Big Match - Large Demo

Rachael Caelie (Rocky) Aikens

10/30/2018

This rmarkdown is meant for running a larger demo of the big_match functionality.

Import Data

This sample cohort data was provided by Justin Lee at the Quantitative Sciences Unit. It contains ~900,000 observations of 112 variables.

```
dat <- read_sas("../sample_data/justincohort_june2017.sas7bdat")

# dimensions: ~900,000 x 112
dim(dat)
```

```
## [1] 893498    112
```

We include only hospitalizations with `totalct` > 1 and `arteryCt < 3`. A patient is considered to have recieved treatment if `arteryCt` is greater than 1. The outcome of this analysis is mortality.

```
# filter and add treatment column
dat <- filter(dat, totalct > 1 & arteryCt < 3) %>%
  mutate(treat = ifelse(arteryCt > 1, 1, 0))

# dimensions: ~900,000 x 112
dim(dat)
```

```
## [1] 833657    113
```

Manual Statify

As a simple example, we can choose to stratify manually by race, sex, and hospital state. This is done as shown below and gives the following strata sizes:

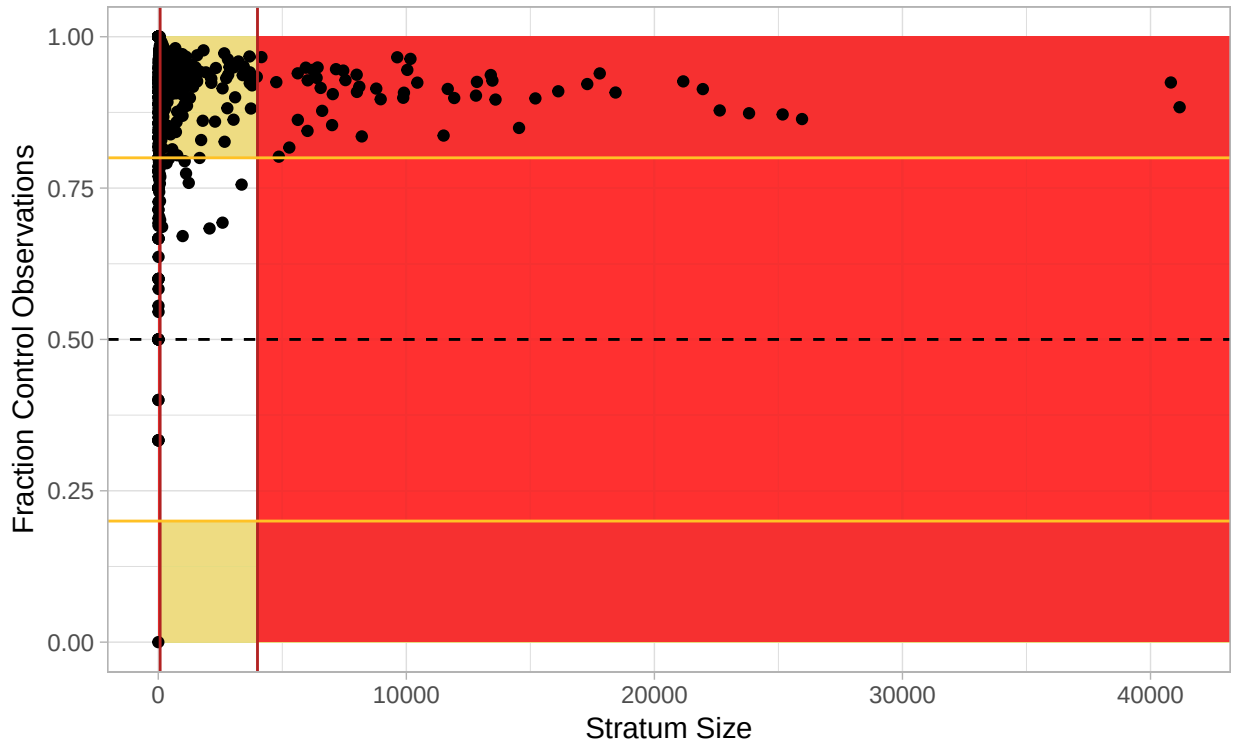
```
m.strat <- manual_stratify(dat, treat = "treat",
                           covariates = c("Male", "race", "hosp_state"))

print(m.strat)
```

```
## manual_strata object from package big_match.
##
## Function call:
## manual_stratify(data = dat, treat = "treat", covariates = c("Male",
##     "race", "hosp_state"))
##
## Analysis set dimensions: 833657 X 114
##
## Number of strata: 704
##
##  Min size: 1     Max size: 41175
```

The minimum size is 1 (far too small) and the maximum is 41175 (too large to be computationally feasible). The diagnostic plot below shows all strata based on their treat/control balance and number of observations. The un-tinted rectangle is the target for strata with good balance and appropriate size.

```
plot(m.strat)
```



If desired, one can use trial and error to select a covariate set for stratification which gives more favorable strata sizes and balances.

Auto Stratify by Prognostic Score

Another option is to stratify based on prognostic score. There are a few ways to do this. The simplest method is to supply a list of covariates to `auto_stratify` which details the features we would like to include in a prognostic model. The `auto_stratify` function will then automatically split the data randomly into a `model_set` and an `analysis_set`. By default, the model set contains 1/10 of the controls from the input dataset. A logistic model for prognostic score is then fit on the `model_set` using the covariates specified. Once the prognostic model is fit, the model set should be kept separate from the data used for the remainder of the analysis.

Using our new prognostic model, we can assign prognostic scores to each observation in the analysis set. Finally, using these scores, the analysis set can be divided into strata of relatively equal size, where observations which appear in the same strata are characterized by similar prognostic scores.

Common Problem: Non-Representative Model Set

The code to automatically stratify the dataset might look like this:

```
set.seed(123)
a.strat <- auto_stratify(data = dat, treat = "treat",
                         outcome = "dead",
                         prog_formula = dead ~ totalct + hosp_state + AMI_7 + COPD_7 +
                           ISCHEMICHEART_7 + STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7
                           ALZH_DEMEN_7 + Male + race)
```

In this example dataset, we see a common problem. When we try to fit a prognostic model which includes the hospital state as a variable, the following error is showm:

```
Error applying prognostic model: Some categorical variable value(s) in the analysis set
do not appear in the modeling set.
```

This occurs because some value(s) of one or more categorical variables appear in the analysis set which were not seen in the model set. This means that when we try to obtain prognostic scores for our analysis set, we run into some new value that our prognostic model was not prepared to handle. For example, suppose all individuals in the model set had `hosp_state` equal to some state in the continental US, and the analysis contained one example from with `hosp_state` equal to "Virgin Islands". We don't know what prognostic score to assign this individual because we did not see any patients in the Virgin Islands when we built our prognostic model.

We have the following options:

1. **Group by this variable in auto_stratify:** add the offending categorical variable to group_vars in the call to auto_stratify. This will ensure that the model set contains the same proportions of individuals from each category as in the original dataset.
2. **Rejection Sampling:** run auto_stratify again and again with different random seeds until this error does not occur
3. **Remove the offending examples from the entire data set**
4. **Remove this variable from the prognostic model**

In this case, there are only two hospitalizations from the Virgin Islands:

```
filter(dat, hosp_state == "Virgin Islands") %>% nrow()
```

```
## [1] 2
```

For simplicity, we will remove these observations for now.

```r
dat <- filter(dat, hosp_state != "Virgin Islands")
```

Basic Auto Strification

Now, we can run the code as before:

```r
set.seed(123)
a.strat <- auto_stratify(data = dat, treat = "treat",
                         outcome = "dead",
                         prog_formula = dead ~ totalct + hosp_state + AMI_7 + COPD_7 +
                           ISCHEMICHEART_7 + STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7
                           ALZH_DEMEN_7 + Male + race)
```

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
```

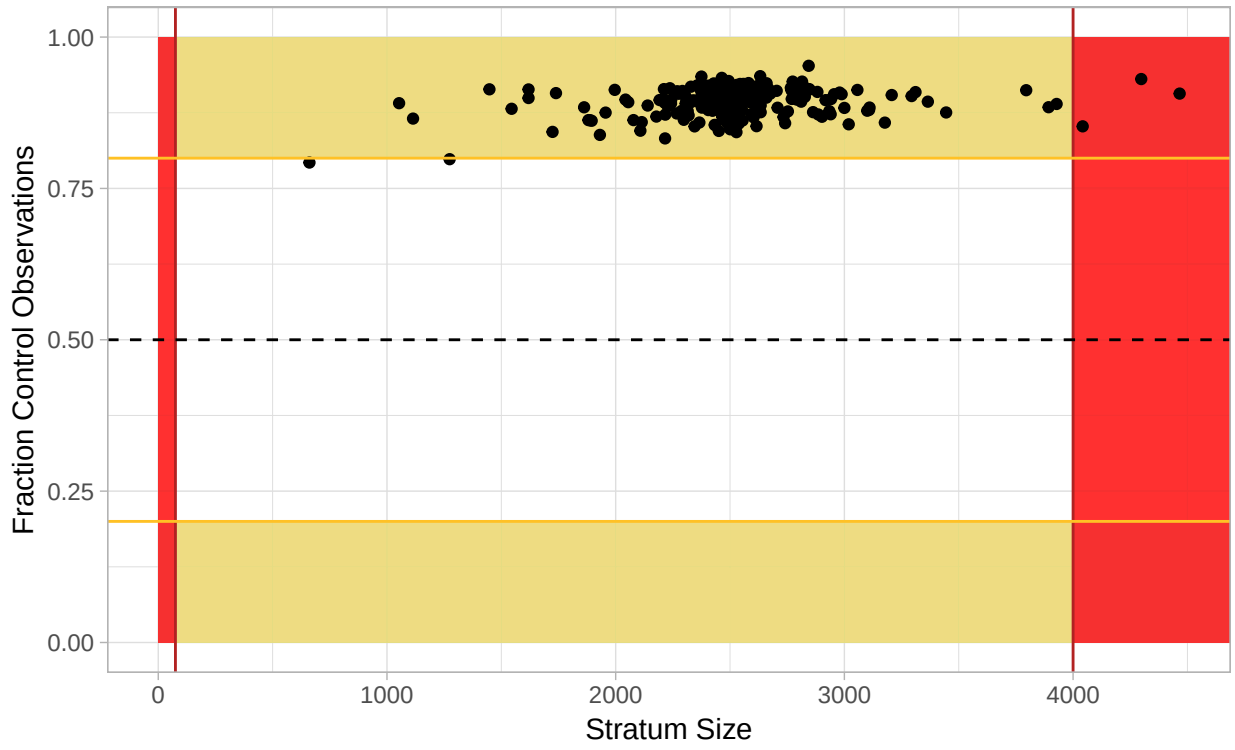
```r
print(a.strat)
```

```
## auto_strata object from package big_match.
##
## Function call:
## auto_stratify(data = dat, treat = "treat", outcome = "dead",
##     covariates = c("totalct", "hosp_state", "AMI_7", "COPD_7",
##         "ISCHEMICHEART_7", "STROKE_TIA_7", "ATRIAL_FIB_7", "CHRONICKIDNEY_7",
##         "DIABETES_7", "ALZH_DEMEN_7", "Male", "race"))
##
## Analysis set dimensions: 758213 X 114
##
## Model set dimensions: 75442 X 113
##
## Prognostic Score Model:
##
## Call:  glm(formula = formula(formula_str), family = "binomial", data = model_set)
##
## Coefficients:
##                     (Intercept)                         totalct
##                        -9.98541                         0.03078
##              hosp_stateAlabama                 hosp_stateAlaska
##                         8.38816                         7.18931
##              hosp_stateArizona                hosp_stateArkansas
##                         8.32274                         8.11693
##           hosp_stateCalifornia                hosp_stateColorado
##                         8.33747                         8.35222
##          hosp_stateConnecticut                hosp_stateDelaware
##                         8.50978                         7.94440
## hosp_stateDistrict of Columbia                 hosp_stateFlorida
##                         8.63687                         8.30760
##              hosp_stateGeorgia                  hosp_stateHawaii
##                         8.30898                         8.76670
##                hosp_stateIdaho                hosp_stateIllinois
##                         7.97234                         8.40512
##              hosp_stateIndiana                    hosp_stateIowa
```

```
##                     8.31327                     8.46555
##           hosp_stateKansas           hosp_stateKentucky
##                     8.14804                     8.26063
##         hosp_stateLouisiana              hosp_stateMaine
##                     8.32117                     8.29869
##          hosp_stateMaryland       hosp_stateMassachusetts
##                     8.34472                     8.31328
##          hosp_stateMichigan          hosp_stateMinnesota
##                     8.36176                     8.32598
##       hosp_stateMississippi           hosp_stateMissouri
##                     8.23361                     8.30468
##          hosp_stateMontana           hosp_stateNebraska
##                     8.57506                     8.35863
##           hosp_stateNevada       hosp_stateNew Hampshire
##                     8.61286                     8.12860
##        hosp_stateNew Jersey          hosp_stateNew Mexico
##                     8.24553                     8.12606
##          hosp_stateNew York       hosp_stateNorth Carolina
##                     8.28634                     8.25697
##       hosp_stateNorth Dakota                hosp_stateOhio
##                     8.46294                     8.42554
##          hosp_stateOklahoma              hosp_stateOregon
##                     8.28569                     8.15285
##       hosp_statePennsylvania         hosp_statePuerto Rico
##                     8.34529                     8.39532
##       hosp_stateRhode Island       hosp_stateSouth Carolina
##                     8.57200                     8.07109
##       hosp_stateSouth Dakota           hosp_stateTennessee
##                     7.90457                     8.23020
##             hosp_stateTexas                hosp_stateUtah
##                     8.31432                     8.29113
##           hosp_stateVermont           hosp_stateVirginia
##                     8.23726                     8.15050
##         hosp_stateWashington       hosp_stateWest Virginia
##                     8.27598                     8.32826
##          hosp_stateWisconsin              hosp_stateWyoming
##                     8.32709                     7.80117
##                       AMI_7                      COPD_7
##                     0.17583                     0.46509
##              ISCHEMICHEART_7                 STROKE_TIA_7
##                     0.02598                     0.21875
##                 ATRIAL_FIB_7               CHRONICKIDNEY_7
##                     0.61505                     0.42109
##                  DIABETES_7                 ALZH_DEMEN_7
##                     0.22020                     0.43783
##                        Male                   raceBlack
##                    -0.04238                     0.35006
##                 raceHispanic      raceNorth American Native
##                     0.19132                    -0.07869
##                   raceOther                  raceUnknown
##                    -0.05135                     0.13405
##                   raceWhite
##                     0.17457
##
```

```
## Degrees of Freedom: 75441 Total (i.e. Null);  75373 Residual
## Null Deviance:       87320
## Residual Deviance: 84630     AIC: 84770
##
## Number of strata: 302
##
##  Min size: 661   Max size: 4466
```

```
plot(a.strat)
```



Matching

Since we're matching on a single computer, we'll start with a very small subset of the original data to demonstrate the optimal matching functionality.

Here, we subsample down to 30,000 examples:

```
small_dat <- sample_n(dat, 30000)
```

Now, stratify using the prognostic score model:

```
dead ~ totalct + AMI_7 + COPD_7 + ISCHEMICHEART_7 + STROKE_TIA_7 +            ATRIAL_FIB_7
+ CHRONICKIDNEY_7 + DIABETES_7 +           ALZH_DEMEN_7 + Male + race
```

```
t1 <- proc.time()
a.strat <- auto_stratify(small_dat, treat = "treat",
                         outcome = "dead",
                         prog_formula = dead ~ totalct + AMI_7 + COPD_7 +
                           ISCHEMICHEART_7 + STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7
                           ALZH_DEMEN_7 + Male + race)
```

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead ~ totalct + AMI_7 + COPD_7 + ISCHEMICHEART_7 + STROKE_TIA_7 +
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
```

```
strat_time = proc.time()-t1
```

This process took 0.142 seconds.

Now, perform a big match within prognostic score strata. When propensity score is not specified (as below), all variables are used (except of course for outcome and strata).

```
t2 <- proc.time()
mymatch <-   big_match_multidplyr(a.strat, propensity_formula = treat ~ totalct + hosp_state +
              AMI_7 + COPD_7 + ISCHEMICHEART_7 + STROKE_TIA_7 + ATRIAL_FIB_7 +
              CHRONICKIDNEY_7 + DIABETES_7 + ALZH_DEMEN_7 + Male + race)
```

```
## Initialising 12 core cluster.

## Warning: group_indices_.grouped_df ignores extra arguments

## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
match_time <- proc.time() - t1
```

Matching 30,000 subjects across ~12 prognostic score strata took 3.42 seconds. Without using strata, this generally takes more than a minute.