Package ‘sitepickR’

December 5, 2022

Type Package

Title Two-Level Sample Selection with Optimal Site Replacement

Version 0.0.1

Date 2022-11-29

Description Carries out a two-level sample selection where the possibility of an initially selected site not wanting to participate is anticipated, and the site is optimally replaced. The procedure aims to reduce bias (and/or loss of external validity) with respect to the target population. In selecting units and sub-units, ‘sitepickR’ uses the cube method developed by ‘Deville & Tillé’, (2004) <http://www.math.helsinki.fi/msm/banocoss/Deville_Tille_2004.pdf> and described in Tillé (2011) <https://www150.statcan.gc.ca/n1/en/pub/12-001-x/2011002/article/11609-eng.pdf?st=5-sx8Q8n>. The cube method is a probability sampling method that is designed to satisfy criteria for balance between the sample and the population. Recent research has shown that this method performs well in simulations for studies of educational programs (see Fay & Olsen (2021, under review). To implement the cube method, ‘sitepickR’ uses the sampling R package <https://cran.r-project.org/package=sampling>. To implement statistical matching, ‘sitepickR’ uses the ‘MatchIt’ R package <https://cran.r-project.org/package=MatchIt>.

Imports MatchIt, sampling, dplyr, ggplot2, reshape2, data.table, stats, stringr, magrittr, tidyselect, scales

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.2

Depends R (>= 2.10)

Suggests knitr, rmarkdown, devtools

VignetteBuilder knitr

NeedsCompilation no

Author Elena Badillo-Goicoechea [aut, cre], Robert Olsen [aut], Elizabeth Stuart [aut]
Description

Build summary tables, with unit/match/sub-unit balance between initially selected units and a target population, for each covariate of interest

Usage

getSummary(smOut, diagnostic)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smOut</td>
<td>list; selectMatch() output</td>
</tr>
<tr>
<td>diagnostic</td>
<td>numeric; balance Diagnostic: &quot;unitBal&quot; = original unit balance, &quot;matchBal&quot; = match balance, &quot;matchFreq&quot; = successful match frequency, &quot;matchCount&quot; = match success count by replacement group, &quot;subunitBal&quot; = sub-unit balance</td>
</tr>
</tbody>
</table>

Value

ggplot object
Examples

```
# Basic usage of getSummary()

rawCCD <- sitepickR::rawCCD

uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCD <- prepDF(rawCCD,
                  unitID="LEAID", subunitID="NCESSCH")
dfCCD <- dplyr::filter(dfCCD, unitID %in% unique(dfCCD$unitID)[1:80])

smOut <- selectMatch(df = dfCCD, # user dataset
                      unitID = "LEAID", # column name of unit ID in user dataset
                      subunitID = "NCESSCH", # column name of sub-unit ID in user dataset
                      unitVars = uSampVarsCCD, # name of unit level covariate columns
                      subunitSampVars = suSampVarsCCD, # name of sub-unit level covariate columns
                      nUnitSamp = 30,
                      nRepUnits = 5,
                      nsubUnits = 2
)

getSummary(smOut, diagnostic="unitBal")
```

---

**matchBalance**

**Match balance**

**Description**

Balance between initially sampled units and their K matches, for each covariate of interest

**Usage**

```
matchBalance(
  smOut,
  title = "Standardized Mean Difference:",
  subtitle = "Replacement Unit Groups (1...K) vs. Originally Selected Units"
)
```

**Arguments**

- `smOut` list; selectMatch() output
- `title` character; user-specified figure title
- `subtitle` character; user-specified figure title
matchCount

Value
ggplot object

Examples

# Basic usage of matchBalance()

rawCCD <- sitepickR::rawCCD

uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCD <- prepDF(rawCCD,
   unitID="LEAID", subunitID="NCESSCH")
dfCCD <- dplyr::filter(dfCCD, unitID %in% unique(dfCCD$unitID)[1:80])

smOut <- selectMatch(df = dfCCD, # user dataset
   unitID = "LEAID", # column name of unit ID in user dataset
   subunitID = "NCESSCH", # column name of sub-unit ID in user dataset
   unitVars = uSampVarsCCD, # name of unit level covariate columns
   subunitSampVars = suSampVarsCCD, # name of sub-unit level covariate columns
   nUnitSamp = 30,
   nRepUnits = 5,
   nsubUnits = 2)

matchBalance(smOut)

matchCount

Successful matches for each replacement group

Description
Percentage of successful matches in each unit replacement group, 1...K

Usage
matchCount(smOut, title = "Percentage of Successful Matches per Unit Group")

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smOut</td>
<td>list; selectMatch() output</td>
</tr>
<tr>
<td>title</td>
<td>character; user-specified figure title</td>
</tr>
</tbody>
</table>
### matchFreq

**Value**

- ggplot object

**Examples**

```r
# Basic usage of matchCount()

rawCCD <- sitepickR::rawCCD

uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCD <- prepDF(rawCCD,
                    unitID="LEAID", subunitID="NCESSCH")
dfCCD <- dplyr::filter(dfCCD, unitID %in% unique(dfCCD$unitID)[1:80])

smOut <- selectMatch(df = dfCCD, # user dataset
                      unitID = "LEAID", # column name of unit ID in user dataset
                      subunitID = "NCESSCH", # column name of sub-unit ID in user dataset
                      unitVars = uSampVarsCCD, # name of unit level covariate columns
                      subunitSampVars = suSampVarsCCD, # name of sub-unit level covariate columns
                      nUnitSamp = 30,
                      nRepUnits = 5,
                      nsubUnits = 2)

matchCount(smOut)
```

---

**matchFreq**  
*Match frequency*

**Description**

Distribution of successful matches among original units

**Usage**

```r
matchFreq(smOut, title = "Match Frequency per Original Unit")
```

**Arguments**

- `smOut` list; selectMatch() output
- `title` character; user-specified figure title
Value

ggplot object

Examples

# Basic usage of matchFreq()
rawCCD <- sitepickR::rawCCD

uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCD <- prepDF(rawCCD, unitID="LEAID", subunitID="NCESSCH")
dfCCD <- dplyr::filter(dfCCD, unitID %in% unique(dfCCD$unitID)[1:80])

smOut <- selectMatch(df = dfCCD, # user dataset
                      unitID = "LEAID", subunitID = "NCESSCH")
smOut <- dplyr::filter(smOut, unitID %in% unique(smOut$unitID)[1:80])

matchFreq(smOut)
**Value**

processed dataframe

**Examples**

```r
# Basic usage of prepDF()
rawCCD <- sitepickR::rawCCD
uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")
dfCCD <- prepDF(rawCCD,
    unitID="LEAID", subunitID="NCESSCH")
```

**Description**

A pre-processed dataset containing key variables from administrative data compiled by the CCD, aggregated at the district and school level for public schools in California for the 2017 and 2018 school years.

**Usage**

```r
data(rawCCD)
```

**Format**

A data frame with 1890 rows and 11 variables.

- **LEAID**  school district unique identifier
- **NCESSCH**  school unique identifier
- **w.pct.frlunch**  percentage of students in the school district who are under free/reduced price lunch program; weighted by school size.
- **w.pct.black**  percentage of students in the school district who are Black; weighted by school size.
- **w.pct.hisp**  percentage of students in the school district who are Hispanic; weighted by school size.
- **w.pct.female**  percentage of students in the school district who are female; weighted by school size.
- **sch.pct.frlunch**  percentage of students in the school who are under free/reduced price lunch program.
- **sch.pct.black**  percentage of students in the school who are Black.
selectMatch

`sch.pct.hisp` percentage of students in the school who are Hispanic.

`sch.pct.female` percentage of students in the school who are female.

`distr.type` school district type (constructed for illustration purposes; (values="A", "B", "C", "D").

`dstrct_size` number of schools in the district

Source

https://nces.ed.gov/ccd/files.asp#FileNameId:15,VersionId:10,FileSchoolYearId:33, Page:1

selectMatch  

Two-level sample selection

Description

Carries out a two-level sample selection where the possibility of an initially selected site not wanting to participate is anticipated, and the site is optimally replaced. The procedure aims to reduce the bias (and/or loss of generalizability) with respect to the target population.

Usage

```r
selectMatch(
  df,
  unitID,
  subunitID,
  subunitSampVars,
  unitVars,
  nUnitSamp,
  nRepUnits,
  nsubUnits,
  exactMatchVars = NULL,
  calipMatchVars = NULL,
  calipValue = 0.2,
  seedN = NA,
  matchDistance = "mahalanobis",
  sizeFlag = TRUE,
  repFlag = TRUE,
  writeOut = FALSE,
  replacementUnitsFilename = "replacementUnits.csv",
  subUnitTableFilename = "subUnitTable.csv"
)
```
selectMatch

Arguments

df dataframe; sub-unit level dataframe with both sub-unit and unit level variables
unitID character; name of unit ID column
subunitID character; name of sub-unit ID column
subunitSampVars vector; column names of unit level variables to sample units on
unitVars vector; column names of unit level variables to match units on
nUnitSamp numeric; number of units to be initially randomly selected
nRepUnits numeric; number of replacement units to find for each selected unit
nsubUnits numeric; number of sub-units to be randomly selected for each unit
exactMatchVars vector; column names of categorical variables on which units must be matched exactly. Must be present in 'unitVars'; default = NULL
calipMatchVars vector; column names of continuous variables on which units must be matched within a specified caliper. Must be present in 'unitVars'; default = NULL
calipValue numeric; number of standard deviations to be used as caliper for matching units on calipMatchVars
seedN numeric; seed number to be used for sampling. If NA, calls set.seed(); default = NA
matchDistance character; MatchIt distance parameter to obtain optimal matches (nearest neighbors); default = "mahalanois"
sizeFlag logical; if TRUE, sampling is made proportional to unit size; default = TRUE
repFlag logical; if TRUE, pick unit matches with/without repetition; default = TRUE
writeOut logical; if TRUE, writes a .csv file for each output table; default = FALSE
replacementUnitsFilename character; csv filename for saving unit:replacement directory when writeOut == TRUE; default = "replacementUnits.csv"
subUnitTableFilename character; csv filename for saving unit:replacement directory when writeOut == TRUE; default = "subUnitTable.csv"

Value

list with: 1) table of the form: selected unit i: (unit i replacements), 2) table of the form: potential unit i:(unit i sub-units), 3) balance diagnostics.

Examples

# Basic usage of selectMatch()

rawCCD <- sitepickR::rawCCD
subUnitBalance <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
subunitSampVarsCCI <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCI <- prepDF(rawCCI,
  unitID="LEAID", subunitID="NCESSCH")
dfCCI <- dplyr::filter(dfCCI, unitID %in% unique(dfCCI$unitID)[1:80])

smOut <- selectMatch(df = dfCCI, # user dataset
  unitID = "LEAID", # column name of unit ID in user dataset
  subunitID = "NCESSCH", # column name of sub-unit ID in user dataset
  unitVars = uSampVarsCCI, # name of unit level covariate columns
  subunitSampVars = suSampSampVarsCCI, # name of sub-unit level covariate columns
  nUnitSamp = 30,
  nRepUnits = 5,
  nsubUnits = 2
)

---

subUnitBalance

Sub-unit balance

Description

Sub-unit balance between initially selected units and all units in population, for each covariate of interest

Usage

subUnitBalance(
  smOut,
  title = "Subunits from Original and Replacement Unit Groups vs. Population (SMD)"
)

Arguments

- smOut: list; selectMatch() output
- title: character; user-specified figure title

Value

ggplot object

Examples

# Balance Diagnostics [sitepickR Package]
# Robert Olsen, Elizabeth A. Stuart & Elena Badillo-Goicoechea (2022)
# Basic usage of subUnitBalance()
rawCCD <- sitepickR::rawCCD

uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCD <- prepDF(rawCCD,
  unitID="LEAID", subunitID="NCESSCH")
dfCCD <- dplyr::filter(dfCCD, unitID %in% unique(dfCCD$unitID)[1:80])

smOut <- selectMatch(df = dfCCD, # user dataset
  unitID = "LEAID", # column name of unit ID in user dataset
  subunitID = "NCESSCH", # column name of sub-unit ID in user dataset
  unitVars = uSampVarsCCD, # name of unit level covariate columns
  subunitSampVars = suSampVarsCCD, # name of sub-unit level covariate columns
  nUnitSamp = 30,
  nRepUnits = 5,
  nsubUnits = 2
)

subUnitBalance(smOut = smOut,
  title = "Standardized Mean Difference: "
  "Sub-units from Original + Replacement Unit Groups vs. Population")

---

### unitLovePlot

**Original units balance**

**Description**

Balance between initially sampled units and all units in the population

**Usage**

```r
unitLovePlot(
  smOut, # selectMatch() output
title = "Standardized Mean Difference",
subtitle = "Initially Selected Units vs. Population"
)
```

**Arguments**

- `smOut` list
- `title` character; user-specified figure title
- `subtitle` character; user-specified figure subtitle

**Value**

`ggplot` object
# Basic usage of unitLovePlot()
rawCCD <- sitepickR::rawCCD

uSampVarsCCD <- c("w.pct.frlunch", "w.pct.black", "w.pct.hisp", "w.pct.female")
suSampVarsCCD <- c("sch.pct.frlunch", "sch.pct.black", "sch.pct.hisp", "sch.pct.female")

dfCCD <- prepDF(rawCCD,
    unitID = "LEAID", subunitID="NCESSCH")
dfCCD <- dplyr::filter(dfCCD, unitID %in% unique(dfCCD$unitID)[1:80])

smOut <- selectMatch(df = dfCCD, # user dataset
    unitID = "LEAID", # column name of unit ID in user dataset
    subunitID = "NCESSCH", # column name of sub-unit ID in user dataset
    unitVars = uSampVarsCCD,# name of unit level covariate columns
    subunitSampVars = suSampVarsCCD,# name of sub-unit level covariate columns
    nUnitSamp = 30,
    nRepUnits = 5,
    nsubUnits = 2
)

unitLovePlot(smOut)
Index

* datasets
  rawCCD, 7

getSummary, 2

matchBalance, 3
matchCount, 4
matchFreq, 5

prepDF, 6

rawCCD, 7

selectMatch, 8
subUnitBalance, 10

unitLovePlot, 11