LARisk: A Package for Life Time Attributable Risk Calculation

The R-package, *LARisk*, to compute lifetime attributable risk (LAR) of radiation-induced cancer can be helpful with enhancement of the flexibility in research of projected risks of radiation-associated cancers. It produces LAR values considering various options or arguments. The package, *LARisk*, can provide a powerful research idea and handle a large size of data fast. In addition, it is possible to compute LAR values by group such as occupation, sex, age group, etc.

Contents

1 Run the functions

Function form In general, the *LAR* function has the form as below. Note that the values in the box are the default setting.

```
LAR(data=data,weight_site="no",weight_value=0,
    current=as.numeric(substr(Sys.time()1,4)),
    sim=300,seed=99,basepy=100000,DDREF=TRUE,
    excel=FALSE,filename,ci=0.9,
    changedata=FALSE,dbaseline="dbaseline",dincidence="dincidence",
    rounddigit=4)
```

2 Description of basic arguments (options)

We can manipulate any of the arguments in **Table1**. The following **Table1** is the simple description of the arguments used in the *LAR* function.

Table 1: Arguments in 'LAR'

data	data frame with id/sex/birth/dosedist/dose1/dose2/dose3/site/exposure_rate
weight_site	cancer sites to give weights possible sites : leukemia, solid cancers except breast, thyroid, gallbladder and brain/CNS
weight_value	a value between 0 and 1 which is a weight on ERR model
current	a current year, default value = (year of the system time) be cautious to put as 4 digits like 2018
sim	the iteration of simulation
seed	a seed number
basepy	a base person year
DDREF	logical, whether to apply the dose and dose-rate effectiveness factor
excel	logical, whether to export the result in a csv file
filename	file name of the csv file to save
ci	confidence level between 0 and 1 of the confidence interval
changedata	logical, whether to change the data of lifetime table and incidence rate
dbaseline	a path or data frame of the new lifetime table
dincidence	a path or data frame of the new incidence rate table
rounddigit	the number of decimal points to print

2.1 data

The data to be put in the *LAR* function should include some prerequisite information such as **id**, **sex** and birth year(s) of person (people) (**birth**), distributions of exposed dose (**dosedist**), exposed radiation dose (fixed value) or parameters of dose distributions (random) (**dose1, dose2, dose3**), sites where exposed (**site**), and exposure rate (**exposure_rate**).

Table 2: Example of the data form

id	sex	birth	dosedist	exposure	dose1	dose2	dose3	site	exposure_rate
101	male	1967	lognormal	1991	3.8167	1.404	0	thyroid	chronic
102	female	1985	fixedvalue	1999	2.1101	0	0	ovary	acute
102	female	1985	normal	2000	1.9822	1.156	0	lung	chronic

Basically, the arguments should be written as expressed. **id** should be composed of numbers or alphabets. Also, data with the same **id** should have the same **birth** (year) which is logically trivial. In the same way, since event dates of exposure must occur after the birth date, **exposure** (year) should be larger than **birth** (year).

```
> data <- data.frame(id=121, birth=c(1900), exposure=c(1980),
         dosedist=c("fixedvalue"), dose1=c(10),dose2=c(0),dose3=c(0),
         sex=c("male"), site=c("colon"), exposure_rate=c("acute"))
> LAR(data)
 $LAR
        Lower   Mean  Upper         LBR    LFR
 colon 0.4976 1.3691 2.7281    814.7624      0
 solid 0.4976 1.3691 2.7281    814.7624      0
 total 0.4976 1.3691 2.7281    814.7624      0

 $Future_LAR
       Lower Mean Upper    BFR     TFR
 colon     0    0     0      0       0
 solid     0    0     0      0       0
 total     0    0     0      0       0
```

[Case that the current age is over 100]

The maximum age in the function is set as 100 years old. If the data contains a birth year which makes attained age(= current - birth) over 100, the result has no useful value. Since the baseline at exposure age is drawn without attained age (which is over 100 now), *LBR* is estimated, while *BFR* using attained age is not estimated. Therefore, the value in *BFR, TFR* and *LFR* is provided as zero.

In *dosedist*, we insert the distribution of the exposed dose. It can have *fixedvalue, normal, lognormal, triangular, logtriangular, uniform, loguniform*. For each distribution, it demands essential parameters of itself. Therefore, for instance, if the exposed dose has a normal distribution with the mean of 2.3 and the standard deviation of 0.8, **dose1** = 2.3, **dose2** = 0.8, and **dose3** =0. If the dose has the fixed value of 3.2, **dose1** = 3.2 and both **dose2**=0 and **dose3**=0.

Table 3: Options for dose distributions

dose distribution	dose1	dose2	dose3
fixedvalue (a)	a	0	0
normal $N(a, b^2)$	a	b	0
lognormal $lognormal(a, b^2)$	a	b	0
triangular $T(a, b, c)$	a	b	c
logtriangular $logT(a, b, c)$	a	b	c
uniform $U(a, b)$	a	b	0
loguniform $logU(a, b)$	a	b	0

For **site**, we put the irradiated organ site or cancer-site. The *LAR* function estimates cases with the site as *stomach, colon, liver, lung, breast, ovary, uterus, prostate, bladder, brain/CNS, thyroid, oral, oesophagus, rectum, gallbladder, pancreas, kidney, leukemia, and remainder*. In addition, in **exposure**, put the exposure rate of radiation as *chronic* for chronic exposure and *acute* for acute exposure.

2.2 weight_site, weight_value

These options are used to estimate LAR values through the weighted average of ERR and EAR models. *weight_site* is to decide organ and *weight_value* is for a specific value of the weight. For example, cancers of breast and thyroid only have weight of 1 for EAR or ERR model, respectively (see **Table 4**).

Table 4: Default weights

Cancer site	ERR	EAR	weight
Most cancer	70%	30%	0.7
Lung	30%	70%	0.3
Breast	0%	100%	0.0
Thyroid	100%	0%	1.0
Gallbladder	100%	0%	1.0
Brain/CNS	100%	0%	1.0

2.3 current

current is the year time to set as the moment of estimation. The default value is set as the system time of the computer. Since it is considered as the current year, we can change the option if we want to set the current time into other years. It recommends that the value should be in form of year in 4 digits.

2.4 sim, seed, and basepy

sim is the number of simulation runs. Note that as *sim* goes larger, the computation time takes longer although the simulation variation is getting smaller. *sim*=300 is recommended considering both computing time and simulation variation. *seed* is the random seed number. As long as the same seed number is provided, we obtain the same result in anytime. *basepy* is the baseline person year.

2.5 DDREF

DDREF, dose and dose-rate effectiveness factor, is the logical option to select whether or not to consider DDREF in the LAR calculation. The value DDREF is to modify the effect of exposure to low-dose. The value of DDREF is considered differently as per exposure rate.

2.6 excel, and filename

One of advantages of this package is that we can get the result in a csv file. Thus, we can save the result in file by assigning *full-directory* filename. In other words, it assigns *a path* where to save the file simultaneously. It is done with options *excel* and *filename*, which will be described in detail later.

```
#not necessary to input the part staring with '#'
#path "D:\user\LARisk\save\\"

#assign location.1
> LAR(data,excel=TRUE, filename="D:\\user\\LARisk\\\save\\out1.csv")

#not assign location.2
> LAR(data,excel=TRUE)
"Error in LAR(data, excel=TRUE) : 'filename' must be  specified"
```

2.7 ci

ci is the level of significance to provide the confidence interval of LAR values, expressed in number between 0 and 1. The default value is 0.9, in other words, the *LAR* function provides the confidence interval at 90% level of significance in default setting.

2.8 changedata, dbaseline, and dincidence

changedata is the logical option where it is TRUE if the new lifetime table and the incidence rate table is to be applied. *A path* of the table file is provided in both *dbaseline* and *dincidence*. See **1.1.6** to input *a path*.

2.9 rounddigit

rounddigit is the number of decimal points to print values. The initial set is to show the value in four decimal places. It is possible that the result appears in different digit with *rounddigit* setting when the value itself ends at shorter digit points.

3 Description of outputs

The *LAR* and *LAR_batch* give three kinds of estimates such as LAR, future LAR, and lifetime baseline risks. The LAR values are given with mean and confidence limits (lower and upper) for each cancer site and for each id, and lifetime baseline risk at exposed age (**LBR**) or baseline risk at attained (current) age (**BFR**), and lifetime fractional risk (**LFR**), and total future risk(**TFR**). Within sites, two additional rows are included in the result. *solid* is the summation of all LAR values for solid cancer sites, and *total* is the summation of LAR values for all cancer sites including leukemia. For instance, if a person is exposed to colon, liver and also has a risk to attain leukemia, *colon and liver* are considered in *solid*, and in *total* including leukemia.

```
> data <- data.frame(id=rep(122,3), birth=rep(1987,3),
                     exposure=c(2007,2008,2009),
                     dosedist=c("fixedvalue","normal","lognormal"),
                     dose1=c(10,3.3,4.7),dose2=c(0,2.1,2.5),dose3=c(0,0,0),
                     sex=rep("male",3), site=c("colon"),
                     exposure_rate=c("acute","chronic","chronic"))
```

```
> data
   id birth exposure  dosedist  dose1 dose2 dose3  sex     site
1 122 1987     2007 fixedvalue  10.0    0.0     0 male     colon
2 122 1987     2008     normal   3.3    2.1     0 male     liver
3 122 1987     2009  lognormal   4.7    2.5     0 male leukemia
   exposure_rate
1          acute
2        chronic
3        chronic

> LAR(data)
 $LAR
        Lower    Mean   Upper       LBR      LFR
 colon 18.1258 39.9514 73.3072  4194.876   0.9524
 solid 18.1258 39.9514 73.3072  4194.876   0.9524
 total 18.1258 39.9514 73.3072  4194.876   0.9524

 $Future_LAR
        Lower    Mean   Upper      BFR      TFR
 colon 18.0306 39.7699 72.9673 4179.677 4219.359
 solid 18.0306 39.7699 72.9673 4179.677 4219.359
 total 18.0306 39.7699 72.9673 4179.677 4219.359
```

4 Functions in the 'LARisk' package

In this chapter, there will be some examples for using the **LAR** package and introduce some useful arguments and methods to estimate lifetime attributable risks of radiation-associated cancers.

4.1 function 'LAR'

First of all, start with single id and single exposure. Let a female person with id 'a100' who was born in 1965 be exposed to radiation at 3.8 dose (mGy, or mSv) in chronic rate in 1995. The exposed site is breast and the exposed dose is a fixed value. Then we make it like below in R.

```
> data1 <- data.frame(id="a100", birth=c(1965),
                      exposure=c(1995),
                      dosedist=c("fixedvalue"),
                      dose1=c(3.8),dose2=c(0),dose3=c(0),
                      sex=c("female"), site=c("breast"),
                      exposure_rate=c("chronic"))

> data1
    id birth exposure  dosedist  dose1 dose2 dose3   sex   site
```

```
1 a100 1965     1995 fixedvalue  3.8    0     0  female  breast

  exposure_rate
1       chronic
```

Note that if any options include some characters, like 'a', we should write it using quotation marks. Otherwise, namely when id is only made up with numbers, it is possible to write without quotation marks. For a single id, it is sufficient to use the *LAR* function and the result is like below.

```
> LAR(data1)
  $LAR
          Lower    Mean   Upper      LBR    LFR
  breast 6.2605 11.4438 18.2795 4354.486 0.2628
  solid  6.2605 11.4438 18.2795 4354.486 0.2628
  total  6.2605 11.4438 18.2795 4354.486 0.2628

  $Future_LAR
          Lower  Mean   Upper      BFR      TFR
  breast 5.0172 9.173 14.5482 2186.134 2195.087
  solid  5.0172 9.173 14.5482 2186.134 2195.087
  total  5.0172 9.173 14.5482 2186.134 2195.087
```

The result shows *LAR and Future_LAR* for each site, all-solid cancers and all cancers including leukemia. In the list(*LAR*, *Future_LAR*), each contains lower bound, mean, and upper bound of (future) lifetime attributable risk, and LBR & LFR or BFR & TFR respectively.

Next, suppose that the 'a100' was exposed to radiation again. The second exposure is for being exposed at dose 2.51 in acute rate in 1998, which results in leukemia. Then the data changes like 'data2' in the table. Also, the result is like below.

```
> data2 <- data.frame(  id=rep("a100",2),
                        birth=rep(1965,2),
                        exposure=c(1995,1998),
                        dosedist=rep("fixedvalue",2),
                        dose1=c(3.8,2.51),dose2=c(0,0),dose3=c(0,0),
                        sex=rep("female",2), site=c("breast","leukemia"),
                        exposure_rate=c("chronic","acute"))

> data2
   id birth exposure  dosedist  dose1 dose2 dose3  sex     site
1 a100 1965    1995 fixedvalue   3.80    0     0 female    breast
2 a100 1965    1998 fixedvalue   2.51    2     0 female leukemia
```

```
  exposure_rate
1       chronic
2         acute

> LAR(data2)
 $LAR
           Lower    Mean   Upper       LBR    LFR
 breast   6.2605 11.4438 18.2795 4354.4864 0.2628
 leukemia 0.4993  1.1348  2.5792  385.7546 0.2942
 solid    6.2605 11.4438 18.2795 4354.4864 0.2628
 total    7.4047 12.5787 19.4251 4740.2410 0.2654

 $Future_LAR
           Lower   Mean   Upper       BFR       TFR
 breast   4.8970 8.9533 14.1998 2186.1337 2195.0870
 leukemia 0.3258 0.7420  1.6898  304.9944  305.7364
 solid    4.8970 8.9533 14.1998 2186.1337 2195.0870
 total    5.6390 9.6953 14.9418 2491.1281 2500.8234
```

In fact, if we want to compute the LAR values for only one person's data, i.e with single id, it is not necessary to input **id**. In other words, it does not make any difference in results with data2 and data2_1 as below.

```
> data2_1 <- data.frame(
                        birth=rep(1965,2),
                        exposure=c(1995,1998),
                        dosedist=rep("fixedvalue",2),
                        dose1=c(3.8,2.51),dose2=c(0,0),dose3=c(0,0),
                        sex=rep("female",2), site=c("breast","leukemia"),
                        exposure_rate=c("chronic","acute"))
```

4.2 function 'LAR_batch'

Suppose that we have 'data3' with id 'a100' and 'a101' whose exposure history is like in the table below.

```
> data3 <- data.frame(  id=c("a100","a101","a101"),
                birth=c(1965,1980,1980),
                exposure=c(1995,1999,2002),
                dosedist=c("fixedvalue","normal","triangular"),
                dose1=c(3.8,4.2,1.2),dose2=c(0,1.05,2.3),dose3=c(0,0,2.9),
                sex=c("female","male","male"),
                site=c("breast","colon","bladder"),
                exposure_rate=c("chronic","chronic","acute"))
```

```
> data3
   id birth exposure  dosedist  dose1 dose2 dose3  sex    site
1 a100 1965     1995 fixedvalue   3.8     0    0 female  breast
2 a101 1980     1999     normal   4.2  1.05    0   male   colon
3 a101 1980     2002 triangular   1.2  2.30  2.9   male bladder
   exposure_rate
1        chronic
2        chronic
3          acute
```

In contrast to a previous case, with several ids, we use the 'LAR_batch' function to estimate lifetime attributable risk of radiation-related cancers

```
> LAR_batch(data3)
 $a100
 $a100$LAR
         Lower    Mean   Upper      LBR    LFR
 breast 6.2605 11.4438 18.2795 4354.486 0.2628
 solid  6.2605 11.4438 18.2795 4354.486 0.2628
 total  6.2605 11.4438 18.2795 4354.486 0.2628

 $a100$Future_LAR
        Lower   Mean   Upper      BFR      TFR
 breast 4.897 8.9533 14.1998 2186.134 2195.087
 solid  4.897 8.9533 14.1998 2186.134 2195.087
 total  4.897 8.9533 14.1998 2186.134 2195.087


 $a101
 $a101$LAR
          Lower    Mean   Upper      LBR    LFR
 colon   4.1266  9.1429 16.6405 4195.173 0.2179
 bladder 0.5552  1.6205  3.4384 1612.994 0.1005
 solid   5.4604 10.7634 18.3455 5808.167 0.1853
 total   5.4604 10.7634 18.3455 5808.167 0.1853

 $a101$Future_LAR
          Lower    Mean   Upper      BFR      TFR
 colon   3.9966  8.8643 16.1838 4149.810 4158.674
 bladder 0.5486  1.6018  3.3905 1601.216 1602.818
 solid   5.3189 10.4661 17.8036 5751.026 5761.492
 total   5.3189 10.4661 17.8036 5751.026 5761.492
```

The result from the *LAR_batch* function includes estimates of *LAR and Future_LAR* for each id.

4.3 function 'LAR_summary'

The function *LAR_summary* is to summarize LAR values according to groups (or data) we have. It offers grouped LAR values, grouped future LAR values and grouped baseline risk values based on values of simulation for each **id**. It takes mean of each LAR values for each group, which makes new LAR values, and then this new LAR values are taken to present summarized LAR values for each group.

Figure 1 illustrates the calculation method to compute summary of grouped LAR values. The result of this function is similar(almost same) with the function *LAR*. However, the result suggests summarization of the LAR values for certain groups, not the individual.

```
> LAR_summary(data3)
 $LAR
          Lower    Mean   Upper       LBR
 breast  3.1303  5.7219  9.1397 2177.2432
 colon   2.0633  4.5714  8.3202 2097.5866
 bladder 0.2776  0.8102  1.7192  806.4968
 solid   7.2659 11.1037 16.1097 5081.3266
 total   7.2659 11.1037 16.1097 5081.3266

 $Future_LAR
          Lower   Mean   Upper       BFR       TFR
 breast  2.4485 4.4767  7.0999 1093.0668 1097.5435
 colon   1.9983 4.4322  8.0919 2074.9050 2079.3372
 bladder 0.2743 0.8009  1.6952  800.6079  801.4088
 solid   6.3537 9.7097 14.0202 3968.5798 3978.2895
 total   6.3537 9.7097 14.0202 3968.5798 3978.2895
```

5 A change of arguments

5.1 weight_site, and weight_value

At first, we can manipulate weights with data2. When we are willing to apply weights for the LAR computation, it is needed to mention which organ site you choose to give weights and how much weight (i.e. the value of weight) will be applied. Given a weight on *colone* with value 0.4, the following is how to input the command.

```
LAR(data2, weight_site = c("colon"),weight_value=0.4)
```

5.2 current, seed, basepy, DDREF, and rounddigit

Now, using data2, we would work on options including current, seed, basepy, and DDREF. As "current" is related to the moment of the computation, it should be valid year which is later than **birth** and **exposure**.

```
LAR(data2, current=2005)
```

It represents that changing the current time affects future lifetime cancer risk, and future baseline risk. In the same way, we can work with other options. The next tables are for comparison of the results dealing with seed, basepy and DDREF respectively.

```
LAR(data2,seed=369)
LAR(data2,basepy=1000000)
LAR(data2,DDREF=FALSE)
LAR(data2,rounddigit=2)
```

It is also possible to change several options simultaneously. There are some examples.

```
LAR(data2,seed=100,current=2015,rounddigit=3)
LAR(data2,basepy=10000000,DDREF=FALSE,seed=963)
```

5.3 lifetime table, and incidence data

In *LARisk* package, we use the fundamental datasets, 'Lifetime Table' and 'Incidence data', which are made in 2010 in Korea. However, these data can be switched to other data. Here, we suggest the form which should be followed.

```
Age    Prob_d_m        Prob_d_f
0       0.00320         0.00780
1       0.00023         0.00026
2       0.00017         0.00016
3       0.00013         0.00011
4       0.00011         0.00009
5       0.00010         0.00008
```

The columns of the Lifetime table are consisted of **Age**, **Prob_d_m**, and **Prob_d_f**. **Prob_d_m**, and **Prob_d_f** are the probabilities of death of male and female, respectively.

```
Site   Age     Rate_m  Rate_f
oral    0          0       0
oral    1          0       0
oral    2          0       0
oral    3          0       0
oral    4          0       0
oral    5        0.2       0
oral    6        0.2       0
oral    7        0.2       0
oral    8        0.2       0
```

The columns of incidence data consist of cancer **Site**, **Age**, **Rate_m**, and **Rate_f**. **Rate_m** and **Rate_f** are incidence rates of some cancer of male and female, respectively. Note that the data should have the range of values from age 0 100 one by one. It is also possible to take the path of data directly to the option.

Hence, in incidence data, cancer sites have to be in appointed order. The order refers to following **Table 5**.

Table 5: Order of sites in the incidence data& the lifetime table

1	oral	6	gallbladder	11	ovary	16	brain/cns
2	oesophagus	7	pancreas	12	uterus	17	thyroid
3	stomach	8	liver	13	prostate	18	remainder
4	colon	9	lung	14	bladder	19	leukemia
5	rectum	10	breast	15	kidney		

After constructing the datasets properly, put lifetime table into **dbaseline** and incidence data into **dincidence**. If there are some mistakes, the program gives error messages(see **section 8.2**).

6 Additional arguments

Note that we can put additional information in data with essential elements for LAR. If we know some functions in R, some useful factors of subjects can be used to compare LAR values in groups which have disparate characteristics like occupations, sex etc. Also, it is possible to use essential elements like sex in

different way case by case. For introduction, here we present two examples which are to estimate LAR values with divided data according to certain criteria.

In previous chapters, we input data in R manually using several R functions. However, it is not effective when dealing with large data set, because most of data are managed in an excel or csv file. Therefore, we will load a data with essential elements and occupation factors after inputting the data in a csv file. First, install the package *xlsx*. This package is used to draw a xlsx file into R.

```
install.packages("xlsx")
```

After installing the package, it is necessary to command *library( )* to use the package we have installed. If the package is loaded, then use *read.xlsx* function to draw the file. Note that the data has no blank, namely, if there are no doses in **dose2** and **dose3**, it is recommended to input 0, not remained as blank.

Table 6: Read a xlsx file

read.xlsx (xlsxFile, header=TRUE, sheetIndex=" ")	
xlsxFile	the path to the file to load
header	a logical, indicating whether the first row corresponding to the first element of the rowIndex vector contains the names of the arguments
sheetIndex	a number representing the sheet index in the workbook

```
> library(xlsx)
> data4 <- read.xlsx("D:\\input\\example1.xlsx",header=T,sheetIndex=1)
```

Then, we can load the data we want in R directly.

```
> data4
      id      sex     occup   birth     dosedist      exposure   dose1
1     a11     male      1     1990    fixedvalue         1999     7.8
2     a12     male      2     1991        normal         1998     5.2
3     a13     male      3     1992        normal         2003     4.7
4     a14     male      1     1993    triangular         2008     2.3
5     a15     male      2     1994    fixedvalue         2004     5.2
6     a16   female      3     1995    lognormal          2012     1.9
7     a17   female      1     1996    fixedvalue         2013     4.2
8     a18   female      2     1997    lognormal          2000     2.8
9     a19   female      3     1998    triangular         2005     3.5
```

```
10      a20      female     1      1999     fixedvalue          2012      6.4

        dose2    dose3    site     exposure_rate
1       0.0      0.0      oral           acute
2       2.3      0.0      bladder        acute
3       1.1      0.0      colon          chronic
4       3.3      5.3      liver          chronic
5       0.0      0.0      stomach        chronic
6       0.3      0.0      liver          acute
7       0.0      0.0      stomach        chronic
8       1.7      0.0      breast         chronic
9       4.6      6.7      thyroid        acute
10      0.0      0.0      ovary          acute
```

In similar way, we can load a csv file. Note that the option of the function is different.

```
> data4_1<-read.csv("D:\\input\\example2.csv",header=T,stringAsFactors=F)
```

Table 7: Read a csv file

read.csv(file,header=TRUE,stringAsFactors=FALSE)	
file	the path to the file to load
header	a logical. whether the file contains the names of the arguments as its first line
stringAsFactors	a logical. whether character vectors be converted to factors

If you want to load the file faster, install the another package *readr* which helps to load a worksheet into R as data frame. It offers a R function *read_csv* in *readr* to load a csv file.

```
install.packages("readr")
library(readr)
data4_2 <- read_csv("D:\\input\\example2.csv",col_names=TRUE)
```

6.1 save the result in a csv file

Some options in LAR functions are related to extract the result in a csv file with some simple tasks. It goes with *excel* and *filename*. Understandable with these literal name, *excel* is for whether or not to save the result in a csv file and *filename* is the direction where to save the file and its name. The initial option is not to export the result as a csv file. Hence, if you only set *excel=TRUE*(same with

excel=T) without indicating *filename*, it occurs error. There are some examples with data2 and data3. These examples show the difference in a csv file of the *LAR* function, and that of the *LAR_batch* function.

```
LAR(data2,excel=T, filename="D:\\output\\for data2.csv")
```

Then, after finishing the calculation, we can find the file at the location we indicate. Note that if there exists the csv file which has the same title with *filename*, it would be overlapped.

```
LAR(data2,excel=T,filename="D:\\output\\for data2.csv")
LAR(data1,excel=T,filename="D:\\output\\for data2.csv")
#delete initial for data2.csv
```

Therefore, before deciding *filename*, be cautious to check whether or not the name is duplicated. In the same way as above, the result from the *LAR_batch* function can be saved as a csv file.

```
LAR_batch(data3,excel=T,filename="D:\\output\\for data3.csv")
```

The csv file saves the result according to a certain form. The *LAR* function suggests estimates of LAR, future LAR, and future baseline risk. Each of them has three kind of values respectively in vertical way in R console. On the contrary, in the csv file, the values are represented in horizontal way for each organ, all-solid-cancer and all-organ. Despite the case of the *LAR* function is somehow intuitive, the *LAR_batch* function is not simple. We make a space for all organs, and values from the function is put in their own space. Therefore, there are 190 columns including ID column, and the number of rows depends on the number of id's in data. The columns are ordered in **(LAR)-(Future LAR)-(Baseline Risk)-(Total Future Risk)** in general. In LAR and Future LAR, each is made up of lower limit, upper limit, and mean values, and for the Baseline Risk, it is made up of baseline risk of exposed age, baseline risk of attained age, and LFR. The last part is total future risk for each sites. Hence, for each component, there are values of all-organ, all-solid-cancer, and each organ, i.e. 21 elements. So that, the file has somehow wide shape with 210 columns.

id	site	sex	occup	birth	exposrue	dosedist	dose1	dose2	dose3	exposure_rate
a11	oral	male	1	1990	1999	fixedvalue	7.8	0	0	acute
a12	bladder	male	2	1991	1998	normal	5.2	2.3	0	acute
a13	colon	male	3	1992	2003	normal	4.7	1.1	0	chronic
a14	liver	male	1	1993	2008	triangular	2.3	3.3	5.3	chronic
a15	stomach	male	2	1994	2004	fixedvalue	5.2	0	0	chronic
a16	liver	female	3	1995	2012	lognormal	1.9	0.3	0	acute
a17	stomach	female	1	1996	2013	fixedvalue	4.2	0	0	chronic
a18	breast	female	2	1997	2000	lognormal	2.8	1.7	0	chronic
a19	thyroid	female	3	1998	2005	triangular	3.5	4.6	6.7	acute
a20	ovary	female	1	1999	2012	fixedvalue	6.4	0	0	acute

Table 8: data4

7 Applications

7.1 prediction of the LAR values using divided data

This section describes how to divide the data according to some criteria using data4. The data4 includes essential variables and additional variable, which offers occupation type(**occup**) of each **id**.

Start with separating it with **sex**. As we input the elements 'male' or 'female', we can divide the data into two groups by input the code like below.

```
data4_m <- data4[data4$sex=="male",]
```

data4$sex indicates the column 'sex' in data4. == is a operator which judge the left side object(in this case, the column 'sex') is the same with the right side object(in this case, 'male'), which returns logic values.

```
> data4$sex=="male"
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

data4[data4$sex=="male",] selects rows in data4 which corresponds to *TRUE* value in above elements. In other words, it selects rows which have 'male' as **sex** value. *data4_m* <- is for call the male data conveniently, considering it as saving data separately. We can find out that *data4_m* successfully includes only male rows.

```
> data4_m
# A tibble: 5 x 11
     id    site   sex occup birth exposure   dosedist dose1 dose2 dose3
  <chr>   <chr> <chr> <dbl> <dbl>    <dbl>      <chr> <dbl> <dbl> <dbl>
1   a11    oral  male     1  1990     1999 fixedvalue   7.8   0.0   0.0
2   a12 bladder  male     2  1991     1998     normal   5.2   2.3   0.0
3   a13   colon  male     3  1992     2003     normal   4.7   1.1   0.0
4   a14   liver  male     1  1993     2008 triangular   2.3   3.3   5.3
5   a15 stomach  male     2  1994     2004 fixedvalue   5.2   0.0   0.0
```

```
# ... with 1 more variables: exposure_rate <chr>
```

Similarly, the female data can be selected.

```
data4_f <- data4[data4$sex=="female",]
```

Likewise, we can divide the data with **occup** composed of three numbers indicating different occupations which is not the essential variable to compute lifetime attributable risk. Based on occupation information, the data can be divided into three groups.

```
data4_o1 <- data4[data4$occup==1,]
data4_o2 <- data4[data4$occup==2,]
data4_o3 <- data4[data4$occup==3,]
```

The example is the result of and its summary. It shows that *data4_o1* is the data of occupation 1. That is, the following result is the LAR-summary of partial data4, which is only for occupation 1.

```
> data4_o1
 # A tibble: 4 x 11
      id    site    sex occup birth exposure   dosedist dose1 dose2 dose3
   <chr>   <chr>  <chr> <dbl> <dbl>    <dbl>      <chr> <dbl> <dbl> <dbl>
 1   a11    oral   male     1  1990     1999 fixedvalue   7.8   0.0   0.0
 2   a14   liver   male     1  1993     2008 triangular   2.3   3.3   5.3
 3   a17 stomach female     1  1996     2013 fixedvalue   4.2   0.0   0.0
 4   a20   ovary female     1  1999     2012 fixedvalue   6.4   0.0   0.0
 # ... with 1 more variables: exposure_rate <chr>

> LAR_summary(data4_o1)
  $LAR
            Lower   Mean  Upper       LBR
  oral    0.1403 0.6586 1.4382  215.4442
  liver   0.5994 1.3465 2.5295 1280.4019
  stomach 1.0880 2.0692 3.4621 1189.7401
  ovary   0.1798 0.5702 1.1555  179.4609
  solid   3.1301 4.6446 6.8081 2865.0471
  total   3.1301 4.6446 6.8081 2865.0471

  $Future_LAR
            Lower   Mean  Upper       BFR       TFR
  oral    0.1318 0.6082 1.3204  213.1809  213.7892
  liver   0.5979 1.3428 2.5229 1278.6725 1280.0153
  stomach 1.0880 2.0684 3.4621 1184.6318 1186.7002
  ovary   0.1778 0.5686 1.1555  175.3428  175.9114
  solid   3.0960 4.5880 6.7400 2851.8280 2856.4160
  total   3.0960 4.5880 6.7400 2851.8280 2856.4160
```

Applying this method, we can gain results of certain groups with dividing data in R program without additionally re-ordering data.

8 Error statements

8.1 data format

8.1.1 consistence in id and birth

To run the functions, the data have to meet some conditions. One of them is *to have consistence in **id** and **birth***. In other words, each id corresponds to *only one* birth year.

The example is the case with the data2 which is mistyped in **birth** as 1940, which has to be 1965.

```
> data2_1
   id birth exposure  dosedist  dose1 dose2 dose3   sex    site
1 a100 1965     1955 fixedvalue  3.80    0    0  female  breast
2 a100 1940     1998 fixedvalue  2.51    0    0  female  leaukaemia

   exposure_rate
1        chronic
2          acute

> LAR(data2_1)
       [1] "Error:Pairs of 'birth' is inconsistent"
```

8.1.2 order of birth and exposure

One of them is *to confirm that **birth** is earlier than **exposure*** .

The next example is the case with the data1 which is mistyped in **exposure** as 1955, the original value of which is 1995, later than birth year.

```
> data1_2
   id birth exposure  dosedist  dose1 dose2 dose3   sex    site
1 a100 1965     1955 fixedvalue  3.80    0    0  female  breast

   exposure_rate
1        chronic

> LAR(data1_1)
"Error:'exposure' values are improper"
```

8.1.3 correspondence of site and sex

Some cancers are specific to certain sex. The case for 'male' should not include *uterus, breast or ovary*, and that for 'female' should not include *prostate* as site information. The below is error statements for each case.

```
"Error:'male' cannot calculate about uterus, breast or ovary"
"Error:'female' cannot calculate about prostate"
```

8.1.4 components of sex

It recommends to input **sex** value as *male* or *female*. If **sex** is composed of otherwise, "1" or "2" as data1_2, the error occurs.

```
> data1_3
   id birth exposure  dosedist  dose1 dose2 dose3  sex    site
1 a100 1965     1995 fixedvalue  3.80    0     0     2  breast
   exposure_rate
1        chronic

> LAR(data1_2)
"Error:'sex' has invalid component"
```

8.1.5 expression of site

It is necessary to put cancer sites as one of the followings. If other values are included, the error occurs. It doesn't matter to put it in capitals or lowercase characters.

Table 9: **Site** components

oral	gallbladder	ovary	brain/cns
oesophagus	pancreas	uterus	thyroid
stomach	liver	prostate	remainder
colon	lung	bladder	leukemia
rectum	breast	kidney	

```
"Error:'site' has invalid component"
```

8.1.6 expression of dosedist

It is necessary to put dose distribution as one of the followings. If other values are included, the error occurs. It doesn't matter to put it in capitals or lowercase characters.

Table 10: **Dosedist** components

fixedvalue	
normal	lognormal
uniform	loguniform
triangular	logtriangular

```
"Error:'dosedist' has invalid component"
```

8.1.7 expression of exposure_rate

It is necessary to put rate of exposure as *chronic* or *acute*. If other values are included, the error occurs. It doesn't matter to put it in capitals or lowercase characters.

```
"Error:'exposure_rate' has invalid component"
```

8.1.8 insert filename when excel=TRUE

filename should be assiged to get the result in csv file.

```
"Error in LAR(data, excel=TRUE) : 'filename' must be specified"
"Error in LAR_batch(data, excel=TRUE) : 'filename' must be specified"
"Error in LAR_summary(data, excel=TRUE) : 'filename' must be specified"
```

8.2 lifetime table, and incidence data

8.2.1 insert replacement for changedata

If the option **chagedata** is set as *TRUE*(of *T*), we should input both lifetime table and incidence data for replacement in **dbaseline** and **dincidence** respectively.

```
"Error : Put the data."
```

8.2.2 lifetime table, and incidence data format

In **section 5.3**, we describe how to change the fundamental datasets. Also, we suggest the formation of them. If you don't consider it, there would be error. A valid lifetime table is composed of 101 rows and 3 columns and a valid incidence data is composed of 1919 rows and 4 columns.

```
"Error : Put the baseline data in the correct format."
"Error : Put the incidence data in the correct format."
```

9 Index

List of Tables