Causal Inference in Case-Control Studies: Vignette

Overview

This vignette describes how to use package "ciccr" that is based on the paper entitled "Causal Inference in Case-Control Studies".

The Model and Inference

This section will be added ...

Tutorials

We first call the ciccr package.

```
library(ciccr)
```

To illustrate the usefulness of the package, we use the dataset ACS that is included the package. This dataset is a extract from American Community Survey (ACS) 2018, restricted to white males residing in California with at least a bachelor's degree. The ACS is an ongoing annual survey by the US Census Bureau that provides key information about US population. We use the following variables:

```
y = ACS$topincome
t = ACS$baplus
x = ACS$age
```

- The binary outcome 'Top Income' (Y) is defined to be one if a respondent's annual total pre-tax wage and salary income is top-coded. In our sample extract, the top-coded income bracket has median income $565,000 and the next highest income that is not top-coded is $327,000.
- The binary treatment (T) is defined to be one if a respondent has a master's degree, a professional degree, or a doctoral degree.
- The covariate (X) is age in years and is restricted to be between 25 and 70.

The original ACS sample is not a case-control sample but we construct one by the following procedure.

1. The case sample ($Y = 1$) is composed of 921 individuals whose income is top-coded.
2. The control sample ($Y = 0$) of equal size is randomly drawn without replacement from the pool of individuals whose income is not top-coded.

We now construct cubic b-spline terms with three inner knots using the age variable.

```
x = splines::bs(x, df = 6)
```

Define $\beta(y) = E[\log \mathrm{OR}(X)|Y = y]$ for $y = 0, 1$, where $\mathrm{OR}(x)$ is the odds ratio conditional on $X = x$:

$$\mathrm{OR}(x) = \frac{P(T = 1|Y = 1, X = x)}{P(T = 0|Y = 1, X = x)} \frac{P(T = 0|Y = 0, X = x)}{P(T = 1|Y = 0, X = x)}.$$

Using the retrospective sieve logistic regression model, we estimate $\beta(1)$ by

```
  results_case = avg_retro_logit(y, t, x, 'case')
  results_case$est
#>         y
#> 0.7286012
  results_case$se
#>         y
#> 0.1013445
```

Here, option `'case'` refers to conditioning on $Y = 1$.

Similarly, we estimate $\beta(0)$ by

```
  results_control = avg_retro_logit(y, t, x, 'control')
  results_control$est
#>         y
#> 0.5469094
  results_control$se
#>         y
#> 0.1518441
```

Here, option `'control'` refers to conditioning on $Y = 1$.

We carry out causal inference by

```
results = cicc(y, t, x, 0.2, 0.9)
```

Here, 0.2 is the specified upper bound for unknown $p = \Pr(Y = 1)$. If it is not specified, the default choice for the upper bound for p is p_upper $= 1$. Here, 0.9 refers to the level of the confidence interval. 0.95 is the default choice.

The point estimate and the upper end of the confidence interval can be obtained from the saved results:

```
 # point estimate
 results$est
#>  [1] 0.5469094 0.5488219 0.5507345 0.5526470 0.5545596 0.5564721 0.5583847 0.5602972 0.5622097
#> [10] 0.5641223 0.5660348 0.5679474 0.5698599 0.5717725 0.5736850 0.5755976 0.5775101 0.5794227
#> [19] 0.5813352 0.5832477
 # point estimate
 results$ci
#>  [1] 0.7415054 0.7413744 0.7412533 0.7411425 0.7410422 0.7409529 0.7408748 0.7408084 0.7407540
#> [10] 0.7407121 0.7406830 0.7406672 0.7406651 0.7406772 0.7407040 0.7407460 0.7408038 0.7408778
#> [19] 0.7409686 0.7410768
```

To be more compatible with the odds ratio, it is useful to transform them by the exponential function:

```
 # point estimate
 exp(results$est)
#>  [1] 1.727904 1.731212 1.734527 1.737847 1.741174 1.744507 1.747847 1.751193 1.754545 1.757904
#> [11] 1.761269 1.764641 1.768019 1.771404 1.774795 1.778193 1.781597 1.785008 1.788425 1.791848
 # point estimate
 exp(results$ci)
#>  [1] 2.099093 2.098818 2.098564 2.098332 2.098121 2.097934 2.097770 2.097631 2.097516 2.097428
#> [11] 2.097367 2.097334 2.097330 2.097355 2.097412 2.097500 2.097621 2.097776 2.097967 2.098194
```

Comparison with Logistic Regression

```
logit = stats::glm(y~t+x, family=stats::binomial("logit"))
est_logit = stats::coef(logit)
ci_logit = stats::confint(logit, level = 0.9)
#> Waiting for profiling to be done...
est_logit[2]
#>         t
#> 0.7232745
ci_logit[2]
#> [1] 0.5605642
```

Reference

Sung Jae Jun and Sokbae Lee. Causal Inference in Case-Control Studies. https://arxiv.org/abs/2004.08318.