Causal Inference Assignment

Soumik Ghosh Moulic

Minerva Schools at KGI

I would like to thank Michelle Hackl, Oscar Engelbrektson, and Zineb Salimi for helping me clear numerous conceputal issues about the assignment, guiding me through and even debugging my code when needed. This assignment would not have been possible without their help. I am humbled. Thank God.

Question 1

A) We don’t have a Match function. So we do a GenMatch and then directly run the MatchBalance function, but Match function should be there because running a MatchBalance with genout will result in the balance not running because genout just gives us the weights.

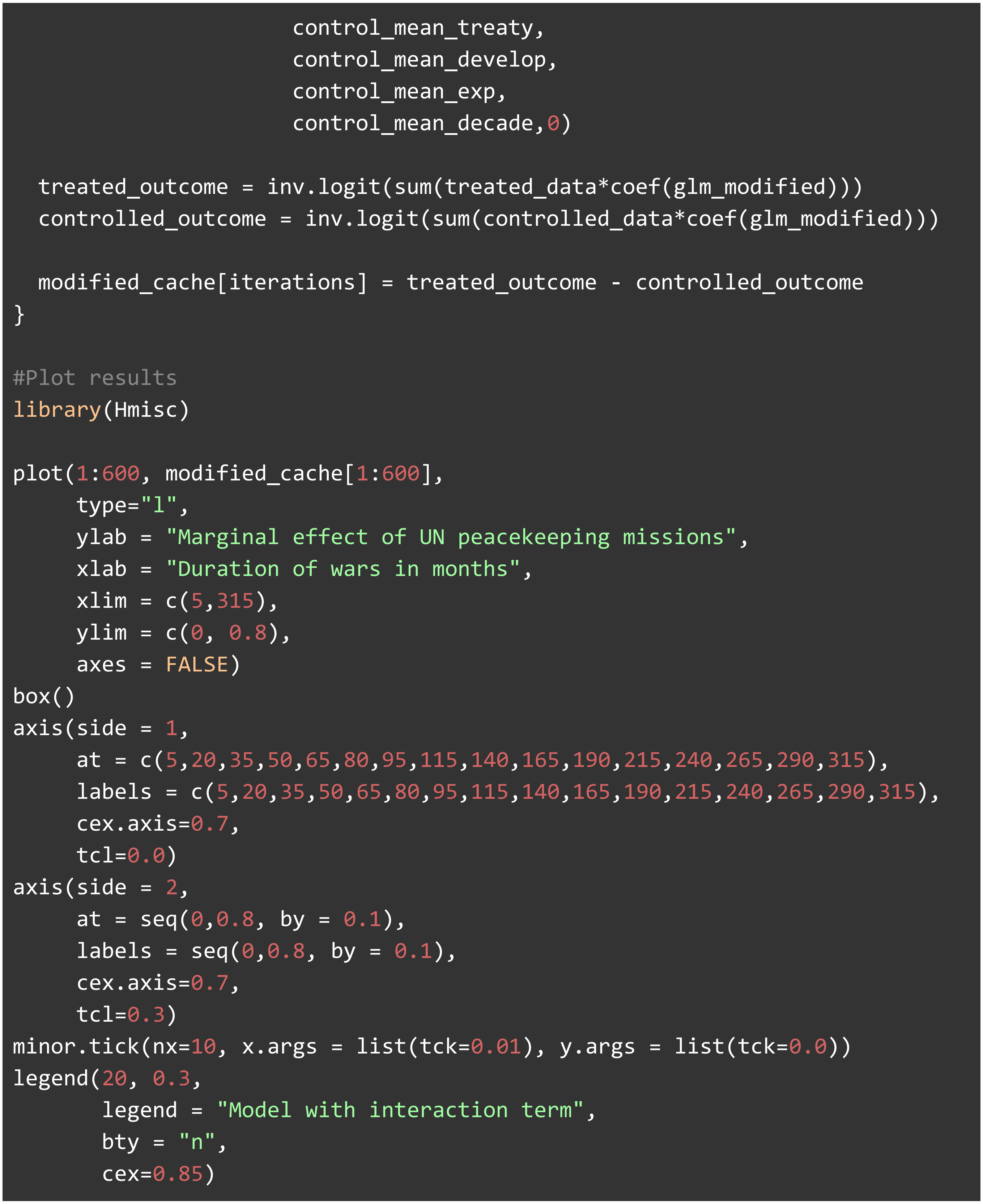
B) We use an ATE estimand in the GenMatch function so we need to define the same estimand for the Match function because match function’s default estimand is ATT. If we don’t do this, then the answer we get from the GenMatch will not be utilized properly by the Match function.

C) Two things, first, we haven't defined additional parameters for the GenMatch function like wait generations or pop size. While these are not mandatory and default values will still run the function, the GenMatch is not optimized and results can be improved. Secondly, the GenMatch function assumes one-to-one matching since M is not defined and the default is one. But in the Match function, M is manually defined as 2, which will create an issue since then the match is done on a one-to-two basis.

Question 2

```
library(Matching)

foo <-
read.csv('https://course-resources.minerva.kgi.edu/uploaded_files/mke/00086677-3767/peace.csv')

# extract relevant columns
foo <- foo[, c(6:8, 11:16, 99, 50, 114, 49, 63, 136, 109, 126, 48, 160,
142, 10)]

# remove 2 rows with missing data (there are better ways to handle missing
data)
foo <- foo[c(-19, -47), ]

#original logit model with no interaction term
glm_original <- glm(pbs2s3 ~ wartype + logcost + wardur + factnum +
factnum2 + trnsfcap + untype4 + treaty + develop + exp + decade, family
                    = binomial, data = foo)
#summary(glm_original)

#logit model with the interaction term wardur*untype4
glm_modified <-  glm(pbs2s3 ~ wartype + logcost + wardur + factnum +
factnum2 + trnsfcap + untype4 + treaty + develop + exp + decade  +
wardur*untype4, family
                     = binomial, data = foo)

#summary(glm_modified)

#From these logit models, we compute the marginal
#effects of UN peacekeeping operations as a function of the duration of the
civil war,
#holding constant all other variables at their means. Figure 8 plots these
results. (PG 207 - https://gking.harvard.edu/files/counterf.pdf)

#First, seperating the treatment and control groups
foo_treatment <- foo[which(foo$untype4==1),]
foo_control <- foo[which(foo$untype4==0),]


#setting other variables at their means in the treated unit
treat_mean_wartype <- mean(foo_treatment$wartype)
treat_mean_logcost <- mean(foo_treatment$logcost)
treat_mean_factnum <- mean(foo_treatment$factnum)
treat_mean_factnum2 <- mean(foo_treatment$factnum2)
treat_mean_trnsfcap <- mean(foo_treatment$trnsfcap)
```

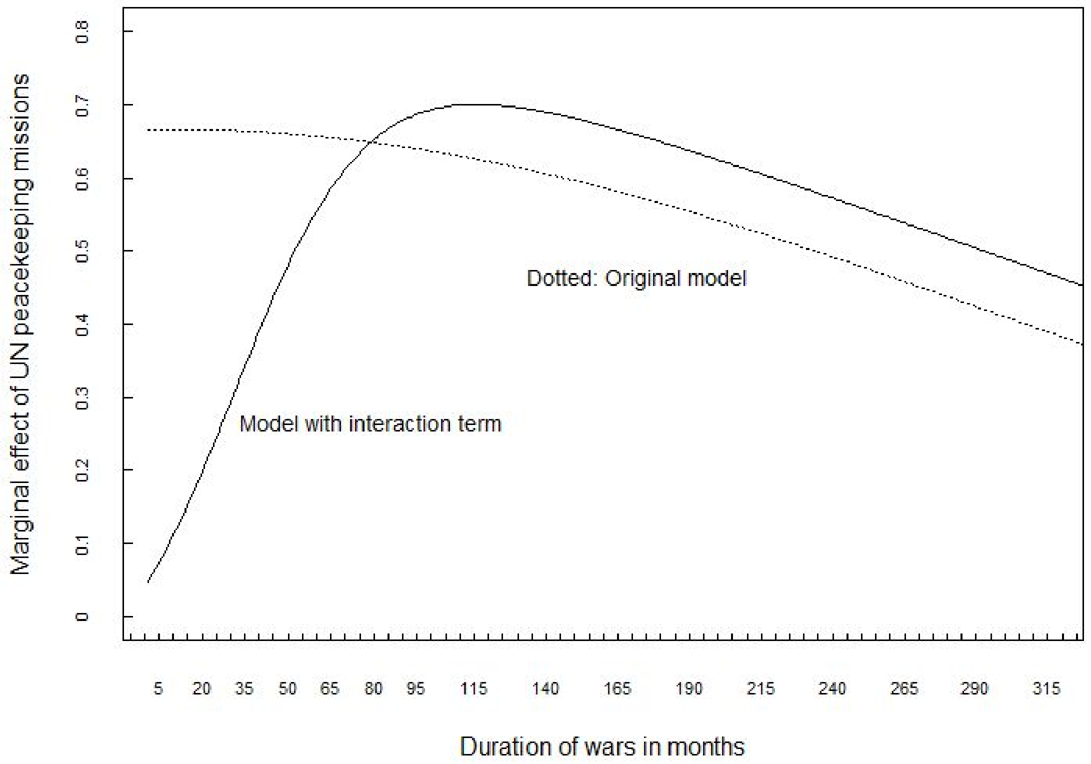
```
treat_mean_untype4 <- mean(foo_treatment$untype4)
treat_mean_treaty <- mean(foo_treatment$treaty)
treat_mean_develop <- mean(foo_treatment$develop)
treat_mean_exp <- mean(foo_treatment$exp)
treat_mean_decade <- mean(foo_treatment$decade)

#setting other variables at their means in the control unit
control_mean_wartype <- mean(foo_control$wartype)
control_mean_logcost <- mean(foo_control$logcost)
control_mean_factnum <- mean(foo_control$factnum)
control_mean_factnum2 <- mean(foo_control$factnum2)
control_mean_trnsfcap <- mean(foo_control$trnsfcap)
control_mean_untype4 <- mean(foo_control$untype4)
control_mean_treaty <- mean(foo_control$treaty)
control_mean_develop <- mean(foo_control$develop)
control_mean_exp <- mean(foo_control$exp)
control_mean_decade <- mean(foo_control$decade)

#creating the original model with inverse logit
library(boot)

original_cache = rep(NA, 600)
for (iterations in min(foo$wardur):max(foo$wardur)){
  treated_data = c(1, treat_mean_wartype,
                   treat_mean_logcost,
                   iterations,
                   treat_mean_factnum,
                   treat_mean_factnum2,
                   treat_mean_trnsfcap, 1,
                   treat_mean_treaty,
                   treat_mean_develop,
                   treat_mean_exp,
                   treat_mean_decade)

  controlled_data = c(1, control_mean_wartype,
                      control_mean_logcost,
                      iterations,
                      control_mean_factnum,
                      control_mean_factnum2,
                      control_mean_trnsfcap,
                      0,
                      control_mean_treaty,
                      control_mean_develop,
                      control_mean_exp,
                      control_mean_decade)

  treated_outcome = inv.logit(sum(treated_data*coef(glm_original)))
  controlled_outcome = inv.logit(sum(controlled_data*coef(glm_original)))

  original_cache[iterations] = treated_outcome - controlled_outcome
}

#doing the same as the original model for the modified model

modified_cache = rep(NA, 600)
for (iterations in min(foo$wardur):max(foo$wardur)){
  treated_data = c(1, treat_mean_wartype,
                   treat_mean_logcost,
                   iterations,
                   treat_mean_factnum,
                   treat_mean_factnum2,
                   treat_mean_trnsfcap, 1,
                   treat_mean_treaty,
                   treat_mean_develop,
                   treat_mean_exp,
                   treat_mean_decade, iterations)

  controlled_data = c(1, control_mean_wartype,
                      control_mean_logcost,
                      iterations,
                      control_mean_factnum,
                      control_mean_factnum2,
                      control_mean_trnsfcap,
                      0,
```



```
                      control_mean_treaty,
                      control_mean_develop,
                      control_mean_exp,
                      control_mean_decade,0)

  treated_outcome = inv.logit(sum(treated_data*coef(glm_modified)))
  controlled_outcome = inv.logit(sum(controlled_data*coef(glm_modified)))

  modified_cache[iterations] = treated_outcome - controlled_outcome
}

#Plot results
library(Hmisc)

plot(1:600, modified_cache[1:600],
     type="l",
     ylab = "Marginal effect of UN peacekeeping missions",
     xlab = "Duration of wars in months",
     xlim = c(5,315),
     ylim = c(0, 0.8),
     axes = FALSE)
box()
axis(side = 1,
     at = c(5,20,35,50,65,80,95,115,140,165,190,215,240,265,290,315),
     labels = c(5,20,35,50,65,80,95,115,140,165,190,215,240,265,290,315),
     cex.axis=0.7,
     tcl=0.0)
axis(side = 2,
     at = seq(0,0.8, by = 0.1),
     labels = seq(0,0.8, by = 0.1),
     cex.axis=0.7,
     tcl=0.3)
minor.tick(nx=10, x.args = list(tck=0.01), y.args = list(tck=0.0))
legend(20, 0.3,
       legend = "Model with interaction term",
       bty = "n",
       cex=0.85)
```

```
legend(120, 0.5,
       legend = "Dotted: Original model",
       bty = "n",
       cex=0.85)
par(new=TRUE)
plot(1:600, original_cache[1:600],
     type="l",
     lty = 3,
     axes = FALSE,
     ylab = "",
     xlab ="",
     xlim = c(5,315),
     ylim = c(0, 0.8))
```

Output



Question 3

The first line will initiate a one-dimensional vector (Tr) with a length equivalent to foo$untype. Then if any element in Tr is not equal to the value of "None", it will be replaced by 1. In simpler words, the treatment (Tr) is untype.

Question 4

A) UN intervening on peacebuilding has a greater effect than no intervention, and intervention for 5 years has a greater effect on the success rate than intervening only for 2 years after the war.

B) Once case, where SUTVA might be violated, is if there is additional interaction/interference across units (countries) that intensifies or dilutes the treatment of the UN.

C)

i)

TABLE 1: Estimating causal effect through different methods

	tmt effect (bias adj)	tmt effect (no bias adj)	p-value (from MatchBalance) with bias adj
logistic regression			
len success 2 years	N.A.	0.5996839	N.A.
len success 5 years	N.A.	0.8233143	N.A.
p-score matching			
len success 2 years	0.3635877	N.A.	0.01
len success 5 years	0.3938908	N.A.	0.006

gen match			
len success 2 years	0.1828086	N.A.	0.024
len success 5 years	0.1828086	N.A.	0.014

Note: A balance matrix was used for GenMatch which had an interaction term foo$develop*foo$treaty other than the normal variables. See appendix (start of Page 15) for the output of the GenMatch.

ii)

Decision Memo for António Guterres

Executive Summary

We wanted to see the impact of UN peace intervention on lenient peacebuilding success over two and five years. We identified three models to evaluate the impact. Models were used in ascending level of accuracy and complexity. First, we tested a simple linear regression model, then we used a propensity score matching based model and finally we used a genetic matching based model with a single interaction between develop and treaty. The results obtained vary from model to model but show a overall positive effect of UN intervention operations.

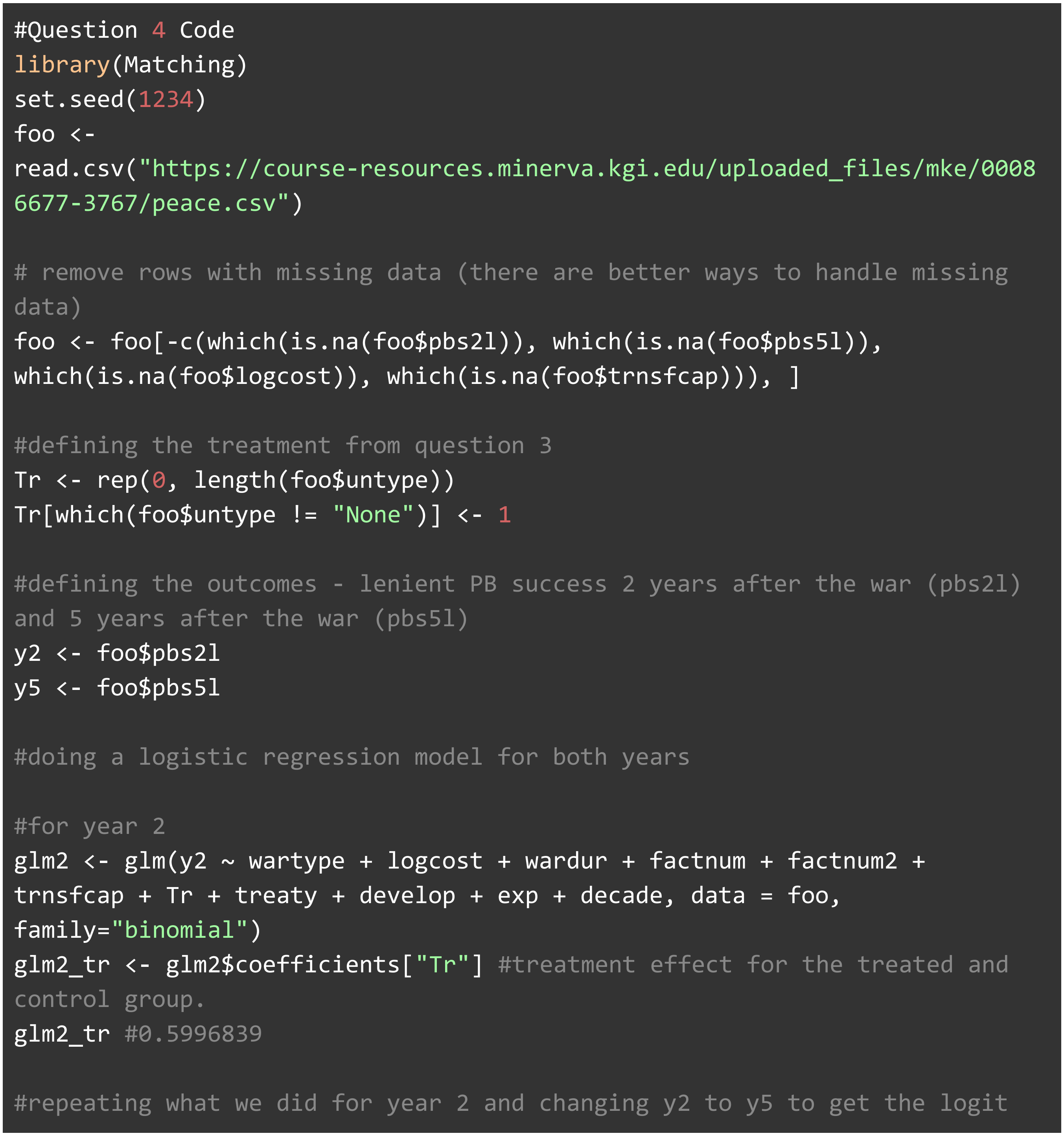
Conclusion

Wanting to look at the impact of the UN peace operation on peacebuilding success over two and five years, three models were chosen to evaluate the impact. The models varied in complexity and accuracy of estimation. All models show a positive effect of UN peace operation on peacebuilding. While the logit and the propensity score model show that there is a higher effect of UN intervention for 5 years than UN intervention for two years, the genetic matching based model doesn’t. It should be noted that there was an involvement of an interaction term between develop and treaty variables in the genetic matching model. Further interactions between different variables should be tested in the models to find out any interdependency and estimate accurate results.

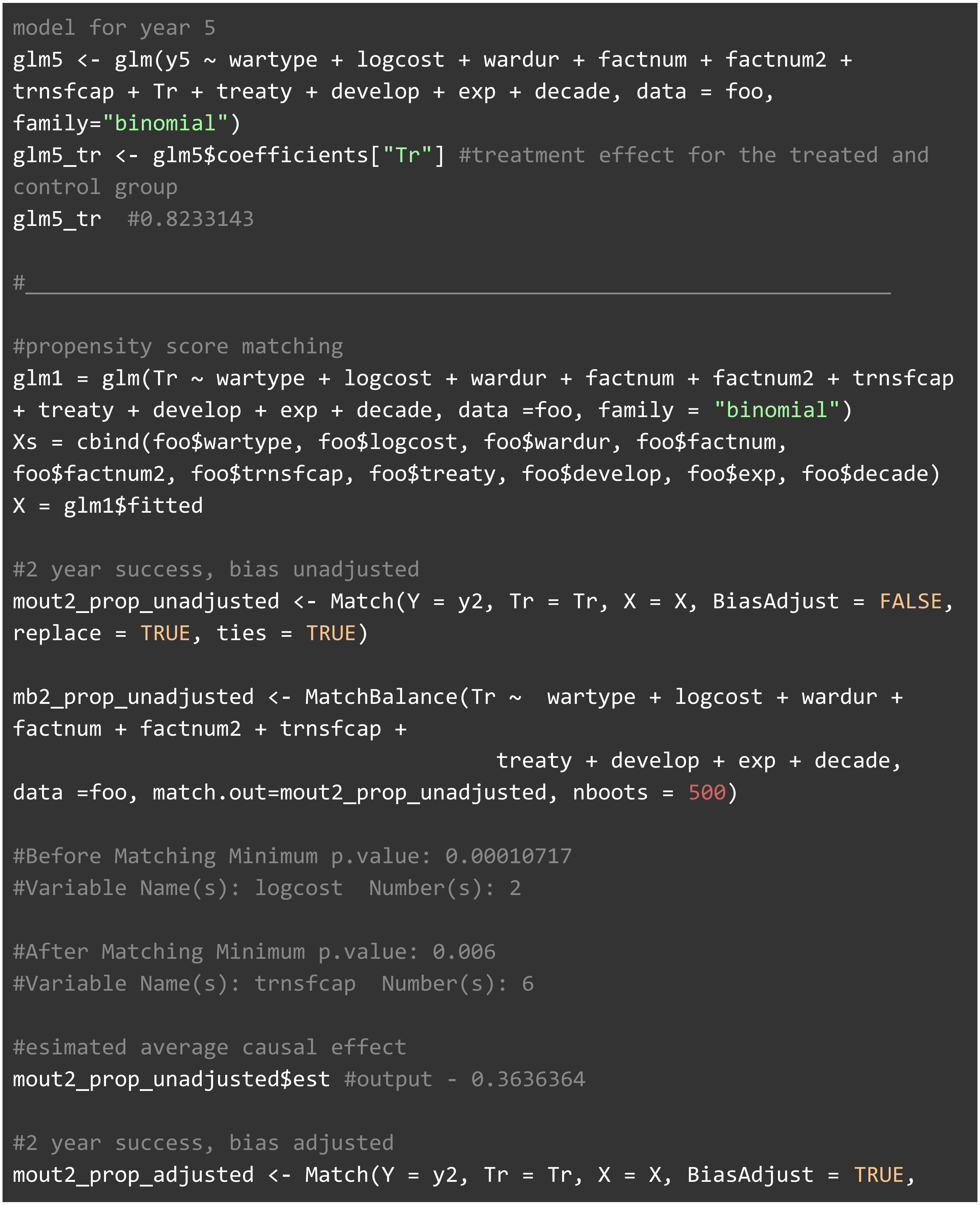
Appendix

Gist with all the code for question 2 and question 4 -

https://gist.github.com/Soumik0833/b385710ecfebb8227580897de741370f



```
#Question 4 Code
library(Matching)
set.seed(1234)
foo <-
read.csv("https://course-resources.minerva.kgi.edu/uploaded_files/mke/0008
6677-3767/peace.csv")

# remove rows with missing data (there are better ways to handle missing
data)
foo <- foo[-c(which(is.na(foo$pbs2l)), which(is.na(foo$pbs5l)),
which(is.na(foo$logcost)), which(is.na(foo$trnsfcap))), ]

#defining the treatment from question 3
Tr <- rep(0, length(foo$untype))
Tr[which(foo$untype != "None")] <- 1

#defining the outcomes - lenient PB success 2 years after the war (pbs2l)
and 5 years after the war (pbs5l)
y2 <- foo$pbs2l
y5 <- foo$pbs5l

#doing a logistic regression model for both years

#for year 2
glm2 <- glm(y2 ~ wartype + logcost + wardur + factnum + factnum2 +
trnsfcap + Tr + treaty + develop + exp + decade, data = foo,
family="binomial")
glm2_tr <- glm2$coefficients["Tr"] #treatment effect for the treated and
control group.
glm2_tr #0.5996839

#repeating what we did for year 2 and changing y2 to y5 to get the logit
```



```
model for year 5
glm5 <- glm(y5 ~ wartype + logcost + wardur + factnum + factnum2 +
trnsfcap + Tr + treaty + develop + exp + decade, data = foo,
family="binomial")
glm5_tr <- glm5$coefficients["Tr"] #treatment effect for the treated and
control group
glm5_tr  #0.8233143

#____________________________________________________________________

#propensity score matching
glm1 = glm(Tr ~ wartype + logcost + wardur + factnum + factnum2 + trnsfcap
+ treaty + develop + exp + decade, data =foo, family = "binomial")
Xs = cbind(foo$wartype, foo$logcost, foo$wardur, foo$factnum,
foo$factnum2, foo$trnsfcap, foo$treaty, foo$develop, foo$exp, foo$decade)
X = glm1$fitted

#2 year success, bias unadjusted
mout2_prop_unadjusted <- Match(Y = y2, Tr = Tr, X = X, BiasAdjust = FALSE,
replace = TRUE, ties = TRUE)

mb2_prop_unadjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap +
                                      treaty + develop + exp + decade,
data =foo, match.out=mout2_prop_unadjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.006
#Variable Name(s): trnsfcap  Number(s): 6

#esimated average causal effect
mout2_prop_unadjusted$est #output - 0.3636364

#2 year success, bias adjusted
mout2_prop_adjusted <- Match(Y = y2, Tr = Tr, X = X, BiasAdjust = TRUE,
replace = TRUE, ties = TRUE)

mb2_prop_adjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap +
                                      treaty + develop + exp + decade,
data =foo, match.out=mout2_prop_unadjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.01
#Variable Name(s): trnsfcap  Number(s): 6


#esimated average causal effect
mout2_prop_adjusted$est #output - 0.3635877
#____________________________________________________________________

#repeating what we did for 2 years with 5 years

#5 year success, bias unadjusted
mout5_prop_unadjusted <- Match(Y = y5, Tr = Tr, X = X, BiasAdjust = FALSE,
replace = TRUE, ties = TRUE)

mb5_prop_unadjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap +
                                      treaty + develop + exp + decade,
data =foo, match.out=mout2_prop_unadjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.012
#Variable Name(s): trnsfcap  Number(s): 6


#esimated average causal effect
```

```
mout5_prop_unadjusted$est #output - 0.3939394

#5 year success, bias adjusted
mout5_prop_adjusted <- Match(Y = y5, Tr = Tr, X = X, BiasAdjust = TRUE,
replace = TRUE, ties = TRUE)

mb5_prop_adjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap +
                                    treaty + develop + exp + decade, data
=foo, match.out=mout2_prop_unadjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.006
#Variable Name(s): trnsfcap  Number(s): 6


#esimated average causal effect
mout5_prop_adjusted$est #output - 0.3938908

#_________________________________________________________________________

#Genetic Matching

#2 year, bias unadjusted
BalanceMat = cbind(foo$wartype, foo$logcost, foo$wardur, foo$factnum,
                   foo$factnum2, foo$trnsfcap, foo$treaty, foo$develop,
foo$exp,
                   foo$decade, I(foo$develop*foo$treaty))

genout2_genmatch_unadjusted <- GenMatch(Tr=Tr, X=Xs, estimand="ATT",
                                        pop.size=200, max.generations=25,
wait.generations=5, nboots = 500, M=2,
                                        BalanceMatrix = BalanceMat,
replace = TRUE, ties = TRUE)
```

```
#'wait.generations' limit reached.
#No significant improvement in 5 generations.

#Solution Lexical Fitness Value:
#  3.408001e-02  4.334361e-02  4.800000e-02  6.200000e-02  7.600000e-02
7.865761e-02  7.865761e-02  1.200000e-01  1.340000e-01  1.340000e-01
1.781227e-01  1.928095e-01  3.455097e-01  4.813030e-01  4.813030e-01
4.900000e-01  6.209529e-01  7.620688e-01  7.680000e-01  8.200000e-01
9.049127e-01  9.480011e-01

#Parameters at the Solution:

#X[ 1] :   3.249646e+02
#X[ 2] :   7.471059e+02
#X[ 3] :   9.105414e+02
#X[ 4] :   3.649107e+01
#X[ 5] :   6.469591e+01
#X[ 6] :   2.603433e+00
#X[ 7] :   4.601945e+02
#X[ 8] :   4.608145e+02
#X[ 9] :   8.104469e+02
#X[10] :   3.164909e+02

#Solution Found Generation 12
#Number of Generations Run 18


mout2_genmatch_unadjusted <- Match(Y = y2, Tr = Tr, X = Xs, Weight.matrix
= genout2_genmatch_unadjusted, BiasAdjust = FALSE,
                                   replace = TRUE, ties = TRUE, M=2,
estimand = "ATT")

mb2_genmatch_unadjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap + treaty + develop + exp +
                                          decade, data =foo,
match.out=mout2_genmatch_unadjusted, nboots = 500)
```

```
#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.014
#Variable Name(s): develop  Number(s): 8




#2 year, bias adjusted
#we need to change only from our match function onwards because that is
where we start adjusting for bias

mout2_genmatch_adjusted <- Match(Y = y5, Tr = Tr, X = Xs, Weight.matrix =
genout2_genmatch_unadjusted, BiasAdjust = TRUE,
                                 replace = TRUE, ties = TRUE, M=2,
estimand = "ATT")

mb2_genmatch_adjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap + treaty + develop + exp +
                                          decade, data =foo,
match.out=mout2_genmatch_adjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.024
#Variable Name(s): develop  Number(s): 8

#estimated average causal effect
mout2_genmatch_adjusted$est #output - 0.1828086

#_________________________________________________________________________
________________

#5 year, bias unadjusted
#Since we start chainging out outcome variable from the Match function
itself, we don't need to do
```

```
#additional genmatching can can use genout2_genmatch_unadjusted in our
Match function

mout5_genmatch_unadjusted <- Match(Y = y5, Tr = Tr, X = Xs, Weight.matrix
= genout2_genmatch_unadjusted, BiasAdjust = FALSE,
                                   replace = TRUE, ties = TRUE, M=2,
estimand = "ATT")

mb5_genmatch_unadjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap + treaty + develop + exp +
                                          decade, data =foo,
match.out=mout5_genmatch_unadjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2

#After Matching Minimum p.value: 0.016
#Variable Name(s): develop  Number(s): 8



#5 year, bias adjusted
#we need to change only from our match function onwards because that is
where we start adjusting for bias

mout5_genmatch_adjusted <- Match(Y = y5, Tr = Tr, X = Xs, Weight.matrix =
genout2_genmatch_unadjusted, BiasAdjust = TRUE,
                                 replace = TRUE, ties = TRUE, M=2,
estimand = "ATT")

mb5_genmatch_adjusted <- MatchBalance(Tr ~  wartype + logcost + wardur +
factnum + factnum2 + trnsfcap + treaty + develop + exp +
                                        decade, data =foo,
match.out=mout5_genmatch_adjusted, nboots = 500)

#Before Matching Minimum p.value: 0.00010717
#Variable Name(s): logcost  Number(s): 2
```

```
#After Matching Minimum p.value: 0.014
#Variable Name(s): develop  Number(s): 8

#estimated average causal effect
mout5_genmatch_adjusted$est #output - 0.1828086
```