Big_match - Testing and Demo

Rachael 'Rocky' Aikens, Voight Lab

August 24, 2018

This R markdown document is used for testing and demoing the current functionality of bigmatch. We'll use the sample data from the MatchIt package for basic testing.

```
library(MatchIt)
library(ggplot2)
library(ggpubr)
library(dplyr)

data("lalonde")
source('big_match.R')
source('class_functions.R')

# adding a binary outcome
lalonde$outcome <- lalonde$re78 > 15000
lalonde$re78 <- NULL

# changing name of treat column to "treated" to demonstrate that any name will suffice
names(lalonde)[names(lalonde) == "treat"] <- "treated"
```

Stratify

Manual Stratify

Testing errors and warnings

This call should return an error because "educ" is a continuous variable.

```
# manual stratification with a continuous variable
m.strat <- manual_stratify(lalonde, treat = "treat",
                           covariates = c("black", "hispan", "educ", "nodegree"))

# Error in warn_if_continuous(data[, covariates[i]], covariates[i]) :
#   There are 19 distinct values for educ. Is it continuous?

# manual stratification with a continuous variable, force = TRUE
m.strat <- manual_stratify(lalonde, treat = "treat",
                           covariates = c("black", "hispan", "educ", "nodegree"), force = TRUE)

# There are 19 distinct values for educ. Is it continuous?
```

Testing Functionality with Valid Inputs

This call should return six strata (since black and hispanic seem to be mutually exclusive categories in this dataset).

```
# allowable manual stratification
m.strat <- manual_stratify(lalonde, treat = "treated",
                           covariates = c("black", "hispan", "nodegree"))
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
m.strat$n_table
```

```
## # A tibble: 6 x 4
##   Stratum Treat Control Total
##     <dbl> <int>   <dbl> <int>
## 1       1     9     127   136
## 2       2     9     154   163
## 3       3     2      15    17
## 4       4     9      46    55
## 5       5    43      31    74
## 6       6   113      56   169
```

Auto Stratify

Testing Errors and Warnings

First, testing error handling. These should fail and/or give warnings.

```
# auto stratification with missing arguements
a.strat <- auto_stratify(lalonde, "treat", "outcome")

# Error in auto_stratify(lalonde, "treat", "outcome") :
#   At least one of covariates and prog_scores should be specified.
```

```
# auto stratification with covariates and prog scores specified, and prog_scores invalid
a.strat <- auto_stratify(lalonde, "treat", "outcome", c("age", "educ"), prog_scores = 1:4)

# covariates and prog_scores are both specified. Using prog_scores; ignoring covariates.
# Error in auto_stratify(lalonde, "treat", "outcome", c("age", "educ"), :
#   prog_scores must be the same length as the data
```

Testing Functionality with Valid Inputs

These should give valid results.

```
# auto stratification with pre-specified prognostic score
myprogscore <- runif(n = dim(lalonde)[1])
a.strat1 <- auto_stratify(data = lalonde, "treated", "outcome",
                          prog_scores = myprogscore, size = 100)

# auto stratification
a.strat2 <- auto_stratify(lalonde, "treated", "outcome",
                          covariates = c("age", "educ", "hispan", "nodegree", "black"),
                          size = 100)

# auto stratification with a non-continuous prognostic score
a.strat3 <- auto_stratify(lalonde, "treated", "outcome",
                          covariates = c("hispan", "nodegree", "black"), size = 100 )
```

Diagnostics

Most of this is implemented with the generic functions `print`, `summary`, and `plot`.

Print

```
print(m.strat)
```

```
## manual_strata object from package big_match.
##
## Function call:
## manual_stratify(data = lalonde, treat = "treated", covariates = c("black",
##     "hispan", "nodegree"))
##
## Strata Sizes:
## # A tibble: 6 x 4
##   Stratum Treat Control Total
##     <dbl> <int>   <dbl> <int>
## 1       1     9     127   136
## 2       2     9     154   163
## 3       3     2      15    17
## 4       4     9      46    55
## 5       5    43      31    74
## 6       6   113      56   169
```

```
print(a.strat1)
```

```
## auto_strata object from package big_match.
##
## Function call:
## auto_stratify(data = lalonde, treat = "treated", outcome = "outcome",
##     prog_scores = myprogscore, size = 100)
##
## Prognostic Scores prespecified.
##
## Strata Sizes:
## # A tibble: 7 x 4
##   Stratum Treat Control Total
##     <int> <int>   <dbl> <int>
## 1       1    24      64    88
## 2       2    27      61    88
## 3       3    30      58    88
## 4       4    25      62    87
## 5       5    27      61    88
## 6       6    22      66    88
## 7       7    30      57    87
```

```
print(a.strat2)
```

```
## auto_strata object from package big_match.
##
## Function call:
## auto_stratify(data = lalonde, treat = "treated", outcome = "outcome",
##     covariates = c("age", "educ", "hispan", "nodegree", "black"),
```

```
##     size = 100)
##
## Prognostic Score Model:
##
## Call:  glm(formula = formula(formula_str), family = "binomial", data = data0)
##
## Coefficients:
## (Intercept)          age         educ       hispan     nodegree
##    -5.00887      0.04639      0.19183      0.09823      0.14705
##       black
##    -0.53706
##
## Degrees of Freedom: 428 Total (i.e. Null);  423 Residual
## Null Deviance:       388.2
## Residual Deviance: 361.1     AIC: 373.1
##
## Strata Sizes:
## # A tibble: 7 x 4
##   Stratum Treat Control Total
##     <int> <int>   <dbl> <int>
## 1       1    40      48    88
## 2       2    37      51    88
## 3       3    39      49    88
## 4       4    26      61    87
## 5       5    22      66    88
## 6       6    16      72    88
## 7       7     5      82    87
```

Summary

```
# TODO: implement summary methods
summary(m.strat)
```

```
##              Length Class      Mode
## data         11     tbl_df     list
## treat         1     -none-     character
## strata_table  5     grouped_df list
## call          4     -none-     call
## n_table       4     tbl_df     list
```

```
summary(a.strat1)
```

```
##             Length Class      Mode
## data         11    data.frame list
## prog_scores 614    -none-     numeric
## prog_model    0    -none-     NULL
## treat         1    -none-     character
## outcome       1    -none-     character
## covariates    0    -none-     NULL
## call          6    -none-     call
## n_table       4    tbl_df     list
```

```
summary(a.strat2)
```

```
##             Length Class      Mode
## data         11    data.frame list
## prog_scores 614    -none-     numeric
## prog_model   30    glm        list
## treat         1    -none-     character
## outcome       1    -none-     character
## covariates    5    -none-     character
## call          6    -none-     call
## n_table       4    tbl_df     list
```

Plot

There are three types of plots: `"scatter"`, `"residual"`, and `"hist"`. Any other plot options for a strata object will throw an error.

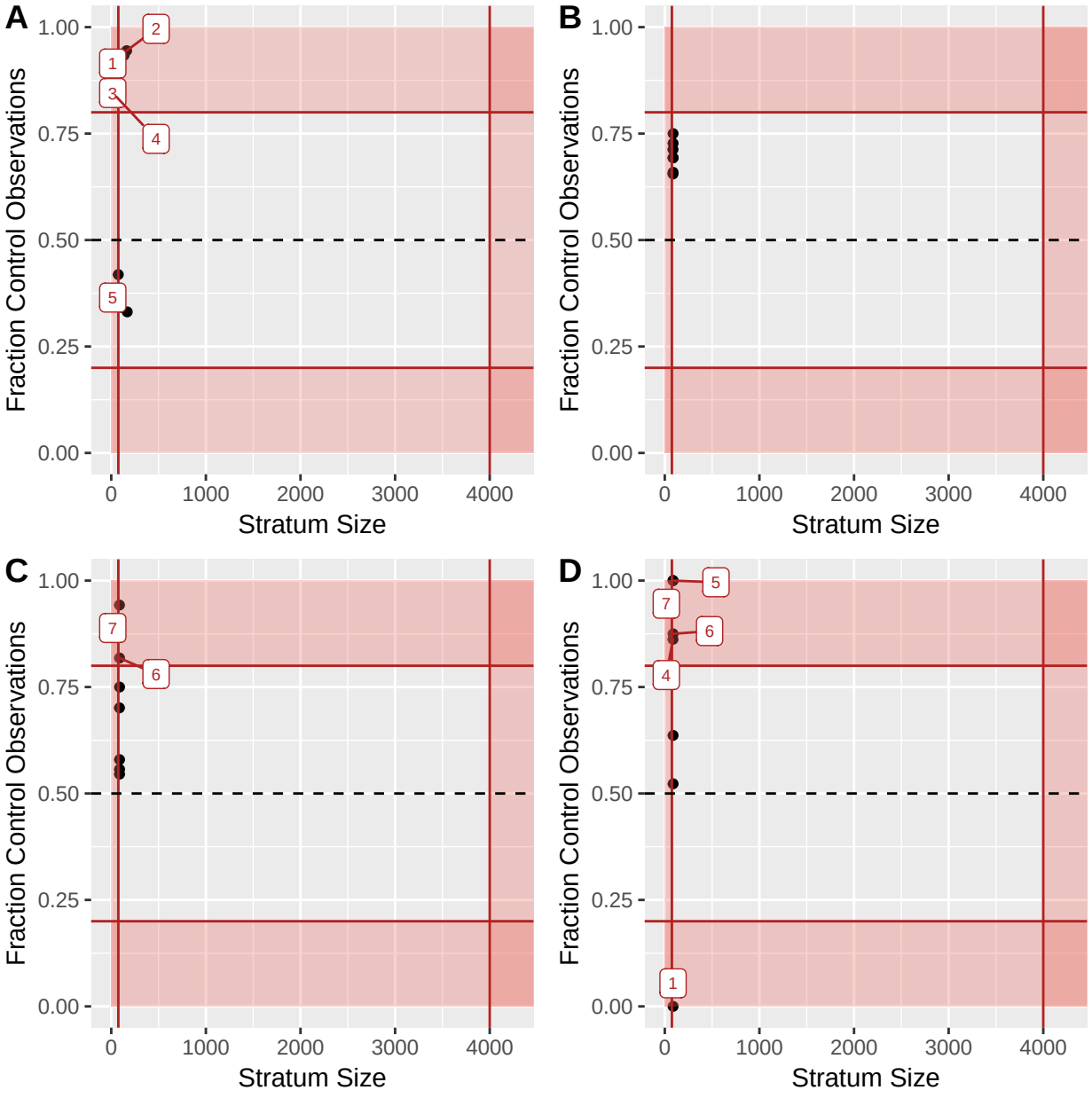
```
plot(m.strat, type = "QQ")
# Error in plot.strata(m.strat, type = "QQ") :
#   Not a recognized plot type.
```

Auto and Manual Stratify : Size-Balance Scatterplot

Below are the basic size × control fraction scatterplots for (A) manual stratification by `black`, `hispan` and `nodegree`, (B) auto-stratification with a uniform random prognostic score, (C) auto-stratification with a prognostic score that is relatively continuous, and (D) auto-stratification with a prognostic score that is discontinuous (built solely from discrete variables with few distinct values). As you can see, this sample data contains a relatively small number of examples to begin with, so most strata are quite small.

```
a <- plot(m.strat) # equivalently: plot(m.strat, type = "scatter")
b <- plot(a.strat1)
c <- plot(a.strat2)
d <- plot(a.strat3)

ggarrange(a, b, c, d, ncol = 2, nrow = 2,
          labels = "AUTO")
```



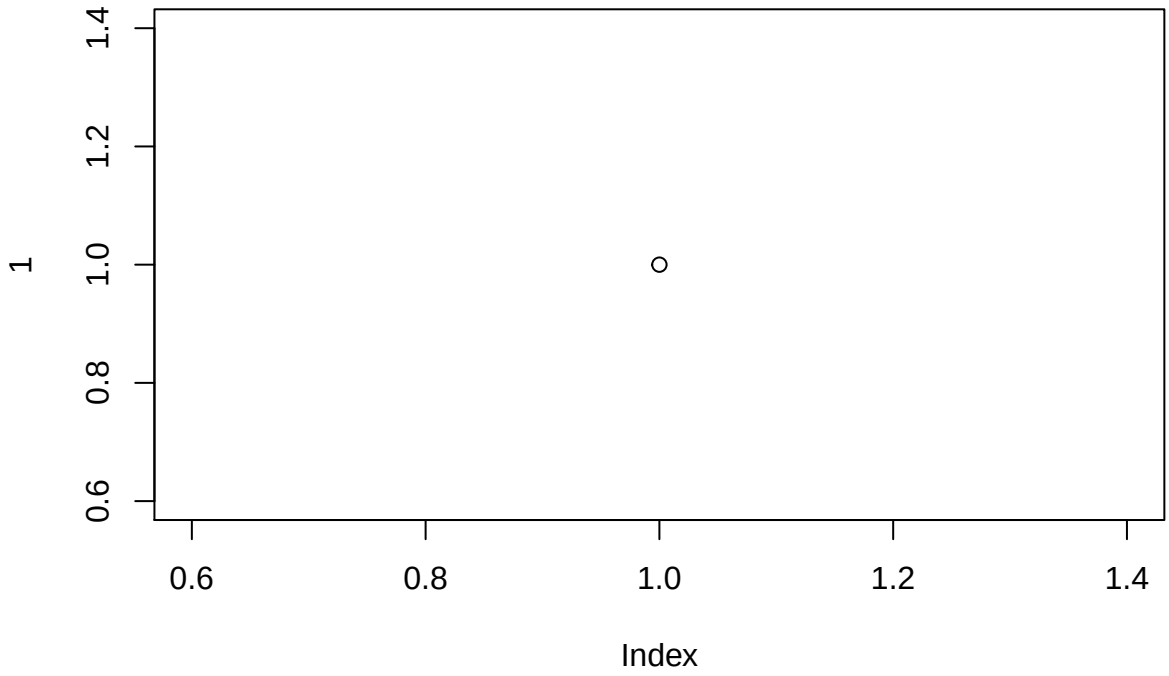
Auto Stratify: Prognostic Score Residual Plot

This function is only meant for auto-stratified data for which prognostic scores were not prespecified. Running it on a `manual_strata` object or an `auto_strata` object where prognostic scores were prespecified will throw an error.

```
plot(m.strat, type = "residual")
# Error in plot.strata(m.strat, type = "residual") :
#   Prognostic score residual plots are only valid for auto-stratified data.

plot(a.strat1, type = "residual")
# Error in plot.strata(a.strat1, type = "residual"):
#  Cannot make prognostic score residual plots. Prognostic model is unknown.
```

```
# TODO: implement this plot
plot(a.strat2, type = "residual")
```



Auto Stratify: Prognostic Score Histograms

This function is only meant for auto-stratified data. Running it on a `manual_strata` object will throw an error.

```
plot(m.strat, type = "hist")

# Error in plot.strata(m.strat, type = "hist"):
#  Prognostic score histograms are only valid for auto-stratified data.
```

```
# uniformly generated prognostic score. Nicely continuous from 0 to 1
a <- plot(a.strat1, type = "hist")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
# prognostic score generated from some continuous and some distcrete variables.
# Fairly continuous
b <- plot(a.strat2, type = "hist")
```

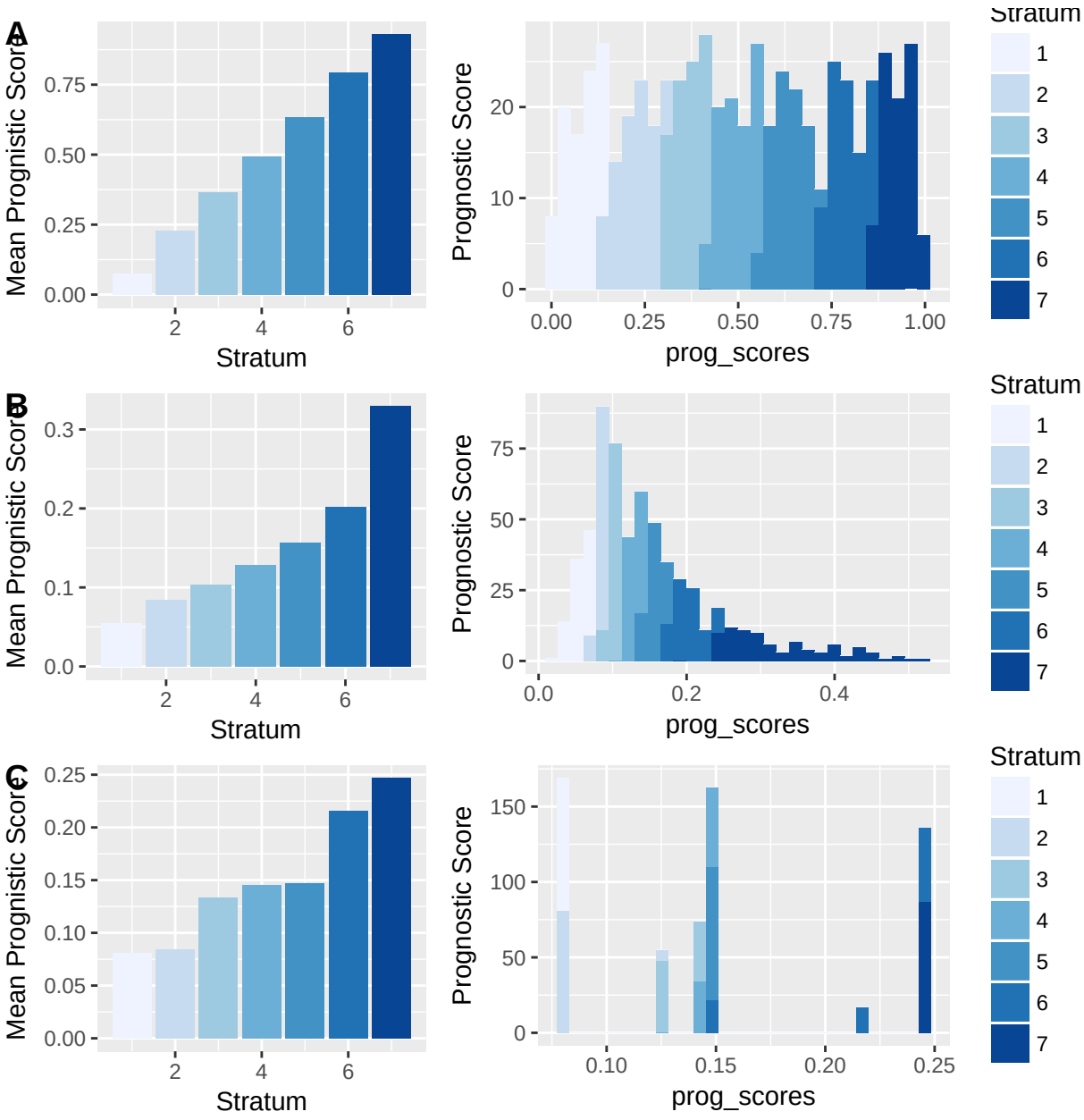
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
# prognostic score generated from only a few discrete variables.
# Since prog_score only takes on a few different values,
# strata quantiles are less evenly distributed from 0 to 1
```

```
c <- plot(a.strat3, type = "hist")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
ggarrange(a, b, c, ncol = 1, nrow = 3, labels = "AUTO")
```



Matching