern?]

2.10.3 (c) Discussion

Tasks:

- Which design (1:1, 2:1, or 3:1) is most appropriate?
- How does observed pattern relate to Abadie & Imbens (2006)?
- What would happen with infinite data and perfect overlap?

Discussion:

[Your analysis here]

2.11 Q6. Robust Post-Matching Inference (Abadie & Spiess, 2021)

2.11.1 Regression with Cluster-Robust Standard Errors

After matching, fit the regression:

$$Y_i = \alpha + \tau T_i + \varepsilon_i$$

using only matched data.

Let $s(i)$ denote the **subclass (pair id)** of observation i.

Compute **cluster-robust standard errors** for $\hat{\tau}$ by clustering on $s(i)$:

$$\widehat{V}_{\text{CR}}(\hat{\tau}) = (X'X)^{-1}\left(\sum_s X_s'\hat{\varepsilon}_s\hat{\varepsilon}_s'X_s\right)(X'X)^{-1}$$

2.11.2 Tasks:

1. Report $\hat{\tau}$ and its cluster-robust standard error
2. Compare results for logit-matched and RF-matched samples

```
# Regression on logit-matched data with cluster-robust SE
```

```
# Regression on RF-matched data with cluster-robust SE
```

```
# Compare results
```

Interpretation:

[Compare point estimates and standard errors across methods]

2.12 Q7. (Optional) Bootstrap Check

2.12.1 Matched-Pair Bootstrap

Warning: Bootstraps are not theoretically valid for matching estimators, but this serves as a check.

Tasks:

1. Resample matched pairs (subclasses) with replacement
2. Recompute $\hat{\tau}^{(b)}$ for each bootstrap sample $b = 1, \dots, B$
3. Report bootstrap mean, SD, and percentile 95% CI
4. Compare to cluster-robust results

```
# Bootstrap procedure
```

Discussion:

[Do bootstrap and cluster-robust results tell a similar story?]

2.13 Q8. Reflection

2.13.1 Tasks:

1. Why does the propensity score $e(X_i)$ act as a **balancing score**?
2. How does random-forest estimation of $e(X_i)$ change matching results compared to logistic regression?
3. Why is overlap $(0 < e(X_i) < 1)$ necessary for identifying the ATT?

Discussion:

[Your reflection here]

3 Part II: Synthetic Control - German Reunification Study

3.1 Background

In 1990, West Germany underwent reunification with East Germany. The question: *What was the economic cost (or benefit) of this event on West Germany's GDP per capita?*

Using the synthetic control method, we construct a counterfactual "synthetic West Germany" from a weighted combination of other OECD countries.

Paper: Abadie, Diamond & Hainmueller (2015), *Comparative Politics and the Synthetic Control Method*, AJPS.

Dataset: Available via Harvard Dataverse (doi:10.7910/DVN/24714)

3.2 Load Data

```r
# Read German reunification dataset
# Load the replication dataset
load("repgermany.RData")
repgermany <- x

# Display structure
#str(repgermany)
#head(repgermany)
```

3.3 (a) Conceptual Questions

3.3.1 Tasks:

1. Explain the intuition behind the synthetic control method. What kind of assignment problem does it address?
2. Why is it particularly suitable for the West Germany case?
3. What is the key identification assumption?

Discussion:

1.)

The synthetic control method serves as a way to bridge the quantitative/qualitative divide in comparative politics and possibly all of social science research, helping to make precise quantitative inference in small-sample studies. Usually, small sample comparative case studies uncover evidence at a level of high granularity that is impossible to establish in large-scale quantitative studies, which in turn provide precise numerical results that can be compared across studies. For these reasons calls became loud to combine the best of both worlds.

The goal of most study designs is to compare outcomes between treated units, that are the main objects of study, and similar but unaffected control units. The comparison units are intended to reproduce the counterfactual of the treated units in the absence of the treatment. When we try to make quantitative causal inference in small sample comparative studies, we usually do not fail due to the small sample size itself, but due to the missing mechanism of how to find suitable comparison units for the small pool of units we have.

Especially in comparative political science, we often look at aggregate entities, such as states, countries or regions, for which obvious single comparisons often do not exist. In that case we often have one aggregate unit receiving treatment, where we try to select a set of similar untreated units from all of the untreated cases ("donor pool") to approximate the counterfactual. Standard designs based on selection on observables try to remove bias by controlling for measurable covariates. This approach fails when unobserved factors confound the result. This is particularly prevalent in comparative political research, as we study large entities as treatment units and control units, being made up of many individuals.

However, the synthetic control method is based on the premise that when the units of analysis are a few aggregate entities, a combination of comparison units (the synthetic control) often does a better job of reproducing the characteristics of the unit or units representing the case of interest than any single comparison unit alone. The comparison unit in the synthetic control method is selected as the weighted average of all potential comparison units. This weighted average is optimized to best reproduce the treated unit's pre-treatment outcome path and other key predictors. The optimally-weighted average is what serves as the counterfactual, which is custom-built to be the treated unit's 'doppelgänger' before treatment occurred. The core intuition is that by replicating the pre-treatment trend, the synthetic control also replicates the unobserved factors that drove that trend, thus creating a valid counterfactual.

2.)

The case of West Germany is particularly suitable because no single country could serve as a good counterfactual comparison on its own. As an aggregate entity (a country), single comparisons often do not exist in comparative political science. The authors demonstrated that no single country from the donor pool (16 OECD countries) could closely approximate the values of pre-reunification economic growth predictors for West Germany. This lack of a good single match perfectly aligns with the synthetic control method's motivation, namely to construct a composite counterfactual from a weighted average of the donor pool. This weighted average (a combination of a few OECD countries) provided a very close approximation to West Germany's economic attributes and pre-treatment outcome path prior to 1990.

3.)

The key identification assumption of the synthetic control method is rooted in the belief that similar pre-treatment trajectories imply similar causal structures. The authors assume that if the unit representing the case of interest and the synthetic control exhibit similar outcomes over an extended period of time prior to the intervention, then any discrepancy in the outcome variable following the intervention is interpreted as a product of the intervention itself. The ultimate goal is for the synthetic control to accurately reproduce the unobserved counterfactual outcome path that the treated unit (West Germany) would have experienced in the absence of the treatment (reunification). Therefore, the method aims to find a weighted average of multiple control units that collectively replicates the entire pre-treatment trajectory of the outcome variable (Y over t), in addition to matching on key predictor variables (X).

3.4 (b) Mathematical/Optimization Questions

3.4.1 The Optimization Problem

The synthetic control method solves:

$$\min_{w} \sum_{t \leq T_0} \left(Y_{1t} - \sum_{j=2}^{J+1} w_j Y_{jt} \right)^2$$

subject to:

$$w_j \geq 0, \quad \sum_{j} w_j = 1$$

3.4.2 Tasks:

1. Write and explain each term in the optimization problem

2. What role do v-weights play in predictor balancing?
3. Why is the convex-combination constraint important? What if weights could be negative or sum $\neq 1$?

Mathematical Discussion:

The optimization problem in synthetic control is the problem to find the ideal set of weights to build the most similar looking control group to the treatment group pre-treatment. To answer the above questions in more detail, we first have to define the meaning of the underlying terminology.

The Synthetic Control

First of all, we have a sample of $J+1$ units of analysis (e.g. countries), where $j=1$ is the case of interest ("treated unit"), and the units ($j=2$ to $j=J$) define the "Donor pool" of potential comparison units. The goal of the Synthetic control is to find a weighted average of the donor pool, that is a $(J \times 1)$ vector of weights $W=(w_2,...,w_{J+1})'$, with $0 \leq w_j \leq 1$ for $j=2,...,J$ and $w_2+...+w_{J+1}=1$, that the characteristics of the treated unit is best resembled by the synthetic control. Let X_1 be a $(k \times 1)$ vector of preintervention characteristics for the treated unit and X_0 be a $(k \times J)$ matrix of the same variables for the donor pool ($j=2$ to $j=J$). The optimization problem is defined as to find a W^* that minimizes $\|X_1 - X_0W\|$, the difference between the treated unit's characteristics and the synthetic control.

Operationalization Given $m=1,...,k$, X_{1m} is the value of the m-th variable for the treated unit and X_{0m} : is a $(1 \times J)$ vector containing the values of the m-th variable for the units in the donor pool. W^* is chosen to minimize:

$\sum_{m=1}^{k} v_m (X_{1m} - X_{0m}W)^2$

Regarding Question 2.)

v_m is a wight that reflects the relative importance assigned to each of the predictor variables, when the difference between X_1 and X_0W is measured. It is crucial that the synthetic control closely reproduces the values that variables with a large predictive power on the outcome of interest take for the unit affected by the intervention. Therefore, variables with high predictive power, logically receive high v-weights.

The v-weights therefore scale the influence of a potential mismatch between the pre-treatment case of interest and the synthetic control. The unit weights w_j however, determine the contribution of each single unit ($j=2$ to $j=J$) in our data (or donor pool), to the overall value of that value in the synthetic control condition. Therefore, the v-weights, scale the relative importance of the w_j weights for a variable m. Selecting a high value for v, also leads to a stronger influence of an individual units (j) difference between treatment and synthetic control. Where variables are assumed to receive a higher weight, values of w_j should also be chosen more carefully. Selecting low values for v, leads to lower influence of difference in pre-treatment characteristics of the treated and the synthetic control.

The Synthetic Control Estimator

Let Y_{it} denote the outcome of unit j at time t, Y_1 be a $(T_1 \times 1)$ vector collecting the post-intervention values of the outcome for the treated unit, and Y_0 be a $(T_1 \times J)$ matrix where column j contains the post-intervention values of the outcome for unit $j+1$. For a post-intervention period t (with $t \geq T_0$), the synthetic control estimator of the effect of the treatment is given by the difference between the outcome of the treated unit (Y_{1t}) and the outcome for the synthetic control at that period, which equals the difference between the post-intervention outcomes of the treated unit and the synthetic control $Y_1 - Y_0W^*$:

$$Y_{1t} - \sum_{j=2}^{J+1} w_j^* Y_{jt}$$

Regarding Question 3:

Recalling the premise and the motivation of the synthetic control method, the goal is to construct a combination of comparison units to reproduce the characteristics of the unit or units representing the case of interest. By definition, the comparison units are constructed by taking the weighted average of all potential comparison units that best resemble the characteristics of the case of interest.

Therefore we can answer the question of what would happen if the weights "could" be negative or sum up to different than one, using just some basic statistics definitions. A mathematical average (or mean) is the sum of all numbers in a set divided by the count of how many numbers are in that set. A weighted average is the sum of a set of numbers where each number has first been multiplied by its own "importance" or "weight" (as explained in the paper). When the weights are negative or greater than one, it is no longer a weighted average, but extrapolation, where a weight greater than one helps to project beyond, and a negative weight means to subtract that unit's contribution, creating a value that has a negative effect.

According to the authors, this is exactly what we see in regression-based approaches. Although they use synthetic comparison units having coefficients that sum up to one as well, the regression approach **does not** restrict the coefficients of the linear combination that define the comparison unit to be between zero and one. If we now assume that our individual weights (w_j) in the synthetic control term could also have values beyond $[0, 1]$, this would mean that we would just have a simple linear regression model, whose weights extrapolate to produce a perfect fit, in order to minimize the error. The findings of the paper show, however, that extrapolation is not necessary in the context of this study, since there exists a synthetic control that closely fits the values of the characteristics of the units and that does not extrapolate outside of the support of the data.

3.5 (c) Estimation, Balance Before & After

3.5.1 Tasks:

1. Estimate synthetic control for West Germany over pre-treatment period
2. Compute balance table of key predictors (GDP, trade openness, inflation, schooling, investment) showing treated vs. synthetic mean **before treatment**
3. Report non-zero weights w_j
4. Interpret: which donor countries dominate and why?
5. Assess whether pre-treatment fit is acceptable for credible inference

```r
# Prepare data and identify West Germany's index
west_germany_id <- repgermany %>%
  filter(country == "West Germany") %>%
  pull(index) %>%
  unique()

# Synthetic control estimation (silently)
invisible(capture.output({
  dataprep_out <- dataprep(
    foo = as.data.frame(repgermany),
    predictors = c("gdp", "trade", "infrate", "industry"),
    predictors.op = "mean",
    time.predictors.prior = 1981:1990,
    special.predictors = list(
      list("invest80", 1980:1985, "mean"),
      list("schooling", 1980:1985, "mean")
    ),
    dependent = "gdp",
    unit.variable = "index",
    unit.names.variable = "country",
    time.variable = "year",
    treatment.identifier = west_germany_id,
    controls.identifier = setdiff(unique(repgermany$index), west_germany_id),
    time.optimize.ssr = 1960:1989,
    time.plot = 1960:2003
  )

  # Optimization
  synth_out <- synth(data.prep.obj = dataprep_out, method = "BFGS")
}))
```

```r
# Create balance table for pre-treatment predictors
balance_table <- synth.tab(
  dataprep.res = dataprep_out,
  synth.res = synth_out
)

# Create a formatted balance table
balance_df <- data.frame(
  Predictor = c("GDP per capita", "Trade openness", "Inflation rate",
                "Industry share", "Schooling", "Investment rate"),
  West_Germany = balance_table$tab.pred[, 1],
  Synthetic = balance_table$tab.pred[, 2],
  OECD = balance_table$tab.pred[, 3]
)

# Format the table
knitr::kable(
  balance_df,
  digits = 1,
  col.names = c("Predictor", "West Germany", "Synthetic", "OECD"),
  caption = "Economic Growth Predictor Means Before German Reunification"
)
```

Table 4: Economic Growth Predictor Means Before German Reunification

	Predictor	West Germany	Synthetic	OECD
gdp	GDP per capita	15808.9	15806.4	13669.4
trade	Trade openness	56.8	58.4	59.8
infrate	Inflation rate	2.6	4.3	7.6
industry	Industry share	34.5	34.5	33.8
special.invest80.1980.1985	Schooling	27.0	27.0	25.9
special.schooling.1980.1985	Investment rate	55.5	53.4	38.7

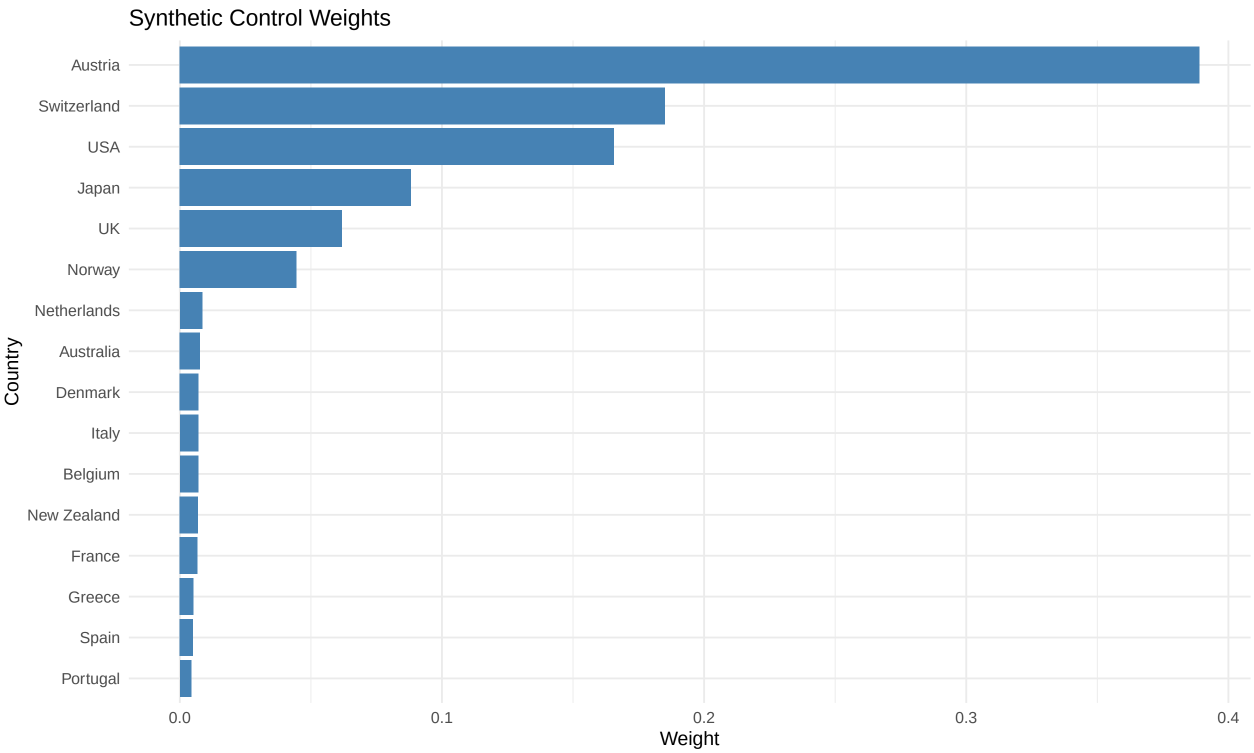
```r
# Report non-zero weights
weights_df <- data.frame(
  Country = dataprep_out$names.and.numbers[-1, 1],
  Weight = as.numeric(synth_out$solution.w)
)

nonzero_weights <- weights_df %>%
  filter(Weight > 0.001) %>%
  arrange(desc(Weight))
```

```r
knitr::kable(
  nonzero_weights,
  digits = 3,
  col.names = c("Country", "Synthetic Control Weight"),
  caption = "Non-Zero Weights for Synthetic West Germany"
)
```

Table 5: Non-Zero Weights for Synthetic West Germany

Country	Synthetic Control Weight
Austria	0.389
Switzerland	0.185
USA	0.166
Japan	0.088
UK	0.062
Norway	0.044
Netherlands	0.009
Australia	0.008
Denmark	0.007
Italy	0.007
Belgium	0.007
New Zealand	0.007
France	0.007
Greece	0.005
Spain	0.005
Portugal	0.004

```r
# Visualize country weights
lookup <- dataprep_out$names.and.numbers
weights_plot_df <- data.frame(
  Unit = as.numeric(rownames(synth_out$solution.w)),
  Weight = synth_out$solution.w[, 1]
) %>%
  left_join(lookup, by = c("Unit" = "unit.numbers")) %>%
  rename(Country = unit.names) %>%
  filter(Weight > 0.001) %>%
  arrange(desc(Weight))

ggplot(weights_plot_df, aes(x = reorder(Country, Weight), y = Weight)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Synthetic Control Weights",
    x = "Country",
    y = "Weight"
  ) +
  theme_minimal()
```



Interpretation:

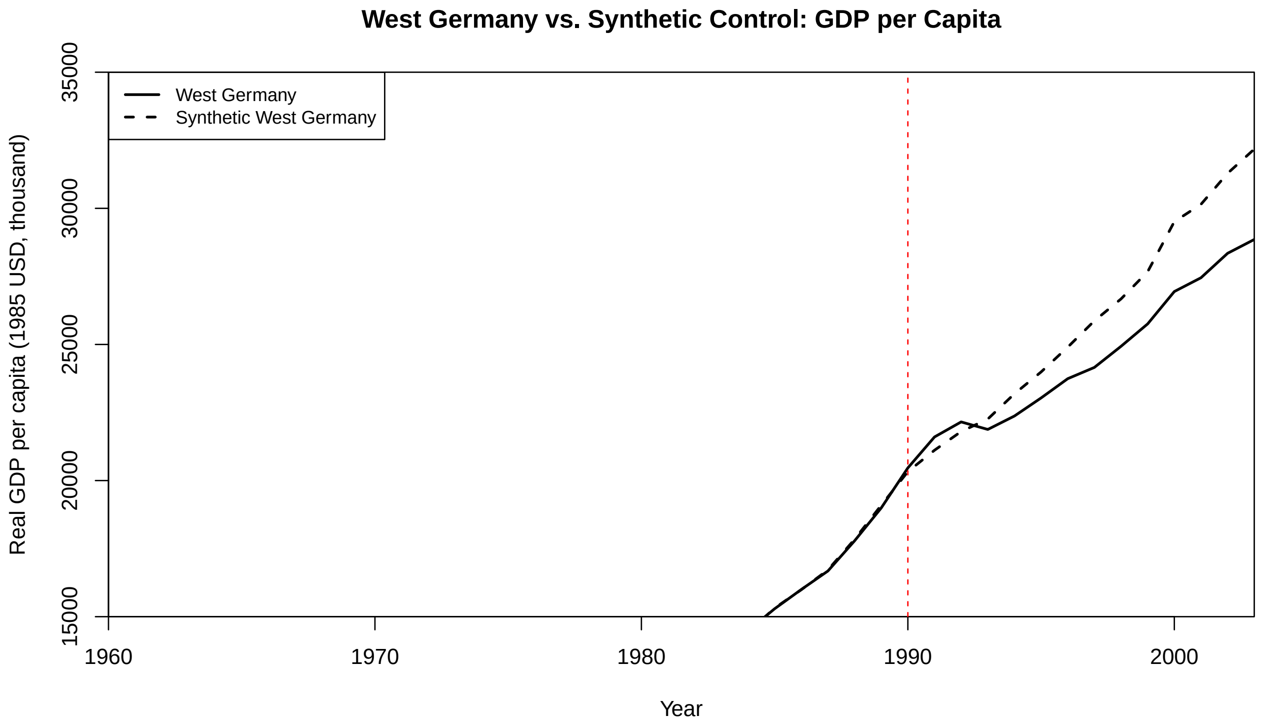
[Which countries contribute most to synthetic West Germany? Is pre-treatment balance good?]

3.6 (d) Effect Size & Permutation Test

3.6.1 Tasks:

1. Plot actual vs. synthetic GDP per capita trajectory (pre- and post-treatment)
2. Calculate estimated effect (gap) in first few post-treatment years and average post-treatment gap
3. Perform **permutation (placebo) test** by reassigning treatment to each control country
4. Report where treated unit's gap falls in the distribution (approximate p-value)
5. Interpret: What does this suggest about the economic impact of reunification?

```r
# Task (d) 1: Plot actual vs synthetic GDP per capita trajectory
path.plot(
  synth.res = synth_out,
  dataprep.res = dataprep_out,
  Ylab = "Real GDP per capita (1985 USD, thousand)",
  Xlab = "Year",
  Ylim = c(15000, 35000),
  Legend = c("West Germany", "Synthetic West Germany"),
  Legend.position = "topleft",
  Main = "West Germany vs. Synthetic Control: GDP per Capita"
)

# Add vertical line at treatment year
abline(v = 1990, lty = 2, col = "red")
```



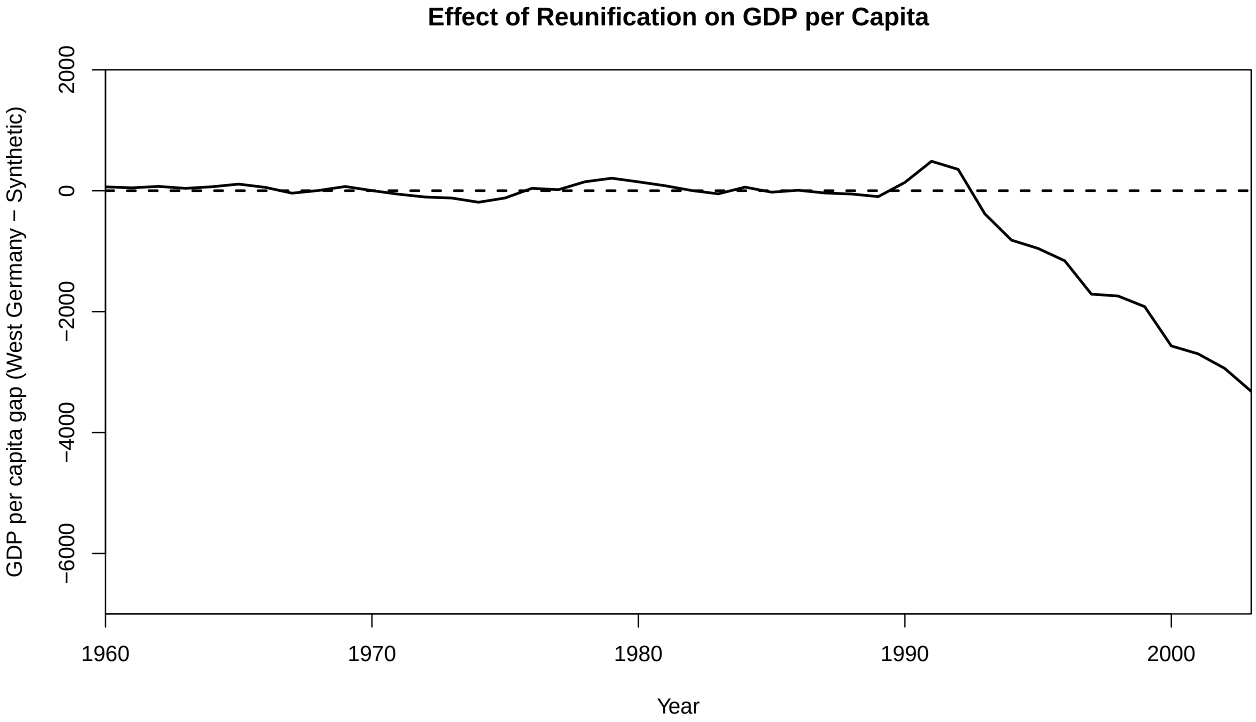
```
# Task (d) 2: Calculate estimated effect gaps

# Calculate the raw gaps (Treated - Synthetic)
gaps <- dataprep_out$Y1plot - (dataprep_out$Y0plot %*% synth_out$solution.w)
years <- as.numeric(rownames(gaps))

# First few post-treatment years (1990-1992)
first_years_idx <- years %in% 1990:1992
avg_gap_first_years <- mean(gaps[first_years_idx])

# Average post-treatment gap (1990-2003)
post_years_idx <- years >= 1990
avg_gap_post <- mean(gaps[post_years_idx])

# Print results
cat("Average gap in first post-treatment years (1990-1992):",
    round(avg_gap_first_years, 2), "USD\n")
```

```
## Average gap in first post-treatment years (1990-1992): 326.02 USD
```

```
cat("Average post-treatment gap (1990-2003):",
    round(avg_gap_post, 2), "USD\n\n")
```

```
## Average post-treatment gap (1990-2003): -1373.17 USD
```

```
# Plot the gap over time
gaps.plot(
  synth.res = synth_out,
  dataprep.res = dataprep_out,
  Ylab = "GDP per capita gap (West Germany - Synthetic)",
  Xlab = "Year",
  Main = "Effect of Reunification on GDP per Capita",
  Ylim = c(-7000, 2000)
)
```



```
# Tasks (d) 3-4: Perform permutation (placebo) test

# Get list of all control units
control_units <- setdiff(unique(repgermany$index), west_germany_id)

# Store results
placebo_gaps <- matrix(NA, nrow = length(years), ncol = length(control_units))
colnames(placebo_gaps) <- control_units
rownames(placebo_gaps) <- years

# Store RMSPE for each placebo
pre_rmspe_placebo <- numeric(length(control_units))
post_rmspe_placebo <- numeric(length(control_units))

cat("Running permutation test for", length(control_units), "control units...\n")
```

```
## Running permutation test for 16 control units...
```

```
# Loop through each control unit and run synthetic control
for (i in 1:length(control_units)) {

  placebo_unit <- control_units[i]

  # Skip if this would cause issues
  tryCatch({

    # Suppress output
    invisible(capture.output({
      # Prepare data with current control unit as "treated"
      dataprep_placebo <- dataprep(
        foo = as.data.frame(repgermany),
        predictors = c("gdp", "trade", "infrate", "industry"),
        predictors.op = "mean",
        time.predictors.prior = 1981:1990,
        special.predictors = list(
          list("invest80", 1980:1985, "mean"),
          list("schooling", 1980:1985, "mean")
        ),
        dependent = "gdp",
        unit.variable = "index",
        unit.names.variable = "country",
        time.variable = "year",
        treatment.identifier = placebo_unit,
        controls.identifier = setdiff(unique(repgermany$index), placebo_unit),
        time.optimize.ssr = 1960:1989,
        time.plot = 1960:2003
      )

      # Run synthetic control
      synth_placebo <- synth(data.prep.obj = dataprep_placebo, method = "BFGS")

    }))

    # Calculate gaps for this placebo
    placebo_gaps[, i] <- dataprep_placebo$Y1plot -
      (dataprep_placebo$Y0plot %*% synth_placebo$solution.w)

    # Calculate RMSPE
    pre_idx <- years < 1990
    post_idx <- years >= 1990
    pre_rmspe_placebo[i] <- sqrt(mean(placebo_gaps[pre_idx, i]^2))
    post_rmspe_placebo[i] <- sqrt(mean(placebo_gaps[post_idx, i]^2))

  }, error = function(e) {
    cat("Error with unit", placebo_unit, "\n")
  })
}

# Calculate RMSPE for actual West Germany
pre_idx <- years < 1990
post_idx <- years >= 1990
pre_rmspe_wg <- sqrt(mean(gaps[pre_idx]^2))
post_rmspe_wg <- sqrt(mean(gaps[post_idx]^2))
rmspe_ratio_wg <- post_rmspe_wg / pre_rmspe_wg

# Calculate RMSPE ratios for placebos
rmspe_ratio_placebo <- post_rmspe_placebo / pre_rmspe_placebo

# Calculate p-value: proportion of placebos with ratio >= West Germany's ratio
p_value <- mean(rmspe_ratio_placebo >= rmspe_ratio_wg, na.rm = TRUE)
```

```r
cat("\n=== PERMUTATION TEST RESULTS ===\n")
```

```
##
## === PERMUTATION TEST RESULTS ===
```

```r
cat("West Germany RMSPE ratio:", round(rmspe_ratio_wg, 3), "\n")
```

```
## West Germany RMSPE ratio: 20.646
```

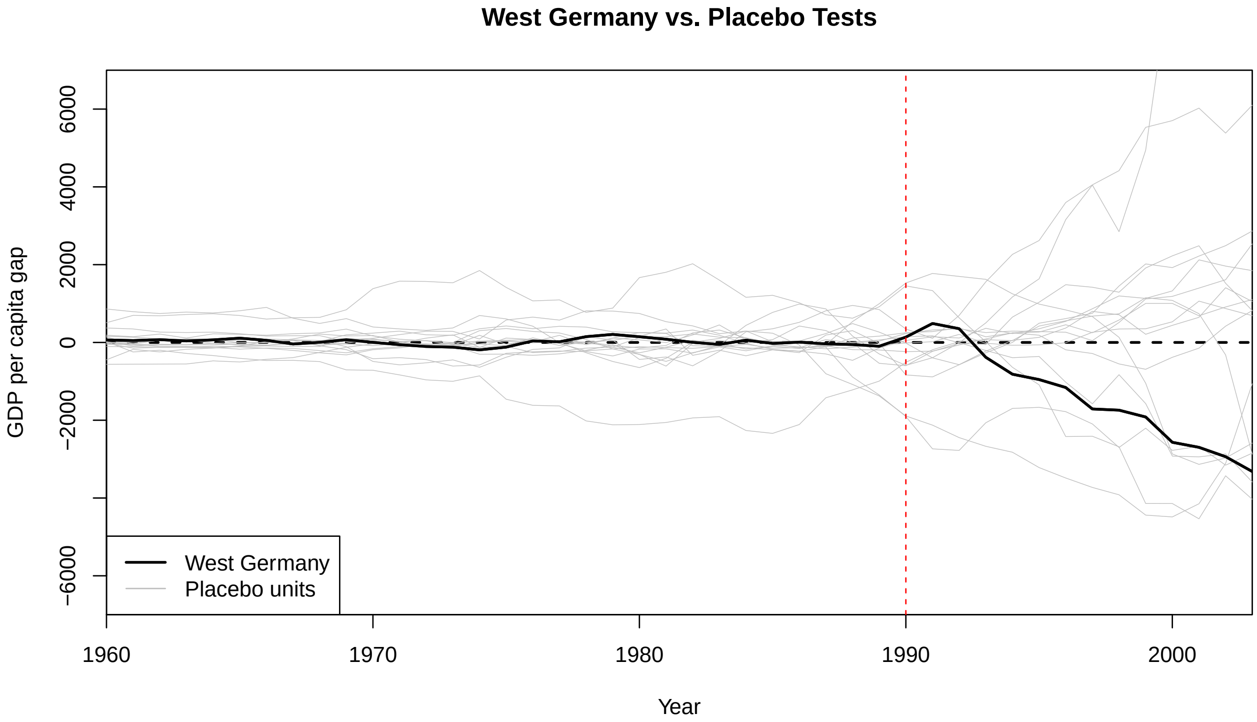
```r
cat("Rank of West Germany:",
    sum(rmspe_ratio_placebo >= rmspe_ratio_wg, na.rm = TRUE) + 1,
    "out of", sum(!is.na(rmspe_ratio_placebo)) + 1, "\n")
```

```
## Rank of West Germany: 1 out of 17
```

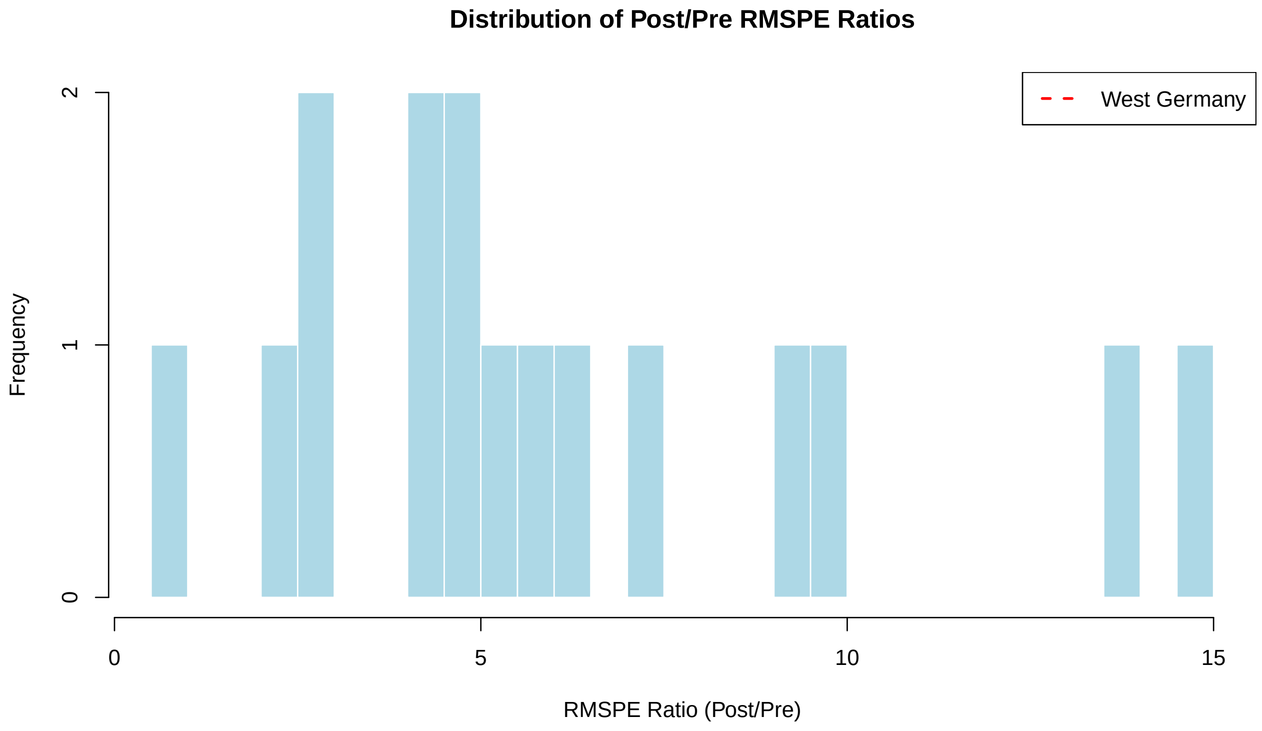
```r
cat("Approximate p-value:", round(p_value, 3), "\n\n")
```

```
## Approximate p-value: 0
```

```r
# Plot: West Germany gap vs all placebo gaps
gaps.plot(
  synth.res = synth_out,
  dataprep.res = dataprep_out,
  Ylab = "GDP per capita gap",
  Xlab = "Year",
  Main = "West Germany vs. Placebo Tests",
  Ylim = c(-7000, 7000)
)

# Add placebo lines
for (i in 1:ncol(placebo_gaps)) {
  lines(years, placebo_gaps[, i], col = "gray", lwd = 0.5)
}

# Re-add West Germany line on top
lines(years, gaps, col = "black", lwd = 2)
abline(v = 1990, lty = 2, col = "red")
legend("bottomleft",
       legend = c("West Germany", "Placebo units"),
       col = c("black", "gray"),
       lwd = c(2, 1))
```



```
# Alternative plot: RMSPE ratio distribution
hist(rmspe_ratio_placebo,
     breaks = 20,
     main = "Distribution of Post/Pre RMSPE Ratios",
     xlab = "RMSPE Ratio (Post/Pre)",
     col = "lightblue",
     border = "white")
abline(v = rmspe_ratio_wg, col = "red", lwd = 2, lty = 2)
legend("topright",
       legend = "West Germany",
       col = "red",
       lwd = 2,
       lty = 2)
```



Interpretation (Task d.5):

[What does this suggest about the economic impact of reunification?]

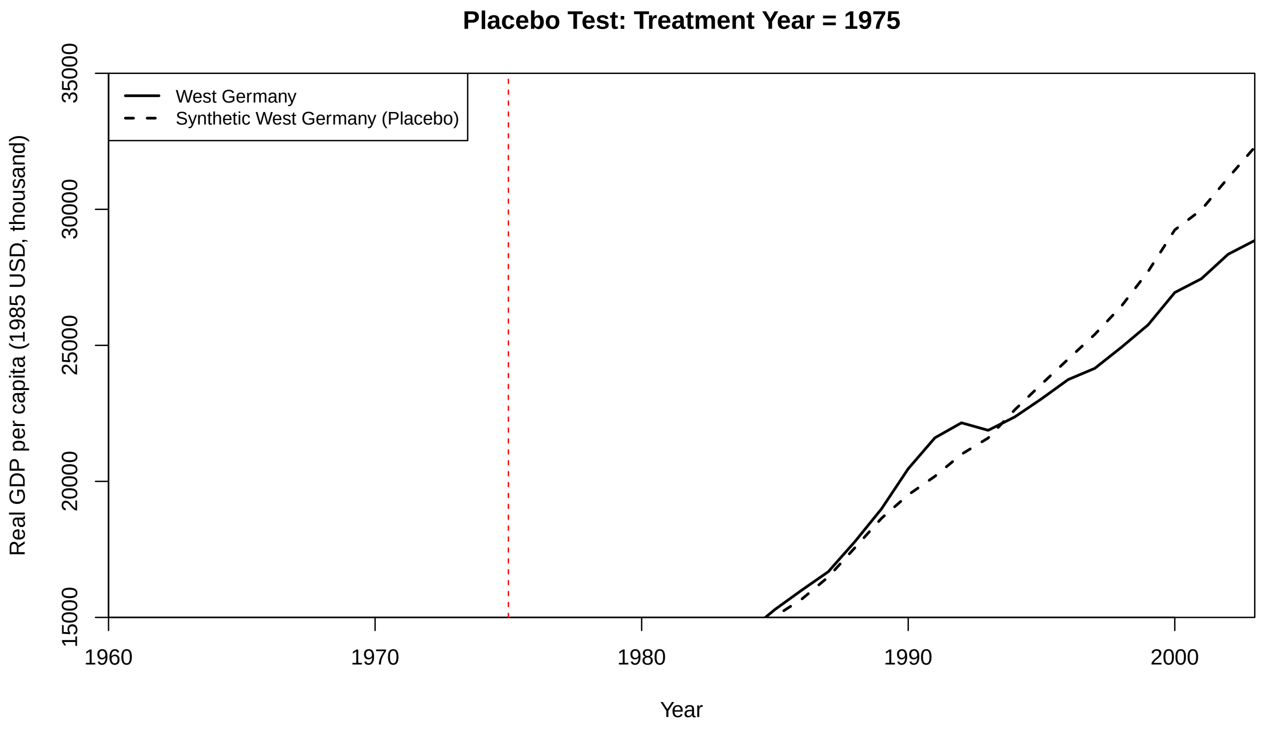
3.7 (e) Placebo Test on Earlier Years

3.7.1 Tasks:

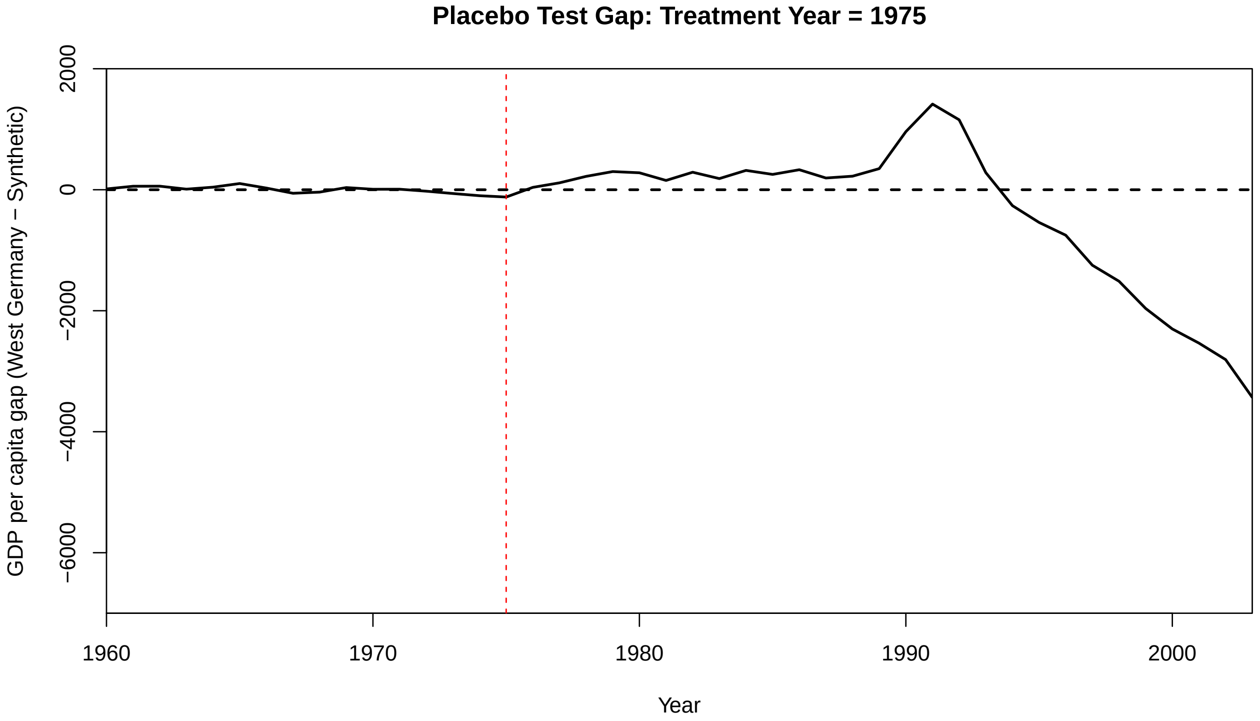
1. Conduct placebo treatment year **before** actual 1990 treatment (e.g., 1975)
2. Re-estimate synthetic control and plot the gap
3. What does pre-treatment gap behavior tell you about parallel-trajectory assumption?
4. Comment on how convincing you find the main causal estimate

```r
# Task (e) 1: Conduct placebo test with earlier treatment year

# Define the placebo year
placebo_year <- 1975

# Prepare data for the placebo run
invisible(capture.output({
  dataprep_placebo_1975 <- dataprep(
    foo = as.data.frame(repgermany),
    predictors = c("gdp", "trade", "infrate", "industry"),
    predictors.op = "mean",
    # Use period prior to placebo year for predictors
    time.predictors.prior = 1965:1974,
    special.predictors = list(
      # Keep original time period for static predictors
      list("invest80", 1980:1985, "mean"),
      list("schooling", 1980:1985, "mean")
    ),
    dependent = "gdp",
    unit.variable = "index",
    unit.names.variable = "country",
    time.variable = "year",
    treatment.identifier = west_germany_id,
    controls.identifier = setdiff(unique(repgermany$index), west_germany_id),
    # Optimize up to placebo year
    time.optimize.ssr = 1960:(placebo_year - 1),
    time.plot = 1960:2003
  )

  # Re-estimate Synthetic Control
  synth_placebo_1975 <- synth(data.prep.obj = dataprep_placebo_1975, method = "BFGS")
}))
```

```
# Task (e) 2: Plot the gap for the placebo run

# Plot trajectory
path.plot(
  synth.res = synth_placebo_1975,
  dataprep.res = dataprep_placebo_1975,
  Ylab = "Real GDP per capita (1985 USD, thousand)",
  Xlab = "Year",
  Ylim = c(15000, 35000),
  Legend = c("West Germany", "Synthetic West Germany (Placebo)"),
  Legend.position = "topleft",
  Main = paste0("Placebo Test: Treatment Year = ", placebo_year)
)
abline(v = placebo_year, lty = 2, col = "red")
```



```r
# Plot the gap
gaps.plot(
  synth.res = synth_placebo_1975,
  dataprep.res = dataprep_placebo_1975,
  Ylab = "GDP per capita gap (West Germany - Synthetic)",
  Xlab = "Year",
  Main = paste0("Placebo Test Gap: Treatment Year = ", placebo_year),
  Ylim = c(-7000, 2000)
)
abline(v = placebo_year, lty = 2, col = "red")
```



```r
# Calculate gaps for placebo test
gaps_placebo_1975 <- dataprep_placebo_1975$Y1plot -
  (dataprep_placebo_1975$Y0plot %*% synth_placebo_1975$solution.w)
years_placebo <- as.numeric(rownames(gaps_placebo_1975))

# Calculate RMSPE before and after placebo treatment
pre_placebo_idx <- years_placebo < placebo_year
post_placebo_idx <- years_placebo >= placebo_year

pre_rmspe_placebo_1975 <- sqrt(mean(gaps_placebo_1975[pre_placebo_idx]^2))
post_rmspe_placebo_1975 <- sqrt(mean(gaps_placebo_1975[post_placebo_idx]^2))
rmspe_ratio_placebo_1975 <- post_rmspe_placebo_1975 / pre_rmspe_placebo_1975

cat("\n=== PLACEBO TEST (1975) RESULTS ===\n")
```

```
##
## === PLACEBO TEST (1975) RESULTS ===
```

```r
cat("Pre-placebo RMSPE (1960-1974):", round(pre_rmspe_placebo_1975, 2), "\n")
```

```
## Pre-placebo RMSPE (1960-1974): 52.09
```

```
cat("Post-placebo RMSPE (1975-2003):", round(post_rmspe_placebo_1975, 2), "\n")
```

```
## Post-placebo RMSPE (1975-2003): 1248.83
```

```
cat("RMSPE Ratio:", round(rmspe_ratio_placebo_1975, 3), "\n\n")
```

```
## RMSPE Ratio: 23.973
```

```
# Compare to actual treatment
cat("For comparison:\n")
```

```
## For comparison:
```

```
cat("Actual treatment (1990) RMSPE ratio:", round(rmspe_ratio_wg, 3), "\n")
```

```
## Actual treatment (1990) RMSPE ratio: 20.646
```

Interpretation (Tasks e.3-4):

[What does pre-treatment gap behavior tell you about parallel-trajectory assumption? How convincing is the main causal estimate?]
