# Using the epi calculator in R

*Version date:*
September 25, 2018

*Author:*
Travis Gerke
tgerke@hsph.harvard.edu

## 1 Installation

After downloading the `epicalc_v2.R` file to a convenient location on your computer, the package can be loaded into your R session with the command

```
source("/Users/Travis/Dropbox/classes/epi202/software/epicalc_v2.R")
```

Of course, the path should be edited to match that on your system. Windows users should note that the forward-slashes ("/") must be replaced by back-slashes ("\").

## 2 Usage

The `epicalc_v2.r` package has three general function types designed for data entry, data viewing, and effect estimation. The data entry functions are comprised of `as.rateTable`, `as.riskTable`, `as.ccTable`, and `as.mccTable`; these allow you to input open cohort, cross-sectional, case-control, and matched case-control data, respectively. The data viewing functions are called simply by typing the name of your data table at any time following data entry. Effect estimation is achieved for each of the data types through the `summary` function. A fourth class of functions convert tables into long-data formats to permit use by other R methods. This option is discussed in the "Extensions" section.

In what follows, an example session is provided for each of the data types. The data and descriptions correspond to those found in the `Statistical_software_2008_update.doc` document. All code in this document is provided in the `examples.R` file.

### 2.1 Closed cohort study

We will analyze the Evans Country Study, a closed cohort study with 487 unexposed subjects and 122 exposed subjects followed over 7 years. Among them, 44 unexposed and 27 exposed subjects developed the disease. We first use the `as.riskTable` function to enter the data. The general input format of the `as.riskTable` function is

```
as.riskTable(exposed cases, exposed non-cases, unexposed cases, unexposed non-cases, ...)
```

where we can repeat this ordering for each stratum in our data. So, we enter the Evans data as

```
evansData <- as.riskTable(1, 7, 17, 257,
                          3, 14, 7, 52,
                          9, 30, 15, 107,
                          14, 44, 5, 27)
```

Alternatively, instead of inputting our data as counts in the example above, we may wish to create a similar table using columns from a data frame. To do this, we can use the command `as.riskTable.new`. In this case, we need to have previously created a dataset in which there is a column with the counts of cases and a separate column with the count of non-cases. In this dataset, the odd rows should be the unexposed individuals and the even rows should be the exposed individuals. This dataset can be either crude (2 rows), or stratified, such that rows corresponding to the exposed and unexposed individuals in a given stratum are consecutive. Instructions on how to produce such

a dataset from raw data are provided at the end of this document. For example, an appropriate dataset for crude analyses would look like:

| | Exposed | Ncase | Nnoncase |
|---|---|---|---|
| 1 | 0 | 28 | 326 |
| 2 | 1 | 43 | 212 |

And an appropriate dataset for stratified analyses would look like:

| | Stratification.factor | Exposed | Ncase | Nnoncase |
|---|---|---|---|---|
| 1 | 0 | 0 | 6 | 122 |
| 2 | 0 | 1 | 11 | 83 |
| 3 | 1 | 0 | 22 | 204 |
| 4 | 1 | 1 | 32 | 129 |

Given an appropriately structured dataset, then, we can construct the corresponding riskTable object by:

```
evansData <- as.riskTable.new(dataset$cases, dataset$non_cases)
```

We can now view the Evans data, with both stratified and crude tables, by simply typing

```
evansData
```

If we would like to only view the second stratum, we can access it with

```
evansData[[2]]
```

A broad summary of effect estimates with 95% confidence intervals is given by

```
summary(evansData, alpha=.05)
```

Note that if we instead wanted 90% confidence intervals, we could have set `alpha=.10`. If we want to summarize a single stratum in isolation, we can do so with

```
summary(evansData[[2]], alpha=.05)
```

## 2.2 Case-control study

We will analyze a case control study of OC use and MI risk with 153 cases and 1418 controls. Among them, 23 cases and 112 controls were exposed. To input case-control data, we use the `as.ccTable` function, which has form

```
as.ccTable(exposed cases, exposed controls, unexposed cases, unexposed controls, ...)
```

where, as before, we can repeat this ordering for each stratum in our data. The MI data is then entered with

```
miData <- as.ccTable(4, 62, 2, 224,
                     9, 33, 12, 390,
                     4, 26, 33, 330,
                     6, 9, 65, 362)
```

Data viewing is accomplished with

```
miData
```

or, in a stratum-specific fashion, by

```
miData[[4]]
```

Similarly, effect estimates can be extracted using the full data with

```
summary(miData, alpha=.05)
```

or, using only one stratum,

```
summary(miData[[4]], alpha=.05)
```

Alternatively, instead of inputting our data as counts like in the example above, we may wish to create a similar table using columns from a data frame. To do this, we can use the command `as.ccTable.new`. In this case, we need to have previously created a dataset in which there is a column with the counts of cases and a separate column with the count of controls. In this dataset, the odd rows should be the unexposed individuals and the even rows should be the exposed individuals. This dataset can be either crude (2 rows), or stratified, such that rows corresponding to the exposed and unexposed individuals in a given stratum are consecutive. Instructions on how to produce such a dataset from raw data are provided at the end of this document. Using the MI data as a example, an appropriate dataset for crude analyses would look like:

| | Exposed | Ncase | Ncontrol |
|---|---|---|---|
| 1 | 0 | 112 | 1306 |
| 2 | 1 | 23 | 130 |

And a stratified version might look like:

| | Stratification.factor | Exposed | Ncase | Ncontrol |
|---|---|---|---|---|
| 1 | 0 | 0 | 2 | 224 |
| 2 | 0 | 1 | 4 | 62 |
| 3 | 1 | 0 | 12 | 390 |
| 4 | 1 | 1 | 9 | 33 |
| 5 | 2 | 0 | 33 | 330 |
| 6 | 2 | 1 | 4 | 26 |
| 7 | 3 | 0 | 65 | 362 |
| 8 | 3 | 1 | 6 | 9 |

With the data formatted to match either of these tables, we can do the crude or stratified analyses respectively with the following command:

```
britishData <- as.ccTable.new(dataset$cases, dataset$controls)
```

With this `ccTable` object, we can use the rest of the commands in the epi 202 calculator package in the same way we used them when inputting the counts for the original function, as.ccTable.

## 2.3 Open cohort study

We will analyze an open cohort study, the British Doctors Study of cigarette smoking and CHD mortality. Among the exposed, 630 subjects develop the disease, while 101 subjects had the disease among unexposed. The rate data input format is

```
as.rateTable(exposed cases, exposed person-time, unexposed cases, unexposed person-time,...)
```

which is repeated for each stratum. Data viewing and effect estimation can be accomplished by the two lines

```
britishData
summary(britishData, alpha=.10)
```

or

```
britishData[[3]]
summary(britishData[[3]], alpha=.10)
```

for a specific stratum.

As with the closed cohort analyses and case-control analyses above, we can do the same set of analyses for open cohort data using alternative inputs to create our `rateTable` object. To do this, we can use the command `as.rateTable.new`. In this case, we need to have previously created a dataset in which there is a column with the counts of cases and a separate column with the sum of person-time accumulated under follow-up. In this dataset, the odd rows should be the unexposed individuals and the even rows should be the exposed individuals. This dataset can be either crude (2 rows), or stratified, such that rows corresponding to the exposed and unexposed individuals in a given stratum are consecutive. Instructions on how to produce such a dataset from raw data are provided at the

end of this document. Using the Evans County data as an example and simulating the person-time, an appropriate dataset for crude analyses would look like:

| | Exposed | Ncase | PY |
|---|---|---|---|
| 1 | 0 | 28 | 2393.356 |
| 2 | 1 | 43 | 1586.654 |

And a stratified version might look like:

| | Stratification.factor | Exposed | 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 |

With the data formatted to match either of these tables, we can do the crude or stratified analyses respectively with the following command:

```
britishData <- as.rateTable.new(dataset$cases, dataset$person_time)
```

With this `rateTable` object, we can use the rest of the commands in the epi 202 calculator package in the same way we used them when inputting the counts for the original function, `as.ccTable`.

### 2.4 Matched case-control study

Matched data is entered by specifying a matching ratio (the number of controls per case), followed by the numbers of matched pairs. The entry style is best understood through examples. If we have a matching ratio of 1 with (i) 4 matched pairs with an exposed case and 1 exposed control, (ii) 8 matched pairs with an exposed case and 0 exposed controls, (iii) 1 matched pair with an unexposed case and 1 exposed control, and (iv) 5 matched pairs with an unexposed case and 0 exposed controls, we would enter

```
matched1Data <- as.mccTable(matchRatio=1, 4, 8, 1, 5)
```

We would then view and summarize the data with

```
matched1Data
summary(matched1Data)
```

A data example with a matching ratio of 2 is provided in the `examples.R` file.

## 3 Extensions

The `epicalc_v2.R` package also provides methods for converting the data tables to so-called data frames, which are the standard data format for analysis by most R functions. This can be useful if an effect measure is desired that is not provided within the framework of `epicalc_v2.R`.

To convert the earlier MI case-control data, we can use the function

```
miLong <- as.data.frame(miData)
```

A quick look at the first few observations with `head(miLong)` shows

```
> head(miLong)
  Y X C
1 1 1 A
2 1 1 A
3 1 1 A
4 1 1 A
5 1 0 A
6 1 0 A
```

revealing that the function has labeled our outcomes with $Y$, our exposure with $X$, and our stratifying variable with $C$. We can now run the sequence

```
fit <- glm(Y ~ X, data=miLong, family=binomial)
summary(fit)
exp(fit$coef)
```

to verify that the crude OR estimate via logistic regression is, indeed, the same as the one we got with `summary(miData)`. Note that we would have also gotten another type of $C$-stratified estimate (i.e. not the Mantel-Haenszel) with

```
fit <- glm(Y ~ X + C, data=miLong, family=binomial)
```

The Evans Country risk data can be similarly converted with

```
evansLong <- as.data.frame(evansData)
```

In this case, the exponentiated crude estimate provided through the Poisson regression

```
evansLong <- as.data.frame(evansData)
fit <- glm(Y ~ X, data=evansLong, family=poisson)
exp(fit$coef)
```

equals the one we got with `summary(evansData)`. Fantastic!

# 4 Manipulating raw data with the 'dplyr' package

In order to get a raw dataset into the format required for the `as.riskTable.new`, `as.rateTable.new`, and `as.ccTable.new` functions, you need the following in your raw dataset:

1. A binary exposure variable that is coded as 0=unexposed, 1=exposed
2. A binary outcome variable that is coded as 0=no outcome/control, 1=outcome/case
3. For stratified analyses only: stratification variable(s) that are coded as factors
4. For person-time based analyses: a variable with the amount of person-time that was observed for each individual.

Given the appropriate variables in your raw data, we can then use the `dplyr` package to manipulate our data to create the inputs needed for the `as.riskTable.new`, `as.rateTable.new`, and `as.ccTable.new` functions. To install and load the `dplyr` package, run the following code:

```
install.packages('dplyr') #only need to install the FIRST time you use dplyr
library(dplyr) #need to load the library EVERY time you use dplyr
```

The generic `dplyr` code to set up the input data for both crude and stratified analyses of count data, person-time data, and case-control data respectively is given below. For more detailed information using an example dataset, please see the document 'Example Example Data Analysis: Epi 202 Calculator in R'.

Crude count data:

```
crude.riskTable.input<-as.data.frame(data %>%
group_by(exposure)%>%
summarise(Ncase=sum(outcome=='1'),
Nnoncase=sum(outcome=='0')))
```

Stratified count data:

```
stratified.riskTable.input<-as.data.frame(data %>%
group_by(stratification.factor(s), exposure)%>%
summarise(Ncase=sum(outcome=='1'),
Nnoncase=sum(outcome=='0')))
```

Crude person-time data:

```
crude.rateTable.input<-as.data.frame(data %>%
group_by(exposure)%>%
summarise(Ncase=sum(outcome=='1'),
PY=sum(person_time)))
```

Stratified person-time data:

```
stratified.rateTable.input<-as.data.frame(data %>%
group_by(stratification.factor(s), exposure)%>%
summarise(Ncase=sum(outcome=='1'),
PY=sum(person_time)))
```

Crude case-control data:

```
crude.ccTable.input<-as.data.frame(data %>%
group_by(exposure)%>%
summarise(Ncase=sum(outcome=='1'),
Ncontrol=sum(outcome=='control')))
```

Stratified case-control data:

```
stratified.ccTable.input<-as.data.frame(data %>%
group_by(stratification.factor(s), exposure)%>%
summarise(Ncase=sum(outcome=='1'),
Ncontrol=sum(outcome=='control')))
```