Example Data Analysis: Epi 202 Calculator in R

Version date:
October 27, 2020

Author:
Samantha Molsberry
sam306@mail.harvard.edu

As researchers, our data is usually not given to us in tabular format. Instead, we work with datasets that have row(s) of observations for each of the participants in our study and columns with different variables we have collected information on from our participants. An example of such a dataset, `evans_example_dat.csv` has been provided to you on the course website. This dataset has been adapted from the Evans County dataset provided by the R package `lbreg`. In the version provided to you, we have kept the three variables of interest and additionally generated a variable for person-time and case-control selection so that you can will be able to conduct closed cohort, open cohort, and case-control analyses all on the same example data. Briefly, the Evans County study followed white males for incident coronary heart disease for 7 years. For the purposes of these examples, we are going to assume no loss to follow-up or competing risks.

To begin, we can set our working directory to the pathway where you saved the dataset. For example, my dataset is saved in my Dropbox in a subfolder called 'Sam Code'

```
setwd("C:/Users/Sam/Dropbox/Epi202 TA Materials Fall 2018/Homework/Sam Code")
```

Next, we will need to load the two packages that will be used in our analyses, `dplyr` and `epicalc_v3`. If you have never installed the `dplyr`, begin by running the following command:

```
install.packages('dplyr')
```

Next, you need to load the `dplyr` library by running:

```
library(dplyr)
```

We also need to load the `epicalc_v3` package. We can this by running the `source` command using the file path at which you saved the `epicalc_v3` file. For example:

```
source("C:/Users/Sam/Documents/R/win-library/3.4/epi202calc/epicalc_v3.R")
```

Now that we have both of the necessary packages loaded, it is time to load our dataset. To do this, we can use the `read.csv` command and save our data as a dataset called 'evansData'. The header=T option tells R that our csv file contains variables names in the first row.

```
evansData<-read.csv("evans_example_dat.csv", header=T)
```

1 Exploring your data

Before beginning our analyses, it is a good idea to become familiar with our dataset. We can look begin by looking at the top of our dataset to make sure it loaded properly:

```
head(evansData)
```

```
  CDH HPT SMK person_time caco
1   0   0   0    7.000000 <NA>
2   0   0   1    7.000000 <NA>
3   1   1   1    2.194249 case
4   0   1   1    7.000000 <NA>
5   0   0   1    7.000000 <NA>
6   0   0   1    7.000000 <NA>
```

For your reference, the variables in the dataset are coded as follows:

- CHD: a binary indicator of outcome status where 1=incident CHD case and 0=non-case
- HTN: a binary indicator of exposure status where 1=hypertensive and 0=not hypertensive at baseline
- SMK: a binary of smoking status where 1=smoker and 0=non-smoker at baseline
- person time: a continuous variable representing amount of person-time under follow-up prior to a CHD event (maximum=7)
- caco: a binary indicator of case status for participants selected into a nested case-control study; 'case'=case and 'control'=control

To use the functions in the `epicalc` package, both the exposure and outcome variables of interest must be binary. We can check that this is true for our data by first getting a table of the exposure:

```
table(evansData$HTN)
```

```
  0   1
354 255
```

And do the same for our outcome:

```
table(evansData$CHD)
```

```
  0   1
538  71
```

For case-control analyses, we can also look at the indicator of case status ('caco'):

```
table(evansData$caco)
```

```
 case control
   71     148
```

Of course, we might also be interested in the distribution of person-time in our dataset:

```
summary(evansData$person_time)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2736  7.0000  7.0000  6.5353  7.0000  7.0000
```

Important note: In the example dataset provided to you, the variables have already been appropriately coded for you to be compatible with the `epicalc` package. If you are working with a different dataset, it is important to note that exposure variable MUST be coded such that the exposed have a higher value of the indicator variable than the unexposed. This will work, for example, if your exposure variable is coded such that the exposed get a value of '1' and the unexposed get a value of '0'.

2 Count Data: Crude Analysis

Now that our data is loaded and we have become familiar with it, we can begin using `dplyr` to manipulate our dataset so that it is in a format that can be used by the `epicalc_v3` package. The `dplyr` package allows us to use 'piping', which is represented by the character combination `'%>%'`, to do several steps at once before saving the final object. In order to do analyses with the `epicalc_v3` package, we need to get the counts of cases, non-cases, and total person-time for each combination of exposure and outcome. Additionally, for stratified analyses, we will also need the stratum-specific values of these variables.

To start, we will do the crude analysis of the data, treating it as a closed cohort. For this example, we will be looking at the association between hypertension status at baseline (`evansData$HTN`) and incident coronary heart disease

(evansData$CHD). To get the counts of cases and non-cases for each level of exposure and store this data as a new dataset, we can run the following code:

```
crude.evans.risk<-as.data.frame(evansData %>%
group_by(HTN)%>%
summarise(Ncase=sum(CHD=='1'),
  Nnoncase=sum(CHD=='0')))
```

Here, we use the group_by command to tell R to group our data by the exposure variable, HTN. We then pipe this grouped dataset to the summarise command and use the sum option to get the number of cases (Ncases) and the number of non-cases (Nnoncase) for each level of exposure. These counts are then saved as a two row data frame called 'crude.evans.risk'. If we use the View command (ex: View(crude.evans.risk)), we can see that our new dataset looks like:

```
  HPT Ncase Nnoncase
1   0    28      326
2   1    43      212
```

With our new dataset, 'crude.evans.risk', we can create a riskTable and do the crude closed cohort analysis with the following two commands:

```
crude.evans.riskTable<-as.riskTable.new(crude.evans.risk$Ncase, crude.evans.risk$Nnoncase)
summary(crude.evans.riskTable, alpha=0.05)
```

Which gives us the following risk table:

```
                Exposed     Not exposed
Cases                43              28
Non-cases           212             326
```

And statistical output:

```
Point estimate of CIR:  2.1319
95% confidence interval for CIR: 1.3622 - 3.3365
Point estimate of CID:  0.08953
95% confidence interval for CID: 0.03566 - 0.1434
H0 test: X^2 = 11.5364 with p = 0.0006825
```

3 Count Data: Stratified Analysis

We can also do stratified analyses using much the same code. For example, if we were concerned about smoking status acting as a confounder of the association between hypertension and coronary heart disease, we could condition on smoking status. To do this in our data, we can use the same dplyr piping sequence as above, simply adding our stratification factor, smoking (SMK) to the group_by command BEFORE our exposure variable. This time we will save our outputted dataset as 'stratified.evans.risk':

```
stratified.evans.risk<-as.data.frame(evansData %>%
group_by(SMK,HTN)%>%
summarise(Ncase=sum(CHD=='1'),
  Nnoncase=sum(CHD=='0')))
```

Now our dataset has 4 rows: 2 rows (1 for each level of exposure) per stratum of smoking:

```
  SMK HPT Ncase Nnoncase
1   0   0     6      122
2   0   1    11       83
3   1   0    22      204
4   1   1    32      129
```

To do the stratified analyses on these data, we can create a `riskTable` and apply the `summary` command:

```
stratified.evans.riskTable<-as.riskTable.new(stratified.evans.risk$Ncase,  stratified.evans.risk$Nnoncase)
summary(stratified.evans.riskTable, alpha=0.05)
```

Which gives us the following riskTable:

```
Stratum 1:
                Exposed     Not exposed
  Cases              11               6
  Non-cases          83             122

Stratum 2:
                Exposed     Not exposed
  Cases              32              22
  Non-cases         129             204

Crude data:
                Exposed     Not exposed
  Cases              43              28
  Non-cases         212             326
```

And statistical output:

```
Crude estimate of CIR:  2.1319
95% confidence interval for crude CIR: 1.3622 - 3.3365
Crude estimate of CID:  0.08953
95% confidence interval for crude CID: 0.03566 - 0.1434
Crude H0 test: X^2 = 11.5364 with p = 0.0006825

Mantel-Haenszel estimate of CIR:  2.1406
95% confidence interval for MH CIR: 1.3704 - 3.3436
Mantel-Haenszel-style estimate of CID:  0.08998
95% confidence interval for MH CID: 0.03619 - 0.1438
Stratified H0 test: X^2 = 11.7629 with p = 0.0006042

Test of CIR homogeneity: X^2 = 0.1327 with p = 0.7156
Test of CID homogeneity: X^2 = 0.3665 with p = 0.5449
```

4 Person-time Data: Crude Analysis

In addition to being able to do analyses of count data, we can do analyses of person-time data with only small modifications to our code. Instead of getting counts of cases and non-cases, now we need to get the count of cases and sum of observed follow-up time for each level of exposure. We can use `dplyr` to do this:

```
crude.evans.rate<-as.data.frame(evansData %>% group_by(HTN)%>%
```

```
summarise(Ncase=sum(CHD=='1'),
  PY=sum(person_time)))
```

The crude.evans.rate data frame looks like:

```
  HPT Ncase       PY
1   0    28 2393.356
2   1    43 1586.654
```

Next, we can use the `as.rateTable.new` command to produce a `rateTable` from our dataset:

```
crude.evans.rateTable<-as.rateTable.new(crude.evans.rate$Ncase, crude.evans.rate$PY)
```

Which will look like:

```
                Exposed     Not exposed
Cases                43              28
Person-time    1586.654         2393.356
```

Before applying the summary command to obtain our statistical analysis results:

```
summary(crude.evans.rateTable, alpha=0.05)
```

Which gives us the following output:

```
Point estimate of IRR:  2.3165
95% confidence interval for IRR: 1.4392 - 3.7285
Point estimate of IRD:  0.0154
95% confidence interval for IRD: 0.006215 - 0.02459
H0 test: X^2 = 12.6878 with p = 0.0003681
```

5 Person-time Data: Stratified Analysis

We can also do stratified analyses of person-time data using much the same code. For example, if we were concerned about smoking status acting as a confounder of the association between hypertension and coronary heart disease, we could condition on smoking status. To do this in our data, we can use the same `dplyr` piping sequence as above, simply adding our stratification factor, smoking (`SMK`) to the `group_by` command BEFORE our exposure variable. This time we will save our outputted dataset as 'stratified.evans.rate':

```
stratified.evans.rate<-as.data.frame(evansData %>% group_by(SMK, HTN)%>%
summarise(Ncase=sum(CHD=='1'),
  PY=sum(person_time)))
```

Now our dataset has 4 rows: 2 rows (1 for each level of exposure) per stratum of smoking:

```
  SMK HPT Ncase        PY
1   0   0     6  866.2547
2   0   1    11  615.6133
3   1   0    22 1527.1018
4   1   1    32  971.0408
```

To do the stratified analyses on these data, we can create a `rateTable` and apply the `summary` command:

```
stratified.evans.rateTable<-as.rateTable.new(stratified.evans.rate$Ncase,  stratified.evans.rate$PY)
summary(stratified.evans.rateTable, alpha=0.05)
```

Which gives us the following rateTable:

```
Stratum 1:
                Exposed     Not exposed
  Cases              11               6
  Person-time    615.6133         866.2547

Stratum 2:
                Exposed     Not exposed
  Cases              32              22
  Person-time    971.0408         1527.102

Crude data:
                Exposed     Not exposed
  Cases              43              28
  Person-time    1586.654         2393.356
```

And statistical output:

```
Crude estimate of IRR:  2.3165
95% confidence interval for crude IRR: 1.4392 - 3.7285
Crude estimate of IRD:  0.0154
95% confidence interval for crude IRD: 0.006215 - 0.02459
Crude H0 test: X^2 = 12.6878 with p = 0.0003681

Mantel-Haenszel estimate of IRR:  2.3534
95% confidence interval for MH IRR: 1.4615 - 3.7897
Mantel-Haenszel-style estimate of IRD:  0.01568
95% confidence interval for MH IRD: 0.006467 - 0.02489
Stratified H0 test: X^2 = 13.1743 with p = 0.0002838

Test of IRR homogeneity: X^2 = 0.04326 with p = 0.8352
Test of IRD homogeneity: X^2 = 0.7957 with p = 0.3724
```

6 Case-control Data: Crude Analysis

We can also do unmatched case-control analyses using raw data and the `epicalc` package. To do this, we need to begin by filter our data to contain only those individuals selected as either case or controls into our case-control study, which is represented by the 'caco' indicator variable in our example data set. To do this, we can use the following code, which tells R to keep only those observations where 'caco' is not NA and store this filtered dataset as a dataset called 'evansCC':

```
evansCC<-evansData%>%filter(!is.na(caco))
```

Now that we have a dataset which contains only the selected cases and controls, we can do our case-controls analyses in much the same way as we did our closed and open cohort analyses above. To begin, we will start with the crude analyses by getting the count of cases and controls for each level of exposure. Even though our 'caco' variable is coded with levels of 'case' and 'control' rather than '1' and '0', we can do this using a set of `dplyr` piping sets as above, this time getting the count of controls rather than non-cases or person-time by simply changing the value of 'caco' that represents a case or control respectively within our `summarise` command:

```
crude.evans.cc<-as.data.frame(evansCC %>%
group_by(HTN)%>%
summarise(Ncase=sum(caco=='case'),
```

```
  Ncontrol=sum(caco=='control')))
```

This new dataset, 'crude.evans.cc' looks like:

	HPT	Ncase	Ncontrol
1	0	28	85
2	1	43	63

Next, we can use the `as.ccTable.new` command to produce a `ccTable` from our dataset:

```
crude.evans.ccTable<-as.ccTable.new(crude.evans.cc$Ncase, crude.evans.cc$Ncontrol)
```

Which will look like:

	Exposed	Not exposed
Cases	43	28
Controls	63	85

Before applying the summary command to obtain our statistical analysis results:

```
summary(crude.evans.ccTable, alpha=0.05)
```

Which gives us the following output:

```
Point estimate of OR:  2.0720
95% confidence interval for OR: 1.1638 - 3.6888
H0 test: X^2 = 6.1935 with p = 0.01282
```

7 Case-control Data: Stratified Analysis

Lastly, we can use the `epicalc` package to do stratified analyses of case-control data. Adapting our code from the crude case-control analyses above, we can obtain a dataset stratified on smoking status by adding smoking (`SMK`) to the `group_by` command BEFORE our exposure variable (`HTN`). For example:

```
stratified.evans.cc<-as.data.frame(evansCC %>%
group_by(SMK,HTN)%>%
summarise(Ncase=sum(caco=='case'),
  Ncontrol=sum(caco=='control')))
```

This new dataset, 'stratified.evans.cc' looks like:

	SMK	HPT	Ncase	Ncontrol
1	0	0	6	35
2	0	1	11	23
3	1	0	22	50
4	1	1	32	40

Next, we can use the `as.ccTable.new` command to produce a `ccTable` from our dataset:

```
stratified.evans.ccTable<-as.ccTable.new(stratified.evans.cc$Ncase, stratified.evans.cc$Ncontrol)
```

Which will look like:

```
Stratum 1:
                Exposed     Not exposed
  Cases              11               6
  Controls           23              35

Stratum 2:
                Exposed     Not exposed
  Cases              32              22
  Controls           40              50

Crude data:
                Exposed     Not exposed
  Cases              43              28
  Controls           63              85
```

Before applying the summary command to obtain our statistical analysis results:

```
summary(stratified.evans.ccTable, alpha=0.05)
```

Which gives us the following output:

```
Crude estimate of OR:  2.0720
95% confidence interval for crude OR: 1.1638 - 3.6888
Crude H0 test: X^2 = 6.1935 with p = 0.01282

Mantel-Haenszel estimate of OR:  2.0430
95% confidence interval for MH OR: 1.1406 - 3.6594
Stratified H0 test: X^2 = 5.8295 with p = 0.01576

Test of OR homogeneity: X^2 = 0.4062 with p = 0.5239
```