Two-phase sampling

Thomas Lumley

9(10) July 2025

Two-phase sampling

Take a sample then based on the data you have, take a subsample

Technically, even traditional case-control designs are two-phase sampling, but they are special because the analysis is based on a likelihood.

Here we cover

- case–control sampling depending on covariates as well as outcome
- stratified and unstratified case–cohort sampling
- other designs

These are analysed based on survey methodology, using sampling weights.

Two worlds

New variables

- Phase I is the cohort
- Phase II measures a few additional variables from eg stored blood samples or coding free-text responses. Phase II probably not highly correlated with Phase I

Electronic Health Record research

- Phase I is The Database
- Phase II is validation based on, eg, audit of medical records. Many variables; highly correlated with Phase I

Full analysis

Two phases of sampling: - specify clusters, strata and sampling probabilities at each phase

Simplified because

- often no clusters
- often no strata at Phase I
- sampling probabilities at Phase I are often equal
- sampling probabilities at Phase II can be computed by the program

In R

`twophase()` function declares a two-phase design

- `id`: list of two `id` formulas (usually `list(~1, ~1)`)
- `strata`: list of two `strata` formulas (first one is usually `NULL`)
- `subset`: logical vector indicating whether an observation is in Phase II
- `data`: data frame with Phase I and Phase II data
- `method`: `"simple"` when there is no clustering or Phase-I strata

Approximate analysis

If phase I variables only used for stratifying Phase II sampling, and population is large or infinite a good **approximate** analysis is

- ignore phase I
- treat phase II as sampled from population

For case–control the approximation is **exact**

The approximation is useful when using software other than R

For case–cohort, the approximation is called “Barlow’s method”

Case-cohort design

If X is an expensive predictor variable in a large cohort study with a low event rate, then

- initially measure X on, eg, 10% of cohort (subcohort)
- follow up and measure X on all cases

Would expect nearly full efficiency because information is mostly in cases.

Can use the same subcohort for different case groups, or after more cases accumulate, save more money

For **very large** cohorts (eg national health data) use case-cohort sampling to save computer time.

Practical problems

X may be measured at different times for different people (even if on stored samples taken at the same time)

For biochemical measurements, laboratory drift may be a problem, and will be confounded with case/subcohort status

In contrast, for matched case-control sampling you would always measure X on case and matched controls at the same time.

Problem doesn’t arise when entire measurement is retrospective and the order of lab processing can be randomised.

Efficiency

If there is only one event variable, case–cohort sampling is less efficient than matched case-control sampling (Langholz & Thomas, Biometrics 1991)

If there are multiple event variables case–cohort sampling is typically more efficient because the subcohort and the cases of the first event can be reused as controls for the second event.

Analysis

Analysis has always been by weighted Cox regression, initially with complicated time-dependent weighting schemes

- Prentice, Self & Prentice

The complications were to make the weights **predictable**, ie, depending on the past and not the future. Necessary for 1990s mathematical theory using martingales.

Modern analysis uses retrospective sampling from the full cohort data set: weights can depend on any phase I data.

- $\pi_i = 1$ for cases (whether or not part of subcohort)
- π_i for non-cases in subcohort is proportion of non-cases that are in subcohort

Modern approach is easier, (slightly) more efficient, allows for clustering, **allows for calibration**

Example

Cohort of 10,000 people, subcohort of 500

203 cases occur, 12 in subcohort members.

Sampling probability is $\pi_i = 1$ for all 203 cases

Sampling probability is $\pi_i = \frac{500-12}{10000-203}$ for non-cases

Stratified case–cohort design

A small subcohort may have very few people with a rare exposure

Makes sense to oversample groups of people who

- do have a rare value of a phase I exposure/confounder Z
- are likely to have a rare value of the phase II variable X

In traditional analysis, the stratifying variable had to be available at baseline (predictable)

In modern survey analysis, the stratifying variable can be measured at any time: π_i can depend on arbitrary Phase I data.

Design examples

- Low potassium is more common in people taking thiazide drugs for blood pressure, so oversample people with high blood pressure
- If medication data is available at Phase I, oversample people taking thiazides.
- in study of genetics and heart attack, oversample people at high predicted risk of heart attack (Framingham risk score)

Worked example: Wilms’ Tumour study

- Wilms’ Tumour is a rare kidney cancer in children
- Most US children with the cancer are in the National Wilms’ Tumour Study Group clinical trials
- We have data for everyone, but we can simulate two-phase sampling strategies such as case–cohort

Simple case–cohort

```
library(survival)
data(nwtco)
dcchs<-twophase(id=list(~seqno,~seqno), strata=list(NULL,~rel),
         subset=~I(in.subcohort | rel), data=nwtco)
dcchs
```

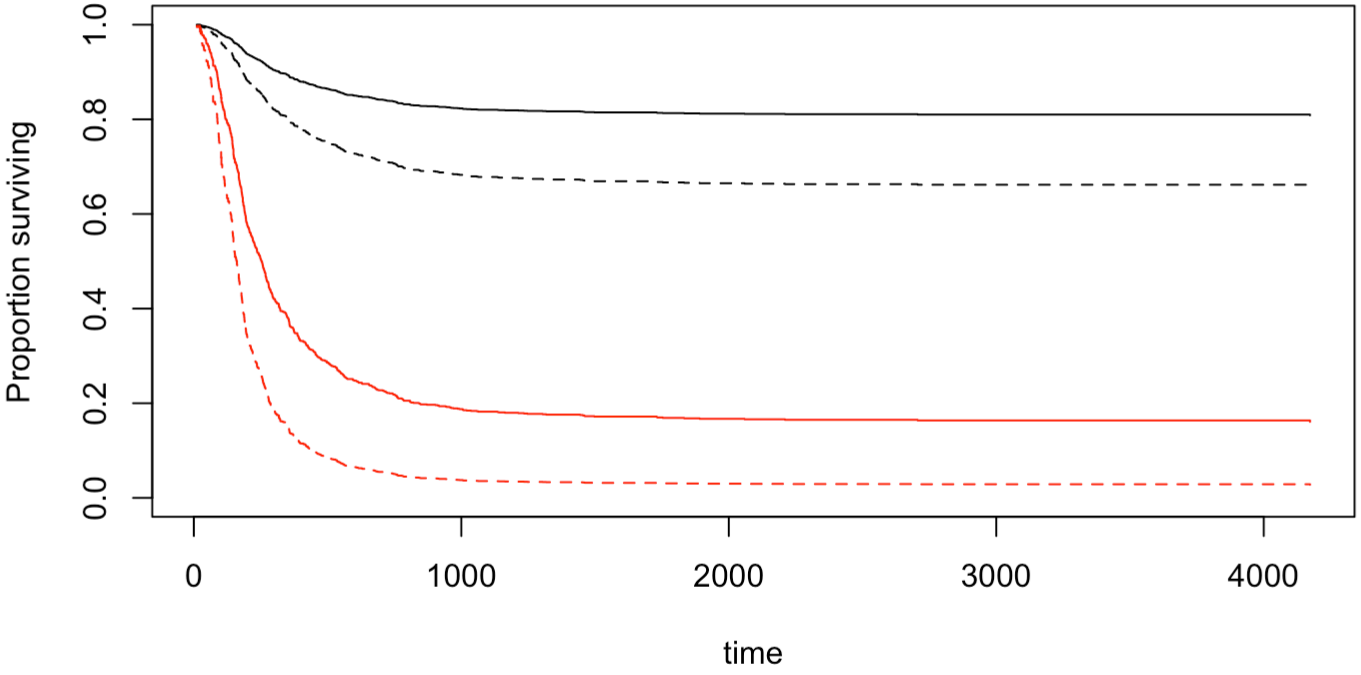
```
## Two-phase sparse-matrix design:
##  twophase(id = list(~seqno, ~seqno), strata = list(NULL, ~rel),
##     subset = ~I(in.subcohort | rel), data = nwtco)
## Phase 1:
## Independent Sampling design (with replacement)
## svydesign(ids = ~seqno)
## Phase 2:
## Stratified Independent Sampling design
## svydesign(ids = ~seqno, strata = ~rel, fpc = `*phase1*`)
```

Cox model

```
model <- svycoxph(Surv(edrel,rel)~factor(stage)+factor(histol)+I(age/12),
                  design=dcchs)
model
```

```
## Call:
## svycoxph(formula = Surv(edrel, rel) ~ factor(stage) + factor(histol) +
##     I(age/12), design = dcchs)
##
##                  coef exp(coef) se(coef) robust se  z      p
## factor(stage)2  0.69      2.00     0.23      0.16  4  2e-05
## factor(stage)3  0.63      1.87     0.23      0.17  4  2e-04
## factor(stage)4  1.30      3.67     0.25      0.19  7  6e-12
## factor(histol)2 1.46      4.30     0.17      0.15 10 <2e-16
## I(age/12)       0.05      1.05     0.03      0.02  2   0.05
##
## Likelihood ratio test=  on 5 df, p=
## n= 1154, number of events= 571
```

```
s<-predict(model, type="curve",newdata=
  data.frame(stage=c(1,1,4,4),histol=c(1,2,1,2),age=c(12,12,12,12)))
plot(s[[1]]); lines(s[[2]],lty=2)
lines(s[[3]],col="red"); lines(s[[4]],lty=2,col="red")
```



Two-phase case–control designs

Ordinary case–control study samples on Y only.

Sample on other Phase I variables Z as well:

- surrogates for exposure
- interaction variables

Survey analysis is easy: $\pi_i(Z, Y)$ are just the sampling probabilities

Semiparametric maximum likelihood analysis is possible, can be more accurate, makes more assumptions

Examples

Gene:environment interaction, with genetic data measured at phase II - sample balanced numbers of cases and controls with and without environmental exposure

Surrogate for exposure: eg self-report vs examination of medical records - sample so that you expect to get equal numbers exposed and unexposed

Worked example: Wilms’ Tumour study

- Wilms’ Tumour is a rare kidney cancer in children
- Most US children with the cancer are in the National Wilms’ Tumour Study Group clinical trials
- One important predictor of survival is **histology** (bad/good)
    - the study group central pathologist is much better at measuring histology
    - could use local hospital measurements as a surrogate
- We have central-lab and local-hospital measurements for everyone, but we can simulate two-phase sampling strategies

Wilms' Tumour: Sampling strategies

Bad histology is rare (about 10%). Relapse is rare (14%)

- **random**: random sample of 1200
- **case–control**: obtain central-lab histology on all cases, random subset of controls
- **risk–based**: obtain central-lab histology on all with bad histology according to local lab, random subset of others
- **balanced**: all cases, all bad histology, subsample of remainder

[exercise: **balanced+stage**: same, but stratified on disease stage (I–IV) as well]

Random

```
data(nwtco,package="survival")
random <- nwtco[sample(nrow(nwtco), 1200),]
coef(summary(glm(rel~factor(histol)*factor(stage),data=random,
                 family=binomial)))
```

```
##                                Estimate Std. Error  z value  Pr(>|z|)
## (Intercept)                     -2.9741     0.2237 -13.2944 2.495e-40
## factor(histol)2                  1.4700     0.4504   3.2640 1.098e-03
## factor(stage)2                   1.0604     0.2862   3.7049 2.115e-04
## factor(stage)3                   1.0381     0.2992   3.4698 5.209e-04
## factor(stage)4                   1.4055     0.3324   4.2287 2.350e-05
## factor(histol)2:factor(stage)2   0.4949     0.5808   0.8522 3.941e-01
## factor(histol)2:factor(stage)3   1.0050     0.5961   1.6859 9.182e-02
## factor(histol)2:factor(stage)4   1.2618     0.7251   1.7402 8.183e-02
```

Case-control

```r
cases <- subset(nwtco, rel==1)
controls <- subset(nwtco,rel==0)[sample(3457, 629),]
casecontrol <- rbind(cases,controls)
coef(summary(glm(rel~factor(histol)*factor(stage),data=casecontrol,
                 family=binomial)))
```

```
##                                 Estimate Std. Error  z value  Pr(>|z|)
## (Intercept)                     -0.97127     0.1224 -7.93443 2.115e-15
## factor(histol)2                  1.41756     0.3428  4.13572 3.539e-05
## factor(stage)2                   0.66899     0.1720  3.88848 1.009e-04
## factor(stage)3                   0.80198     0.1814  4.42212 9.774e-06
## factor(stage)4                   1.12542     0.2222  5.06473 4.090e-07
## factor(histol)2:factor(stage)2 -0.03772     0.4646 -0.08118 9.353e-01
## factor(histol)2:factor(stage)3  0.38934     0.4698  0.82875 4.072e-01
## factor(histol)2:factor(stage)4  1.24170     0.7108  1.74697 8.064e-02
```

Balanced, ignoring two-phase structure

```
full <- subset(nwtco, (rel==1) | (instit==2))
sampled <- subset(nwtco,(rel==0) & (instit==1))[sample(3207, 379),]
balanced <- rbind(full,sampled)
coef(summary(glm(rel~factor(histol)*factor(stage),data=balanced,
                 family=binomial)))
```

```
##                                 Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                      -0.6430     0.1288 -4.9931 5.943e-07
## factor(histol)2                  -0.3576     0.2670 -1.3394 1.804e-01
## factor(stage)2                    0.7493     0.1853  4.0439 5.258e-05
## factor(stage)3                    0.7891     0.1937  4.0733 4.635e-05
## factor(stage)4                    0.6272     0.2193  2.8605 4.229e-03
## factor(histol)2:factor(stage)2   0.1697     0.3604  0.4707 6.379e-01
## factor(histol)2:factor(stage)3   0.3773     0.3500  1.0780 2.810e-01
## factor(histol)2:factor(stage)4   1.6463     0.4407  3.7358 1.871e-04
```

Balanced with weights

```
balanced$wt <- with(balanced, ifelse(rel==1 | instit==2, 1, 3207/379))
bdesign <- svydesign(id=~1, strata=~interaction(rel,instit),
                    weights=~wt,data=balanced)
coef(summary(svyglm(rel~factor(histol)*factor(stage),design=bdesign,
                 family=quasibinomial)))
```

```
##                                Estimate Std. Error  t value   Pr(>|t|)
## (Intercept)                     -2.7000     0.1099 -24.5634 4.308e-108
## factor(histol)2                  1.3354     0.2961   4.5101  7.122e-06
## factor(stage)2                   0.7657     0.1888   4.0560  5.317e-05
## factor(stage)3                   0.9069     0.2003   4.5278  6.560e-06
## factor(stage)4                   0.9298     0.2337   3.9792  7.333e-05
## factor(histol)2:factor(stage)2   0.2605     0.4222   0.6170  5.374e-01
## factor(histol)2:factor(stage)3   0.2251     0.4154   0.5419  5.880e-01
## factor(histol)2:factor(stage)4   1.2805     0.5793   2.2104  2.727e-02
```

Balanced: proper two-phase

```
balanced2 <- merge(balanced[,c("seqno","histol")], nwtco[,-3],
                   by="seqno",all=TRUE)
balanced2$insample <- !is.na(balanced2$histol)
summary(balanced2[,1:4])
```

```
##      seqno          histol          instit          stage     
##  Min.   :   1   Min.   :1.0    Min.   :1.0    Min.   :1.00  
##  1st Qu.:1009   1st Qu.:1.0    1st Qu.:1.0    1st Qu.:1.00  
##  Median :2022   Median :1.0    Median :1.0    Median :2.00  
##  Mean   :2026   Mean   :1.3    Mean   :1.1    Mean   :2.07  
##  3rd Qu.:3039   3rd Qu.:2.0    3rd Qu.:1.0    3rd Qu.:3.00  
##  Max.   :4088   Max.   :2.0    Max.   :2.0    Max.   :4.00  
##                 NA's   :2828                                
```

```
b2design <- twophase(id=list(~1,~1),
                     strata=list(NULL,~interaction(rel,instit)),
                     subset=~insample,data=balanced2, method="simple")
coef(summary(svyglm(rel~factor(histol)*factor(stage),design=b2design,
                 family=quasibinomial)))
```

```
##                                 Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)                      -2.7000     0.1214 -22.2489 5.791e-92
## factor(histol)2                   1.3354     0.3045   4.3855 1.259e-05
## factor(stage)2                    0.7657     0.1887   4.0587 5.257e-05
## factor(stage)3                    0.9069     0.2002   4.5305 6.479e-06
## factor(stage)4                    0.9298     0.2336   3.9809 7.281e-05
## factor(histol)2:factor(stage)2    0.2605     0.4217   0.6176 5.370e-01
## factor(histol)2:factor(stage)3    0.2251     0.4151   0.5422 5.878e-01
## factor(histol)2:factor(stage)4    1.2805     0.5790   2.2115 2.719e-02
```

[Very little impact of proper two-phase analysis]

Related designs: real examples

- Maternal weight gain during pregnancy and asthma & obesity in young child, in Memphis, TN †
- Efavirenz (vs nevirapine) in Kenyan women with HIV and potential interference with contraceptive implants (3-phase, some non-response)
- Association between CD4 count and hazard of AIDS-defining event in people on ART
- CHARGE Consortium genomic sequencing follow-up of GWAS (multinational)
- TGF-B1 and heart failure in stratified (HF, diabetes, ACE inhibitor) subsample of Cardiovascular Health Study (USA)

† Shepherd et al (2023) *Biometrics* 79(3):2649-2663

Basic message

If you can’t sample everyone, you can choose the most interesting people to sample **in any way you like** as long as everyone has a chance of being chosen.

You know the sampling probabilities for each person, they can be used in the analysis.

Optimal design

To think about optimal design we want to reduce the problem to means.

Influence functions are the solution!

$$\hat{\beta} - \beta = \frac{1}{\sqrt{n}} \sum_{i=1}^{N} h_i(\beta) + o_p(n^{-1/2})$$

We want to maximise information about h: perhaps with imputation of phase-2

Optimal design

The optimal (stratified) design has $\pi_k \propto N_k\sigma_k$ where σ_k is the standard deviation **of the influence functions** of the estimator $\hat{\beta}$ you're trying to estimate

Optimal design typically depends on the value of β

- prior distribution
- multiwave sampling, re-estimating parameters at each wave

R package `optimall` can help with designing a sample

Using the whole cohort…

Available information

In a classical case-control study, only Y is known for the whole cohort.

Often we know more

- variables Z that are in our outcome model, maybe as confounders
- variables A that are not in our outcome model but are predictive either of X or of Y

We'd like to use this information

How to use the information

A general approach is to adjust the weights slightly, so that the estimated whole-cohort total for *auxiliary variables* like A and Z matches the known truth.

You get improved estimation for totals of anything correlated with A or Z.

In survey statistics this is called *calibration* or *raking*. In theoretical biostatistics these are *AIPW* estimators. Direct standardisation (in epidemiology) is a special case.

We want regression coefficients

Totals of variables Y or A or Z are not strongly correlated with regression coefficients

We need to use *influence functions* or *delta-betas* as the auxiliary variables for reweighting

To do this, we need to *impute* X for everyone in the cohort

Procedure

1. Impute X to get $\hat{X}$.
2. Fit your outcome model $Y \sim \hat{X} + Z$ using $\hat{X}$ instead of X for **everyone**. Call this the **whole-cohort model**
3. Extract the influence functions from the whole-cohort model and calibrate using them
4. Fit your outcome model $Y \sim X + Z$ to the calibrated phase-two subsample. Using `twophase()` **does** matter now.

[It's even better to use multiple imputation]

National Wilms' Tumour Group data

```
set.seed(2017-12-3)
nwts <- read.table("nwts-share.txt", header=TRUE)
names(nwts)
```

```
##  [1] "trel"    "tsur"    "relaps"  "dead"    "study"   "stage"   "histol" 
##  [8] "instit"  "age"     "yr"      "specwgt" "tumdiam"
```

New variables

A linear spline in age

```
nwts$age1 <- with(nwts, pmin(age, 1))
nwts$age2 <- with(nwts, pmax(age, 1))
```

The full-cohort model

Histology:age interaction and stage:tumour-diameter interaction

```
fullmodel <- glm(relaps~histol*(age1+age2)+ I(stage>2)*tumdiam,
                 family=binomial, data=nwts)
coef(summary(fullmodel))
```

```
##                            Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                -2.61048    0.34241 -7.6238 2.463e-14
## histol                      5.66253    0.97790  5.7905 7.018e-09
## age1                       -0.73020    0.34976 -2.0877 3.682e-02
## age2                        0.12198    0.01920  6.3544 2.093e-10
## I(stage > 2)TRUE            1.50690    0.29682  5.0768 3.838e-07
## tumdiam                     0.07219    0.01591  4.5386 5.663e-06
## histol:age1                -4.19225    1.05085 -3.9894 6.625e-05
## histol:age2                -0.03064    0.04752 -0.6448 5.191e-01
## I(stage > 2)TRUE:tumdiam -0.08947    0.02355 -3.7984 1.456e-04
```

Case-control sample

Here we take all the cases and a random sample of controls

```
nwts$id <- 1:nrow(nwts)
cases <- subset(nwts, relaps==1)
noncases <- subset(nwts, relaps==0)
controlsample <- sample(noncases$id, nrow(cases))
ccsample<- rbind(cases, noncases[noncases$id %in% controlsample,])
ccsample$weight<-with(ccsample,
                  ifelse(relaps==1, 1, nrow(noncases)/nrow(cases)))
```

We can compare the maximum likelihood estimator and the survey estimator:

```
library(survey)
ccmle <- glm(relaps~offset(log(weight))+histol*(age1+age2)+
               I(stage>2)*tumdiam, family=binomial, data=ccsample)
coef(summary(ccmle))
```

```
##                          Estimate Std. Error z value  Pr(>|z|)
## (Intercept)              -2.20612    0.42691 -5.1677 2.370e-07
## histol                    5.13479    1.34403  3.8204 1.332e-04
## age1                     -1.18201    0.44027 -2.6847 7.259e-03
## age2                      0.20206    0.02855  7.0774 1.468e-12
## I(stage > 2)TRUE          2.41562    0.43219  5.5893 2.280e-08
## tumdiam                   0.09628    0.02117  4.5477 5.423e-06
## histol:age1              -3.44287    1.47755 -2.3301 1.980e-02
## histol:age2               0.08089    0.09217  0.8777 3.801e-01
## I(stage > 2)TRUE:tumdiam -0.13922    0.03429 -4.0601 4.906e-05
```

```
survey_cc <- svydesign(id=~1, weights=~weight, strata=~relaps,
                       data=ccsample)
ccest <- svyglm(relaps~histol*(age1+age2)+I(stage>2)*tumdiam,
                family=quasibinomial, design=survey_cc)
coef(summary(ccest))
```

```
##                          Estimate Std. Error t value  Pr(>|t|)
## (Intercept)              -2.56853    0.40096  -6.406 2.070e-10
## histol                    3.88912    1.21947   3.189 1.460e-03
## age1                     -0.87602    0.39907  -2.195 2.832e-02
## age2                      0.14053    0.02705   5.196 2.355e-07
## I(stage > 2)TRUE          1.86148    0.41186   4.520 6.743e-06
## tumdiam                   0.07198    0.02152   3.345 8.444e-04
## histol:age1              -2.86948    1.33514  -2.149 3.180e-02
## histol:age2               0.10865    0.08409   1.292 1.965e-01
## I(stage > 2)TRUE:tumdiam -0.10940    0.03318  -3.297 1.004e-03
```

The two seem fairly comparable: we know that asymptotically the maximum likelihood estimator must be better, but the difference is small enough to not show up in a single comparison

```
round(cbind(coef(ccmle), coef(ccest))-coef(fullmodel),3)
```

```
##                              [,1]   [,2]
## (Intercept)                 0.404  0.042
## histol                     -0.528 -1.773
## age1                       -0.452 -0.146
## age2                        0.080  0.019
## I(stage > 2)TRUE            0.909  0.355
## tumdiam                     0.024  0.000
## histol:age1                 0.749  1.323
## histol:age2                 0.112  0.139
## I(stage > 2)TRUE:tumdiam -0.050 -0.020
```

Using a twophase() objet

The simple survey estimator does not use the full cohort; we can declare a `twophase` object that does. We do not need to specify weights because the software can work out what they are.

```
nwts_twophase <- twophase(id=list(~1,~1), strata=list(NULL, ~relaps),
                          subset=~I((relaps==1)| id %in% controlsample),
                          data=nwts)
twophaseest <- svyglm(relaps~histol*(age1+age2)+ I(stage>2)*tumdiam,
                      family=quasibinomial, design=nwts_twophase)
```

```
coef(summary(twophaseest))
```

```
##                            Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                -2.56853    0.40318  -6.371 2.589e-10
## histol                      3.88912    1.21920   3.190 1.456e-03
## age1                       -0.87602    0.39888  -2.196 2.825e-02
## age2                        0.14053    0.02704   5.198 2.331e-07
## I(stage > 2)TRUE            1.86148    0.41175   4.521 6.704e-06
## tumdiam                     0.07198    0.02151   3.347 8.402e-04
## histol:age1                -2.86948    1.33484  -2.150 3.176e-02
## histol:age2                 0.10865    0.08407   1.292 1.964e-01
## I(stage > 2)TRUE:tumdiam -0.10940    0.03317  -3.298 1.000e-03
```

We still aren't using the whole cohort for anything, so the two analyses are almost identical

Using the whole cohort

We'll try to use the whole cohort now. First, just use `instit` instead of `histol`

First, fit the model to the full data

```
phase1model <- glm(relaps~instit*(age1+age2)+ I(stage>2)*tumdiam,
                   family=binomial, data=nwts)
```

Extract the influence functions and create a new design object

```r
inffun<-model.matrix(phase1model)*resid(phase1model, type="response")
colnames(inffun)<-paste0("if",1:ncol(inffun))
aug_twophase <- twophase(id=list(~1,~1), strata=list(NULL, ~relaps),
                         subset=~I((relaps==1)| id %in% controlsample),
                         data=cbind(nwts,inffun), method="simple")
```

Calibrate, and fit the model of interest (ie, with `histol`) to the calibrated sample

```
calformula <- make.formula(colnames(inffun))
cal_twophase <- calibrate(aug_twophase, calformula, phase=2)
svyest_instit<-svyglm(relaps~histol*(age1+age2)+ I(stage>2)*tumdiam,
                      family=quasibinomial, design=cal_twophase)
```

```
coef(summary(svyest_instit))
```

```
##                            Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                -2.58154    0.33237 -7.7671 1.597e-14
## histol                      6.47280    1.30594  4.9564 8.106e-07
## age1                       -0.73222    0.32328 -2.2650 2.367e-02
## age2                        0.11680    0.02021  5.7801 9.293e-09
## I(stage > 2)TRUE            1.59303    0.31440  5.0669 4.614e-07
## tumdiam                     0.06967    0.01717  4.0575 5.249e-05
## histol:age1                -5.27895    1.42959 -3.6926 2.310e-04
## histol:age2                 0.06956    0.07685  0.9051 3.656e-01
## I(stage > 2)TRUE:tumdiam -0.09329    0.02521 -3.6999 2.245e-04
```

Comparing the uncalibrated and calibrated estimates, the coefficients have nearly all moved closer to the true full cohort value.

```
round(cbind(coef(twophaseest), coef(svyest_instit))-coef(fullmodel),3)
```

```
##                              [,1]   [,2]
## (Intercept)                 0.042  0.029
## histol                     -1.773  0.810
## age1                       -0.146 -0.002
## age2                        0.019 -0.005
## I(stage > 2)TRUE            0.355  0.086
## tumdiam                     0.000 -0.003
## histol:age1                 1.323 -1.087
## histol:age2                 0.139  0.100
## I(stage > 2)TRUE:tumdiam -0.020 -0.004
```

Calibration by imputation

It is always valid to just use a surrogate such as `instit` in calibration, but it is probably not optimal.

The attenuation bias in using a mismeasured predictor translates into a loss of precision in the calibrated estimate. We can try to construct a regression imputation of histology instead:

```
impmodel<-svyglm(histol~instit*(relaps+I(stage>3))+I(age>10)+factor(study),
                 family=quasibinomial,design=nwts_twophase)
nwts$imphistol <-as.vector(predict(impmodel,newdata=nwts,
                                   type="response",se.fit=FALSE))
with(nwts, by(imphistol, histol, summary))
```

```
## histol: 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0119  0.0191  0.0262  0.0419  0.0262  0.9873
## ------------------------------------------------------------
## histol: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0140  0.0991  0.8789  0.6718  0.9810  0.9873
```

We now proceed as before. In particular, note that it is important to use `imphistol` for all observations in the phase-1 model, **even those where `histol` is available** – it was not available at phase 1.

```
phase1model_imp <- glm(relaps~imphistol*(age1+age2)+ I(stage>2)*tumdiam,
                       family=binomial, data=nwts)
```

Extract the influence functions and create a new design object

```
inffun_imp<-model.matrix(phase1model_imp)*
  resid(phase1model_imp, type="response")
colnames(inffun_imp)<-paste0("if",1:ncol(inffun_imp))
aug_twophase_imp <- twophase(id=list(~1,~1), strata=list(NULL, ~relaps),
                             subset=~I((relaps==1)| id %in% controlsample),
                             data=cbind(nwts,inffun_imp), method="simple")
```

Calibrate, and fit the model of interest

```
calformula <- make.formula(colnames(inffun_imp))
cal_twophase_imp <- calibrate(aug_twophase_imp, calformula, phase=2)
svyest_imp<-svyglm(relaps~histol*(age1+age2)+ I(stage>2)*tumdiam,
                   family=quasibinomial, design=cal_twophase_imp)
coef(summary(svyest_imp))
```

```
##                           Estimate Std. Error t value  Pr(>|t|)
## (Intercept)               -2.61615    0.33200 -7.8800 6.773e-15
## histol                     6.45150    1.39665  4.6193 4.225e-06
## age1                      -0.74017    0.32235 -2.2961 2.182e-02
## age2                       0.11742    0.02027  5.7913 8.707e-09
## I(stage > 2)TRUE           1.63215    0.31786  5.1348 3.245e-07
## tumdiam                    0.07328    0.01729  4.2379 2.412e-05
## histol:age1               -5.01482    1.52036 -3.2984 9.980e-04
## histol:age2               -0.04630    0.07894 -0.5865 5.576e-01
## I(stage > 2)TRUE:tumdiam -0.09501    0.02533 -3.7514 1.835e-04
```

There has been a slight additional improvement; slight, because `instit` is overwhelmingly the best predictor of histology.

```
round(cbind(coef(twophaseest), coef(svyest_instit),
            coef(svyest_imp))-coef(fullmodel),3)
```

```
##                              [,1]   [,2]   [,3]
## (Intercept)                 0.042  0.029 -0.006
## histol                     -1.773  0.810  0.789
## age1                       -0.146 -0.002 -0.010
## age2                        0.019 -0.005 -0.005
## I(stage > 2)TRUE            0.355  0.086  0.125
## tumdiam                     0.000 -0.003  0.001
## histol:age1                 1.323 -1.087 -0.823
## histol:age2                 0.139  0.100 -0.016
## I(stage > 2)TRUE:tumdiam -0.020 -0.004 -0.006
```

Technical digression

The optimal designs mentioned earlier are optimal if you *don't* use the whole cohort.

Optimal designs for the calibration estimator are hard

Fortunately, it doesn't matter much: optimising for the simple IPW design tends to give near-optimal calibration designs

Optimal design matters less when you do calibration (they both extract phase I information)

Further reading

McIsaac, M. A., & Cook, R. J. (2015). Adaptive sampling in two-phase designs: a biomarker study for progression in arthritis. *Statistics in medicine*, 34(21), 2899–2912.

Lumley T, Shaw PA, Dai JY. (2011) Connections between survey calibration estimators and semiparametric models for incomplete data. *Int Stat Rev.* 79(2):200-220.

Chen T, Lumley T (2020) Optimal multiwave sampling for regression modeling in two-phase designs *Statistics in Medicine* 39(30) 4912-4921

Han K, Shaw PA, Lumley T (2021) Combining multiple imputation with raking of weights: An efficient and robust approach in the setting of nearly-true models *Statistics in Medicine* 40(30) 6777-6791

Chen T, Lumley T (2022) Optimal sampling for design-based estimators of regression models *Statistics in Medicine* 41(8) 1482-1497

Shepherd BE, Shaw PA et al (2023) Analysis of Error-prone Electronic Health Records with Multi-wave Validation Sampling: Association of Maternal Weight Gain during Pregnancy with Childhood Outcomes *Biometrics* 79(3):2649-2663

Yang JB, Shepherd BE, Lumley T, Shaw PA (2021) Optimum Allocation for Adaptive Multi-Wave Sampling in R: The R Package `optimall`. *arXiv:2106.09494*