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)
require(optmatch)


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 = "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", outcome ~ 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", outcome ~ age + :
#   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)
```

```
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
```

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

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: outcome ~ age + educ + hispan + nodegree + black"
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
```

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

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: outcome ~ hispan + nodegree + black"
## [1] "Generating strata assignments based on prognostic score."
```

```
## Warning in min(xx[xx > upper]): no non-missing arguments to min; returning
## Inf
```

```
## [1] "Completing strata diagnostics."
```

Diagnostics

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

Print

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

```
## manual_strata object from package big_match.
##
## Function call:
## manual_stratify(data = lalonde, treat = "treated", covariates = c("black",
##     "hispan", "nodegree"))
##
## Analysis set dimensions: 614 X 11
##
## Number of strata: 6
##
##  Min size: 17    Max size: 169
##
## Strata 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
```

```r
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)
##
## Analysis set dimensions: 614 X 11
##
## Prognostic Scores prespecified.
##
## Number of strata: 7
##
##  Min size: 87    Max size: 88
##
## Strata issue table:
## # A tibble: 7 x 6
##   Stratum          Treat Control Total Control_Proportion Potential_Issues
##   <fct>            <int>   <dbl> <int>              <dbl> <chr>
## 1 [0.00109,0.175)    24      64    88              0.727 none
## 2 [0.17499,0.309)    28      60    88              0.682 none
## 3 [0.30861,0.425)    30      58    88               0.659 none
## 4 [0.42527,0.557)    23      64    87               0.736 none
## 5 [0.55703,0.686)    22      66    88               0.75  none
## 6 [0.68649,0.859)    28      60    88               0.682 none
## 7 [0.85903,1.000]    30      57    87               0.655 none
```

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

```
## auto_strata object from package big_match.
##
## Function call:
## auto_stratify(data = lalonde, treat = "treated", outcome = "outcome",
##     prog_formula = outcome ~ age + educ + hispan + nodegree +
##         black, size = 100)
##
## Analysis set dimensions: 571 X 11
##
## Model set dimensions: 43 X 10
##
## Prognostic Score Model:
##
## Call:  glm(formula = prog_formula, family = "binomial", data = model_set)
##
## Coefficients:
## (Intercept)          age         educ       hispan     nodegree
##   -14.55219      0.02945      0.91509    -14.70832      2.83839
##       black
##     0.55390
##
## Degrees of Freedom: 42 Total (i.e. Null);  37 Residual
## Null Deviance:       34.75
## Residual Deviance: 27.61     AIC: 39.61
##
## Number of strata: 6
##
##  Min size: 93    Max size: 96
##
## Strata issue table:
## # A tibble: 6 x 6
##   Stratum       Treat Control Total Control_Proporti~ Potential_Issues
##   <fct>         <int>   <dbl> <int>             <dbl> <chr>
## 1 [1.15e-11,0.~    18      78    96             0.812 Not enough treated ~
## 2 [5.83e-03,0.~    17      79    96             0.823 Not enough treated ~
## 3 [4.97e-02,0.~    26      70    96             0.729 none
## 4 [8.64e-02,0.~    36      57    93             0.613 none
## 5 [1.38e-01,0.~    39      56    95             0.589 none
## 6 [2.87e-01,0.~    49      46    95             0.484 none
```

Summary

```r
# 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"
```

```r
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 [0.00109,0.175)     24      64    88              0.727 none
## 2 [0.17499,0.309)     28      60    88              0.682 none
## 3 [0.30861,0.425)    30      58    88              0.659 none
## 4 [0.42527,0.557)    23      64    87              0.736 none
## 5 [0.55703,0.686)    22      66    88              0.75  none
## 6 [0.68649,0.859)    28      60    88              0.682 none
## 7 [0.85903,1.000]    30      57    87              0.655 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",
##     prog_formula = outcome ~ age + educ + hispan + nodegree +
##         black, size = 100)
##
## $issue_table
## # A tibble: 6 x 6
##   Stratum       Treat Control Total Control_Proporti~ Potential_Issues
##   <fct>         <int>   <dbl> <int>             <dbl> <chr>
## 1 [1.15e-11,0.~    18      78    96             0.812 Not enough treated ~
## 2 [5.83e-03,0.~    17      79    96             0.823 Not enough treated ~
## 3 [4.97e-02,0.~    26      70    96             0.729 none
## 4 [8.64e-02,0.~    36      57    93             0.613 none
## 5 [1.38e-01,0.~    39      56    95             0.589 none
## 6 [2.87e-01,0.~    49      46    95             0.484 none
##
## $sum_before
##            Treat_Mean  Contol_Mean
## treat       0.0000000 1.000000e+00
## age        28.0207254 2.581622e+01
## educ       10.2150259 1.034595e+01
## black       0.1994819 8.432432e-01
## hispan      0.1502591 5.945946e-02
## married     0.4922280 1.891892e-01
## nodegree    0.5906736 7.081081e-01
## re74     5630.7449109 2.095574e+03
## re75     2451.1418009 1.532055e+03
## outcome     0.1709845 9.729730e-02
## stratum            NA           NA
##
## attr(,"class")
## [1] "summary.strata"
```

Plot

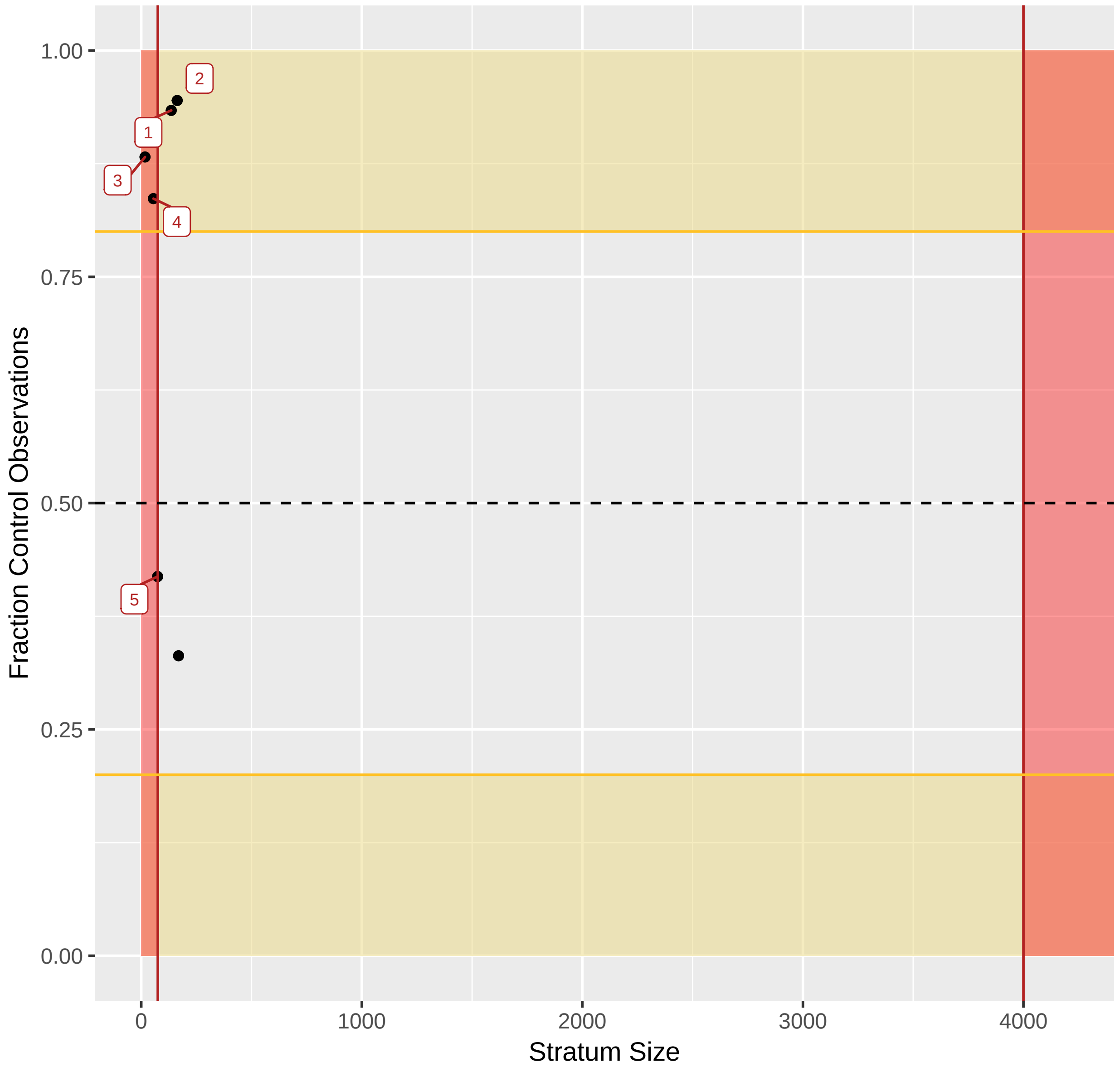
There are three types of plots: `"scatter"`, `"residual"`, and `"hist"`. Any other plot options for a strata object will throw an error.

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

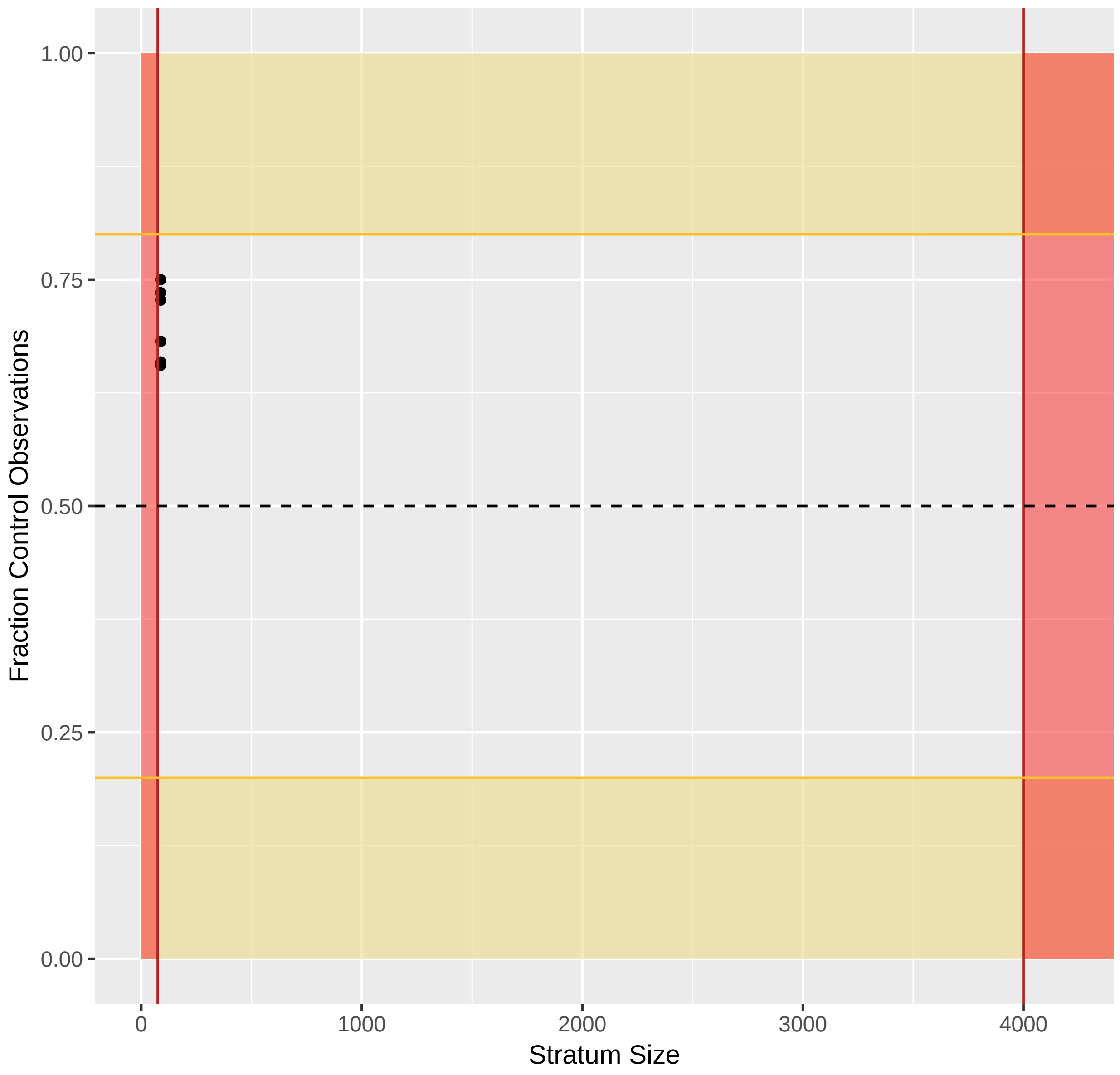
Auto and Manual Stratify : Size-Balance Scatterplot

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.

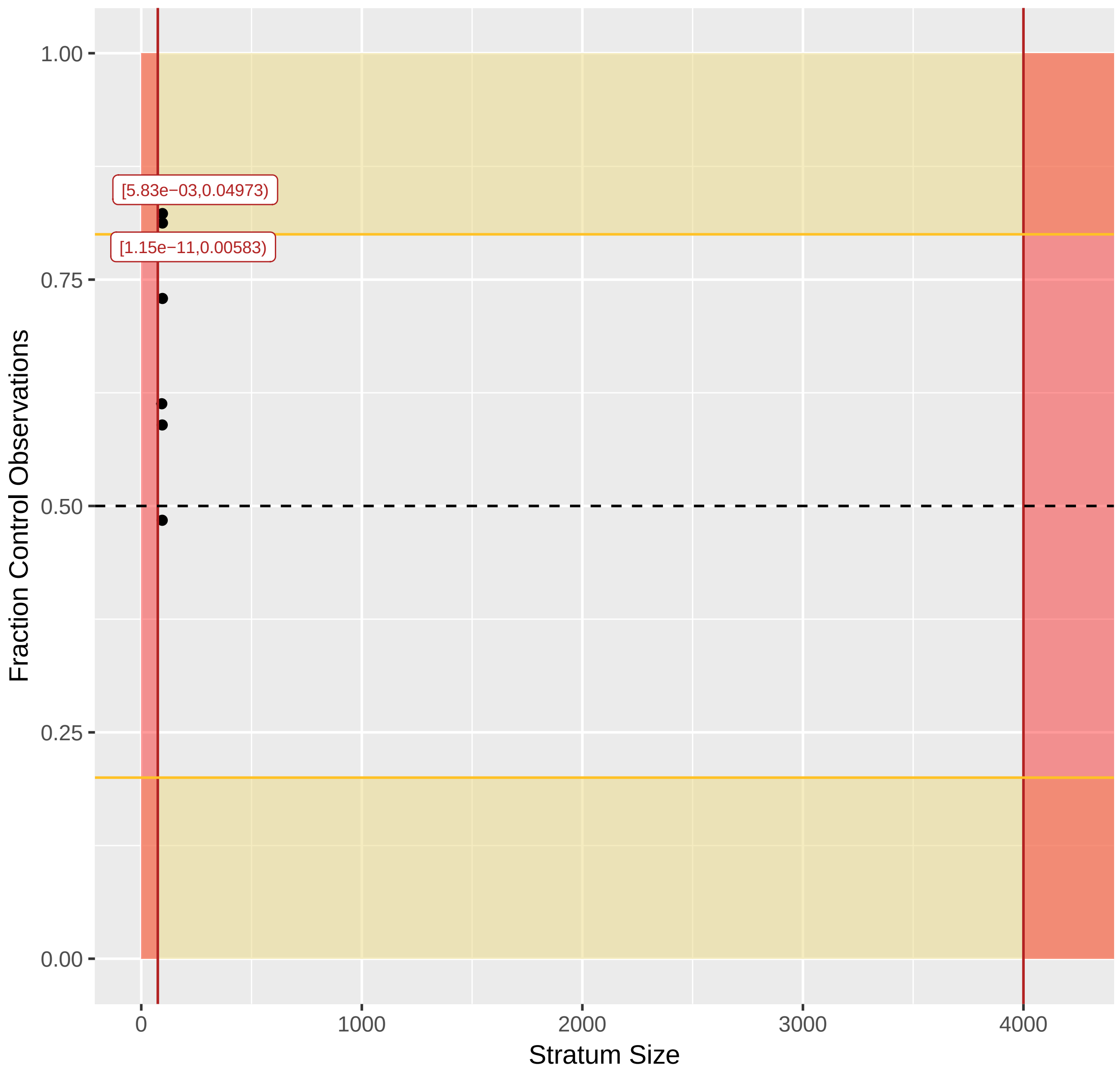
```r
a <- plot(m.strat, label = TRUE) # equivalently: plot(m.strat, type = "scatter")
```



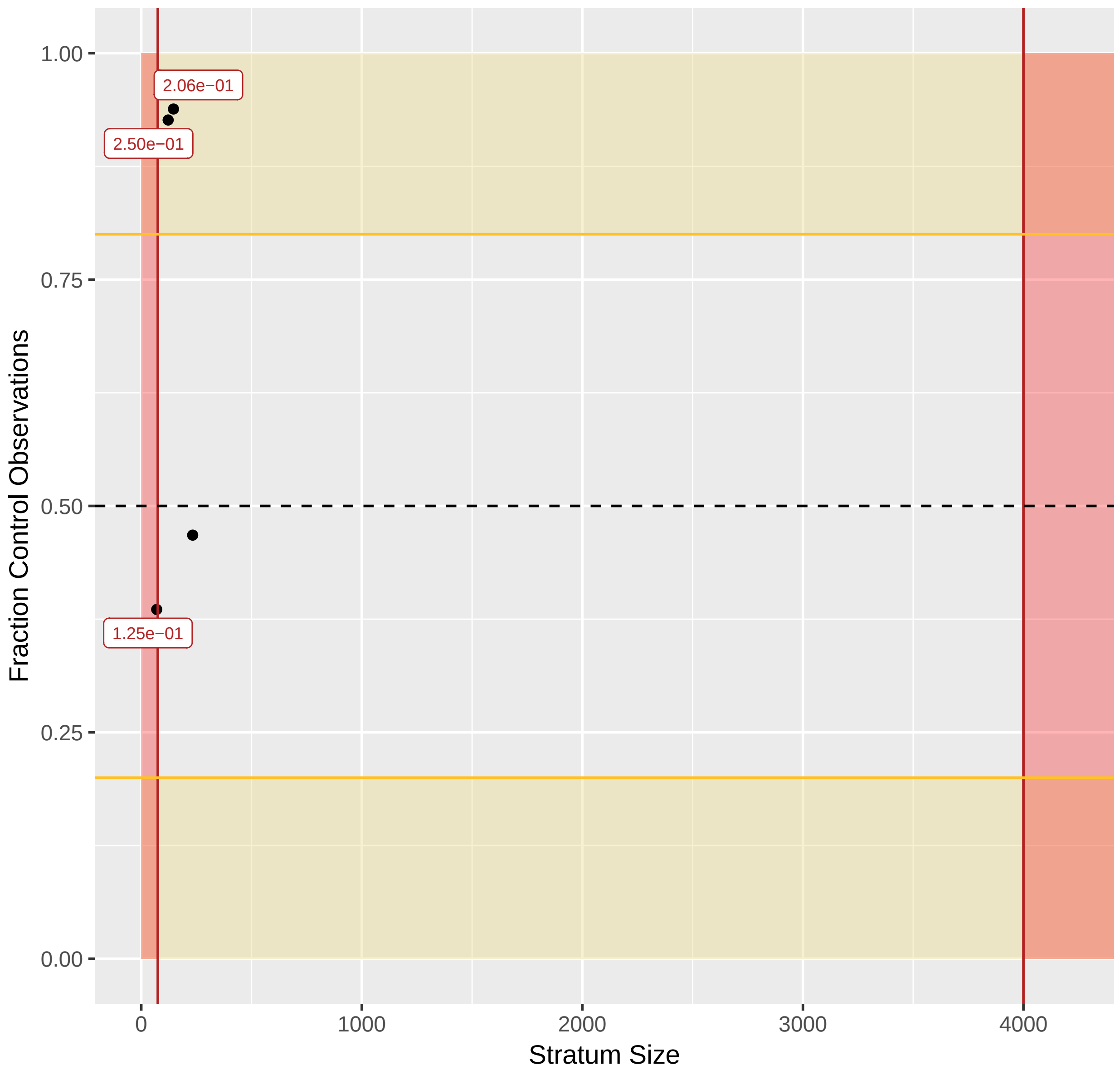
```r
b <- plot(a.strat1, label = TRUE)
```



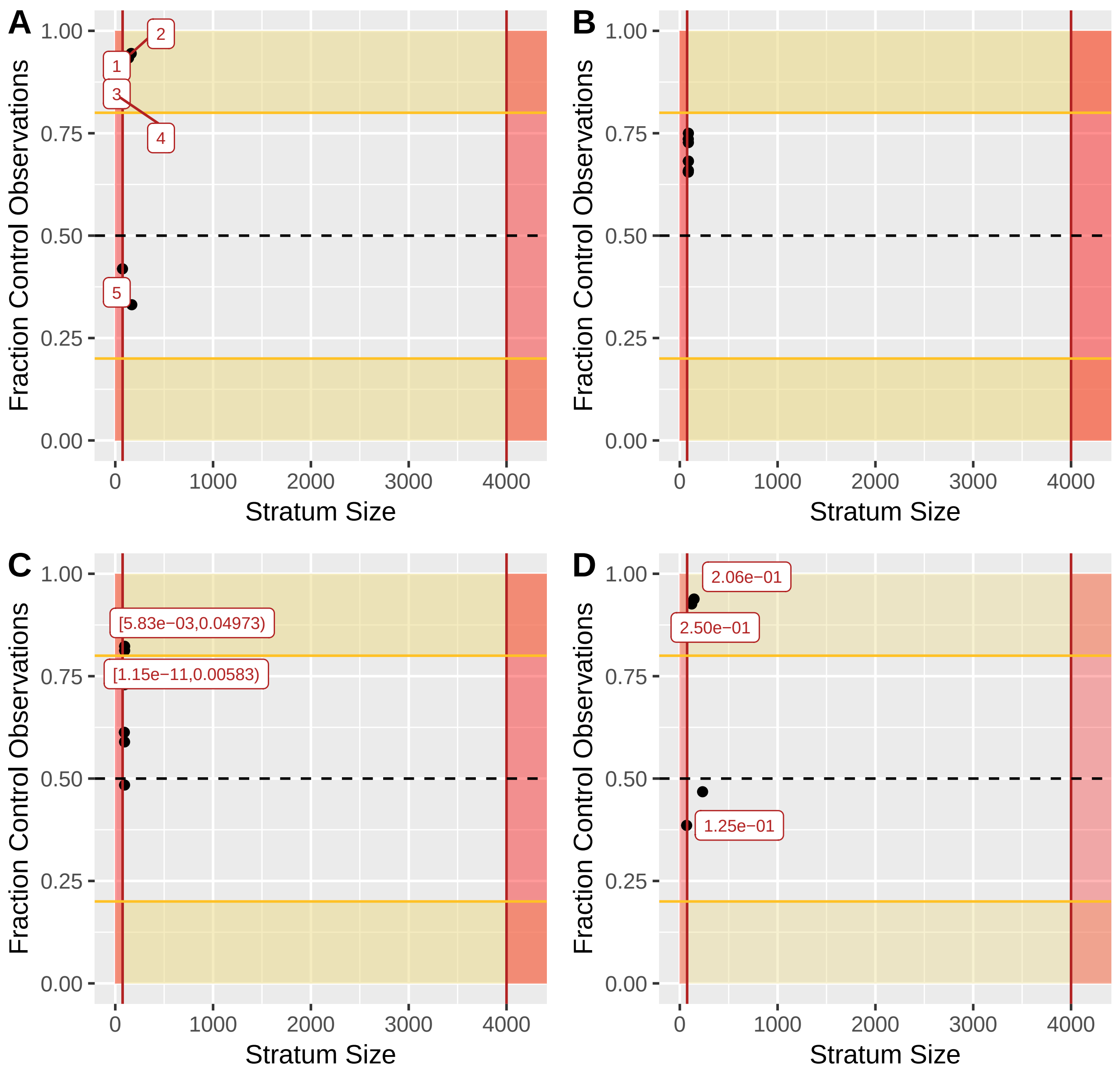
```r
c <- plot(a.strat2, label = TRUE)
```



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



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



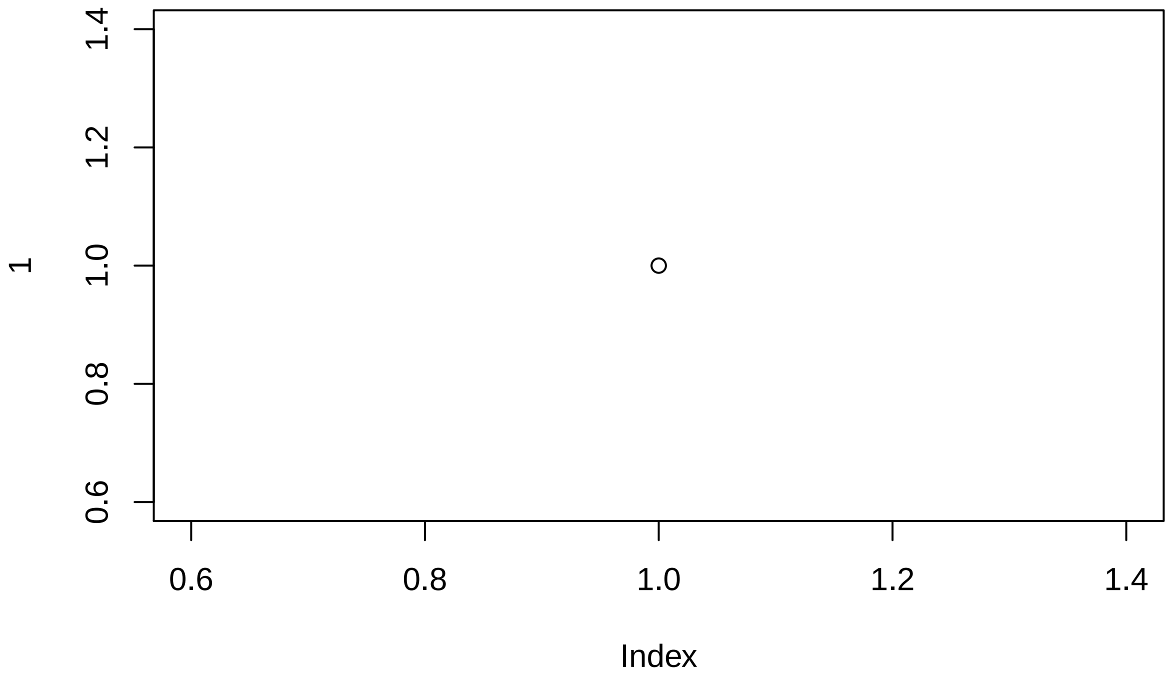
Auto Stratify: Prognostic Score Residual Plot

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.

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

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

```r
# 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`.
```

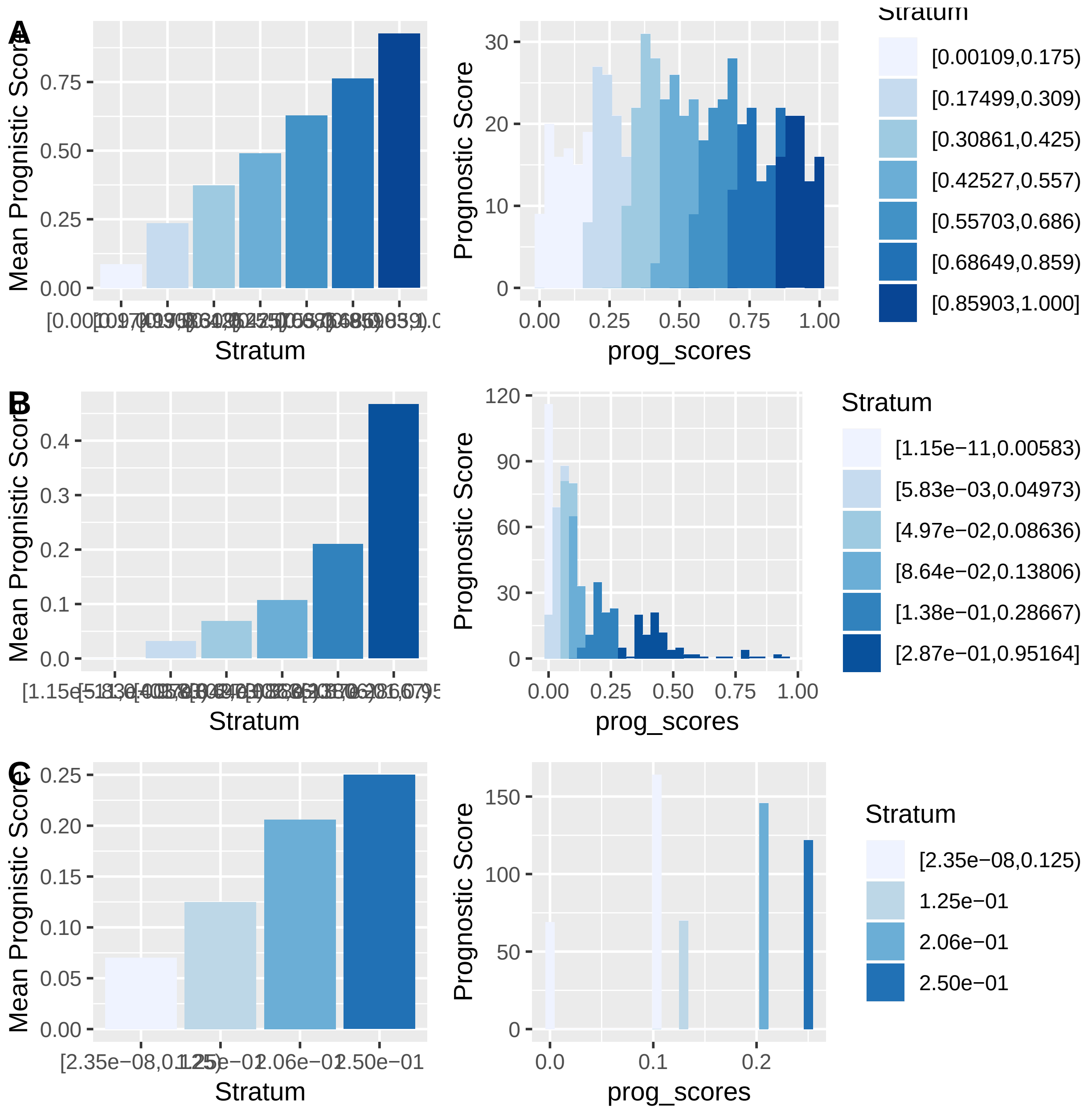
```r
# prognostic score generated from some continuous and some distcrete variables.
# Fairly continuous
b <- plot(a.strat2, type = "hist")
```

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

```r
# 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`.
```

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



Matching

Below, we run a default big match: the propensity score is built on the analysis set based on all covariates, stratified by stratum assignment. This implementation of big_match is essentially a wrapper for pairmatch from the optmatch package.

```r
summary(big_match(a.strat1))
```

```
## treated ~ . - outcome - stratum + strata(stratum)
## <environment: 0x7ffe4361b340>

## Structure of matched sets:
```

```
## 1:1 0:1
## 185 244
## Effective Sample Size:  185
## (equivalent number of matched pairs).
```

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

```
## treated ~ . - outcome - stratum + strata(stratum)
## <environment: 0x7ffe4637ae80>

## Structure of matched sets:
## 1:0 1:1 0:1
##   3 182 204
## Effective Sample Size:  182
## (equivalent number of matched pairs).
```

```
summary(big_match(a.strat3)) # throws a warning
```

```
## treated ~ . - outcome - stratum + strata(stratum)
## <environment: 0x7ffe42ef2820>

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Structure of matched sets:
## 1:0 1:1 0:1
##  31 154 232
## Effective Sample Size:  154
## (equivalent number of matched pairs).
```

```
summary(big_match(m.strat)) # throws a warning
```

```
## treated ~ . - -stratum + strata(stratum)
## <environment: 0x7ffe48b34820>

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Structure of matched sets:
## 1:0 1:1 0:1
##  69 116 313
## Effective Sample Size:  116
## (equivalent number of matched pairs).
```

One can also specify the propensity score formula:

```
summary(big_match(a.strat2, propensity_formula = treated ~ educ))
```

```
## treated ~ educ + strata(stratum)
## <environment: 0x7ffe44f6dab8>

## Structure of matched sets:
## 1:0 1:1 0:1
##   3 182 204
## Effective Sample Size:  182
## (equivalent number of matched pairs).
```