

Solution to the practicals - measuring disease frequency and association

Epidemiological methods in medical research

19 January 2023

During the exercise we will use the following R package:

```
library(Epi) ## effx function
library(survival) ## survfit function
```

and the dataset

```
bissau <- read.table(
  file = "https://bozenne.github.io/doc/Teaching/bissau.txt",
  header=TRUE
)
## only keep relevant column
bissau <- bissau[,c("id","fuptime","fupstatus","bcg")]
## convert categorical variable from numeric to factor
bissau$id <- as.factor(bissau$id)
bissau$fupstatus <- as.factor(bissau$fupstatus)
bissau$bcg <- as.factor(bissau$bcg)
```

Part A: by hand calculation

In this section we use the following dataset:

```
bissau.10 <- rbind(bissau[bissau$bcg=="no",][c(3:5,15,182),],  
                  bissau[bissau$bcg=="yes",][c(1,25:28),])  
bissau.10
```

	id	fuptime	fupstatus	bcg
20	20	183	censored	no
25	25	147	dead	no
31	31	183	censored	no
59	59	183	censored	no
526	526	177	dead	no
1	1	65	dead	yes
29	29	183	censored	yes
30	30	183	censored	yes
32	32	183	censored	yes
33	33	183	censored	yes

1. To obtain the first table, we count the number of lines with "dead" and "censored" per bcg group or in total:

```
t22.10 <- table(bcg = bissau.10$bcg, status = bissau.10$fupstatus)  
t22.10t <- rbind(t22.10, total = colSums(t22.10))  
t22.10t
```

	censored	dead
no	3	2
yes	4	1
total	7	3

For the second table, the first two columns can be retrieved from the previous table. The last column is obtained by summing the `fuptime` per bcg group or in total:

```
t23.10 <- xtabs(cbind("n" = 1,  
                     "death" = fupstatus=="dead",  
                     "person-day" = fuptime) ~ bcg,  
               data = bissau.10)  
t23.10t <- addmargins(t23.10, margin = 1)  
t23.10t
```

bcg	n	death	person-day
no	5	2	873
yes	5	1	797
Sum	10	3	1670

- The risk of death is obtained by dividing the number of deaths by the number of children (e.g. $2/5=0.4$).

The odds is the risk of death divided by 1 minus the risk of death (e.g. $0.4/(1-0.4)=0.67\dots$).

The rate per day is the number of death divided by the number of person-days (e.g. $2/873=0.0023\dots$).

The rate per year is the number of death divided by the number of person-years where the number of person-years is the number of person-days divided by 365.25 (e.g. $2/(873/365.25)=0.84\dots$).

```
D <- t22.10t[, "dead"]
n <- rowSums(t22.10t)
Y <- t23.10t[, "person-day"]
estimate.10 <- rbind(risk = D/n,
                     odds = (D/n)/(1-D/n),
                     "rate (per day)" = D/Y,
                     "rate (per year)" = D/(Y/365.25))
colnames(estimate.10) <- paste0("bcg ", colnames(estimate.10))
estimate.10
```

	bcg no	bcg yes	bcg total
risk	0.400000000	0.200000000	0.300000000
odds	0.666666667	0.250000000	0.428571429
rate (per day)	0.002290951	0.001254705	0.001796407
rate (per year)	0.836769759	0.458281054	0.656137725

- Risk and rates point estimates are lower in the vaccinated group compared to the non-vaccinating group, pointing toward a protective vaccine effect. The point estimate of the pooled sample lies in between the point estimate of each vaccination group.

- There are several limitations:

- we are working on a very small sample, which may not be representative of the whole population. Moreover, the estimates we get are probably very uncertain.
- we have not computed the uncertainty associated with the estimates, so it is unclear if the observed difference are due to the vaccine or due to sampling variability (i.e. chance).
- the study was not described as randomized so the vaccinated and un-vaccinated children may not be comparable, e.g. vaccinated may have easier access to healthcare. So it is unclear if the observed difference are due to the vaccine or these external factors (confounding).

5. We can estimate the incidence rate (per day) from the full population using a similar approach:

```
t23 <- xtabs(cbind("n" = 1,
                  "death" = fupstatus=="dead",
                  "person-day" = fuptime) ~ bcg,
            data = bissau)
t23
```

```
bcg      n  death person-day
no    1973    97    325258
yes   3301   125    554929
```

```
t23[, "death"]/t23[, "person-day"]
```

```
no      yes
0.0002982248 0.0002252540
```

A large proportion of children left the study (i.e. was right-censored) before 183 days:

```
100*mean((bissau$fuptime<183)*(bissau$fupstatus=="censored"))
```

```
[1] 47.47819
```

so we cannot directly compute the risk on the whole cohort, as assuming that none of the children who left the studied dies is an unrealistic assumption.

6. Among the first children there are no ties. To obtain the risk we:

- first calculate the number at risk at each timepoint and add it to the dataset

```
bissau.order <- bissau[order(bissau$fuptime),]
bissau.orderY8 <- bissau.order[bissau.order$bcg=="yes",][1:8,]
bissau.orderY8$atRisk <- sum(bissau.order$bcg=="yes") - 0:7
bissau.orderY8
```

```
id fuptime fupstatus bcg atRisk
2876 2876    2  censored yes  3301
89    89    4  censored yes  3300
1908 1908    5  censored yes  3299
2551 2551    6    dead yes  3298
2786 2786    9  censored yes  3297
1344 1344   12    dead yes  3296
598   598   15  censored yes  3295
3736 3736   16    dead yes  3294
```

- compute the hazard rates:

```
lambda <- (bissau.orderY8$fupstatus=="dead")/bissau.orderY8$atRisk
print(lambda, digits = 3)
```

```
[1] 0.000000 0.000000 0.000000 0.000303 0.000000 0.000303 0.000000 0.000304
```

- deduce the risk

```
M.risk <- rbind(time = bissau.orderY8$fuptime,
               risk = 1 - cumprod(1-lambda))
print(M.risk, digits = 4)
```

```
      [,1] [,2] [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
time      2    4    5 6.0000000 9.0000000 1.200e+01 1.500e+01 1.600e+01
risk      0    0    0 0.0003032 0.0003032 6.065e-04 6.065e-04 9.099e-04
```

So the risk at 5 days is 0, at 10 days is 0.0003 (i.e. the same as at 9 days), and the risk at 15 days is 0.0006.

Part B: using dedicated functions of a statistical software

Incidence rate

7. The 2 by 3 table can be obtained using

```
t23 <- xtabs(cbind("n" = 1,
                  "death" = fupstatus=="dead",
                  "person-day" = fuptime) ~ bcg,
            data = bissau)
t23
```

```
bcg      n  death person-day
no      1973    97    325258
yes     3301   125    554929
```

The incidence rate can be deduced by dividing the number of deaths per the person-day (or person-year divided by 365.25):

```
D <- t23[, "death"]
Y.day <- t23[, "person-day"]
Y.year <- t23[, "person-day"] / 365.25

M.I <- cbind("rate per day" = D/Y.day,
            "rate per year" = D/Y.year)
M.I
```

```
      rate per day rate per year
no  0.0002982248    0.10892661
yes 0.0002252540    0.08227404
```

The rate per person day looks very small but it is not. The ratio is a scaled quantity so it is only meaningful with a scale and a day is a very short time interval. A rate of 0.109 per person year is equivalent to 1.09 per 10-person year i.e. we would expect about 1 child out of 10 to die every year if not vaccinated. This interpretation is here to illustrate what the value is, as we only have 6-month follow-up - we would need to assume a constant rate over time for this interpretation to hold.

8. The incidence difference and ratio can be obtained by subtracting or dividing the previous estimates, e.g.:

```
setNames(M.I[2,] - M.I[1,],
        c("rate difference per day", "rate difference per year")
        )
```

```
rate difference per day rate difference per year
      -7.297075e-05      -2.665257e-02
```

```
setNames(M.I[2,] / M.I[1,],
        c("rate ratio", "rate ratio")
        )
```

```
rate ratio rate ratio
0.7553163 0.7553163
```

Note that the rate difference keeps its unit (per person.day or per person.year) while the rate ratio is unitless. Confidence intervals can be obtained using dedicated functions:

```
effx(response = bissau$fupstatus=="dead",
      exposure = bissau$bcg,
      fup = bissau$fuptime/365.25, type = "failure", eff = "RD")
```

```
-----  
response      : bissau$fupstatus == "dead"  
type          : failure  
exposure      : bissau$bcg
```

```
bissau$bcg is a factor with levels: no / yes  
baseline is  no  
effects are measured as rate differences  
-----
```

```
effect of bissau$bcg on bissau$fupstatus == "dead"  
number of observations  5274
```

```
      Effect      2.5%      97.5%  
-0.026700 -0.052700 -0.000616
```

```
Test for no effects of exposure on 1 df: p-value= 0.0395  
Der var 50 eller flere advarsler (brug warnings() for at se den første 50)
```

```
effx(response = bissau$fupstatus=="dead",  
      exposure = bissau$bcg,  
      fup = bissau$fuptime/365.25, type = "failure", eff = "RR")
```

```
-----  
response      : bissau$fupstatus == "dead"  
type          : failure  
exposure      : bissau$bcg
```

```
bissau$bcg is a factor with levels: no / yes  
baseline is  no  
effects are measured as rate ratios  
-----
```

```
effect of bissau$bcg on bissau$fupstatus == "dead"  
number of observations  5274
```

```
Effect   2.5%   97.5%  
  0.755   0.579   0.985
```

```
Test for no effects of exposure on 1 df: p-value= 0.0395  
There were 50 or more warnings (use warnings() to see the first 50)
```

The confidence intervals for the rate ratio barely do not overlap 1 (or 0 for the rate difference). So there is some evidence for a vaccine effect. We should keep in mind that the point estimate is

about a 24% reduction in rate, which is not that impressive for a vaccine and the data is compatible with a risk reduction from nearly nothing to 40%.

9. Using a Poisson model we retrieve the same rate ratio:

```
e.Poisson <- glm(fupstatus=="dead" ~ bcg, data = bissau,  
                family = poisson(link = "log"), offset = log(fuptime))  
cbind(estimate = exp(coef(e.Poisson)), exp(confint(e.Poisson)))
```

Waiting for profiling to be done...

	estimate	2.5 %	97.5 %
(Intercept)	0.0002982248	0.0002427451	0.0003615709
bcgyes	0.7553162786	0.5799959555	0.9865547965

An equivalent syntax is:

```
e.PoissonBis <- glm(cbind(fupstatus=="dead",fuptime) ~ bcg,  
                   data = bissau, family = poisreg)  
ci.exp(e.PoissonBis)
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.0002982248	0.0002444092	0.0003638899
bcgyes	0.7553162791	0.5793649166	0.9847035350

Omitting the intercept output the group-specific incidence rates instead of the incidence rate of the reference group and the rate ratio:

```
e2.Poisson <- glm(cbind(fupstatus=="dead",fuptime) ~ 0 + bcg,  
                 data = bissau, family = poisreg)  
ci.exp(e2.Poisson)
```

	exp(Est.)	2.5%	97.5%
bcgno	0.0002982248	0.0002444092	0.0003638899
bcgyes	0.0002252540	0.0001890341	0.0002684140

Incidence rates can be obtained per year by dividing by 365.25:

```
e3.Poisson <- glm(cbind(fupstatus=="dead",fuptime/365.25) ~ 0 + bcg,  
                 data = bissau, family = poisreg)  
ci.exp(e3.Poisson)
```

	exp(Est.)	2.5%	97.5%
bcgno	0.10892661	0.08927365	0.13290604
bcgyes	0.08227404	0.06904509	0.09803763

183-day risk of death:

10. 'Ignoring censoring', we can get an estimate of the risks and risk ratio simply subtracting or dividing the risks:

```
D <- t23[, "death"]
n <- t23[, "n"]
r <- D/n
c(r, ratio = unname(r[2]/r[1]))
```

```
      no      yes      ratio
0.04916371 0.03786731 0.77022895
```

- or using `effx`

```
effx(response = bissau$fupstatus=="dead",
      exposure = bissau$bcg,
      type = "binary", eff = "RR")
```

```
-----
response      : bissau$fupstatus == "dead"
type          : binary
exposure      : bissau$bcg
```

```
bissau$bcg is a factor with levels: no / yes
baseline is  no
effects are measured as relative risk
-----
```

```
effect of bissau$bcg on bissau$fupstatus == "dead"
number of observations  5274
```

```
Effect    2.5%  97.5%
0.770    0.594  0.998
```

```
Test for no effects of exposure on 1 df: p-value= 0.0501
```

- or using `glm`

```
e.logit <- glm(fupstatus ~ bcg, data = bissau,
              family = binomial(link = "log"))
cbind(estimate = exp(coef(e.logit)), exp(confint(e.logit)))
```

Waiting for profiling to be done...

```
              estimate      2.5 %      97.5 %  
(Intercept) 0.04916371 0.04020354 0.05929258  
bcgyes       0.77022897 0.59498657 1.00017261
```

Either way this approach assumes that none of the children who left the study died within 183 days which is not realistic.

11. The computation 'by hand' of the Kaplan Meier estimates was shown in the answer to question 6.

12. We use the `survfit` function to obtain the Kaplan Meier estimates:

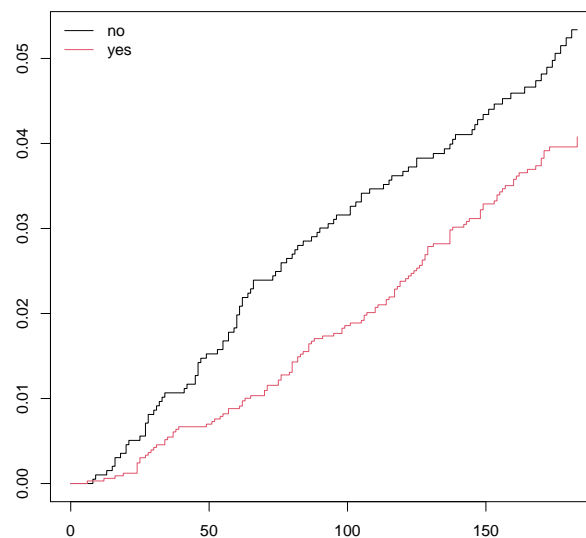
```
e.KM <- survfit(Surv(fuptime,fupstatus) ~ bcg, data = bissau)  
e.KM
```

```
Call: survfit(formula = Surv(fuptime, fupstatus) ~ bcg, data = bissau)
```

```
              n nevent      rmean*  
bcg=no, (s0) 1973      0 178.004315  
bcg=yes, (s0) 3301      0 179.728609  
bcg=no, dead 1973     97   4.995685  
bcg=yes, dead 3301    125   3.271391  
*restricted mean time in state (max time = 183 )
```

For a graphical display:

```
plot(e.KM, col=1:2)  
legend("topleft", legend=levels(bissau$bcg), col=1:2, lty=1, horiz=FALSE, bty='n')
```



Estimates of the risk can be extracted using `summary`:

```
eS.KM <- summary(e.KM, times = c(5,10,15,183))
print(eS.KM, digits = 4)
```

Call: `survfit(formula = Surv(fuptime, fupstatus) ~ bcg, data = bissau)`

```

          bcg=no
time n.risk n.event Pr((s0)) Pr(dead)
  5   1973      0   1.0000 0.000000
 10   1969      2   0.9990 0.001014
 15   1967      2   0.9980 0.002029
183   935     93   0.9466 0.053374
```

```

          bcg=yes
time n.risk n.event Pr((s0)) Pr(dead)
  5   3299      0   1.0000 0.0000000
 10   3296      1   0.9997 0.0003032
 15   3295      1   0.9994 0.0006065
183   1615     123  0.9592 0.0407863
```

For the earlier timepoints, we get the same values as those computed question 6 and 11. As expected the final risk is higher than the one computed 'ignoring censoring', about 0.003 higher.

We can get standard errors and confidence intervals for the risk¹:

```
df.riskKM <- data.frame(time = eS.KM$time,
                        bcg = eS.KM$strata,
                        estimate = eS.KM$pstate[,2],
                        se = eS.KM$std.err[,2],
                        lower = eS.KM$lower[,2],
                        upper = eS.KM$upper[,2])
df.riskKM
```

	time	bcg	estimate	se	lower	upper
1	5	bcg=no	0.0000000000	0.0000000000	0.000000e+00	0.0000000000
2	10	bcg=no	0.0010141988	0.0007167831	2.538271e-04	0.004052361
3	15	bcg=no	0.0020294283	0.0010136844	7.624386e-04	0.005401850
4	183	bcg=no	0.0533744092	0.0053653648	4.382961e-02	0.064997785
5	5	bcg=yes	0.0000000000	0.0000000000	NA	NA
6	10	bcg=yes	0.0003032141	0.0003031681	4.272449e-05	0.002151899
7	15	bcg=yes	0.0006065201	0.0004287444	1.517530e-04	0.002424114
8	183	bcg=yes	0.0407863345	0.0036161646	3.428042e-02	0.048526974

¹The confidence interval are computed using a transformation (log by default) but the standard error is the untransformed one. This can be verified by re-fitting the `survfit` object with the argument `conf.type="plain"`

Getting the confidence intervals for the risk difference require specific calculations, e.g.:

```
RD <- eS.KM$pstate[8,2] - eS.KM$pstate[4,2]
sigma_RD <- sqrt(eS.KM$std.err[4,2]^2 + eS.KM$std.err[8,2]^2)
c(estimate = RD,
  lower = RD - 1.96*sigma_RD,
  upper = RD + 1.96*sigma_RD)
```

estimate	lower	upper
-1.258807e-02	-2.526971e-02	9.356241e-05

Appendix : Exact test

Consider 20 realizations where the event of interest happened in 5 of them. The "classic", asymptotically valid, confidence interval would look like:

```
n <- 20
D <- 3
c(D/n - 1.96*sqrt(D/n*(1-D/n)/n), D/n + 1.96*sqrt(D/n*(1-D/n)/n))
```

```
[1] -0.00649345  0.30649345
```

with a lower bound below 0. This is not optimal and is likely to be wide on the left of the estimate and to narrow on the right of the estimate (coverage of about 90% instead of 95%). An exact confidence interval would be:

```
binom.test(x = D, n = n)
```

Exact binomial test

```
data: D and n
number of successes = 3, number of trials = 20, p-value = 0.002577
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.03207094 0.37892683
sample estimates:
probability of success
               0.15
```

To derive this confidence interval we identify the probability:






- giving 2.5% chance of observing between 0 and 5 events
- giving 2.5% chance of observing between 5 and 20 events

```
sum(dbinom(0:5, size = 20, prob = 0.49104587))
sum(dbinom(5:20, size = 20, prob = 0.08657147))
```

```
[1] 0.025
[1] 0.025
```

In practice we would need to try out several probabilities until we find a reasonable approximation of 2.5% on each side of the confidence interval.

Appendix : Summary table

Parameter	by hand	regression model
risk  no drop-out	(r_1, r_2)	Logistic <pre>e <- glm(Y ~ G-1,family=binomial(link="identity"),data=...) coef(e)</pre>
odds of risk  no drop-out	$\left(\frac{r_1}{1-r_1}, \frac{r_2}{1-r_2}\right)$	Logistic <pre>e <- glm(Y ~ G-1,family=binomial(link="logit"),data=...) exp(coef(e)) or ci.exp(e)</pre>
rate	(λ_1, λ_2)	Poisson <pre>e <- glm(Y ~ G-1,family=poisson(link="log"),offset = log(time), data=...) exp(coef(e)) or e <- glm(cbind(Y,time) ~ G-1,family=poisreg, data=...) ci.exp(e)</pre>
risk difference  no drop-out	$r_2 - r_1$	Logistic <pre>e <- glm(Y ~ G,family=binomial(link="identity"),data=...) coef(e)[2]</pre>
risk ratio  no drop-out	r_2/r_1	Logistic <pre>e <- glm(Y ~ G,family=binomial(link="log"),data=...) exp(coef(e)[2]) or ci.exp(e, subset = "G")</pre>
odds ratio  no drop-out	$\frac{(r_2/(1-r_2))}{(r_1/(1-r_1))}$	Logistic <pre>e <- glm(Y ~ G,family=binomial(link="logit"),data=...) exp(coef(e)[2]) or ci.exp(e, subset = "G")</pre>
rate ratio	λ_2/λ_1	Poisson <pre>e <- glm(Y ~ G,family=poisson(link="log"),offset = log(time), data=...) exp(coef(e)[2]) or e <- glm(cbind(Y,time) ~ G,family=poisreg, data=...) ci.exp(e, subset = "G")</pre>