**Chapter 7**

**Other Useful R packages for Epidemiologic Research**

* 1. **nmle: linear and nonlinear mixed effects models**

**lme**: This function fits a linear mixed-effects model in the formulation described in Laird and Ware (1982) but allowing for nested random effects. The within-group errors are allowed to be correlated and/or have unequal variances.

lme(fixed, data, random, correlation, weights, subset, method, na.action, control, contrasts = NULL)

### fixed: a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a '~' operator and the predictors, separated by '+' operators, on the right.

### random: any of the following: (i) a one-sided formula of the form '~x1+...+xn | g1/.../gm', with 'x1+...+xn' specifying the model for the random effects and 'g1/.../gm' the grouping structure ('m' may be equal to 1, in which case no '/' is required). (ii) a list of one-sided formulas of the form '~x1+...+xn | g', with possibly different random effects models for each grouping level. (iii) a one-sided formula of the form '~x1+...+xn', or a 'pdMat' object with a formula (iv) a named list of formulas or 'pdMat' objects as in (iii), with the grouping factors as names. (v) an 'reStruct' object.

### correlation: an optional 'corStruct' object describing the within-group correlation structure. Defaults to 'NULL', corresponding to no within-group correlations.

### weights: an optional 'varFunc' object or one-sided formula describing the within-group heteroscedasticity structure.

### method: estimation method; if '"REML"' the model is fit by maximizing the restricted log-likelihood. If '"ML"' the log-likelihood is maximized. Defaults to '"REML"'.

DEFINE THE CORRELATION STRUCTURE OF THE MODEL

lmeStruct(reStruct, corStruct, varStruct): a linear mixed effect structure is a list of model components representing different sets of parameters in the model. An lmeStruct must contain at least a reStruct, but may also contain corStruct and varFunc objects.

reStruct: representing a random effects structrure

corStruct : representing a correlation structure. Default is NULL.

varStruct: a varFunc object, representing a variance function structure. Default is NULL.

CorClasses: standard classes of correlation structure:

CorAR1: autoregressive process of order 1

CorARMA: autoregressive moving average process

corCAR1:continuos autoregressive process

corCompSymm: compound symmetry structure corresponding to a constant correlation. The correlation model does not depend on the position of the observation but just on the group.

corExp, corLin, corGaus, corRatio, corSpher: exponential, linear,Gaussian, rational,spherical spatial correlation

corSymm: general correlation matrix, with no additional structure

All of these structures are functions in which the form needs to be specified. For example:

CorCompSymm(value, form, fixed) – this function is a constructor for the corComSymm class. Value is the correlation between any 2 correlated observation. Form is a formula of the form ~t, or ~t|g, specifying a time covariate and a grouping factor.

VarCorr(x,sigma,rdig) – This is a function that calculates the estimated variances, standard deviations and correlations between the random-effects terms.

1. Introduction

The NHANES study was not a simple random sample of the US population, since providing standardized medical examinations would have been too costly. **Rather a limited number of counties or county clusters were selected at the first stage and then subjects were selected within these locations.** Persons from the same area are often more alike than two randomly selected persons reflecting **the clustering of ethnic immigrants or racial minorities** or **the commonality of exposures and socioeconomic differences**. If all the causes of these similarities are not included in the model, the errors will be correlated within the same location (strata+PSU), but uncorrelated between locations.

The design effect of sampling subjects within a limited number of strata may be accommodated by **allowing the intercept to be a random effect**. Since the distribution of the random effects is constrained, this is not entirely equivalent to a separate intercept for each stratum that is using a dummy variable for each stratum in a fixed effect model (glm, gam, lm).

A mixed effect model can also be used in **a longitudinal study**, where there are multiple measurements of a determined health indicator for each subject. In that case the subject could be included as the random effect in the model, and in the fixed effect part of the model one can control for predictors, including subject’s characteristic.

# Load these packages all the time

library(foreign)

library(Hmisc)

library(epicalc)

## Try linear mixed effects models using nlme

library(nlme)

# Set working directory, load and attach the nhanes3 data.

setwd("C:/epid674")

load("nhanes3.rda")

attach(nhanes3)

1. First see the distributions of PSU and strata. These variables indicate geographic locations (counties) in two phases of NHANES-3. Let’s combine these and create a single variable (**locode**).

tab1(psu) # 2 PSU

tab1(strata) # 49 strata

nhanes3$locode<-ifelse(nhanes3$psu==1, nhanes3$strata, nhanes3$strata+49)

tab1(nhanes3$locode)

table(nhanes3$psu,nhanes3$locode)

1. **Random intercepts only model**. Let’s create a random intercepts model for SBP. In **lme** you can run only data with a Gaussian error distribution.

sbp.lme<-lme(fixed=log(sbp)~log(bpb)+age+I(age^2)+bmi +factor(race)+factor(sex)+factor(educ)+factor(alc), random=~1|locode, na.action=na.omit, data=nhanes3)

summary(sbp.lme)

##Return variance-covariance matrix for random variables

getVarCov(sbp.lme)

1. **Random intercepts and slopes model**. The effect of blood lead on SBP may not be the same within all **locode**. We can model this by allowing stratum-specific effect estimates (random effects) that vary around a common mean effect (fixed effect). We can fit both a fixed effect and a random effect for **bpb** across the various **locode** using the **lme()** function. We can fit a model with a random effect for **locode** and a random slope for bpb, extract and plot the random slope.

sbp.lme2<-lme(fixed=log(sbp)~log(bpb)+age+I(age^2)+bmi +factor(race)+factor(sex)+factor(smk)+factor(educ)+factor(alc), random=~1+log(bpb)|locode, na.action=na.omit, data=nhanes3)

summary(sbp.lme2)

getVarCov(sbp.lme2)

1. Compare the model with a random intercept only with the model with a random intercept and a random slope

anova(sbp.lme, sbp.lme2)

* 1. **survey: analysis of complex survey samples**

The package ‘survey’ was developed by Dr. Thomas Lumley of the University of Washington. Useful information on complex sampling analyses in R can be found at his website, <http://faculty.washington.edu/tlumley/survey>.

Functions available in the survey package

**svydesign**: Specify a complex sampling design.

svydesign(ids, probs=NULL, strata = NULL, variables = NULL, fpc=NULL,

data = NULL, nest = FALSE, check.strata = !nest, weights=NULL,pps=FALSE,...)

|  |  |
| --- | --- |
| ids | Formula or data frame specifying cluster ids from largest level to smallest level, ~0 or ~1 is a formula for no clusters. PSU can be specified here. |
| probs | Formula or data frame specifying cluster sampling probabilities |
| strata | Formula or vector specifying strata, use NULL for no strata |
| variables | Formula or data frame specifying the variables measured in the survey. If NULL, the data argument is used. |
| fpc | Finite population correction: see Details below |
| weights | Formula or vector specifying sampling weights as an alternative to prob |
| data | Data frame to look up variables in the formula arguments, or database table name, or imputationList object, see below |
| nest | If TRUE, relabel cluster ids to enforce nesting within strata |

**svytable** and **svychisq**: Contingency tables and chisquared tests of association for survey data.

svytable(formula, design, Ntotal = NULL, round = FALSE,...)

svychisq(formula, design, statistic = c("F","Chisq","Wald","adjWald", "lincom","saddlepoint"),na.rm=TRUE,...)

svychisq computes first and second-order Rao-Scott corrections to the Pearson chi-squared test, and two Wald-type tests.

The default (statistic="F") is the Rao-Scott second-order correction. The p-values are computed with a Satterthwaite approximation to the distribution. The alternative statistic="Chisq" adjusts the Pearson chi-squared statistic by a design effect estimate and then compares it to the chi-squared distribution it would have under simple random sampling.

The statistic="Wald" test is that proposed by Koch et al (1975) and used by the SUDAAN software package. It is a Wald test based on the differences between the observed cells counts and those expected under independence. The adjustment given by statistic="adjWald" reduces the statistic when the number of PSUs is small compared to the number of degrees of freedom of the test. Rao and Thomas (1990) compare these tests and find the adjustment beneficial.

statistic="lincom" uses the exact asymptotic distribution, which is a linear combination of chi-squared variables, and statistic="saddlepoint" uses a saddlepoint approximation to this distribution.

**svyglm**: Fit a generalized linear model to data from a complex survey design, with inverse-probability weighting and design-based standard errors.

svyglm(formula, design, subset=NULL, ...)

summary(object, correlation = FALSE, df.resid=NULL,...)

predict(object,newdata=NULL,total=NULL,type=c("link","response","terms"), se.fit=(type != "terms"),vcov=FALSE,...)

|  |  |
| --- | --- |
| formula | Model formula |
| design | Survey design from [svydesign](about:blank) or [svrepdesign](about:blank). Must contain all variables in the formula |
| subset | Expression to select a subpopulation |
| ... | Other arguments passed to glm or summary.glm |
| rho | For replicate BRR designs, to specify the parameter for Fay's variance method, giving weights of rho and 2-rho |
| return.replicates | Return the replicates as a component of the result? |
| object | A svyglm object |
| correlation | Include the correlation matrix of parameters? |
| na.action | Handling of NAs |
| multicore | Use the multicore package to distribute replicates across processors? |
| df.resid | Optional denominator degrees of freedom for Wald tests |
| newdata | new data frame for prediction |
| total | population size when predicting population total |
| type | linear predictor (link) or response |
| se.fit | if TRUE, return variances of predictions |
| vcov | if TRUE and se=TRUE return full variance-covariance matrix of predictions |

For binomial and Poisson families use **family=quasibinomial()** and **family=quasipoisson()** to avoid a warning about non-integer numbers of successes. The `quasi' versions of the family objects give the same point estimates and standard errors and do not give the warning.

If df.resid is not specified the df for the null model is computed by [degf](http://faculty.washington.edu/tlumley/survey/html/svychisq.html) and the residual df computed by subtraction. It's not that these are particularly good approximations in a regression model but they are relatively standard. To get tests based on a Normal distribution use df.resid=Inf.

Parallel processing with multicore=TRUE is helpful only for fairly large data sets and on computers with sufficient memory. It may be incompatible with GUIs, although the Mac Aqua GUI appears to be safe.

predict gives fitted values and sampling variability for specific new values of covariates. When newdata are the population mean it gives the regression estimator of the mean, and when newdata are the population totals and total is specified it gives the regression estimator of the population total. Regression estimators of mean and total can also be obtained with [calibrate](http://faculty.washington.edu/tlumley/survey/html/calibrate.html).

## Survival analysis is not a part of this class, but a Cox model can be fit under the survey package.

**svycoxph**: Fit a proportional hazards model to data from a complex survey design.

svycoxph(formula, design,subset=NULL, ...)

predict(object, newdata, se=FALSE, type=c("lp", "risk", "expected", "terms","curve"),...)

|  |  |
| --- | --- |
| formula | Model formula. Any cluster() terms will be ignored. |
| design | survey.design object. Must contain all variables in the formula |
| subset | Expression to select a subpopulation |
| object | A svycoxph object |
| newdata | New data for prediction |
| se | Compute standard errors? This takes a lot of memory for type="curve" |
| type | "curve" does predicted survival curves. The other values are passed to predict.coxph() |
| ... | Other arguments passed to coxph |

**svykm**: Estimates the survival function using a weighted Kaplan-Meier estimator.

svykm(formula, design,se=FALSE, ...)

plot(x,xlab="time",ylab="Proportion surviving", ylim=c(0,1), ci=NULL, lty=1, ...)

lines(x,xlab="time",type="s",ci=FALSE,lty=1,...)

|  |  |
| --- | --- |
| formula | Two-sided formula. The response variable should be a right-censored Surv object |
| design | survey design object |
| se | Compute standard errors? This is slow for moderate to large data sets |
| ... | in plot and lines methods, graphical parameters |
| x | a svykm or svykmlist object |
| xlab,ylab,ylim,type | as for plot |
| lty | Line type, see par |
| ci | Plot (or return, forquantile) the confidence interval |

### load the survey package

library(survey)

1. As we learned before, the NHANES is based on the multi-stage survey design (complex sampling design) and therefore, special analytic methods are required to take such a design effect into account. R allows a variety of complex sampling designs which can be specified using the **”svydesign”** function. First, let’s specify the survey design.

bpdsn<-svydesign(id=~psu, strata=~strata, weights=~wt\_mh, data=nhanes3, nest=T)

#nest=T: relabel cluster ids to enforce nesting within strata

1. Descriptive analyses in the R survey package. Calculate weighted means, medians and standard errors of continuous variables and weighted prevalences of categorical variables.

svymean(~age, design=bpdsn)

summ(age)

svymean(~age+sbp+bpb, design=bpdsn)

svymean(~sbp+bpb+bmi, design=bpdsn)

svymean(~sbp+bpb+bmi, design=bpdsn, na.rm=T)

svytable(~sex, design=bpdsn)

svytable(~sex, Ntotal=100, design=bpdsn)

## To compute percentage, Ntotal=100 should be specified.

svytable(~sex+race, design=bpdsn)

summary(svytable(~sex+race, Ntotal=100, design=bpdsn))

svychisq(~sex+race, design=bpdsn)

chisq.test(sex, race)

svytable(~htn+smk, design=bpdsn)

svychisq(~htn+smk, design=bpdsn, statistic="Wald")

## Survey Chi-square test used in SUDAAN

1. Examine univariate distributions and bivariate associations using graphical analyses.

svyhist(~sbp, design=bpdsn)

svyhist(~log(sbp), design=bpdsn)

svyhist(~bpb, bpdsn)

svyhist(~log(bpb), bpdsn)

svyboxplot(log(sbp)~1, bpdsn)

svyboxplot(log(sbp)~as.factor(smk), bpdsn)

bp.bysexrace<-svyby(~sbp+dbp, ~sex+race, bpdsn, svymean)

barplot(bp.bysexrace)

barplot(bp.bysexrace, legend=TRUE)

## change the label of x-axis and ylim

xlabel<-c("Male, White","Female, White","Male, Black","Female, Black")

barplot(bp.bysexrace, legend=TRUE, ylim=c(0,160), col=c("purple","violet"), names=xlabel)

#simple scatterplot with circles whose area is proportional to the sampling weight

svyplot(log(sbp)~age, bpdsn, style="bubble")

svyplot(log(sbp)~log(bpb), bpdsn, style="transparent", pch=19, xlab="Blood lead", ylab="log(SBP)")

#style="transparent" plots points with opacity proportional to sampling weight

## Scatterplot smoothing

smth.sbp.bpb1<-svysmooth(log(sbp)~bpb, bpdsn, bandwidth=10)

#fit local polynomial kernel plot with a bandwidth=10 bpb unit

plot(smth.sbp.bpb1)

smth.sbp.bpb2<-svysmooth(log(sbp)~bpb, bpdsn, bandwidth=20)

plot(smth.sbp.bpb2)

1. Create our linear model for SBP using the **lm()** and **lme()** functions and then fit using the **svyglm()** function.

sbp.lm<-lm(log(sbp)~log(bpb)+age+I(age^2)+bmi+factor(race) +factor(sex)+factor(smk)+factor(educ)+factor(alc), data=nhanes3)

summary(sbp.lm)

sbp.lme<-lme(fixed=log(sbp)~log(bpb)+age+I(age^2)+bmi +factor(race)+factor(sex)+factor(smk)+factor(educ)+factor(alc), random=~1|strata, na.action=na.omit, data=nhanes3)

summary(sbp.lme)

sbp.svy<-svyglm(log(sbp)~log(bpb)+age+I(age^2)+bmi+factor(race) +factor(sex)+factor(smk)+factor(educ)+factor(alc),bpdsn)

summary(sbp.svy)

plot(sbp.svy)

par(mfrow=c(2,2))

plot(sbp.lm, which=1)

plot(sbp.lm, which=2)

plot(sbp.svy, which=1)

plot(sbp.svy, which=2)

par(mfrow=c(1,1))

##Let’s compare the beta coefficients for bpb

summary(sbp.lm)$coef[2,]

summary(sbp.lme)$tTable[2,]

summary(sbp.svy)$coef[2,]

1. Logistic regression can be fit in **svyglm()** as well. Let’s create a logistic regression model for hypertension and compute OR and 95% CI for an IQR increase in log(bpb).

htn.svy<-svyglm(htn~log(bpb)+ns(age,df=5)+ns(bmi,df=5) +factor(race)+factor(sex)+factor(educ)+hematoc+chol+packyrs +diag\_dm,family=quasibinomial(),bpdsn1)

summary(htn.svy)

* 1. **rmeta: Meta-analysis in R**

Computes a summary estimate and confidence interval from a collection of treatment effect estimates and standard errors. Allows fixed or random effects, optional quality weights.

meta.summaries(d, se, method=c("fixed", "random"), weights=NULL,

logscale=FALSE, names=NULL, data=NULL,

conf.level=0.95, subset=NULL,na.action=na.fail)

|  |  |
| --- | --- |
| d | Effect estimates |
| se | standard errors for d |
| method | Standard errors and default weights from fixed or random-effects? |
| weights | Optional weights (eg quality weights) |
| logscale | Effect is on a log scale? (for plotting) |
| names | labels for the separate studies |
| data | optional data frame to find variables in |
| conf.level | level for confidence intervals |
| subset | Which studies to use |
| na.action | a function which indicates what should happen when the data contain NAs. Defaults to [na.fail](http://rss.acs.unt.edu/Rdoc/library/modeltools/html/ModelEnv-class.html). |
| x,object | a meta.summaries object |
| summary | Plot the summary odds ratio? |
| summlabel | Label for the summary odds ratio |
| colors | see [meta.colors](http://rss.acs.unt.edu/Rdoc/library/rmeta/html/meta.colors.html) |
| xlab | label for the effect estimate axis. |

DETAILS

* The summary estimate is a weighted average. If weights are specified they are used, otherwise the reciprocal of the estimated variance is used.
* The estimated variance is the square of se for a fixed analysis. For a **random** analysis a heterogeneity variance is estimated and added.
* The variance of a weighted average is a weighted average of the estimated variances using the squares of the weights. This is the square of the summary standard error.
* With the default weights these are the standard fixed and random effects calculations.

### load the survey package

library(rmeta)

1. Suppose you have the estimates from 16 cities of the effect of air pollution (PM2.5) on Myocardial Infarction (MI) hospital admissions. We want to obtain the combined effect across all the cities.

city<-1:16

coef<-c(0.00155,0.00445,-0.00597,0.00237,0.00031,-0.00035,0.00206, 0.00127,-0.00395,-0.00434,0.00241,0.00730,-0.00175,0.00117, 0.00007,0.00350)

se<-c(0.013589379,0.003224905,0.002400626,0.001797566, 0.001019834,0.002658267,0.001529252,0.002388748,0.002583907, 0.005885021,0.00122501,0.00597465,0.002546777,0.002406304, 0.000913846,0.00517299)

model<-meta.summaries(coef, se, names=city,method="random")

##method="random" estimates and adds a heterogeneity variance

summary(model)

1. Plot the city-specific and summary results

plot(model)

metaplot(coef, se)

1. Get combined estimate, its SE and RR and 95% CI for 10 μg/m3 increase in PM2.5.

model$summary

model$summ

model$se.summary

model$se

exp(model$summ\*10)

exp((model$summ-1.96\*model$se)\*10)

exp((model$summ+1.96\*model$se)\*10)

* 1. **Other R packages for epidemiology?**

A number of more useful R packages for epidemiologic research are available. You know R languages now, so why not try? Manuals and sample codes are available for free, so just try!

* **ipw**: Estimate inverse probability weights
* **surveillance**: Temporal and spatio-temporal modeling and monitoring of epidemic phenomena
* **OutbreakTools**: Basic tools for the analysis of disease outbreaks
* **EpiModel**: Mathematical modeling of infectious disease
* **sna**: R for social network analysis
* **igraph**: a package for network analysis
* **SpatialEpi**: Methods and data for spatial epidemiology
* **sem**: Structural Equation Models
* **gap**: Genetic analysis package
* **CGEN**: an R package for analysis of case-control studies in genetic epidemiology

Thank you!