Big_match - Testing and Small 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.

```
require(MatchIt)
require(ggplot2)
require(ggpubr)
require(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"
```

Strata Objects

The functions `auto_stratify` and `manual_stratify` return `auto_strata` and `manual_strata` S3 objects, respectively, which both inherit from class `strata`. The `plot`, `summary` and `print` methods for `strata` objects implement most of the diagnsostics we have for assessing the outcome of a stratification. The contents of `strata` objects is shown below:

Info for all strata objects:

- `data` **data.frame** with strata specified
- `treat` **string** of the name of the treatment assignment column in the data frame
- `covariates` **character vector** giving names of all covariates for stratification (may be NULL if prognostic scores were pre-specified)
- `call` **call object** of function call which produced object
- `issue_table` **data.frame** of Treated, Controls, and Total observations for each stratum

Additional info for manual_strata objects:

- `strata_table` **data.frame** of strata definitions

Additional info for auto_strata objects:

- `prog_scores` **numeric vector** of prognostic scores for each observation
- `prog_model` **glm object** prognostic score model (may be NULL if prognostic scores were pre-specified)
- `outcome` **string** with the name of the outcome variable
- `discarded` **???** rows deleted

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 = "treated",
                           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 = "treated",
                           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"))
```

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, "treated", "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, "treated", "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)
```

```
## [1] 571  10
## [1] "Fitting prognostic model: outcome~age+educ+hispan+nodegree+black"
```

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

```
## [1] 571  10
## [1] "Fitting prognostic model: outcome~hispan+nodegree+black"
```

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 6
##   Stratum Treat Control Total Control_Proporti~ Potential_Issues
##     <dbl> <int>   <dbl> <int>             <dbl> <chr>
## 1       1     9     127   136             0.934 Not enough treated samples
## 2       2     9     154   163             0.945 Not enough treated samples
## 3       3     2      15    17             0.882 Too few samples; Not enou~
## 4       4     9      46    55             0.836 Too few samples; Not enou~
## 5       5    43      31    74             0.419 Too few samples
## 6       6   113      56   169             0.331 none
```

```
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 6
##   Stratum         Treat Control Total Control_Proportion Potential_Issues
##   <fct>           <int>   <dbl> <int>              <dbl> <chr>
## 1 [0.00127,0.134]    20      68    88              0.773 none
## 2 [0.13372,0.273]    23      65    88              0.739 none
## 3 [0.27308,0.424]    27      61    88              0.693 none
## 4 [0.42406,0.535]    33      54    87              0.621 none
## 5 [0.53490,0.687]    28      60    88              0.682 none
## 6 [0.68720,0.838]    28      60    88              0.682 none
## 7 [0.83818,1.000]    26      61    87              0.701 none
```

```
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 = model_set)
##
## Coefficients:
## (Intercept)          age         educ       hispan     nodegree
##     -7.6333       0.0472       0.3021       0.5000       2.4107
##       black
##      0.2329
##
## Degrees of Freedom: 42 Total (i.e. Null);  37 Residual
## Null Deviance:       44.12
## Residual Deviance: 38.46     AIC: 50.46
##
## Strata Sizes:
## # A tibble: 6 x 6
##   Stratum         Treat Control Total Control_Proportion Potential_Issues
##   <fct>           <int>   <dbl> <int>              <dbl> <chr>
## 1 [0.0317,0.0696)    24      76   100              0.76  none
## 2 [0.0696,0.1178)    27      64    91              0.703 none
## 3 [0.1178,0.1876)    21      74    95              0.779 none
## 4 [0.1876,0.2514)    30      65    95              0.684 none
## 5 [0.2514,0.3274)    34      61    95              0.642 none
## 6 [0.3274,0.6899]    49      46    95              0.484 none
```

Summary

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

```
## $call
## manual_stratify(data = lalonde, treat = "treated", covariates = c("black",
##     "hispan", "nodegree"))
##
## $issue_table
## # A tibble: 6 x 6
##   Stratum Treat Control Total Control_Proporti~ Potential_Issues
##     <dbl> <int>   <dbl> <int>             <dbl> <chr>
## 1       1     9     127   136             0.934 Not enough treated samples
## 2       2     9     154   163             0.945 Not enough treated samples
## 3       3     2      15    17             0.882 Too few samples; Not enou~
## 4       4     9      46    55             0.836 Too few samples; Not enou~
## 5       5    43      31    74             0.419 Too few samples
## 6       6   113      56   169             0.331 none
##
## $sum_before
##            Treat_Mean  Contol_Mean
## treat       0.0000000 1.000000e+00
## age        28.0303030 2.581622e+01
## educ       10.2354312 1.034595e+01
## black       0.2027972 8.432432e-01
## hispan      0.1421911 5.945946e-02
## married     0.5128205 1.891892e-01
## nodegree    0.5967366 7.081081e-01
## re74     5619.2365064 2.095574e+03
## re75     2466.4844431 1.532055e+03
## outcome     0.1678322 9.729730e-02
## stratum     2.6923077 5.200000e+00
##
## attr(,"class")
## [1] "summary.strata"
```

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

```
## Warning in mean.default(stratum): argument is not numeric or logical:
## returning NA

## Warning in mean.default(stratum): argument is not numeric or logical:
## returning NA
```

```
## $call
## auto_stratify(data = lalonde, treat = "treated", outcome = "outcome",
##     prog_scores = myprogscore, size = 100)
##
## $issue_table
## # A tibble: 7 x 6
##   Stratum         Treat Control Total Control_Proportion Potential_Issues
##   <fct>           <int>   <dbl> <int>              <dbl> <chr>
## 1 [0.00127,0.134)    20      68    88              0.773 none
## 2 [0.13372,0.273)    23      65    88              0.739 none
```

```
## 3 [0.27308,0.424)    27      61    88              0.693 none
## 4 [0.42406,0.535)    33      54    87              0.621 none
## 5 [0.53490,0.687)    28      60    88              0.682 none
## 6 [0.68720,0.838)    28      60    88              0.682 none
## 7 [0.83818,1.000]    26      61    87              0.701 none
##
## $sum_before
##            Treat_Mean  Contol_Mean
## treat       0.0000000 1.000000e+00
## age        28.0303030 2.581622e+01
## educ       10.2354312 1.034595e+01
## black       0.2027972 8.432432e-01
## hispan      0.1421911 5.945946e-02
## married     0.5128205 1.891892e-01
## nodegree    0.5967366 7.081081e-01
## re74     5619.2365064 2.095574e+03
## re75     2466.4844431 1.532055e+03
## outcome     0.1678322 9.729730e-02
## stratum            NA           NA
##
## attr(,"class")
## [1] "summary.strata"
```

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

```
## Warning in mean.default(stratum): argument is not numeric or logical:
## returning NA

## Warning in mean.default(stratum): argument is not numeric or logical:
## returning NA
```

```
## $call
## auto_stratify(data = lalonde, treat = "treated", outcome = "outcome",
##     covariates = c("age", "educ", "hispan", "nodegree", "black"),
##     size = 100)
##
## $issue_table
## # A tibble: 6 x 6
##   Stratum         Treat Control Total Control_Proportion Potential_Issues
##   <fct>           <int>   <dbl> <int>              <dbl> <chr>
## 1 [0.0317,0.0696)    24      76   100              0.76  none
## 2 [0.0696,0.1178)    27      64    91              0.703 none
## 3 [0.1178,0.1876)    21      74    95              0.779 none
## 4 [0.1876,0.2514)    30      65    95              0.684 none
## 5 [0.2514,0.3274)    34      61    95              0.642 none
## 6 [0.3274,0.6899]    49      46    95              0.484 none
##
## $sum_before
##            Treat_Mean  Contol_Mean
## treat       0.0000000 1.000000e+00
## age        27.9119171 2.581622e+01
## educ       10.2927461 1.034595e+01
## black       0.1943005 8.432432e-01
## hispan      0.1450777 5.945946e-02
## married     0.5233161 1.891892e-01
```

```
## nodegree    0.5880829 7.081081e-01
## re74     5731.1294532 2.095574e+03
## re75     2498.9146305 1.532055e+03
## outcome     0.1632124 9.729730e-02
## stratum            NA           NA
##
## attr(,"class")
## [1] "summary.strata"
```

Plot

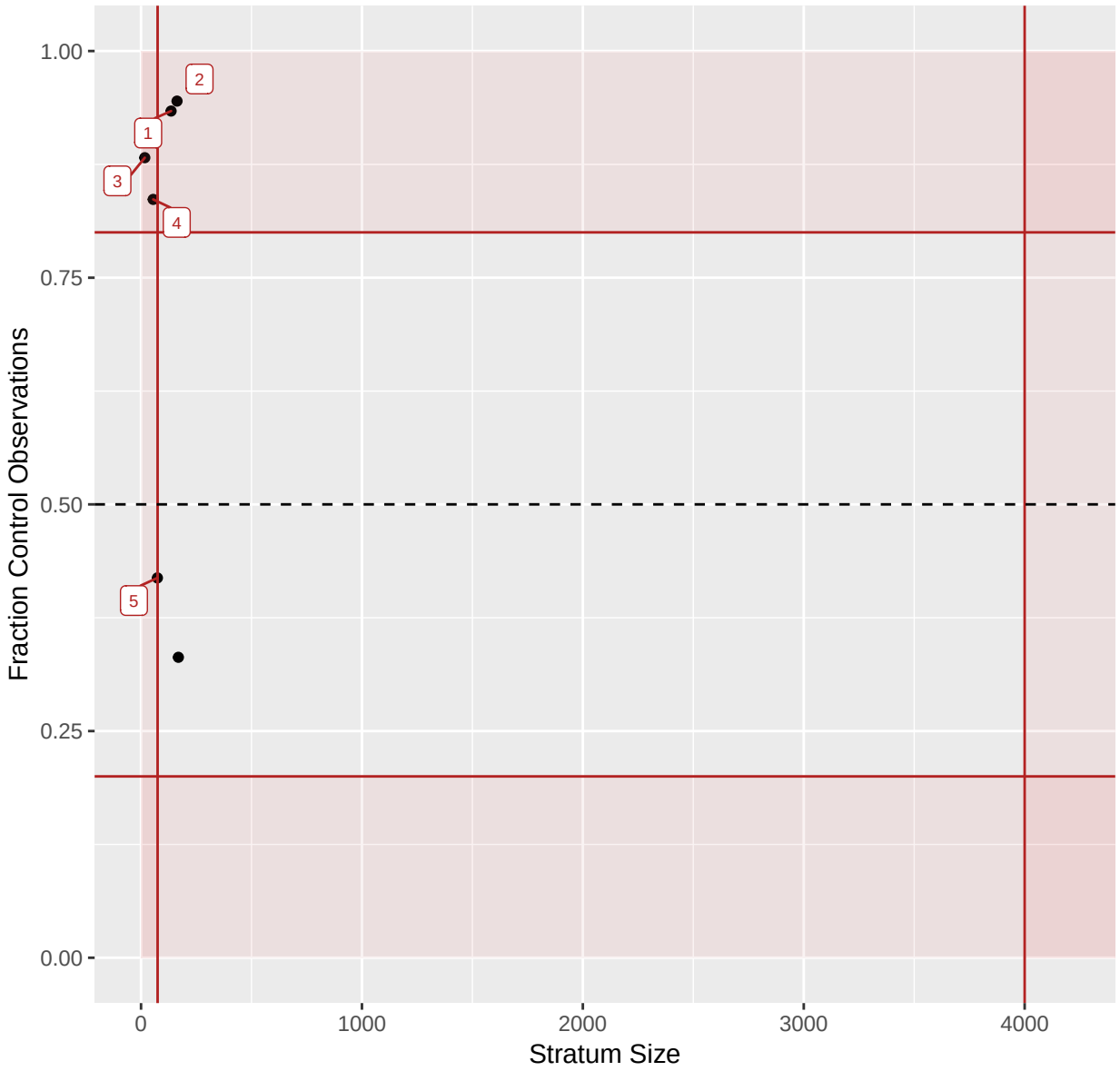
There are three types of plots: `"scatter"`, `"residual"`, and `"hist"`. Any other plot options for a strata object will throw an error.

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

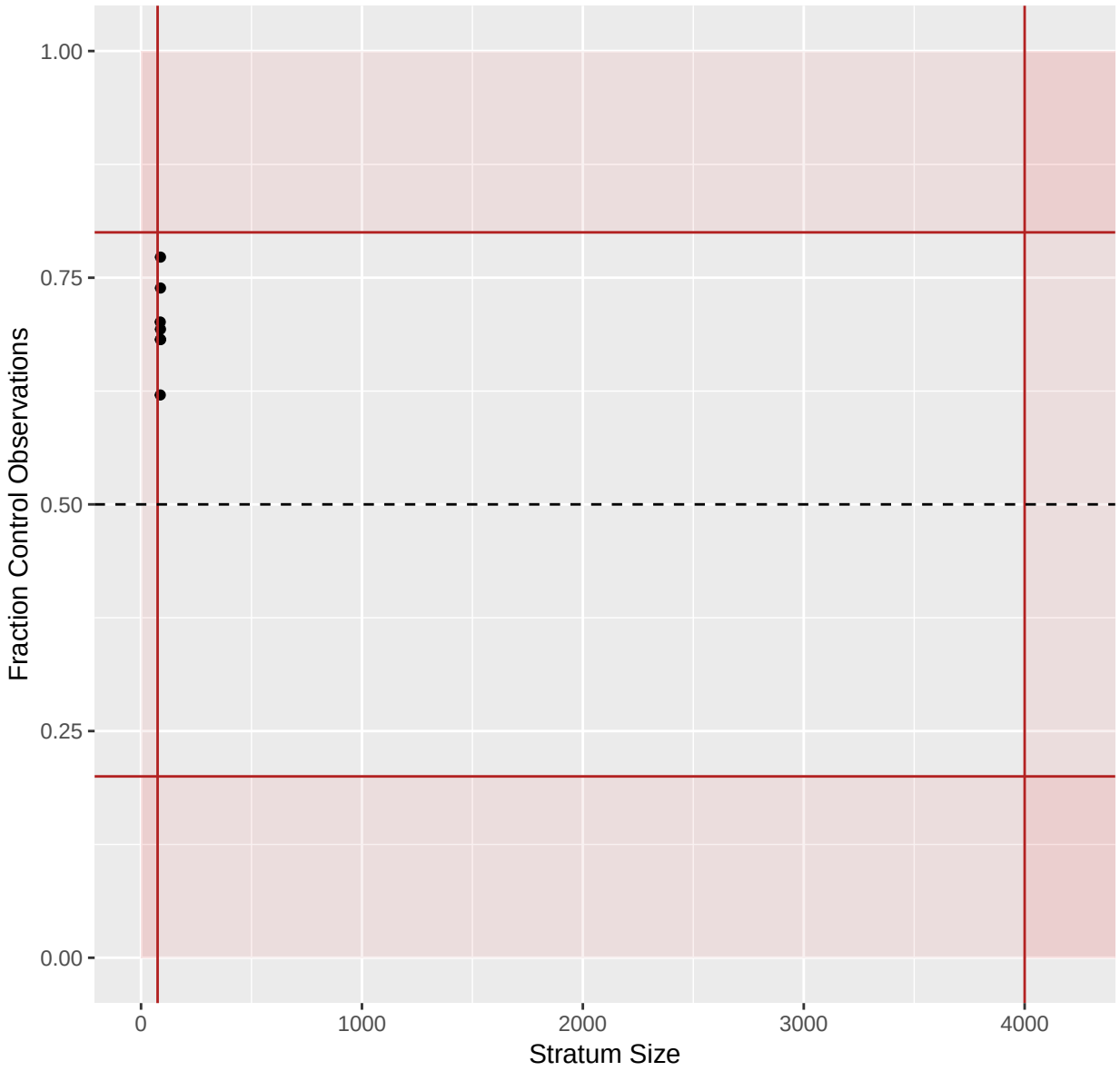
Auto and Manual Stratify : Size-Balance Scatterplot

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.

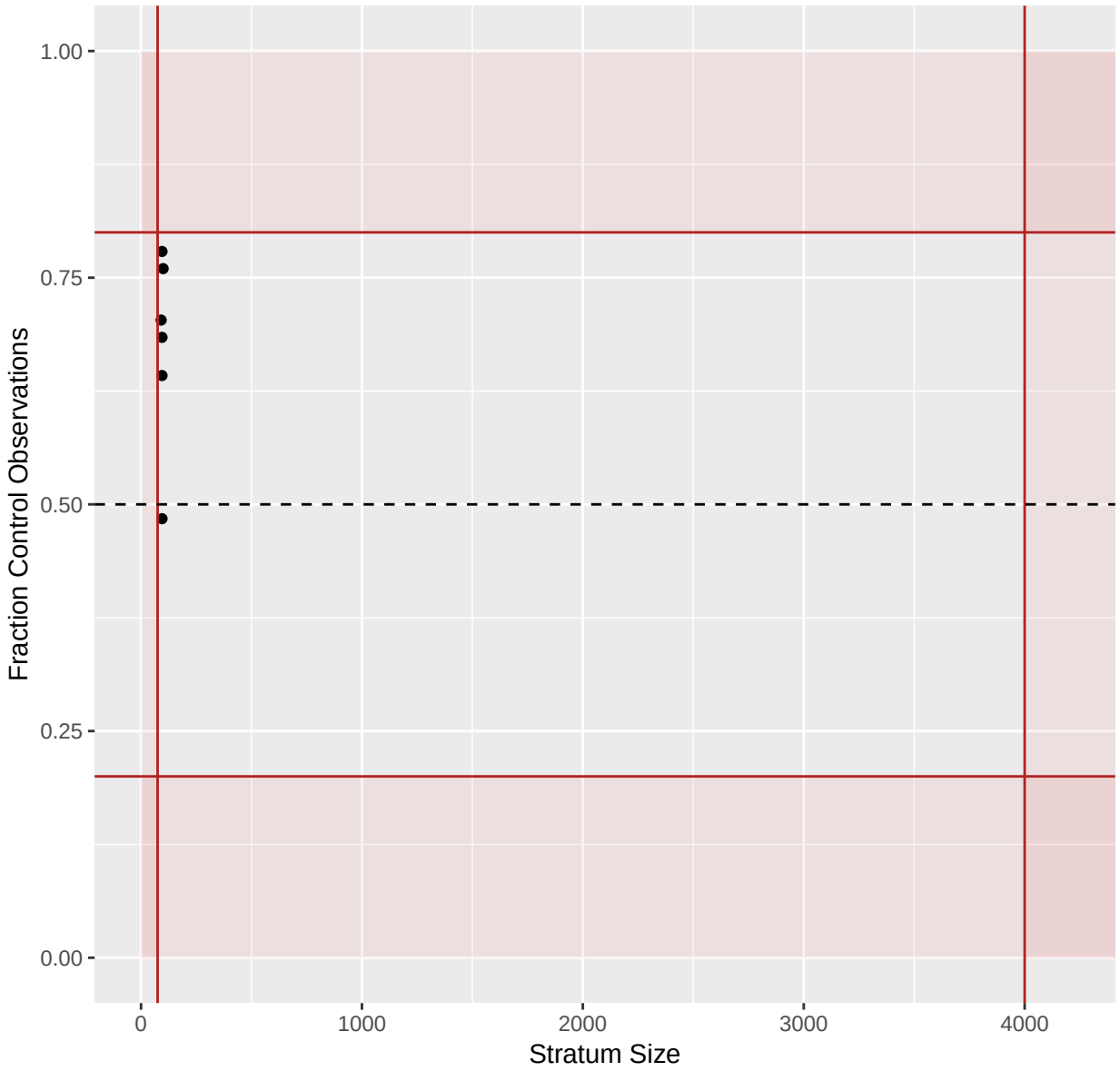
```
a <- plot(m.strat, label = TRUE) # equivalently: plot(m.strat, type = "scatter")
```



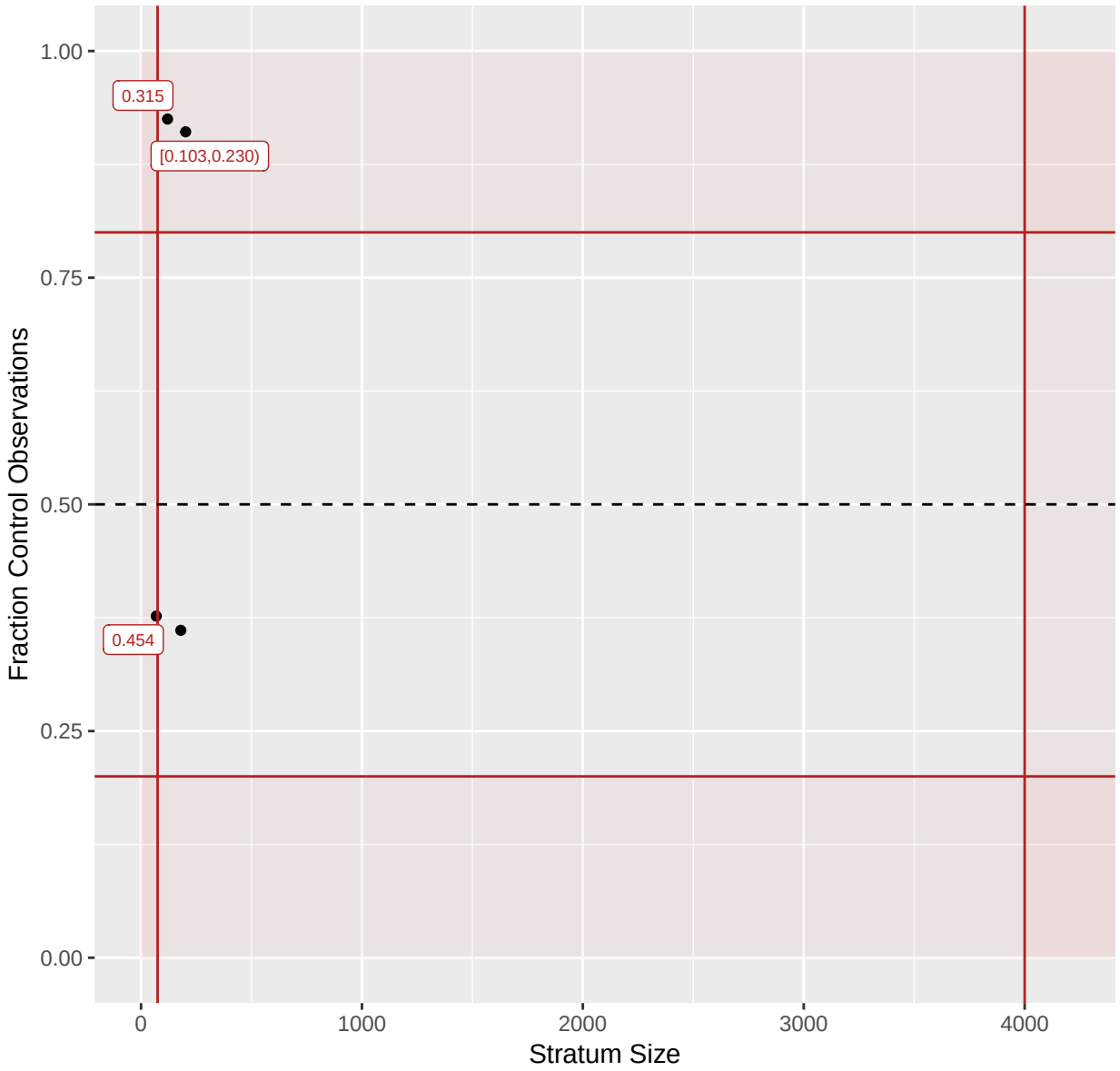

```
b <- plot(a.strat1, label = TRUE)
```



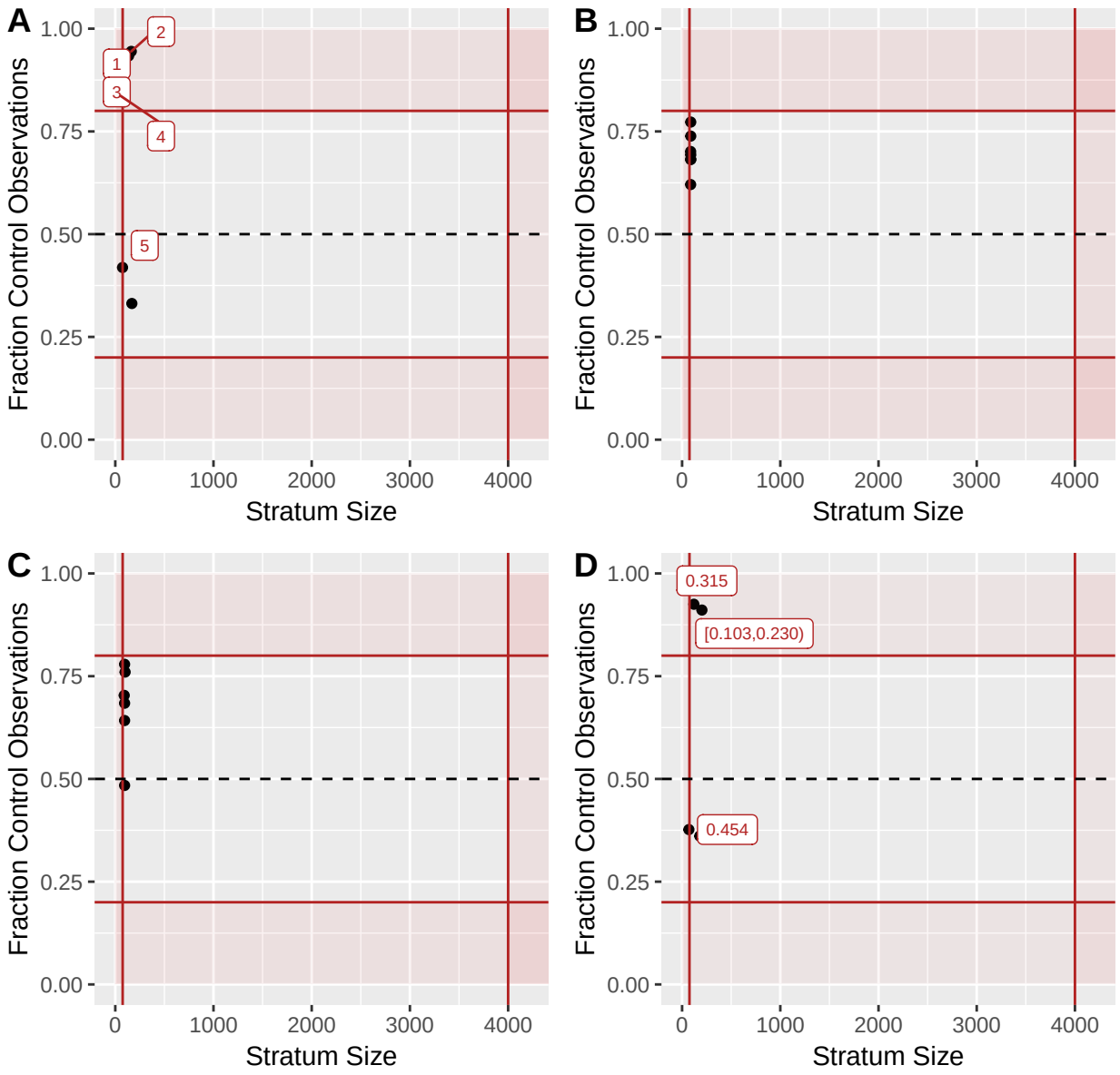

```
c <- plot(a.strat2, label = TRUE)
```




```
d <- plot(a.strat3, label = TRUE)
```




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



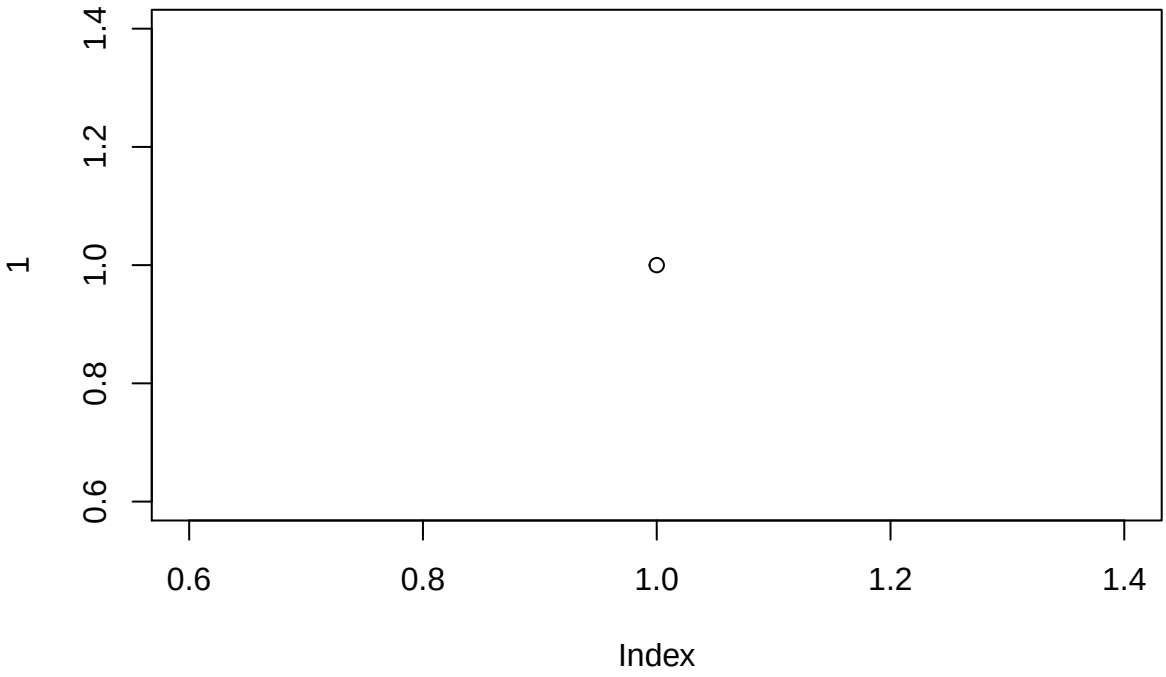
Auto Stratify: Prognostic Score Residual Plot

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`.
```

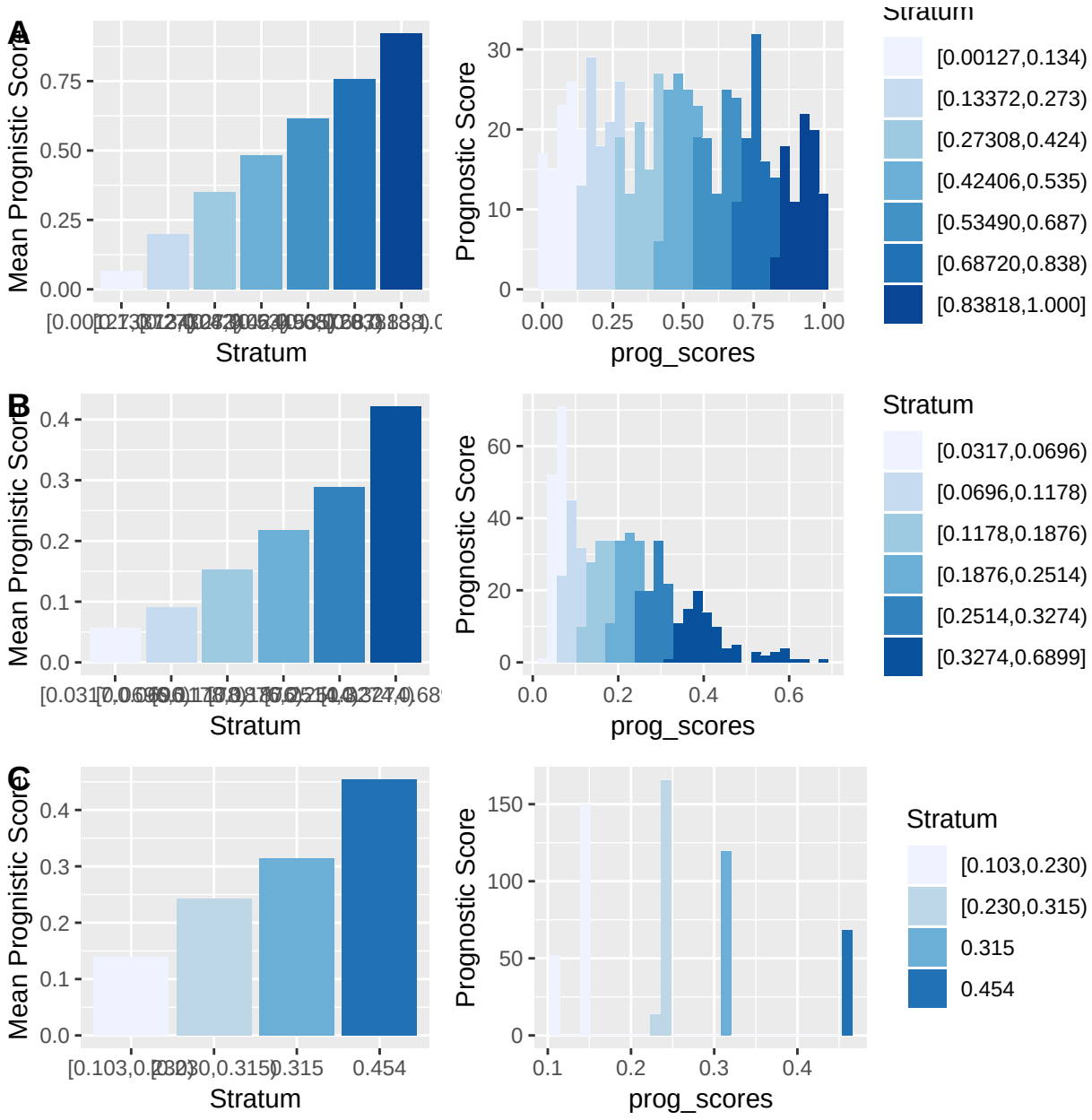
```
# prognostic score generated from some continuous and some distcrete variables.
# Fairly continuous
b <- plot(a.strat2, type = "hist")
```

```
## `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

This is not implemented yet.