

# Propensity Scores Analysis Using R

## Zhenghao Xiao

### #1. Import and prepare data

Let's get started with importing packages and data set: We have patients' data on surgery types as treatment variable, died or not as outcome variable, along with some covariates.

```
library(tidyverse)
library(readxl)
library(mice)
library(party)
library(mitools)
library(twang)
library(lattice)
library(car)
library(broom)
library(treatSens)
library(Matching)
library(MatchIt)
library(optmatch)
library(rgenoud)
library(rbounds)
library(survey)
e3 <- read_excel("~/Documents/R/510/Exam3Q1.xlsx")
str(e3)
```

All the data looks good, but some of the kg is 0, which is weird, we should replace it with NA:

```
e3$kg <- na_if(e3$kg, 0)
```

Check how many NA we have in kg column:

```
round(sum(is.na(e3$kg)) / length(e3$kg), 3)
[1] 0.09
```

We have 9% missing data in kg column, we need to create a NA indicator for this column, because the reason of causing missing data can be a true confounder.

```
kg_na <- data.frame(is.na(e3$kg))
names(kg_na) <- c("kg_na")
e3 <- cbind(e3, kg_na)
```

Now, we need to scale and factorize our data:

```
factor_cols <- c("severe", "cognitivedecline", "depression", "cancer", "autoimmune", "transferredin",
                 "sex", "surgerytype", "nolifesupportorder", "insurancetype", "resp",
                 "infection", "trauma", "race", "kg_na")
scale_cols <- c("age", "yearseducation", "bloodpressure", "temperature", "creatinelvels",
```

```

"sodiumlevels", "urineweight", "kg", "income")
e3_adj <- e3 %>%
  mutate_at(factor_cols, factor) %>%
  mutate_at(scale_cols, scale) %>%
  mutate_at(scale_cols, as.numeric)

```

## #2. Dealing with NAs: imputation

Use multiple imputation to replace NAs in kg column:

```

imputed_data <- mice(e3_adj, m = 5, maxit = 50, method = "pmm", seed = 500)
complete_data1 <- complete(imputed_data, 1)
complete_data2 <- complete(imputed_data, 2)
complete_data3 <- complete(imputed_data, 3)
complete_data4 <- complete(imputed_data, 4)
complete_data5 <- complete(imputed_data, 5)

```

Create a list of all imputed datasets, this object was specifically designed to be analyzed with the survey package.

```

all_imputations <- imputationList(list(complete_data1, complete_data2, complete_data3,
                                         complete_data4, complete_data5))

```

## #3. Get propensity scores

### #3.1 get propensity scores with logistic regression

Let's get our formula first: this is just a test formula with all possible covariates.

```

cov_names <- subset(colnames(e3_adj), !(colnames(e3_adj) %in% c("died", "surgerytype")))
ps_formula <- paste(cov_names, collapse = "+")
ps_formula <- formula(paste("surgerytype ~ ", ps_formula, sep = ""))
ps_formula
surgerytype ~ severe + cognitivedecline + depression + cancer +
  autoimmune + transferredin + age + sex + yearseducation +
  bloodpressure + temperature + creatinelevels + sodiumlevels +
  urineweight + kg + nolifesupportorder + insurancetype + resp +
  infection + trauma + race + income + kg_na

```

Since our data has no strata no clusters, we can just use the glm function, let's start with 1st imputation data:

```

ps_model1 <- glm(ps_formula, data = complete_data1, family = "binomial")

```

Have a look on the ps model summary:

```
(sum_1 <- summary(ps_model1))
```

Call:

```
glm(formula = ps_formula, family = "binomial", data = complete_data1)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.1222	-0.9444	-0.6403	1.1279	2.4568

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.638303	0.128927	-4.951	7.39e-07 ***
severe1	0.197291	0.078251	2.521	0.011694 *
cognitivedecline1	-0.393524	0.109451	-3.595	0.000324 ***
depression1	-0.494581	0.127808	-3.870	0.000109 ***
cancer1	-0.293721	0.072445	-4.054	5.03e-05 ***
autoimmune1	0.054109	0.067085	0.807	0.419914
transferredin1	0.419771	0.089572	4.686	2.78e-06 ***
age	0.027556	0.040218	0.685	0.493241
sexMale	-0.008139	0.061541	-0.132	0.894779
yearseducation	0.063165	0.032654	1.934	0.053066 .
bloodpressure	-0.462896	0.032482	-14.251	< 2e-16 ***
temperature	-0.008874	0.030130	-0.295	0.768358
creatinelevels	0.173220	0.030582	5.664	1.48e-08 ***
sodiumlevels	-0.072019	0.029410	-2.449	0.014335 *
urineweight	-0.070740	0.030324	-2.333	0.019659 *
kg	0.237212	0.031103	7.627	2.41e-14 ***
nolifesupportorderYes	-0.673323	0.104871	-6.420	1.36e-10 ***
insurancetypeMedicare	0.222079	0.122674	1.810	0.070246 .
insurancetypeMedicare & Medicaid	0.249647	0.154203	1.619	0.105458
insurancetypeNo insurance	0.424862	0.152657	2.783	0.005384 **
insurancetypePrivate	0.419656	0.110109	3.811	0.000138 ***
insurancetypePrivate & Medicare	0.394192	0.125479	3.141	0.001681 **
respYes	-0.426753	0.062984	-6.776	1.24e-11 ***
infectionYes	0.364333	0.076801	4.744	2.10e-06 ***
traumaYes	1.056936	0.310204	3.407	0.000656 ***
raceother	0.077129	0.139060	0.555	0.579138
racewhite	-0.005705	0.085535	-0.067	0.946819
income	0.013231	0.032484	0.407	0.683778
kg_naTRUE	-0.261482	0.109807	-2.381	0.017252 *

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 7621.4 on 5734 degrees of freedom  
Residual deviance: 6873.2 on 5706 degrees of freedom  
AIC: 6931.2

Number of Fisher Scoring iterations: 4

Some of the predictors are not significant, we may decide to remove them. Yeadeducation is just at verge, we can test it with other imputation data.

```
ps_model2 <- glm(ps_formula, data = complete_data2, family = "binomial")
(sum_2 <- summary(ps_model2))
```

```
ps_model3 <- glm(ps_formula, data = complete_data3, family = "binomial")
(sum_3 <- summary(ps_model3))
```

```
ps_model4 <- glm(ps_formula, data = complete_data4, family = "binomial")
(sum_4 <- summary(ps_model4))
```

```
ps_model5 <- glm(ps_formula, data = complete_data5, family = "binomial")
(sum_5 <- summary(ps_model5))
```

To save some space, I cut other 4 summaries off. As a result, yearseducation is significant in 2 of the 5 models, and the p-value of it is very close to 0.05, so I decide to put it in our final ps model.

Let's check multicolliearity of our ps model:

```
vif(ps_model1)
```

	GVIF	Df	GVIF^(1/(2*Df))
severe	1.111411	1	1.054235
cognitivedecline	1.072454	1	1.035594
depression	1.027456	1	1.013635
cancer	1.071400	1	1.035084
autoimmune	1.068362	1	1.033616
transferredin	1.025188	1	1.012516
age	1.875079	1	1.369335
sex	1.100803	1	1.049192
yearseducation	1.268377	1	1.126223
bloodpressure	1.084051	1	1.041178
temperature	1.098332	1	1.048013
creatinelevels	1.112559	1	1.054779
sodiumlevels	1.023631	1	1.011747
urineweight	1.091063	1	1.044540
kg	1.126934	1	1.061571
nolifesupportorder	1.064218	1	1.031610
insurancetype	2.087064	5	1.076350
resp	1.053906	1	1.026599
infection	1.093590	1	1.045749
trauma	1.027783	1	1.013796
race	1.168117	2	1.039613
income	1.298566	1	1.139546
kg_na	1.042631	1	1.021093

All the vif values if close to 1, we have no problem in multicollinearity.

Time to find the significant predictors for treatment variable now:

```
names(subset(coef(sum_1)[, 4], coef(sum_1)[, 4] < 0.05))
[1] "(Intercept)"                  "severe1"
[3] "cognitivedecline1"           "depression1"
[5] "cancer1"                      "transferredin1"
[7] "bloodpressure"                "creatinelevels"
[9] "sodiumlevels"                 "urineweight"
[11] "kg"                           "nolifesupportorderYes"
[13] "insurancetypeNo insurance"   "insurancetypePrivate"
[15] "insurancetypePrivate & Medicare" "respYes"
[17] "infectionYes"                "traumaYes"
[19] "kg_naTRUE"
```

We need to add yearseducation into this list:

```
cov_names_final <- c("severe", "cognitivedecline", "depression",
                     "cancer", "transferredin", "bloodpressure", "yearseducation",
                     "creatinelevels", "sodiumlevels", "urineweight", "kg",
                     "nolifesupportorder", "insurancetype", "resp", "infection",
                     "trauma", "kg_na")
ps_formula_final <- paste(cov_names_final, collapse = "+")
ps_formula_final <- formula(paste("surgerytype ~ ", ps_formula_final, sep = ""))
ps_formula_final

surgerytype ~ severe + cognitivedecline + depression + cancer +
  transferredin + bloodpressure + yearseducation + creatinelevels +
  sodiumlevels + urineweight + kg + nolifesupportorder + insurancetype +
  resp + infection + trauma + kg_na
```

Get propensity scores from logistic regression:

```
ps_model_final1 <- glm(ps_formula_final, data = complete_data1, family = "binomial")
summary(ps_model_final1)
```

```

Call:
glm(formula = ps_formula_final, family = "binomial", data = complete_data1)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-2.1513 -0.9427 -0.6390  1.1292  2.4693 

Coefficients:
                                         Estimate Std. Error z value Pr(>|z|)    
(Intercept)                         -0.65335  0.09858 -6.628 3.41e-11 ***  
severe1                                0.20939  0.07704  2.718 0.006571 **  
cognitivedecline1                   -0.38571  0.10851 -3.555 0.000379 ***  
depression1                            -0.50407  0.12703 -3.968 7.24e-05 ***  
cancer1                                 -0.29187  0.07232 -4.036 5.45e-05 ***  
transferredin1                          0.41898  0.08938  4.688 2.76e-06 ***  
bloodpressure                           -0.46273  0.03236 -14.302 < 2e-16 ***  
yearseducation                         0.06547  0.03016  2.171 0.029922 *  
creatinelevels                          0.17603  0.02994  5.879 4.13e-09 ***  
sodiumlevels                            -0.07263  0.02928 -2.481 0.013107 *  
urineweight                             -0.07212  0.03000 -2.404 0.016227 *  
kg                                      0.23444  0.02995  7.828 4.98e-15 ***  
nolifesupportorderYes                 -0.66860  0.10395 -6.432 1.26e-10 ***  
insurancetypeMedicare                  0.26035  0.10965  2.374 0.017580 *  
insurancetypeMedicare & Medicaid     0.28253  0.14937  1.891 0.058571 .  
insurancetypeNo insurance              0.43011  0.15199  2.830 0.004656 **  
insurancetypePrivate                  0.43328  0.10712  4.045 5.24e-05 ***  
insurancetypePrivate & Medicare       0.43245  0.11197  3.862 0.000112 ***  
respYes                                 -0.43006  0.06261 -6.868 6.49e-12 ***  
infectionYes                            0.36275  0.07622  4.759 1.94e-06 ***  
traumaYes                               1.03550  0.30943  3.346 0.000819 ***  
kg_naTRUE                               -0.26447  0.10951 -2.415 0.015739 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

(Dispersion parameter for binomial family taken to be 1)

```

Null deviance: 7621.4  on 5734  degrees of freedom
Residual deviance: 6875.0  on 5713  degrees of freedom
AIC: 6919

```

Number of Fisher Scoring iterations: 4

```

p_score1 <- fitted(ps_model_final1)
complete_data1$p_scores <- p_score1

```

Now let's do it with all imputation data:

```

svy_design_all <- svydesign(ids = ~0, data = all_imputations)
ps_model_all <- with(svy_design_all, svyglm(ps_formula_final, family = binomial()))

```

We use the average of propensity scores from 5 imputation data as our final p score.

```

p_score_all <- sapply(ps_model_all, fitted)
p_scores <- apply(p_score_all, 1, mean)

```

Bind the propensity scores into our data frame:

```
all_imputations <- update(all_imputations, p_scores = p_scores)
str(all_imputations)
```

### #3.2 Get propensity scores from recursive partitioning

First we need to check the distributions of treatment variable and outcome variable, if they are balanced, the partitioning will be fine.

```
table(e3_adj$died)
table(e3_adj$surgerytype)
> table(e3_adj$died)
```

0	1
2013	3722

```
> table(e3_adj$surgerytype)
```

0	1
3551	2184

We are OK according to the result.

Also need to check for appropriate mtry value:

Recommended number of covariates per tree is the square root of the number of predictors:

```
sqrt(ncol(e3) - 1)
```

```
[1] 4.898979
```

So, we can set mtry to 5.

Random forest:

```
set.seed(2019)
my_controls <- cforest_unbiased(ntree = 1000, mtry = 5)
my_cforest <- cforest(ps_formula_final, data = complete_data1, controls = my_controls)

p_score_rf <- predict(my_cforest, type = "prob")
complete_data1$p_score_rf <- matrix(unlist(p_score_rf), , 2, byrow = TRUE)[, 2]
```

### #3.3 Get propensity scores using genearilize boosted modeling

First we need to convert our treatment variable back to numeric for ps function, and also set a seed for reproducibility:

```
set.seed(2019)
complete_data1$surgerytype <- as.numeric(complete_data1$surgerytype == 1)
my_gbm <- ps(ps_formula_final, data = complete_data1, n.trees = 10000, interaction.depth = 4,
              stop.method = c("es.max"), estimand = "ATT", verbose = TRUE)

p_score_gbm <- my_gbm$ps
names(p_score_gbm) = "p_score_gbm"
complete_data1$p_score_gbm <- unlist(p_score_gbm)
```

Check if we succeed in store all three kinds of propensity scores:  
`str(complete_data1[, 26:28])`

```
'data.frame': 5735 obs. of 3 variables:  
 $ p_scores : num 0.289 0.633 0.337 0.368 0.365 ...  
 $ p_score_rf: num 0.246 0.511 0.528 0.232 0.537 ...  
 $ p_score_gbm: num 0.288 0.442 0.53 0.334 0.521 ...
```

I didn't convert the propensity scores to linear propensity scores, because the advantage in using linear propensity scores is for matching, but I'm only using weighting and stratification.

Factor treatment variable back:

```
complete_data1$surgerytype <- factor(complete_data1$surgerytype)
```

#### #4. Evaluation of common support

#4.1 For the imputation data set 1:

#4.1.1 Descriptive statistics:

```
with(complete_data1, by(p_scores, surgerytype, summary))  
with(complete_data1, by(p_score_rf, surgerytype, summary))  
with(complete_data1, by(p_score_gbm, surgerytype, summary))  
> with(complete_data1, by(p_scores, surgerytype, summary))  
surgerytype: 0  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.02345 0.20937 0.31516 0.33366 0.43782 0.90113  
-----  
surgerytype: 1  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.04743 0.33836 0.46255 0.45749 0.57676 0.93947  
> with(complete_data1, by(p_score_rf, surgerytype, summary))  
surgerytype: 0  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.04071 0.18219 0.26836 0.28623 0.37338 0.72949  
-----  
surgerytype: 1  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.1271 0.4230 0.5366 0.5350 0.6481 0.8862  
> with(complete_data1, by(p_score_gbm, surgerytype, summary))  
surgerytype: 0  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.03276 0.18859 0.28032 0.30876 0.40249 0.82382  
-----  
surgerytype: 1  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.06882 0.36241 0.49343 0.49799 0.63810 0.91109
```

```
by(complete_data1[, 26:28], complete_data1$surgerytype, summary)
```

From the statistics, we can find that propensity scores getting from logistic regression has best common support.

```

> by(complete_data1[, 26:28], complete_data1$surgerytype, summary)
complete_data1$surgerytype: 0
  p_scores      p_score_rf      p_score_gbm
Min.   :0.02345   Min.   :0.04071   Min.   :0.03276
1st Qu.:0.20937   1st Qu.:0.18219   1st Qu.:0.18859
Median :0.31516   Median :0.26836   Median :0.28032
Mean    :0.33366   Mean    :0.28623   Mean    :0.30876
3rd Qu.:0.43782   3rd Qu.:0.37338   3rd Qu.:0.40249
Max.    :0.90113   Max.    :0.72949   Max.    :0.82382
-----
complete_data1$surgerytype: 1
  p_scores      p_score_rf      p_score_gbm
Min.   :0.04743   Min.   :0.1271   Min.   :0.06882
1st Qu.:0.33836   1st Qu.:0.4230   1st Qu.:0.36241
Median :0.46255   Median :0.5366   Median :0.49343
Mean    :0.45749   Mean    :0.5350   Mean    :0.49799
3rd Qu.:0.57676   3rd Qu.:0.6481   3rd Qu.:0.63810
Max.    :0.93947   Max.    :0.8862   Max.    :0.91109

```

#### #4.1.2 Graphical checks:

For logistic regression:

```

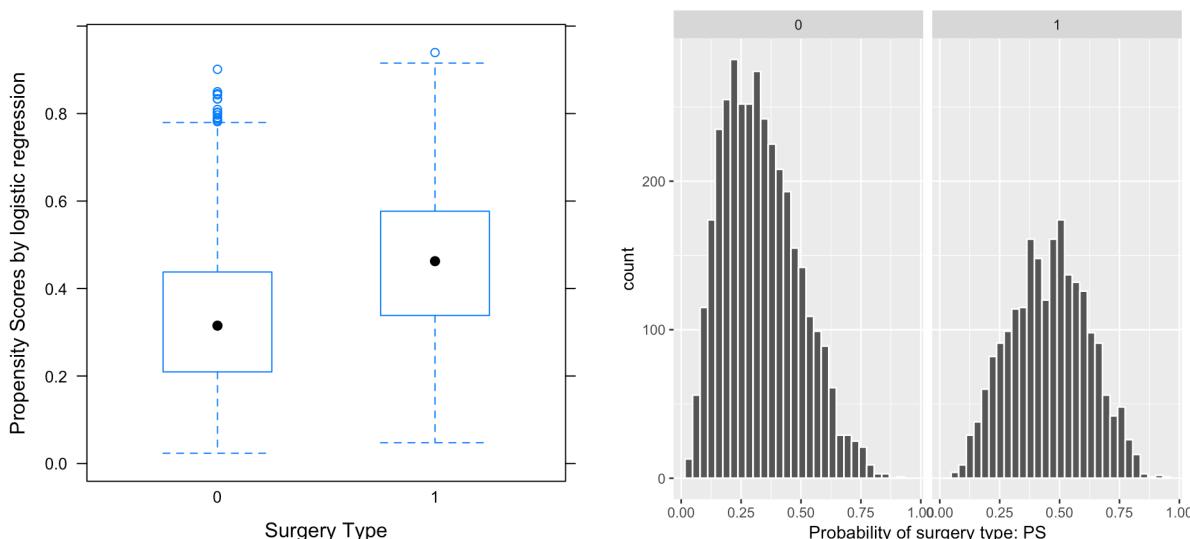
bwplot(p_scores ~ surgerytype, data = complete_data1,
       ylab = "Propensity Scores by logistic regression",
       xlab = "Surgery Type", auto.key = TRUE)

```

```

complete_data1 %>%
  ggplot(aes(x = p_scores)) +
  geom_histogram(color = "white") +
  facet_wrap(~ surgerytype) +
  xlab("Probability of surgery type: PS")

```



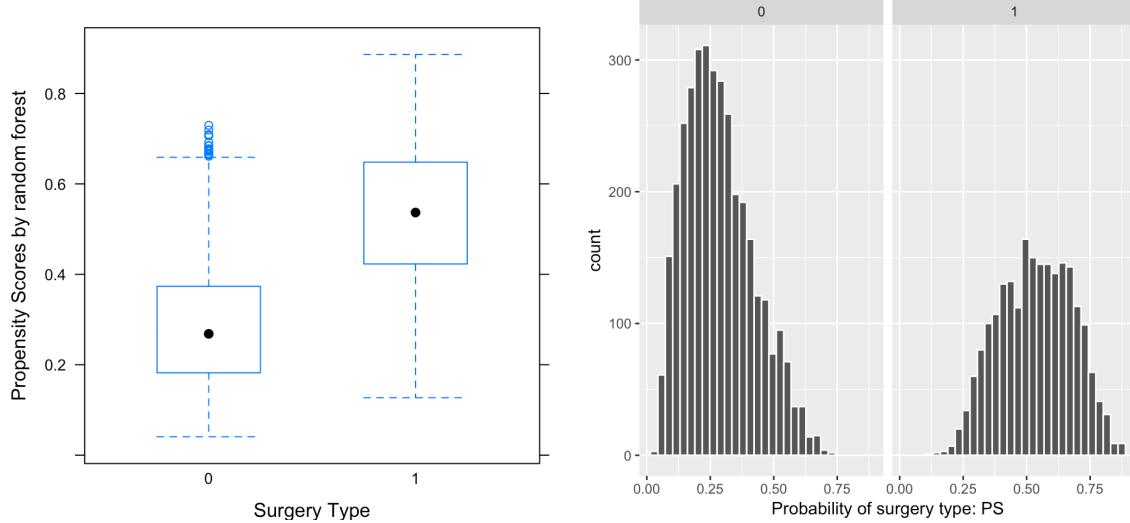
For random forest:

```

bwplot(p_score_rf ~ surgerytype, data = complete_data1,
       ylab = "Propensity Scores by random forest",
       xlab = "Surgery Type", auto.key = TRUE)

```

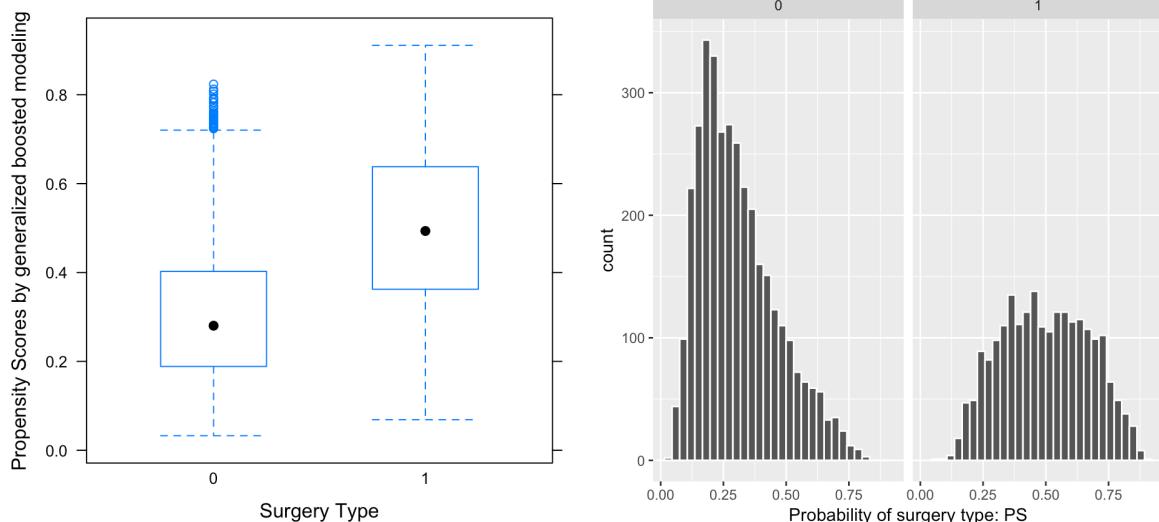
```
complete_data1 %>%
  ggplot(aes(x = p_score_rf)) +
  geom_histogram(color = "white") +
  facet_wrap(~ surgerytype) +
  xlab("Probability of surgery type: PS")
```



For generalize boosted modeling:

```
bwplot(p_score_gbm ~ surgerytype, data = complete_data1,
       ylab = "Propensity Scores by generalized boosted modeling",
       xlab = "Surgery Type", auto.key = TRUE)
```

```
complete_data1 %>%
  ggplot(aes(x = p_score_gbm)) +
  geom_histogram(color = "white") +
  facet_wrap(~ surgerytype) +
  xlab("Probability of surgery type: PS")
```

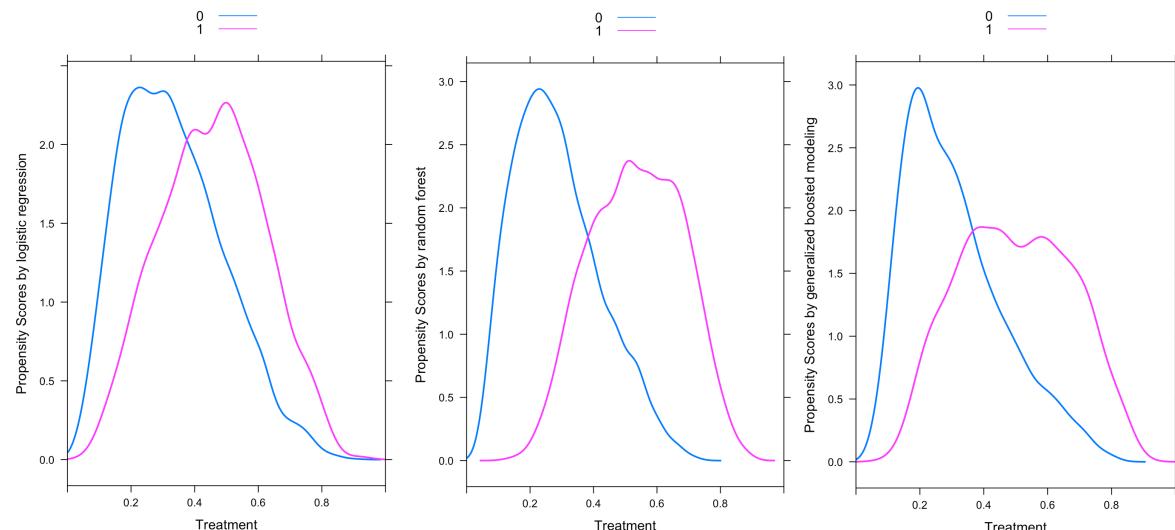


Compare them at once:

```
densityplot( ~ p_scores, groups = surgerytype, plot.points = FALSE,  
            xlim = c(0,1), lwd = 2,  
            data = complete_data1,  
            ylab = "Propensity Scores by logistic regression",  
            xlab = "Treatment",auto.key = TRUE)
```

```
densityplot( ~ p_score_rf, groups = surgerytype, plot.points = FALSE,  
            xlim = c(0,1), lwd = 2,  
            data = complete_data1,  
            ylab = "Propensity Scores by random forest",  
            xlab = "Treatment",auto.key = TRUE)
```

```
densityplot( ~ p_score_gbm, groups = surgerytype, plot.points = FALSE,  
            xlim = c(0,1), lwd = 2,  
            data = complete_data1,  
            ylab = "Propensity Scores by generalized boosted modeling",  
            xlab = "Treatment",auto.key = TRUE)
```



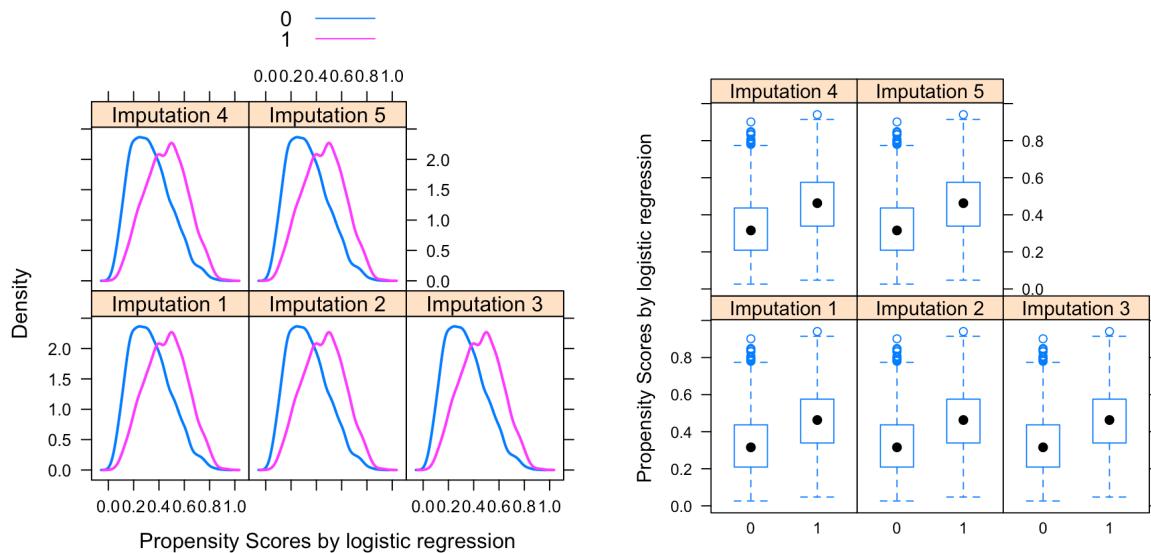
From the comparison we can conclude that propensity scores get from logistic regression has most common support.

#4.2 Common support for all imputed data sets

```
all_imputations_stacked <- data.frame()  
for (imp in 1:5) { temp <- cbind(all_imputations$imputations[[imp]],imputation = imp)  
all_imputations_stacked = rbind(all_imputations_stacked, temp) }  
all_imputations_stacked$surgerytype <- factor(all_imputations_stacked$surgerytype)  
all_imputations_stacked$imputation <- factor(all_imputations_stacked$imputation,  
                                              labels=paste("Imputation",1:5))
```

```
densityplot( ~ p_scores | imputation, data = all_imputations_stacked,  
            plot.points = FALSE, lwd = 2,  
            groups = surgerytype, xlab = "Propensity Scores by logistic regression",  
            auto.key = TRUE)
```

```
bwplot(p_scores ~ surgerytype | imputation, data = all_imputations_stacked, lwd = 2,
       ylab = "Propensity Scores by logistic regression", auto.key = TRUE)
```



Common supports among all imputation data set are basically the same.

## #5. Propensity score weighting

#5.1 Propensity scores weighting from logistic regression  
ATT weights:

```
complete_data1$weight_att <- with(complete_data1,
                                   ifelse(surgerytype == 1, 1, p_scores/(1 - p_scores)))

with(complete_data1, by(weight_att, surgerytype, summary))

> with(complete_data1, by(weight_att, surgerytype, summary))
surgerytype: 0
  Min. 1st Qu. Median Mean 3rd Qu. Max.
0.02401 0.26482 0.46020 0.62165 0.77878 9.11472
-----
surgerytype: 1
  Min. 1st Qu. Median Mean 3rd Qu. Max.
1 1 1 1 1 1
```

ATE weights:

```
complete_data1$weight_ate <- with(complete_data1,
                                   ifelse(surgerytype == 1, 1/p_scores, 1/(1 - p_scores)))

with(complete_data1, by (weight_ate, surgerytype, summary))
```

```

> with(complete_data1, by (weight_ate, surgerytype, summary))
surgerytype: 0
  Min. 1st Qu. Median Mean 3rd Qu. Max.
1.024 1.265 1.460 1.622 1.779 10.115
-----
surgerytype: 1
  Min. 1st Qu. Median Mean 3rd Qu. Max.
1.064 1.734 2.162 2.612 2.955 21.086

```

Now let's do the covariate balance check for both ATT and ATE weights:

Covariate balance check for ATT weights:

```

balance_table1 <- bal.stat(complete_data1, vars = cov_names_final,
                           treat.var = "surgerytype", w.all = complete_data1$weight_att,
                           get.ks = FALSE, sampw = 1, estimand = "ATT", multinom = FALSE)

```

```

round(balance_table1$results[, 1:5], 3)
> round(balance_table1$results[, 1:5], 3)
      tx.mn tx.sd ct.mn ct.sd std.eff.sz
severe:0          0.796 0.403 0.796 0.403   0.001
severe:1          0.204 0.403 0.204 0.403  -0.001
cognitivedecline:0 0.931 0.254 0.934 0.249  -0.011
cognitivedecline:1 0.069 0.254 0.066 0.249   0.011
depression:0       0.954 0.209 0.956 0.206  -0.007
depression:1       0.046 0.209 0.044 0.206   0.007
cancer:0           0.797 0.402 0.796 0.403   0.001
cancer:1           0.203 0.402 0.204 0.403  -0.001
transferredin:0    0.850 0.357 0.853 0.354  -0.007
transferredin:1    0.150 0.357 0.147 0.354   0.007
bloodpressure      -0.271 0.900 -0.269 0.872  -0.003
yearseducation     0.058 1.003 0.057 0.978   0.001
creatinelevels     0.166 1.000 0.257 1.479  -0.091
sodiumlevels       -0.057 0.993 -0.068 1.005   0.012
urineweight        -0.075 1.037 -0.080 1.037   0.006
kg                 0.156 0.982 0.157 1.113  -0.001
nolifesupportorder>No 0.929 0.257 0.929 0.257   0.001
nolifesupportorder>Yes 0.071 0.257 0.071 0.257  -0.001
insurancetype:Medicaid 0.088 0.284 0.088 0.284   0.000
insurancetype:Medicare 0.234 0.423 0.234 0.423   0.000
insurancetype:Medicare & Medicaid 0.056 0.231 0.054 0.226   0.010
insurancetype>No insurance 0.062 0.242 0.060 0.238   0.007
insurancetype:Private 0.335 0.472 0.332 0.471   0.005
insurancetype:Private & Medicare 0.224 0.417 0.230 0.421  -0.015
resp>No            0.711 0.453 0.714 0.452  -0.008
resp>Yes           0.289 0.453 0.286 0.452   0.008
infection>No      0.764 0.425 0.749 0.434   0.036
infection>Yes     0.236 0.425 0.251 0.434  -0.036
trauma>No          0.984 0.124 0.983 0.130   0.014
trauma>Yes         0.016 0.124 0.017 0.130  -0.014
kg_na:FALSE       0.931 0.254 0.937 0.244  -0.023
kg_na:TRUE        0.069 0.254 0.063 0.244   0.023

```

This table is too long, to save some spaces, I will skip that, only give the summary of the standard effect size of this balance table.

I cut off test statistics from the balance table result because it is not recommended: first because covariate balance is a property of the sample, and hypothesis tests refer to the population. Second,

inferential measures depend on sample size, as our data set is big, the test statistics will always be significant.

We can see that the tx.sd and ct.sd are very close, and the absolute values of standard effect size are lower than 0.1, which indicating we have achieved adequate covariate balance.

```
summary(abs(balance_table1$results[, 5]))  
> summary(abs(balance_table1$results[, 5]))  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.0003675 0.0011414 0.0070614 0.0113786 0.0121017 0.0908287
```

Covariate balance check for ATE weights:

```
balance_table2 <- bal.stat(complete_data1, vars = cov_names_final,  
                           treat.var = "surgerytype", w.all = complete_data1$weight_ate,  
                           get.ks = FALSE, sampw = 1, estimand = "ATE", multinom = FALSE)  
  
summary(abs(balance_table2$results[, 5]))  
> summary(abs(balance_table2$results[, 5]))  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.0006937 0.0022701 0.0084464 0.0087630 0.0113345 0.0235498
```

We can see that propensity score weighting from logistic regression reached covariates balance in both average treatment effect of treatment (ATT) and average treatment effect (ATE). Which means we can get estimation of treatment effect on both ATT and ATE.

#5.2 Propensity scores weighting from random forest

ATT weights:

```
complete_data1$weight_att_rf <- with(complete_data1,  
                                      ifelse(surgerytype == 1, 1, p_score_rf / (1 - p_score_rf)))  
  
with(complete_data1, by(weight_att_rf, surgerytype, summary))  
> with(complete_data1, by(weight_att_rf, surgerytype, summary))  
surgerytype: 0  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.04244 0.22278 0.36680 0.46412 0.59586 2.69675  
-----  
surgerytype: 1  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
1 1 1 1 1 1
```

ATE weights:

```
complete_data1$weight_ate_rf <- with(complete_data1,  
                                      ifelse(surgerytype == 1, 1/p_score_rf, 1/(1 - p_score_rf)))  
  
with(complete_data1, by (weight_ate_rf, surgerytype, summary))
```

```

> with(complete_data1, by (weight_ate_rf, surgerytype, summary))
surgerytype: 0
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  1.042   1.223   1.367  1.464   1.596  3.697
-----
surgerytype: 1
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  1.128   1.543   1.863  2.039   2.364  7.869

```

Balance table for ATT random forest weights:

```

balance_table3 <- bal.stat(complete_data1, vars = cov_names_final,
                           treat.var = "surgerytype", w.all = complete_data1$weight_att_rf,
                           get.ks = FALSE, sampw = 1, estimand = "ATT", multinom = FALSE)

summary(abs(balance_table3$results[, 5]))
> summary(abs(balance_table3$results[, 5]))
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  0.01061  0.03067  0.04688  0.04352  0.05628  0.07601

```

Maximum absolute value of the standard effect size is less than 0.1, passed balance check.

Balance table for ATE random forest weights:

```

balance_table4 <- bal.stat(complete_data1, vars = cov_names_final,
                           treat.var = "surgerytype", w.all = complete_data1$weight_ate_rf,
                           get.ks = FALSE, sampw = 1, estimand = "ATE", multinom = FALSE)

summary(abs(balance_table4$results[, 5]))
> summary(abs(balance_table4$results[, 5]))
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  0.02115  0.04870  0.06876  0.07457  0.09089  0.16340

```

Maximum absolute value of the standard effect size is greater than 0.1, failed balance check on ATE.

#5.3 Propensity weighting from generalize boosted modeling

ATT weights:

```

complete_data1$weight_att_gbm <- with(complete_data1,
                                         ifelse(surgerytype == 1, 1, p_score_gbm / (1 - p_score_gbm)))

with(complete_data1, by(weight_att_gbm, surgerytype, summary))
> with(complete_data1, by(weight_att_gbm, surgerytype, summary))
surgerytype: 0
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  0.03387  0.23242  0.38952  0.55348  0.67361  4.67610
-----
surgerytype: 1
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  1         1         1         1         1         1

```

ATE weights:

```
complete_data1$weight_ate_gbm <- with(complete_data1,
                                         ifelse(surgerytype == 1, 1/p_score_gbm, 1/(1 - p_score_gbm)))
```

```
with(complete_data1, by (weight_ate_gbm, surgerytype, summary))
> with(complete_data1, by (weight_ate_gbm, surgerytype, summary))
```

surgerytype: 0

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	1.034	1.232	1.390	1.553	1.674	5.676

surgerytype: 1

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	1.098	1.567	2.027	2.353	2.759	14.531

Balance table for ATT gbm weights:

```
balance_table5 <- bal.stat(complete_data1, vars = cov_names_final,
                           treat.var = "surgerytype", w.all = complete_data1$weight_att_gbm,
                           get.ks = FALSE, sampw = 1, estimand = "ATT", multinom = FALSE)
```

```
summary(abs(balance_table5$results[, 5]))
> summary(abs(balance_table5$results[, 5]))
Min. 1st Qu. Median Mean 3rd Qu. Max.
0.003365 0.017605 0.023434 0.026273 0.034675 0.049658
```

Balance check with ATT passed.

Balance table for ATE gbm weights:

```
balance_table6 <- bal.stat(complete_data1, vars = cov_names_final,
                           treat.var = "surgerytype", w.all = complete_data1$weight_ate_gbm,
                           get.ks = FALSE, sampw = 1, estimand = "ATE", multinom = FALSE)
```

```
summary(abs(balance_table6$results[, 5]))
> summary(abs(balance_table6$results[, 5]))
Min. 1st Qu. Median Mean 3rd Qu. Max.
0.0009294 0.0381332 0.0440621 0.0454768 0.0566532 0.0847269
```

Reached balance in ATE. But the standardize effect size is higher than these from logistic regression.

According to the standard effect size, weighting using propensity scores get from logistic regression achieved best performance in covariate balance, so our final estimation of treatment effects will be based on propensity scores weighting from logistic regression.

## #6. Estimation of treatment effects by weighting

According to our covariate balance check results in last part, we know that we reached balance in both ATT and ATE by using propensity scores weighting derived from logistic regression, so we can do both estimation of treatment effect for ATT and ATE.

ATT:

```
svy_design_att <- svydesign(ids = ~0, weights = complete_data1$weight_att,  
                           data = complete_data1)
```

```
svy_design_boot_att <- as.svrepdesign(svy_design_att, type = c("bootstrap"), replicates = 2000)
```

Svyglm model:

```
model_att <- svyglm(died ~ surgerytype + severe + cognitivedecline + depression + cancer +  
                     autoimmune + transferredin + age + sex + yearseducation +  
                     bloodpressure + temperature + creatinelevels + sodiumlevels +  
                     urineweight + kg + nolifesupportorder + insurancetype + resp +  
                     infection + trauma + race + income + kg_na, svy_design_att, family = binomial())
```

summary(model\_att)

```
Call:  
svyglm(formula = died ~ surgerytype + severe + cognitivedecline +  
       depression + cancer + autoimmune + transferredin + age + sex +  
       yearseducation + bloodpressure + temperature + creatinelevels +  
       sodiumlevels + urineweight + kg + nolifesupportorder + insurancetype +  
       resp + infection + trauma + race + income + kg_na, design = svy_design_att,  
       family = binomial())
```

Survey design:

```
svydesign(ids = ~0, weights = complete_data1$weight_att, data = complete_data1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.1390752	0.1521196	0.914	0.360625
surgerytype1	0.3320852	0.0689996	4.813	1.53e-06 ***
severe1	0.0554104	0.0945706	0.586	0.557955
cognitivedecline1	0.2979086	0.1399329	2.129	0.033302 *
depression1	-0.0480666	0.1475368	-0.326	0.744592
cancer1	1.0275864	0.0984641	10.436	< 2e-16 ***
autoimmune1	0.2998529	0.0812454	3.691	0.000226 ***
transferredin1	-0.3005033	0.1024596	-2.933	0.003372 **
age	0.4052153	0.0487419	8.313	< 2e-16 ***
sexMale	0.2218287	0.0726540	3.053	0.002274 **
yearseducation	-0.0004878	0.0394459	-0.012	0.990134
bloodpressure	-0.1143096	0.0370465	-3.086	0.002041 **
temperature	-0.1322614	0.0404629	-3.269	0.001087 **
creatinelevels	0.0789827	0.0447443	1.765	0.077584 .
sodiumlevels	-0.0149459	0.0377901	-0.395	0.692490
urineweight	-0.0213743	0.0354584	-0.603	0.546667
kg	-0.1743989	0.0381844	-4.567	5.04e-06 ***
nolifesupportorderYes	1.2981734	0.1697686	7.647	2.40e-14 ***
insurancetypeMedicare	0.0070067	0.1452759	0.048	0.961534
insurancetypeMedicare & Medicaid	-0.0100405	0.1863304	-0.054	0.957028
insurancetypeNo insurance	0.1714766	0.1712632	1.001	0.316750
insurancetypePrivate	0.0260159	0.1250703	0.208	0.835228
insurancetypePrivate & Medicare	-0.0068858	0.1463387	-0.047	0.962472
respYes	-0.0984889	0.0731337	-1.347	0.178132
infectionYes	0.0518409	0.0920313	0.563	0.573255
traumaYes	-0.7660714	0.3403769	-2.251	0.024445 *
raceother	0.0538154	0.1688862	0.319	0.750005
racewhite	-0.1211261	0.1041575	-1.163	0.244913
income	-0.0448386	0.0364185	-1.231	0.218298
kg_naTRUE	-0.0063842	0.1217614	-0.052	0.958187

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1.017032)

Number of Fisher Scoring iterations: 5

We can see that our treatment variable is significant on predicting outcome variable. Let's do a tidy version of summary for later usage including the odd-ratio for the estimate coefficients.

```
outcome_att <- tidy(model_att) %>%
  filter(p.value < 0.05) %>%
  mutate(odd_ratio = exp(estimate))

outcome_att
> outcome_att
# A tibble: 12 x 6
  term      estimate std.error statistic p.value odd_ratio
  <chr>     <dbl>    <dbl>     <dbl>    <dbl>    <dbl>
1 surgerytype1  0.332   0.0690    4.81  1.53e- 6  1.39
2 cognitivedecline1 0.298   0.140     2.13  3.33e- 2  1.35
3 cancer1       1.03    0.0985    10.4   2.86e-25  2.79
4 autoimmune1    0.300   0.0812    3.69  2.26e- 4  1.35
5 transferredin1 -0.301   0.102    -2.93  3.37e- 3  0.740
6 age            0.405   0.0487    8.31  1.15e-16  1.50
7 sexMale        0.222   0.0727    3.05  2.27e- 3  1.25
8 bloodpressure  -0.114   0.0370   -3.09  2.04e- 3  0.892
9 temperature    -0.132   0.0405   -3.27  1.09e- 3  0.876
10 kg             -0.174   0.0382   -4.57  5.04e- 6  0.840
11 nolifesupportorderYes 1.30    0.170     7.65  2.40e-14  3.66
12 traumaYes     -0.766   0.340    -2.25  2.44e- 2  0.465
```

Bootstrap model for ATT:

```
model_att_boot <- svyglm(died ~ surgerytype + severe + cognitivedecline + depression + cancer +
                           autoimmune + transferredin + age + sex + yearseducation +
                           bloodpressure + temperature + creatinelevels + sodiumlevels +
                           urineweight + kg + nolifesupportorder + insurancetype + resp +
                           infection + trauma + race + income + kg_na, svy_design_boot_att,
                           family = binomial())
summary(model_att_boot)

outcome_att_boot <- tidy(model_att_boot) %>%
  filter(p.value < 0.05) %>%
  mutate(odd_ratio = exp(estimate))
outcome_att_boot
```

It is obvious that the results from svyglm model and bootstrap model are very similar.

```

Call:
svyglm(formula = died ~ surgerytype + severe + cognitivedecline +
depression + cancer + autoimmune + transferredin + age + sex +
yearseducation + bloodpressure + temperature + creatinelevels +
sodiumlevels + urineweight + kg + nolifesupportorder + insurancetype +
resp + infection + trauma + race + income + kg_na, svy_design_boot_att,
family = binomial())

```

```

Survey design:
as.svrepdesign(svy_design_att, type = c("bootstrap"), replicates = 2000)

```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.1390752	0.1497018	0.929	0.352995
surgerytype1	0.3320852	0.0693774	4.787	1.82e-06 ***
severe1	0.0554104	0.0972026	0.570	0.568709
cognitivedecline1	0.2979086	0.1442065	2.066	0.038973 *
depression1	-0.0480666	0.1525242	-0.315	0.752688
cancer1	1.0275864	0.0991119	10.368	< 2e-16 ***
autoimmune1	0.2998529	0.0806255	3.719	0.000206 ***
transferredin1	-0.3005033	0.1022329	-2.939	0.003327 **
age	0.4052153	0.0493487	8.211	3.91e-16 ***
sexMale	0.2218287	0.0722697	3.069	0.002174 **
yearseducation	-0.0004878	0.0402287	-0.012	0.990327
bloodpressure	-0.1143096	0.0376149	-3.039	0.002405 **
temperature	-0.1322614	0.0412221	-3.209	0.001356 **
creatinelvels	0.0789827	0.0454542	1.738	0.082432 .
sodiumlevels	-0.0149459	0.0383619	-0.390	0.696872
urineweight	-0.0213743	0.0360564	-0.593	0.553383
kg	-0.1743989	0.0387955	-4.495	7.35e-06 ***
nolifesupportorderYes	1.2981734	0.1794656	7.234	6.71e-13 ***
insurancetypeMedicare	0.0070067	0.1455616	0.048	0.961613
insurancetypeMedicare & Medicaid	-0.0100405	0.1873172	-0.054	0.957258
insurancetypeNo insurance	0.1714766	0.1757692	0.976	0.329393
insurancetypePrivate	0.0260159	0.1257097	0.207	0.836068
insurancetypePrivate & Medicare	-0.0068858	0.1448642	-0.048	0.962093
respYes	-0.0984889	0.0736952	-1.336	0.181561
infectionYes	0.0518409	0.0964360	0.538	0.590936
traumaYes	-0.7660714	0.3649289	-2.099	0.035923 *
raceother	0.0538154	0.1678566	0.321	0.748545
racewhite	-0.1211261	0.1046249	-1.158	0.247120
income	-0.0448386	0.0359230	-1.248	0.212112
kg_naTRUE	-0.0063842	0.1224166	-0.052	0.958414

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 5831.761)

Number of Fisher Scoring iterations: 5

> outcome\_att\_boot

# A tibble: 12 x 6

term	estimate	std.error	statistic	p.value	odd_ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 surgerytype1	0.332	0.0694	4.79	1.82e- 6	1.39
2 cognitivedecline1	0.298	0.144	2.07	3.90e- 2	1.35
3 cancer1	1.03	0.0991	10.4	1.47e-24	2.79
4 autoimmune1	0.300	0.0806	3.72	2.06e- 4	1.35
5 transferredin1	-0.301	0.102	-2.94	3.33e- 3	0.740
6 age	0.405	0.0493	8.21	3.91e-16	1.50
7 sexMale	0.222	0.0723	3.07	2.17e- 3	1.25
8 bloodpressure	-0.114	0.0376	-3.04	2.41e- 3	0.892
9 temperature	-0.132	0.0412	-3.21	1.36e- 3	0.876
10 kg	-0.174	0.0388	-4.50	7.35e- 6	0.840
11 nolifesupportorderYes	1.30	0.179	7.23	6.71e-13	3.66
12 traumaYes	-0.766	0.365	-2.10	3.59e- 2	0.465

ATE:

```
svy_design_ate <- svydesign(ids = ~0, weights = complete_data1$weight_ate,
                           data = complete_data1)
svy_design_boot_ate <- as.svrepdesign(svy_design_ate, type = c("bootstrap"), replicates = 2000)
```

Svyglm model

```
model_ate <- svyglm(died ~ surgerytype + severe + cognitivedecline + depression + cancer +
                     autoimmune + transferredin + age + sex + yearseducation +
                     bloodpressure + temperature + creatinelevels + sodiumlevels +
                     urineweight + kg + nolifesupportorder + insurancetype + resp +
                     infection + trauma + race + income + kg_na, svy_design_ate, family = binomial())
summary(model_ate)

Call:
svyglm(formula = died ~ surgerytype + severe + cognitivedecline +
       depression + cancer + autoimmune + transferredin + age + sex +
       yearseducation + bloodpressure + temperature + creatinelevels +
       sodiumlevels + urineweight + kg + nolifesupportorder + insurancetype +
       resp + infection + trauma + race + income + kg_na, design = svy_design_ate,
       family = binomial())
```

Survey design:

```
svydesign(ids = ~0, weights = complete_data1$weight_ate, data = complete_data1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.100458	0.146434	0.686	0.492722
surgerytype1	0.357418	0.066910	5.342	9.56e-08 ***
severe1	0.013838	0.093211	0.148	0.881983
cognitivedecline1	0.323016	0.134540	2.401	0.016387 *
depression1	-0.076669	0.146352	-0.524	0.600391
cancer1	0.975033	0.096092	10.147	< 2e-16 ***
autoimmune1	0.327612	0.077796	4.211	2.58e-05 ***
transferredin1	-0.286914	0.097623	-2.939	0.003306 **
age	0.381394	0.045345	8.411	< 2e-16 ***
sexMale	0.230842	0.069204	3.336	0.000856 ***
yearseducation	0.009139	0.037465	0.244	0.807293
bloodpressure	-0.084130	0.036374	-2.313	0.020762 *
temperature	-0.134303	0.037335	-3.597	0.000324 ***
creatinelevels	0.108638	0.040935	2.654	0.007979 **
sodiumlevels	0.005322	0.033701	0.158	0.874531
urineweight	0.010498	0.033534	0.313	0.754242
kg	-0.219558	0.034799	-6.309	3.01e-10 ***
nolifesupportorderYes	1.271721	0.159983	7.949	2.25e-15 ***
insurancetypeMedicare	0.069309	0.136819	0.507	0.612471
insurancetypeMedicare & Medicaid	0.059413	0.177601	0.335	0.737992
insurancetypeNo insurance	0.227374	0.163816	1.388	0.165196
insurancetypePrivate	0.064363	0.121207	0.531	0.595431
insurancetypePrivate & Medicare	-0.058888	0.140819	-0.418	0.675831
respYes	-0.055229	0.071340	-0.774	0.438868
infectionYes	0.089826	0.088160	1.019	0.308295
traumaYes	-0.756742	0.330152	-2.292	0.021936 *
raceother	-0.162922	0.159434	-1.022	0.306884
racewhite	-0.103448	0.094850	-1.091	0.275473
income	-0.059454	0.034648	-1.716	0.086233 .
kg_naTRUE	-0.128275	0.114456	-1.121	0.262446

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1.0237)

Number of Fisher Scoring iterations: 4

```

outcome_ate <- tidy(model_ate) %>%
  filter(p.value < 0.05) %>%
  mutate(odd_ratio = exp(estimate))

outcome_ate
> outcome_ate
# A tibble: 13 x 6
   term      estimate std.error statistic p.value odd_ratio
   <chr>     <dbl>    <dbl>     <dbl>    <dbl>    <dbl>
1 surgerytype1  0.357    0.0669    5.34  9.56e- 8    1.43
2 cognitivedecline1 0.323    0.135     2.40  1.64e- 2    1.38
3 cancer1      0.975    0.0961   10.1   5.46e-24    2.65
4 autoimmune1   0.328    0.0778    4.21  2.58e- 5    1.39
5 transferredin1 -0.287    0.0976   -2.94  3.31e- 3    0.751
6 age          0.381    0.0453    8.41  5.08e-17    1.46
7 sexMale       0.231    0.0692    3.34  8.56e- 4    1.26
8 bloodpressure -0.0841   0.0364   -2.31  2.08e- 2    0.919
9 temperature   -0.134   0.0373   -3.60  3.24e- 4    0.874
10 creatinelevels 0.109    0.0409    2.65  7.98e- 3    1.11
11 kg          -0.220    0.0348   -6.31  3.01e-10    0.803
12 nolifesupportorderYes 1.27     0.160     7.95  2.25e-15    3.57
13 traumaYes    -0.757    0.330    -2.29  2.19e- 2    0.469

```

We can see that the ATE result is slightly different from the ATE result on statistics and coefficients estimates, all the significant predictors remains the same. Let's keep going with the bootstrap model for ATE:

```

model_ate_boot <- svyglm(died ~ surgerytype + severe + cognitivedecline + depression + cancer +
                          autoimmune + transferredin + age + sex + yearseducation +
                          bloodpressure + temperature + creatinelevels + sodiumlevels +
                          urineweight + kg + nolifesupportorder + insurancetype + resp +
                          infection + trauma + race + income + kg_na, svy_design_boot_ate,
                          family = binomial())
summary(model_ate_boot)

outcome_ate_boot <- tidy(model_ate_boot) %>%
  filter(p.value < 0.05) %>%
  mutate(odd_ratio = exp(estimate))

outcome_ate_boot
> outcome_ate_boot
# A tibble: 13 x 6
   term      estimate std.error statistic p.value odd_ratio
   <chr>     <dbl>    <dbl>     <dbl>    <dbl>    <dbl>
1 surgerytype1  0.357    0.0675    5.30  1.30e- 7    1.43
2 cognitivedecline1 0.323    0.132     2.45  1.42e- 2    1.38
3 cancer1      0.975    0.0977   9.98  6.23e-23    2.65
4 autoimmune1   0.328    0.0779    4.21  2.71e- 5    1.39
5 transferredin1 -0.287    0.0992   -2.89  3.85e- 3    0.751
6 age          0.381    0.0460    8.29  2.01e-16    1.46
7 sexMale       0.231    0.0681    3.39  7.19e- 4    1.26
8 bloodpressure -0.0841   0.0369   -2.28  2.26e- 2    0.919
9 temperature   -0.134   0.0383   -3.51  4.59e- 4    0.874
10 creatinelevels 0.109    0.0413    2.63  8.60e- 3    1.11
11 kg          -0.220    0.0365   -6.02  2.04e- 9    0.803
12 nolifesupportorderYes 1.27     0.156     8.17  5.51e-16    3.57
13 traumaYes    -0.757    0.349    -2.17  3.05e- 2    0.469

```

```

Call:
svyglm(formula = died ~ surgerytype + severe + cognitivedecline +
    depression + cancer + autoimmune + transferredin + age + sex +
    yearseducation + bloodpressure + temperature + creatinelevels +
    sodiumlevels + urineweight + kg + nolifesupportorder + insurancetype +
    resp + infection + trauma + race + income + kg_na, svy_design_boot_ate,
    family = binomial())

```

Survey design:

```
as.svrepdesign(svy_design_ate, type = c("bootstrap"), replicates = 2000)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.100458	0.147937	0.679	0.497181
surgerytype1	0.357418	0.067466	5.298	1.30e-07 ***
severe1	0.013838	0.094590	0.146	0.883700
cognitivedecline1	0.323016	0.131602	2.454	0.014194 *
depression1	-0.076669	0.149518	-0.513	0.608167
cancer1	0.975033	0.097662	9.984	< 2e-16 ***
autoimmune1	0.327612	0.077882	4.207	2.71e-05 ***
transferredin1	-0.286914	0.099164	-2.893	0.003854 **
age	0.381394	0.045986	8.294	< 2e-16 ***
sexMale	0.230842	0.068142	3.388	0.000719 ***
yearseducation	0.009139	0.037539	0.243	0.807683
bloodpressure	-0.084130	0.036880	-2.281	0.022643 *
temperature	-0.134303	0.038267	-3.510	0.000459 ***
creatinelevels	0.108638	0.041304	2.630	0.008600 **
sodiumlevels	0.005322	0.033465	0.159	0.873665
urineweight	0.010498	0.034386	0.305	0.760163
kg	-0.219558	0.036456	-6.023	2.04e-09 ***
nolifesupportorderYes	1.271721	0.155684	8.169	5.51e-16 ***
insurancetypeMedicare	0.069309	0.134903	0.514	0.607471
insurancetypeMedicare & Medicaid	0.059413	0.176412	0.337	0.736314
insurancetypeNo insurance	0.227374	0.162227	1.402	0.161198
insurancetypePrivate	0.064363	0.120301	0.535	0.592700
insurancetypePrivate & Medicare	-0.058888	0.138497	-0.425	0.670744
respYes	-0.055229	0.072778	-0.759	0.448019
infectionYes	0.089826	0.091012	0.987	0.323778
traumaYes	-0.756742	0.349448	-2.166	0.030466 *
raceother	-0.162922	0.160363	-1.016	0.309775
racewhite	-0.103448	0.098123	-1.054	0.291886
income	-0.059454	0.034975	-1.700	0.089313 .
kg_naTRUE	-0.128275	0.113121	-1.134	0.256949

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 5870.023)

Number of Fisher Scoring iterations: 4

Before our final summary of estimation of treatment effect, we can do a weighting with all multiple imputation data to see if only include one of the imputation data lead to any difference.

Weighting with multiple imputed data sets:

Weight ATT:

```
all_imputations <- update(all_imputations,
    weigh_att = ifelse(surgerytype == 1, 1, p_scores/(1 - p_scores)))
```

Weight ATE:

```
all_imputations <- update(all_imputations,
    weigh_ate = ifelse(surgerytype == 1, 1/p_scores, 1/(1 - p_scores)))
```

Model with weight ATT:

```
svy_design_mi_att <- svydesign(ids = ~ 0, weights = ~ weigh_att, data = all_imputations)
model_mi_att <- with(svy_design_mi_att, svyglm(died ~ surgerytype, family = binomial()))
result_model_mi_att <- MIcombine(model_mi_att)
summary(result_model_mi_att)
> summary(result_model_mi_att)

Multiple imputation results:
  with(svy_design_mi_att, svyglm(died ~ surgerytype, family = binomial()))
  MIcombine.default(model_mi_att)
    results      se   (lower     upper) missInfo
(Intercept) 0.4558225 0.04723182 0.3632498 0.5483952      0 %
surgerytype1 0.2998016 0.06585457 0.1707290 0.4288742      0 %

Model with weight ATE:
```

```
svy_design_mi_ate <- svydesign(ids = ~ 0, weights = ~ weigh_ate, data = all_imputations)
model_mi_ate <- with(svy_design_mi_ate, svyglm(died ~ surgerytype, family = binomial()))
result_model_mi_ate <- MIcombine(model_mi_ate)
summary(result_model_mi_ate)
> summary(result_model_mi_ate)

Multiple imputation results:
  with(svy_design_mi_ate, svyglm(died ~ surgerytype, family = binomial()))
  MIcombine.default(model_mi_ate)
    results      se   (lower     upper) missInfo
(Intercept) 0.5019437 0.03690303 0.4296151 0.5742724      0 %
surgerytype1 0.3190726 0.06332952 0.1949491 0.4431962      0 %
```

Used a simple formula here, we can see that there is 0% missing information, which means both the estimate and standard error for all imputation data are very similar to those obtained with a single imputed data set.

Thus, I will use the model results obtained from single imputed data set as our final model results and build a summary based on them.

## #7. Sensitivity analysis

Before our final summary for propensity scores weighting, we still need to run the sensitivity analysis:

Start with ATT:

```
sens_att <- treatSens(formula = died ~ surgerytype + p_scores +
  I(p_scores ^ 2) + I(p_scores ^ 3), resp.family = binomial(),
  trt.family = binomial(link = "probit"), grid.dim = c(5, 5), nsim = 20,
  weights = complete_data1$weight_att, data = complete_data1)
```

Error in warnings(formula, resp.family, trt.family, theta, grid.dim, standardize, :  
An outcome variable needs to be continuous.

Received an error message indicating that the outcome variable needs to be continuous. As we know, the response variable “died” is binary. Then I went back to the documentation of treatSens function and get that information: “resp.family: an assumed family for GLM of the response model. Currently only supports gaussian.” Which means the author of that package hasn’t included binomial method for the response variable yet, so, unfortunately, I had to give up the sensitivity analysis.

## #8. Summary on propensity scores weighting

We built a logistic regression on imputed patients data set focusing on the effect of treatment variable (surgery type) on outcome variable (died: whether patients died in a six-month period from the start of surgery to the end of surgery) with application of propensity scores weighting.

In the data preparation stage, we used multivariate imputation via chained equation to get imputed data. Based on the imputed data, we used logistic regression, random forest, and generalize boosted modeling to get propensity scores. Logistic regression propensity scores have the highest common support among them. In the covariate balance check with propensity weighting, logistic regression propensity scores outperformed the others (maximum absolute value of standardized effect size for ATT = 0.091, for ATE = 0.024), for random forest (maximum absolute value of standardized effect size for ATT = 0.076, for ATE = 0.163), for generalize boosted modeling (maximum absolute value of standardized effect size for ATT = 0.050, for ATE = 0.084). Thus, our final model for estimation of treatment effect is based on propensity scores weighting obtained from logistic regression.

Our final result is based on the bootstrap svyglm model, as it is more robust compared to simple svyglm model. For average treatment effect on the treated (ATT), surgerytype1 has a significant effect ( $b = 0.332$ , std.error = 0.069, t-value = 4.79, p-value < 0.001, odds-ratio = 1.39) on the outcome variable (died), which means for ATT, the application of surgery type 1 has an odds-ratio of 1.39 compared to surgery type 0 leading to death in a six-month period from the start of surgery to the end of surgery. For average treatment effect (ATE), surgerytype1 has a significant effect ( $b = 0.357$ , std.error = 0.068, t-value = 5.30, p-value < 0.001, odds-ratio = 1.43), which means for ATE, the application of surgery type 1 compared to surgery type 0 has an odds-ratio of 1.43 leading to death in a six-month period from the start of surgery to the end of surgery.

Unfortunately, given the outcome variable is binary, sensitivity analysis could not be performed.

In a nutshell, in order to reduce the death rate, patients and the hospital should choose surgery type 0 instead of surgery type 1.

## #9. Propensity scores stratification

In this part, I will focus on estimation of ATT to save some space. I may have been too comprehensive in the weighting part.

### #9.1 General stratification

Stratification of ATT with logistic regression propensity scores:

We need to coerce the treatment variable to numeric again:

```
complete_data1$surgerytype <- as.numeric(complete_data1$surgerytype == 1)
stratification <- matchit(ps_formula_final, distance = complete_data1$p_scores,
                           data = complete_data1, method = "subclass", sub.by = "treat",
                           subclass = 5)
```

stratification

```

Call:
matchit(formula = ps_formula_final, data = complete_data1, method = "subclass",
        distance = complete_data1$p_scores, sub.by = "treat", subclass = 5)

```

Sample sizes by subclasses:

	Control	Treated
All	3551	2184
Subclass 1	1702	437
Subclass 2	779	437
Subclass 3	511	436
Subclass 4	334	437
Subclass 5	225	437

Covariate balance check:

```

balance_stratification <- summary(stratification, standardize = TRUE)
strat_diff <- data.frame(balance_stratification$q.table[, 3, ])
summary(strat_diff)
> summary(strat_diff)
  Subclass.1      Subclass.2      Subclass.3      Subclass.4      Subclass.5 
Min.   :-0.208656  Min.   :-0.1048866  Min.   :-0.113704  Min.   :-0.08025  Min.   :-0.394330 
1st Qu.:-0.086486  1st Qu.:-0.0401929  1st Qu.:-0.038027  1st Qu.:-0.04041  1st Qu.:-0.042485 
Median : 0.020536  Median :-0.0022537  Median : 0.011608  Median :-0.01503  Median : 0.023288 
Mean   :-0.007618  Mean   :-0.0009989  Mean   : 0.008219  Mean   :-0.01347  Mean   : 0.008772 
3rd Qu.: 0.077797  3rd Qu.: 0.0408501  3rd Qu.: 0.045617  3rd Qu.: 0.02002  3rd Qu.: 0.079277 
Max.   : 0.164431  Max.   : 0.0983545  Max.   : 0.135399  Max.   : 0.07141  Max.   : 0.286959 

```

It can be concluded that adequate covariate balance was only achieved in subclass 4 with the criterion that standardized mean differences should be less than 0.1. We can use marginal mean weighting through stratification to mitigate this problem later.

Stratification of ATT with random forest propensity scores:

```

stratification_rf <- matchit(ps_formula_final, distance = complete_data1$p_score_rf,
                               data = complete_data1, method = "subclass", sub.by = "treat",
                               subclass = 5)

```

stratification\_rf

```

Call:
matchit(formula = ps_formula_final, data = complete_data1, method = "subclass",
        distance = complete_data1$p_score_rf, sub.by = "treat", subclass = 5)

```

Sample sizes by subclasses:

	Control	Treated
All	3551	2184
Subclass 1	2831	437
Subclass 2	414	437
Subclass 3	213	436
Subclass 4	80	437
Subclass 5	13	437

Balance check:

```

balance_stratification_rf <- summary(stratification_rf, standardize = TRUE)
strat_diff_rf <- data.frame(balance_stratification_rf$q.table[, 3, ])
summary(strat_diff_rf)

```

```
> summary(strat_diff_rf)
   Subclass.1      Subclass.2      Subclass.3      Subclass.4      Subclass.5
Min.   :-0.0940524  Min.   :-0.33968  Min.   :-0.52861  Min.   :-0.31099  Min.   :-0.75859
1st Qu.:-0.0396341 1st Qu.:-0.04934  1st Qu.:-0.06621  1st Qu.:-0.17675  1st Qu.:-0.12420
Median : 0.0007616  Median : 0.01885  Median : 0.02402  Median : 0.09539  Median : 0.04378
Mean   : 0.0307615  Mean   : 0.01687  Mean   : 0.01170  Mean   : 0.01953  Mean   :-0.02964
3rd Qu.: 0.0325740  3rd Qu.: 0.06650  3rd Qu.: 0.10701  3rd Qu.: 0.16217  3rd Qu.: 0.13551
Max.   : 0.6724606  Max.   : 0.45900  Max.   : 0.36982  Max.   : 0.36004  Max.   : 0.35030
```

Well, the performance of random forest propensity scores is even worse compared with logistic regression propensity scores.

Stratification of ATT with generalize boosted modeling propensity scores:

```
stratification_gbm <- matchit(ps_formula_final, distance = complete_data1$p_score_gbm,
                                data = complete_data1, method = "subclass", sub.by = "treat",
                                subclass = 5)
```

stratification\_gbm

```
Call:
matchit(formula = ps_formula_final, data = complete_data1, method = "subclass",
         distance = complete_data1$p_score_gbm, sub.by = "treat",
         subclass = 5)
```

Sample sizes by subclasses:

	Control	Treated
All	3551	2184
Subclass 1	2208	437
Subclass 2	653	437
Subclass 3	383	436
Subclass 4	211	437
Subclass 5	96	437

Covariate balance check:

```
balance_stratification_gbm <- summary(stratification_gbm, standardize = TRUE)
```

```
strat_diff_gbm <- data.frame(balance_stratification_gbm$q.table[, 3, ])
```

summary(strat\_diff\_gbm)

```
> summary(strat_diff_gbm)
```

	Subclass.1	Subclass.2	Subclass.3	Subclass.4	Subclass.5
Min.	-0.21114	-0.153268	-0.105413	-0.11448	-0.20130
1st Qu.	-0.05781	-0.031599	-0.034030	-0.01800	-0.01617
Median	-0.01845	0.002219	-0.014010	0.03996	0.02912
Mean	-0.01431	-0.004808	-0.003538	0.03191	0.03225
3rd Qu.	0.03630	0.030032	0.025127	0.08815	0.10386
Max.	0.26325	0.099764	0.145010	0.13555	0.20474

Seems like the generalize boosted modeling method generates the best result in covariate balance, we can just try to improve the subclass numbers to see if it can reach the loose criterion in all subclasses which is 0.25:

```
stratification_gbm1 <- matchit(ps_formula_final, distance = complete_data1$p_score_gbm,
                                 data = complete_data1, method = "subclass", sub.by = "treat",
                                 subclass = 8)
```

```
balance_stratification_gbm1 <- summary(stratification_gbm1, standardize = TRUE)
```

```
strat_diff_gbm1 <- data.frame(balance_stratification_gbm1$q.table[, 3, ])
```

summary(strat\_diff\_gbm1)

```

> summary(strat_diff_gbm1)
   Subclass.1      Subclass.2      Subclass.3      Subclass.4
Min. : -0.17394  Min. : -0.210383  Min. : -2.113e-01  Min. : -0.114708
1st Qu.: -0.06446 1st Qu.: -0.042637  1st Qu.: -3.738e-02  1st Qu.: -0.049467
Median : -0.01898 Median : 0.011108  Median : 2.580e-04  Median : 0.005043
Mean   : -0.02164 Mean   : 0.004763  Mean   : 4.404e-05  Mean   : 0.002560
3rd Qu.: 0.01023  3rd Qu.: 0.046302  3rd Qu.: 4.687e-02  3rd Qu.: 0.040107
Max.   : 0.20546  Max.   : 0.287244  Max.   : 1.324e-01  Max.   : 0.101370
   Subclass.5      Subclass.6      Subclass.7      Subclass.8
Min. : -0.18507  Min. : -0.14537  Min. : -0.12311  Min. : -0.20553
1st Qu.: -0.06663 1st Qu.: -0.02847  1st Qu.: -0.03946  1st Qu.: -0.05913
Median : -0.03620 Median : 0.03779  Median : 0.01602  Median : 0.02571
Mean   : -0.02071 Mean   : 0.04385  Mean   : 0.03129  Mean   : 0.02199
3rd Qu.: 0.01724  3rd Qu.: 0.10211  3rd Qu.: 0.09359  3rd Qu.: 0.11554
Max.   : 0.22392  Max.   : 0.22648  Max.   : 0.24540  Max.   : 0.22109

```

Yes, we reached the cut off of 0.25 by using 8 subclasses. We can still try with the marginal mean weighting through stratification for better covariate balance.

#9.2 Marginal mean weighting through stratification for ATT

```

stratum_working <- match.data(stratification)
stratum_working_rf <- match.data(stratification_rf)
stratum_working_gbm <- match.data(stratification_gbm)

```

For logistic regression propensity scores:

```
design_norm <- svydesign(ids = ~0, data = stratum_working)
```

```

ntreat_norm <- data.frame(table(stratum_working$subclass[stratum_working$surgerytype == 1]))
names(ntreat_norm) <- c("subclass", "N.1s")
ncontrol_norm <- data.frame(table(stratum_working$subclass[stratum_working$surgerytype == 0]))
names(ncontrol_norm) <- c("subclass", "N.0s")

```

```

scounts <- merge(ntreat_norm, ncontrol_norm)
stratum_working <- merge(stratum_working, scounts)
propt_norm <- svymean(~factor(surgerytype), design_norm)

```

```

stratum_working$w_norm <- with(stratum_working,
  ifelse(surgerytype == 1, 1,
         stratum_working$N.1s * propt_norm[1] / stratum_working$N.0s *
  propt_norm[2]))

```

```

xtabs(~w_norm + subclass, stratum_working)
> xtabs(~w_norm + subclass, stratum_working)

```

	subclass				
w_norm	1	2	3	4	5
0.0605422201655125	1702	0	0	0	0
0.132275813506678	0	779	0	0	0
0.20118798964055	0	0	511	0	0
0.308511553058989	0	0	0	334	0
0.457968260985343	0	0	0	0	225
1	437	437	436	437	437

Covariate balance check:

```
stratum_working$norm_norm_weight <- stratum_working$w_norm /  
mean(stratum_working$w_norm)  
norm_table <- bal.stat(stratum_working, estimand = "ATT", w.all =  
stratum_working$norm_norm_weight,  
    vars = cov_names_final, sampw = 1, get.ks = FALSE,  
    treat.var = "surgerytype", multinom = FALSE)  
round(norm_table$results[, 1:5], 3)  
summary(abs(norm_table$results[, 5]))  
> round(norm_table$results[, 1:5], 3)
```

	tx.mn	tx.sd	ct.mn	ct.sd	std.eff.sz
severe:0	0.796	0.403	0.798	0.402	-0.005
severe:1	0.204	0.403	0.202	0.402	0.005
cognitive decline:0	0.931	0.254	0.926	0.261	0.018
cognitive decline:1	0.069	0.254	0.074	0.261	-0.018
depression:0	0.954	0.209	0.951	0.216	0.016
depression:1	0.046	0.209	0.049	0.216	-0.016
cancer:0	0.797	0.402	0.789	0.408	0.020
cancer:1	0.203	0.402	0.211	0.408	-0.020
transferredin:0	0.850	0.357	0.859	0.348	-0.025
transferredin:1	0.150	0.357	0.141	0.348	0.025
bloodpressure	-0.271	0.900	-0.215	0.901	-0.063
yearseducation	0.058	1.003	0.041	0.986	0.017
creatinelevels	0.166	1.000	0.168	1.347	-0.002
sodiumlevels	-0.057	0.993	-0.048	1.014	-0.009
urineweight	-0.075	1.037	-0.055	1.028	-0.019
kg	0.156	0.982	0.115	1.071	0.042
nolifesupportorder:No	0.929	0.257	0.917	0.276	0.046
nolifesupportorder:Yes	0.071	0.257	0.083	0.276	-0.046
insurancetype:Medicaid	0.088	0.284	0.093	0.291	-0.018
insurancetype:Medicare	0.234	0.423	0.234	0.423	0.001
insurancetype:Medicare & Medicaid	0.056	0.231	0.057	0.231	-0.002
insurancetype:No insurance	0.062	0.242	0.059	0.235	0.014
insurancetype:Private	0.335	0.472	0.327	0.469	0.016
insurancetype:Private & Medicare	0.224	0.417	0.230	0.421	-0.015
resp:No	0.711	0.453	0.703	0.457	0.018
resp:Yes	0.289	0.453	0.297	0.457	-0.018
infection:No	0.764	0.425	0.767	0.423	-0.007
infection:Yes	0.236	0.425	0.233	0.423	0.007
trauma:No	0.984	0.124	0.986	0.117	-0.013
trauma:Yes	0.016	0.124	0.014	0.117	0.013
kg_na:FALSE	0.931	0.254	0.931	0.254	0.000
kg_na:TRUE	0.069	0.254	0.069	0.254	0.000

```
> summary(abs(norm_table$results[, 5]))
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.0004945	0.0065514	0.0164599	0.0172781	0.0192252	0.0630060

Now the result is close to perfect. To save some space, I will no longer display this whole table, instead using the summary of the absolute value of standardized effect size.

For random forest propensity scores

```
design_norm_rf <- svydesign(ids = ~0, data = stratum_working_rf)

ntreat_norm_rf <- data.frame(table(stratum_working_rf$subclass[stratum_working_rf$surgerytype == 1]))
names(ntreat_norm_rf) <- c("subclass", "N.1s")
ncontrol_norm_rf <-
data.frame(table(stratum_working_rf$subclass[stratum_working_rf$surgerytype == 0]))
names(ncontrol_norm_rf) <- c("subclass", "N.0s")

scounts_rf <- merge(ntreat_norm_rf, ncontrol_norm_rf)
stratum_working_rf <- merge(stratum_working_rf, scounts_rf)
propt_norm_rf <- svymean(~factor(surgerytype), design_norm_rf)

stratum_working_rf$w_norm <- with(stratum_working_rf,
ifelse(surgerytype == 1, 1,
stratum_working_rf$N.1s * propt_norm_rf[1] /
stratum_working_rf$N.0s * propt_norm_rf[2]))

xtabs(~w_norm + subclass, stratum_working_rf)
> xtabs(~w_norm + subclass, stratum_working_rf)
  subclass
  w_norm      1   2   3   4   5
  0.0363980426427772 2831   0   0   0   0
  0.248895794013774    0 414   0   0   0
  0.482662266226858    0   0 213   0   0
  1                 437 437 436 437 437
  1.28803573402128     0   0   0  80   0
  7.92637374782325     0   0   0   0  13
  
```

Covariate balance check:

```
stratum_working_rf$rf_norm_weight <- stratum_working_rf$w_norm /
mean(stratum_working_rf$w_norm)
rf_table <- bal.stat(stratum_working_rf, estimand = "ATT", w.all =
stratum_working_rf$rf_norm_weight,
vars = cov_names_final, sampw = 1, get.ks = FALSE,
treat.var = "surgerytype", multinom = FALSE)

summary(abs(rf_table$results[, 5]))
> summary(abs(rf_table$results[, 5]))
  Min. 1st Qu. Median Mean 3rd Qu. Max.
 0.01853 0.04432 0.06457 0.10288 0.17535 0.28353
  
```

This one is still not good enough.

For generalize boosted modeling propensity scores:

```
design_norm_gbm <- svydesign(ids = ~0, data = stratum_working_gbm)
```

```
ntreat_norm_gbm <-
data.frame(table(stratum_working_gbm$subclass[stratum_working_gbm$surgerytype == 1]))
names(ntreat_norm_gbm) <- c("subclass", "N.1s")
```

```

ncontrol_norm_gbm <-
  data.frame(table(stratum_working_gbm$subclass[stratum_working_gbm$surgerytype == 0]))
  names(ncontrol_norm_gbm) <- c("subclass", "N.0s")

scounts_gbm <- merge(ntreat_norm_gbm, ncontrol_norm_gbm)
stratum_working_gbm <- merge(stratum_working_gbm, scounts_gbm)
propt_norm_gbm <- svymean(~factor(surgerytype), design_norm_gbm)

stratum_working_gbm$w_norm <- with(stratum_working_gbm,
  ifelse(surgerytype == 1, 1,
         stratum_working_gbm$N.1s * propt_norm_gbm[1] /
  stratum_working_gbm$N.0s * propt_norm_gbm[2]))

xtabs(~w_norm + subclass, stratum_working_gbm)
> xtabs(~w_norm + subclass, stratum_working_gbm)
  subclass
  w_norm      1   2   3   4   5
  0.0466679613775826 2208   0   0   0   0
  0.157799171089896   0  653   0   0   0
  0.268425751191438   0   0  383   0   0
  0.488354780671575   0   0   0  211   0
  1                 437  437  436  437  437
  1.0733631116844     0   0   0   0   96

```

Covariate balance check:

```

stratum_working_gbm$gbm_norm_weight <- stratum_working_gbm$w_norm /
mean(stratum_working_gbm$w_norm)
gbm_table <- bal.stat(stratum_working_gbm, estimand = "ATT", w.all =
stratum_working_gbm$gbm_norm_weight,
  vars = cov_names_final, sampw = 1, get.ks = FALSE,
  treat.var = "surgerytype", multinom = FALSE)

summary(abs(gbm_table$results[, 5]))
> summary(abs(gbm_table$results[, 5]))
  Min. 1st Qu. Median Mean 3rd Qu. Max.
0.000101 0.005274 0.017842 0.018605 0.027008 0.064929

```

The result is good and very close to logistic regression propensity scores. As the mean of standardized effect size in logistic regression propensity scores is slightly lower, we will choose marginal mean weighting through stratification using logistic regression propensity scores to run our final model.

## #10. Estimation of treatment effect by stratification

```

final_design <- svydesign(ids = ~0, weights = stratum_working$norm_norm_weight, data =
stratum_working)
final_design_boot <- as.svrepdesign(final_design, type = c("bootstrap"), replicates = 2000)
model_boot <- svyglm(died ~ factor(surgerytype) + severe + cognitivedecline + depression +
cancer +
  autoimmune + transferredin + age + sex + yearseducation +

```

```

bloodpressure + temperature + creatinelevels + sodiumlevels +
urineweight + kg + nolifesupportorder + insurancetype + resp +
infection + trauma + race + income + kg_na,
final_design_boot, family = binomial())
summary(model_boot)

Call:
svyglm(formula = died ~ factor(surgerytype) + severe + cognitivedecline +
depression + cancer + autoimmune + transferredin + age + sex +
yearseducation + bloodpressure + temperature + creatinelevels +
sodiumlevels + urineweight + kg + nolifesupportorder + insurancetype +
resp + infection + trauma + race + income + kg_na, final_design_boot,
family = binomial())

Survey design:
as.svrepdesign(final_design, type = c("bootstrap"), replicates = 2000)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0945997 0.1833288 0.516 0.605904
factor(surgerytype)1 0.3198338 0.0695470 4.599 4.52e-06 ***
severe1 0.1188963 0.1111709 1.069 0.284980
cognitivedecline1 0.3493427 0.1788429 1.953 0.050920 .
depression1 -0.0647077 0.2034614 -0.318 0.750493
cancer1 0.9272522 0.1204664 7.697 2.18e-14 ***
autoimmune1 0.3382756 0.0972940 3.477 0.000518 ***
transferredin1 -0.3188619 0.1126339 -2.831 0.004688 **
age 0.3400237 0.0562320 6.047 1.76e-09 ***
sexMale 0.1271564 0.0858733 1.481 0.138835
yearseducation -0.0235260 0.0465525 -0.505 0.613359
bloodpressure -0.1240603 0.0452194 -2.744 0.006134 **
temperature -0.1285663 0.0420711 -3.056 0.002274 **
creatinelevels 0.1429334 0.0447636 3.193 0.001430 **
sodiumlevels 0.0003357 0.0416705 0.008 0.993573
urineweight -0.0427548 0.0419262 -1.020 0.307966
kg -0.1765680 0.0424596 -4.158 3.34e-05 ***
nolifesupportorderYes 1.3965068 0.2359648 5.918 3.83e-09 ***
insurancetypeMedicare 0.2042223 0.1736928 1.176 0.239830
insurancetypeMedicare & Medicaid 0.0525440 0.2287807 0.230 0.818372
insurancetypeNo insurance 0.1080374 0.2025390 0.533 0.593806
insurancetypePrivate 0.0556922 0.1537655 0.362 0.717249
insurancetypePrivate & Medicare 0.0340642 0.1769447 0.193 0.847360
respYes -0.0908912 0.0882998 -1.029 0.303443
infectionYes 0.0535118 0.1024429 0.522 0.601480
traumaYes -0.5804867 0.3630572 -1.599 0.110007
raceother 0.0214369 0.1959524 0.109 0.912898
racewhite -0.0925585 0.1223024 -0.757 0.449260
income -0.0593067 0.0418302 -1.418 0.156409
kg_naTRUE -0.0198671 0.1556603 -0.128 0.898454
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 5827.175)

Number of Fisher Scoring iterations: 4

Create a tidy version of the summary for later report:

```

```

outcome_strat <- tidy(model_boot) %>%
  filter(p.value < 0.05) %>%
  mutate(odd_ratio = exp(estimate))
outcome_strat

> outcome_strat
# A tibble: 10 × 6
  term            estimate std.error statistic p.value odd_ratio
  <chr>          <dbl>     <dbl>      <dbl>    <dbl>      <dbl>
1 factor(surgerytype)1 0.320     0.0695     4.60 4.52e- 6    1.38
2 cancer1           0.927     0.120      7.70 2.18e-14   2.53
3 autoimmune1       0.338     0.0973     3.48 5.18e- 4    1.40
4 transferredin1    -0.319     0.113      -2.83 4.69e- 3    0.727
5 age                0.340     0.0562     6.05 1.76e- 9    1.40
6 bloodpressure      -0.124     0.0452     -2.74 6.13e- 3    0.883
7 temperature        -0.129     0.0421     -3.06 2.27e- 3    0.879
8 creatinelevels     0.143     0.0448     3.19 1.43e- 3    1.15
9 kg                 -0.177     0.0425     -4.16 3.34e- 5    0.838
10 nolifesupportorderYes 1.40      0.236      5.92 3.83e- 9    4.04

```

We can see that the surgerytype1 is a significant predictor on died variable.

Before the final summary of stratification, we still need to check for sensitivity analysis. But based on the same reason I mentioned in weighting, there is still no reliable method for sensitivity analysis with binary outcome variable. So, I will just leave the code here for reference:

Sensitivity analysis:

```

sens_strat <- treatSens(formula = died ~ surgerytype + p_scores +
  I(p_scores ^ 2) + I(p_scores ^ 3), resp.family = binomial(),
  trt.family = binomial(link = "probit"), grid.dim = c(5, 5), nsim = 20,
  weights = stratum_working$norm_norm_weight, data = stratum_working)

```

## #11. Summary on propensity scores stratification

I'm reporting both weighting and stratification because both of them have passed the covariate balance check, for now, I think both of them are worth reporting.

We built a logistic regression on imputed patients data set focusing on the effect of treatment variable (surgery type) of treated (ATT) on outcome variable (died: whether patients died in a six-month period from the start of surgery to the end of surgery) with application of propensity scores stratification.

Continued on the propensity scores obtained in the weighting part, we first used basic stratification. In the covariate balance check with basic propensity stratification, all the propensity scores performed not good enough: for logistic regression propensity scores stratification, only subclass 4 is below the criterion of standardized effect size (maximum absolute value of standardized effect size of subclass 4 < 0.1); for random forest, all subclasses are above the criterion; and the situation is the same as for generalized boosted modeling.

To improve the covariate balance, we used the marginal mean weighting through stratification method. By doing so, the covariate balance is improved: for logistic regression propensity scores (maximum absolute value of standardized effect size for ATT = 0.063);; for random forest (maximum absolute value of standardized effect size for ATT = 0.284), it is improved, but still not good enough; for generalize boosted modeling (maximum absolute value of standardized effect size for ATT = 0.065). Although the covariate balance difference between logistic regression propensity scores and the generalize boosted modeling propensity scores is small, the logistic one is still outperformed generalize boosted modeling by a little bit. So, our final model for estimation of treatment effect is based on marginal mean weighting through stratification using propensity scores obtained from logistic regression.

Our final result is based on the bootstrap svyglm model, as it is more robust compared to simple svyglm model. As for average treatment effect on the treated (ATT), surgerytype1 has a significant effect ( $b = 0.320$ , std.error = 0.070, t-value = 4.60, p-value < 0.001, odds-ratio = 1.38) on the outcome variable (died), which means for ATT, the application of surgery type 1 has an odds-ratio of 1.38 compared to surgery type 0 leading to death in a six-month period from the start of surgery to the end of surgery. Unfortunately, given the outcome variable is binary, sensitivity analysis could not be performed.

In a nutshell, in order to reduce the death rate, patients and the hospital should choose surgery type 0 instead of surgery type 1.

PS: if we combine the result from weighting and stratification, we will find that the treatment effect is basically the same with respect to ATT.