**Chapter 6**

**Regression Models in R**

# Load these packages all the time

library(foreign)

library(Hmisc)

library(epicalc)

# Set working directory

setwd("C:/epid674")

You can start R afresh if you didn't save the workspace image. Load the save R data and then attach.

load("nhanes3.rda")

attach(nhanes3)

**6.1. Linear Models in R**

Linear regression models can be fit using **lm()**. Let’s examine the association between systolic blood pressure (SBP) and blood lead levels with adjustment for potential confounding factors.

1. First check the distribution of SBP. Does the distribution of **log(sbp)** look closer to the normal distribution?

par(mfrow=c(2,1))

hist(sbp)

hist(log(sbp))

1. Let’s start with non-log transformed SBP first. Look at bivariate association between **sbp** and continuous covariates

par(mfrow=c(1,1))

plot(age,sbp)

plot(bmi,sbp)

plot(bpb,sbp)

1. First start creating a simple regression model for SBP, including only blood lead (**bpb**) in the model (crude).

sbp.model<- lm(sbp~bpb, na.action=na.omit, data=nhanes3)

summary(sbp.model)

summary.aov(sbp.model)

anova(sbp.model)

> summary(sbp.model)

Call:

lm(formula = sbp ~ bpb, data = nhanes3, na.action = na.omit)

Residuals:

Min 1Q Median 3Q Max

-51.586 -13.964 -3.757 10.612 114.521

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 121.66581 0.43552 279.36 <2e-16 \*\*\*

bpb 1.16166 0.08528 13.62 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.6 on 5072 degrees of freedom

Multiple R-squared: 0.03529, Adjusted R-squared: 0.0351

F-statistic: 185.6 on 1 and 5072 DF, p-value: < 2.2e-16

# From the output you get: the model you run, **estimates**, **Std. Error** and **t values** of the parameters, and a description of the model in terms of **Residuals standard error**, **R-squared** (proportion of total variation in the data which is explained by the model, and R-squared adjusted for the number of the parameter used in the fitted model) and **F-statistic**.

1. Age is an important predictor of SBP as well as a potential confounder of the association between SBP and blood lead.

sbp.model1<-lm(sbp~bpb+age, na.action=na.omit, data=nhanes3)

summary(sbp.model1)

1. The we can add race, which is a categorical variable, therefore we should use **as.factor()** or **factor()** which creates indicator variables for each category of race.

table(race)

sbp.model2<-lm(sbp~bpb+age+as.factor(race), na.action=na.omit, data=nhanes3)

summary(sbp.model2)

**WARNING:** R provides Type I sequential SS, not the default Type III marginal SS reported by SAS and SPSS. We will need to use the **drop1()** function to produce the familiar Type III results. It will compare each term with the full model.

anova(sbp.model2)

drop1(sbp.model2, ~., test="F")

## alternative

library(car)

Anova(sbp.model2, type="III")

1. We can also use the **update()** function to add variables to a pre-existing model rather than type all variables and arguments.

sbp.model2<-update(sbp.model1,.~.+ factor(race))

summary(sbp.model2)

1. Add other covariates to the model that are biologically plausible and previous literatures have suggested as important predictors (determinants): which variables would you include? Sex, BMI, education and cigarette smoking might be important. Let’s add these variables and see if they are significantly associated with SBP.

sbp.model3<- lm(sbp~bpb+age+factor(race)+factor(sex)+bmi +factor(educ)+factor(smk), na.action=na.omit, data=nhanes3)

summary(sbp.model3)

1. Some studies suggest alcohol consumption might be a confounder but others don’t. Let’s check if it is a statistical confounder using the 10% rule.

sbp.model4<-update(sbp.model3,.~.+ factor(alc))

summary(sbp.model4)

(summary(sbp.model4)$coef[2,1]-summary(sbp.model3)$coef[2,1])/summary(sbp.model3)$coef[2,1]

1. You could use **packyrs** (pack-years of smoking) rather than **smk** (smoking status) as a smoking variable. Does the variable **packyrs** give a better estimate than **smk**? Run the 2 models and look at the output.

sbp.model5<-update(sbp.model4,.~.-factor(smk)+packyrs)

summary(bp.model5)

# Both variables appear significant predictors of systolic blood pressure. There is no improvement in the model fit in terms of adjusted R2 (even lower adjusted R2), and more subjects were excluded due to missing values.

1. Model goodness-of-fit can be assessed using **AIC()**, especially two models are not nested. Smaller is better!

AIC(sbp.model4,sbp.model5)

1. Run two models one with **age** and one adding **age** as a quadratic function. Does the quadratic term improve the fit of the model?

Note: we need to use **I(expression)** to add a mathematical expression involving one or more variables as an independent predictor.

sbp.model6<-update(sbp.model4,.~.+I(age^2))

summary(sbp.model6)

1. We could use the **anova()** function to compare the 2 models and see if the quadratic term improves the model. This is useful when two models are nested.

anova(sbp.model4, sbp.model6, test="F")

1. Let us say the **sbp.model6** is the final model. What is the relationship between blood lead and SBP? Interpret your result.

**Regression Diagnostic**

1. Now let’s see if the model follows the assumptions of a linear model. Four assumptions:
   1. Linearity: constant slope
   2. Normality
   3. Independence
   4. Constant variance

To check a): check partial residual plots.

To check b): look at the residuals. Use other plots as the Q-Q plot (normal probability plots): the scatterplot should lie on the diagonal straight line.

To check c) and d) plot of residuals vs fitted data. If the data are independent there should be no pattern in the data; if the variance is not constant you will see an increasing or decreasing cloud. (As in **plot(bpb ,sbp)**). We will learn what to do when this assumption are violated.

In the case of linear model, the plot of the model gives diagnostic plots

par(mfrow=c(1,1))

plot(sbp.model6)

par(mfrow=c(2,2))

plot(sbp.model6 , id.n = 5) # "**id.n**": number of points to be labelled in each plot, starting with the most extreme

* Panel A: a plot of residuals vs. fitted values – if there may be a pattern in the residuals that suggests that we should be fitting a curve rather than a line.

Plot(sbp.model6, which=1)

* Panel B: a normal probability plot of residuals (Q-Q plot) – if residuals are from f normal distribution points should lie, to within statistical error, close to a line.

Plot(sbp.model6, which=2)

* Panel C: a plot of standardized residuals vs. fitted values – this is designed for examining the constancy of the variance. Look for standardized residuals that are >2 or <-2.

Plot(sbp.model6, which=3)

* Panel D: a plot of residuals vs. leverage – this identifies residuals that are influential in determining the form of the regression line. Influential points are commonly taken to be those with Cook’s distances that are >1.

Plot(sbp.model6, which=5)

* Cook’s distance which measures the extent to which the line would change if the point were omitted can be obtained with an option which=4.

Plot(sbp.model6, which=4)

1. How about log transformed blood pressure (sbp)? Fit for log(sbp) and plot regression diagnosis.

sbp.model6.log<-update(sbp.model6,.-sbp+log(sbp)~.)

summary(sbp.model6.log)

plot(sbp.model6.log)

hist(residuals(sbp.model6.log),nclass=20)

par(mfrow=c(2,2))

plot(sbp.model6, which=1)

plot(sbp.model6.log, which=1)

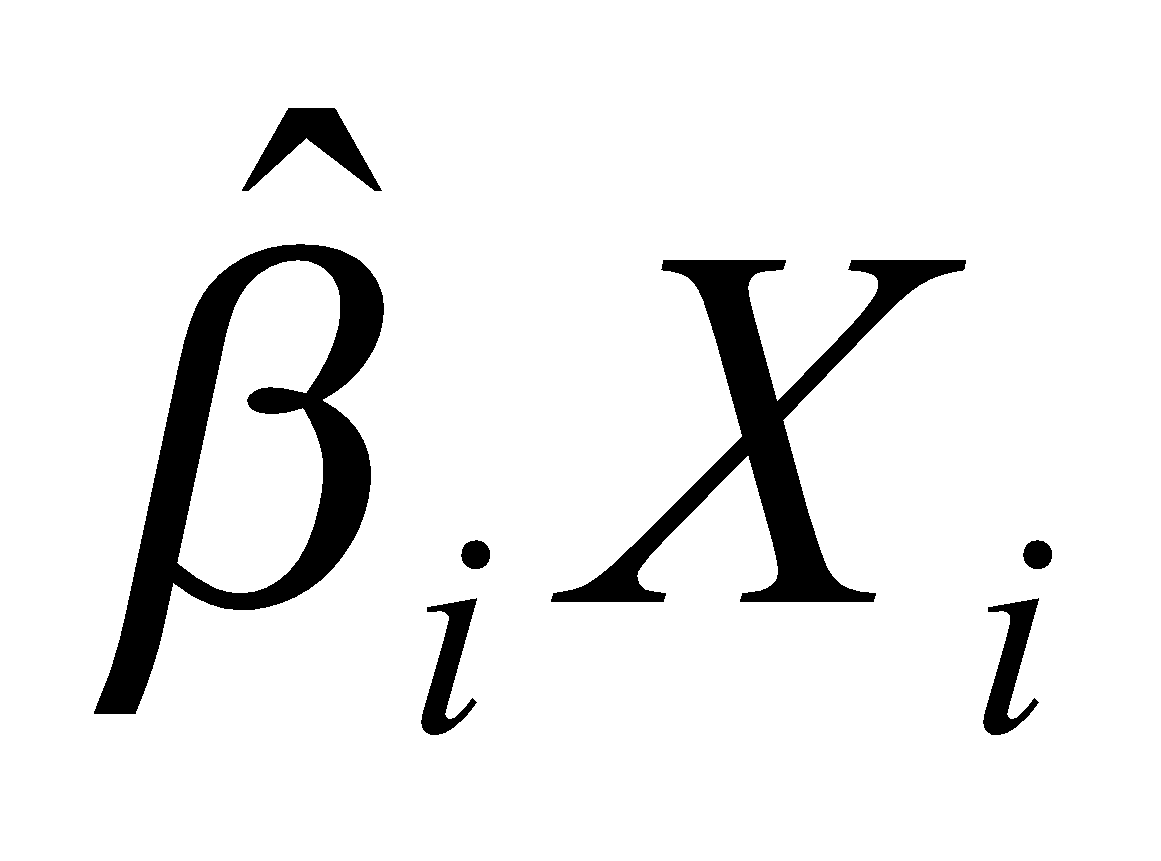
hist(residuals(sbp.model6), main="Histogram of res(SBP)")

hist(residuals(sbp.model6.log), main="Histogram of res(log(SBP))")

par(mfrow=c(1,1))

1. Partial Residual Plot: plot that shows the relationship between a given independent variable and the response variable given that other independent variables are also in the model.

Partial residual plots are formed as

Residuals +  versus *Xi*

This can be done by '**termplot()**'.

termplot(sbp.model6, partial.resid=TRUE, col.res="gray30")

1. Suppose you decided to go with non-log transformed SBP. Compute an effect estimate (difference in SBP) and 95% confidence intervals per one unit and IQR increase in blood lead.

summary(sbp.model6)$coef

summary(sbp.model6)$coef[2,1]

summary(sbp.model6)$coef[2,2]

# for one unit increase

change<-summary(sbp.model6)$coef[2,1]

l95ci<-summary(sbp.model6)$coef[2,1]-1.96\*summary(sbp.model6)$coef[2,2]

u95ci<-summary(sbp.model6)$coef[2,1]+1.96\*summary(sbp.model6)$coef[2,2]

change

l95ci

u95ci

# for an IQR increase

IQR(bpb)

change.iqr<-IQR(bpb)\*summary(sbp.model6)$coef[2,1]

l95ci.iqr<-IQR(bpb)\*(summary(sbp.model6)$coef[2,1]-1.96\*summary(sbp.model6)$coef[2,2])

u95ci.iqr<-IQR(bpb)\*(summary(sbp.model6)$coef[2,1] +1.96\*summary(sbp.model6)$coef[2,2])

change.iqr

l95ci.iqr

u95ci.iqr

1. What to report if the outcome is log-transformed? Because beta coefficients indicate changes in log(SBP), they are not directly interpretable. Consider what to report in logistic regression models where the binary outcome is transformed using the logit function. To make effect estimates interpretable, back-transform beta’s (i.e., exponentiate) to the actual scale. They look similar to relative risk. In this case, percent differences are typically computed and reported. Let’s compute percent increases in SBP (95% confidence intervals) per one unit and IQR increase in blood lead.

summary(sbp.model6.log)$coef

summary(sbp.model6.log)$coef[2,1]

summary(sbp.model6.log)$coef[2,2]

exp(summary(sbp.model6.log)$coef[2,1])

#exp(sbp.model6.log$coef[2])

p.change<-100\*(exp(summary(sbp.model6.log)$coef[2,1])-1)

l95ci<-100\*(exp(summary(sbp.model6.log)$coef[2,1]-1.96\*summary(sbp.model6.log)$coef[2,2])-1)

u95ci<-100\*(exp(summary(sbp.model6.log)$coef[2,1]+ 1.96\*summary(sbp.model6.log)$coef[2,2])-1)

p.change

l95ci

u95ci

IQR(bpb)

p.change.iqr<-100\*(exp(IQR(bpb)\*summary(sbp.model6.log)$coef[2,1])-1)

l95ci.iqr<-100\*(exp(IQR(bpb)\*(summary(sbp.model6.log)$coef[2,1]-1.96\*summary(sbp.model6.log)$coef[2,2]))-1)

u95ci.iqr<-100\*(exp(IQR(bpb)\*(summary(sbp.model6.log)$coef[2,1]+ 1.96\*summary(sbp.model6.log)$coef[2,2]))-1)

p.change.iqr

l95ci.iqr

u95ci.iqr

1. Our final model assumes a linear dose-response relationship. Is it true? The linearity assumption can be evaluated using a smoothing function. The package ‘**mgcv**’ supports penalized splines and smoothing splines. See more details at <https://cran.r-project.org/web/packages/mgcv/mgcv.pdf>.

library(mgcv)

sbp.model6.gam<-gam(sbp~s(bpb)+age+factor(race)+factor(sex)+bmi +factor(educ)+factor(smk)+factor(alc)+I(age^2), na.action=na.omit, data=nhanes3)

summary(sbp.model6.gam)

plot(sbp.model6.gam)

plot(sbp.model6.gam, xlab="blood lead (ug/dL)", ylab="Change in SBP")

sbp.model7<-lm(sbp~log(bpb)+age+factor(race)+factor(sex)+bmi +factor(educ)+factor(smk)+factor(alc)+I(age^2), na.action=na.omit, data=nhanes3)

summary(sbp.model7)

summary(sbp.model6)

## compare model6 and model7

AIC(sbp.model6, sbp.model7)

## **Effect modification by sex**

table(sex)

1. Create a simple regression model for blood pressure among men: **subset=(SEX==1)** and women **subset=(SEX==2)**.

sbp.model6.male<-lm(sbp~bpb+age+factor(race)+bmi+factor(educ)

+factor(smk)+factor(alc)+I(age^2),data=nhanes3,subset=(sex==1))

summary(sbp.model6.male)

1. Now run the regression for women and compare the results for men and women.

sbp.model6.female<-lm(sbp~bpb+age+factor(race)+bmi +factor(educ)+factor(smk)+factor(alc)+I(age^2),data=nhanes3, subset=(sex==2))

summary(sbp.model6.female)

1. Effect modification can also be assessed using an interaction term.

sbp.model6.int<-lm(sbp~bpb\*factor(sex)+age+factor(race)+bmi +factor(educ)+factor(smk)+factor(alc)+I(age^2),data=nhanes3)

# below is the same

sbp.model6.int<-lm(sbp~bpb+factor(sex)+bpb\*factor(sex)+age +factor(race)+bmi+factor(educ)+factor(smk)+factor(alc) +I(age^2),data=nhanes3)

summary(sbp.model6.int)

**6.2. Exercises**

Using the dataset “**bpa.sas7bdat**”, answer the followings.

Problem 1. Build a regression model for the association between BMI and urinary BPA. Include urinary creatinine (ucr), age, gender, race-ethnicity, education and household income as important confounders. Check if smoking status is a confounder using a 10% rule. Based on your final model, what is the association between BMI and urinary BPA? Compute an effect estimate (difference in BMI) and 95% confidence intervals per one unit and IQR increase in urinary BPA.

Problem 2. Verify the assumptions of a linear model for the final model. Create a histogram of the residuals and check if the distribution of residuals is skewed. Do you think log-transformation of BMI would improve the model fit?

Problem 3. Check if gender modifies the association between urinary BPA and BMI. Which group is more susceptible?

**6.3. Generalized Linear Models in R**

Generalized linear models can be fit using **glm()**. The error distribution and link function can be defined by **family=gaussian**, **family=binomial**, **family=poisson**, etc. Let’s examine the association between hypertension (**htn**) and **bpb** using logistic regression models.

## **Logistic regression**

1. First check the distribution of SBP. Does the distribution of **log(sbp)** look closer to the normal distribution?

##Look at hypertension (htn)

tab1(htn, graph=F)

1. GLM are a unified approach to fit data that have a different error distribution, such as normal, binomial, Poisson, etc. Create a logistic regression model for hypertension with main effects for the variables fit in the linear models above. Note that for the binomial error distribution the default link is logit for a logistic model.

htn.model<-glm(htn~bpb+age+factor(sex)+factor(race)+bmi +factor(educ)+factor(smk)+factor(alc), family=binomial, na.action=na.omit, data=nhanes3)

summary(htn.model)

1. Suppose that you have no a priori knowledge about potential confounders. Using ‘step()’, you can select a formula-based model by AIC.

# first, define a matrix of predictors

X<-nhanes3[,c(3:8, 13:25, 29, 30)]

Y<-htn

dat=data.frame(cbind(Y,X))

dat<-na.omit(dat) #note: dataset should be complete

fit.start=lm(Y~1,data=dat)

summary(fit.start)

fit.full=lm(Y~.,data=dat)

summary(fit.full)

# forward

full<-formula(glm(Y~.,data=dat, family=binomial))

full

fit.forward=step(fit.start,direction='forward',scope=full)

summary(fit.forward)

# if you want to keep bpb

fit.forward1=step(fit.start,direction='forward',scope=list(lower=~bpb, upper=full)) # error

fit.forward1=step(glm(Y~bpb, family=binomial, data=dat), direction='forward', scope=list(upper=full))

summary(fit.forward1)

# backward (always keep bpb in the model)

fit.backward=step(fit.full,direction='backward', scope=list(lower=~bpb))

summary(fit.backward)

# stepwise (always keep bpb in the model)

fit.step=step(fit.full,direction='both', scope=list(lower=~bpb))

summary(fit.step) # this is exactly the same as backward

fit.step1=step(glm(Y~bpb, family=binomial, data=dat), direction='both', scope=list(lower=~bpb, upper=full))

summary(fit.step1)

1. Let’s compute odds ratios. The **logistic.display()** function from epicalc will provide exp(beta).

logistic.display(htn.model)

1. Regression diagnostics

plot(htn.model)

plot(htn.model, which=4)

par(mfrow=c(2,2))

plot(htn.model)

par(mfrow=c(1,1))

termplot(htn.model)

termplot(htn.model, se=T)

termplot(htn.model, se=T, partial.resid=T)

## **Poisson regression**

Poisson regression deals with outcome variables that are counts in nature (whole numbers or integers). Independent covariates are similar to those encountered in linear and logistic regression. In epidemiology, Poisson regression is used for analyzing grouped cohort data, looking at incidence density among person-time contributed by subjects of similar characteristics of interest.

The dataset **Montana** from **epicalc** was extracted from an occupational cohort study conducted to test the association between respiratory deaths and exposure to arsenic in the industry, after adjusting for various other risk factors. The main outcome variable is '**respdeath**'. This is the count of the number of deaths among '**personyrs**' or person-years of subjects in each category. The other variables are independent covariates including age group '**agegr**', period of employment '**period**', starting time of employment '**start**' and the level of exposure to arsenic during the study period '**arsenic**'.

1. Read in the data first and examine the variables

data(Montana)

summ(Montana)

head(Montana, 10)

hist(Montana$respdeath)

par(mfrow=c(2,2))

tab1(Montana$agegr)

tab1(Montana$period)

tab1(Montana$start)

tab1(Montana$arsenic)

1. Using **factor()**, label the categorical variables, **agegr**, **period**, **start** and **arsenic**.

Montana$agegr<-factor(Montana$agegr, labels=c("40-49","50-59","60-69","70-79"))

Montana$period<-factor(Montana$period, labels=c("1938-1949", "1950-1959", "1960-1969", "1970-1977"))

Montana$start<-factor(Montana$start, labels=c("pre-1925", "1925 & after"))

Montana$arsenic<-factor(Montana$arsenic, labels=c("<1 year", "1-4 years","5-14 years", "15+ years"))

tab1(Montana$agegr, missing=F)

tab1(Montana$period, missing=F)

tab1(Montana$start, missing=F)

tab1(Montana$arsenic, missing=F)

par(mfrow=c(1,1))

1. Let’s compute incidence rate by age and period. Firstly, create a table for total person-yrs; then create a table for the number of death; finally compute incidence per 10,000 person-yrs for each cell.

table.pyears<-tapply(Montana$personyrs, list(Montana$period, Montana$agegr), sum)

table.deaths<-tapply(Montana$respdeath, list(Montana$period, Montana$agegr), sum)

table.inc10000<-table.deaths/table.pyears\*10000

table.inc10000

1. Now, create a time-series plot of the incidence

plot.ts(table.inc10000, plot.type="single", xlab="", ylab="#/10,000 person-years", xaxt="n", col=c("black", "blue","red","green"), lty=c(2,1,1,2), las=1)

points(rep(1:4,4), table.inc10000, pch=22, cex=table.pyears/sum(table.pyears)\*20)

title(main = "Incidence by age and period")

axis(side = 1, at = 1:4, labels = levels(Montana$period))

legend("topleft", legend=levels(Montana$agegr)[4:1], col=c("green","red","blue","black"),bg="white",lty=c(2,1,1,2))

# check arsenic

tab1(Montana$arsenic)

tapply(Montana$respdeath, Montana$arsenic, mean)

tapply(Montana$personyrs, Montana$arsenic, mean)

1. Let’s fit a Poisson model. The option '**offset=log(personyrs)**' allows the variable '**personyrs**' to be the denominator for the counts of '**respdeath**'. This is because we need to account for different population sizes in each group unless data are from same-size populations.

resp.mode11<-glm(respdeath~period, offset=log(personyrs), family=poisson, data=Montana)

summary(resp.mode11)

resp.mode12<-glm(respdeath~agegr, offset=log(personyrs), family=poisson, data=Montana)

summary(resp.mode12)

resp.mode13<-glm(respdeath~period+agegr, offset=log(personyrs), family=poisson, data=Montana)

summary(resp.mode13)

AIC(resp.mode11, resp.mode12, resp.mode13)

## model2 is better

resp.mode14<-glm(respdeath~agegr+arsenic, offset=log(personyrs), family=poisson, data=Montana)

summary(resp.mode14)

# is there a linear trend across arsenic exposure?

resp.mode14.lin<-glm(respdeath~agegr+as.numeric(arsenic), offset=log(personyrs), family=poisson, data=Montana)

summary(resp.mode14.lin)

1. Let’s compute incidence rate (density) ratios. The **idr.display()** function from epicalc will provide exp(beta).

## compute IRR

idr.display(resp.mode14)

**6.4. Exercises**

The association between BPA and type-2 diabetes in a human population was reported first by Dr. David Melzer group (Lang et al., JAMA 2008). Using NHANES 2003-2004 data, they found that a 1-SD increase in BPA was associated with a 39% increased odds of type-2 diabetes (OR=1.39, 95% CI, 1.21 to 1.60, p<0.001) after controlling for age, gender, race/ethnicity, education, income, smoking status, BMI, waist circumference, and urinary creatinine. Let’s try to examine the same research question using an expanded dataset we have (NHANES 2003-2008, **bpa.sas7bdat**).

Problem 1. Construct your models as done by Lang: **Model 1** adjusted for age, gender, and urinary creatinine; and **Model 2** additionally adjusted for race/ethnicity, education, income, smoking, BMI, waist circumference. Compute odds ratios and 95% confidence intervals for a 1-SD increase in BPA in each model. Interpret your results.

**6.5. Matched Case-Control Study in R**

For a matched case-control study, when a case is recruited, a control, or a set of controls, can be selected to match with the case in some parameters such as age and sex. In the analysis of matched sets, comparison is made within each matched set rather than one series against the other. For this exercise, we will use the datasets **VC1to1** and **VC1to6** from **epicalc** where a matched case-control study testing whether smoking, drinking alcohol and working in the rubber industry are risk factors for *esophageal cancer*. Each case was matched with his/her neighbors of the same sex and age group. The matching ratio varies from 1:1 to 1:6. **VC1to6** is the full dataset whereas **VC1to1** has the number of controls per case reduced to 1 for all matched sets.

1. Read in the data **VC1to1** first and examine the variables.

data(VC1to1)

summ(VC1to1)

head(VC1to1)

1. There are 26 matched pairs (‘**matset**’). The variable ‘**case**’ is coded as 1 for case and 0 for control. To facilitate data exploration, let’s reshape the data. The ‘**reshape()**’ function reshapes a data frame between ‘wide’ format with repeated measurements in separate columns of the same record (subject) and ‘long’ format with the repeated measurements in separate rows. After reshaping the ‘long’ format data to ‘wide’, check the smoking status between cases and controls in each matched pair.

wide <- reshape(VC1to1, timevar="case", v.names=c("smoking","rubber", "alcohol"), idvar="matset", direction="wide")

head(wide,3)

table(wide$smoking.1, wide$smoking.0, dnn=c("smoking in case", "smoking in control"))

#dnn: dimnames names (the names to be given to the dimensions in the result)

table(wide$rubber.1, wide$rubber.0, dnn=c("rubber in case", "rubber in control"))

table(wide$alcohol.1, wide$alcohol.0, dnn=c("alcohol in case", "alcohol in control"))

1. The ratio of discordant counts indicates the conditional odds ratio (aka McNemar’s OR). In **epicalc**, the **matchTab()** function can be used to analyze the matched set.

matchTab(VC1to1$case, VC1to1$smoking, strata=VC1to1$matset)

matchTab(VC1to1$case, VC1to1$rubber, strata=VC1to1$matset)

matchTab(VC1to1$case, VC1to1$alcohol, strata=VC1to1$matset)

1. If there is no serious problem on scarcity of diseased cases, the best ratio of matching is one case per control. However, when the disease of interest is rare, it is often cost-effective to increase the number of controls per case. Given that esophageal cancer is rare, the full dataset with up to 1:6 matching may be better.

data(VC1to6)

summ(VC1to6)

VC1to6[,]

1. Let’s explore the effect of smoking in the full data using **matchTab()**.

matchTab(VC1to6$case, VC1to6$smoking, strata=VC1to6$matset)

1. Conditional logistic regression can be run using **clogit()** from the **survival** package.

library(survival)

clogit1<-clogit(case~smoking+alcohol+strata(matset), data= VC1to1)

summary(clogit1)

clogit2<-clogit(case~smoking+alcohol+strata(matset), data= VC1to6)

summary(clogit2)

1. Let’s compute odds ratios. The c**logistic.display()** function from epicalc will provide exp(beta).

clogistic.display(clogit1)

clogistic.display(clogit2)

**6.6. Survival Analysis in R**

Survival analysis examines and models the time it takes for events to occur. In a cohort study, a person is followed up from a starting time to the end of the study or to the time the follow-up has been terminated by the outcome event, whichever comes first. For subjects whose events take place before the end of the study, the total duration of time is known. For the subjects whose follow-up times end without the event, the end status is called ‘censored’ because the actual duration of time to the event is unknown or ‘censored’ by the study. The outcome variable for each subject is therefore composed of ‘time’ and the ‘status’ at the end.

In R, the package ‘survival’ provides all the functions necessary to analyze survival type data, including descriptive statistics, two-sample tests, parametric accelerated failure models, Cox model, and interval censoring for parametric models. The functions of interest are ‘**Surv**’, ‘**survfit**’, ‘**survdiff**’, ‘**survreg**’, and ‘**coxph**’.

### load the survival package

library(survival)

Let’s examine the association between total mortality (**d\_total**) and blood lead levels from **nhnaes3**.

1. First, let’s examine univariate distributions of total mortality (d\_total) and person months of follow-up (pmon\_mec).

tab1(d\_total)

summ(pmon\_mec)

1. In order to analyze survival data in R, we need to create an object of class **Surv(time, event)**, which combines the information of time and status of event in a single object. The status variable must be either numeric or logical. If numeric, there are two options. Values must be either 0=censored and 1=event, or 1=censored and 2=event. If logical, FALSE=censored and TRUE=event. In our dataset, mortality status is defined as 0=censored and 1=death.

surv.total<-Surv(pmon\_mec, d\_total)

surv.total

# The plus (+) sign in the example above indicates that the actual ‘time’ is beyond those values but were censored.

1. **Kaplan-Meier Life Table and Curve (plot)**: a tabulation of the survival, event and survival probability over time. This table can be obtained by ‘**survfit()**’.

fit.total<-survfit(Surv(pmon\_mec, d\_total)~1)

summary(fit.total)

1. **Survival by different levels of covariates**: by sex.

fit.total.sex<-survfit(Surv(pmon\_mec, d\_total)~sex)

fit.total.sex

summary(fit.total.sex)

plot(fit.total.sex, col=c("blue","red"), lty=c(1,2))

plot(fit.total.sex, ylim=c(0.6,1.0), col=c("blue","red"), lty=c(1,2), mark.time=F)

title(main="Time since follow-up", xlab="Time (months)", ylab="Survival probability")

legend("topright", legend=c("Men","Women"), lty=c(1,2), col=c("blue","red"))

# Statistical comparison among survival curves: Survival curves can be tested for statistical difference with ‘**survdiff()**’.

survdiff(Surv(pmon\_mec, d\_total)~sex)

1. **Cox regression**: Survival outcomes can be tested for more than one predictor using regression modelling. There are many parametric regression choices for the survival object. Each of them has a specific assumption about the distribution of the survival probability over time. In epidemiologic studies, the most popular regression choice for survival analysis is Cox regression, which has no assumption regarding the hazard function.

cox.bpb<-coxph(Surv(pmon\_mec, d\_total)~bpb)

summary(cox.bpb)

## fit tertiles of blood lead

bpb3<-cut2(bpb, g=3)

tab1(bpb3)

# K-M Life table and curve

fit.total.bpb3<-survfit(Surv(pmon\_mec, d\_total)~bpb3)

summary(fit.total.bpb3)

plot(fit.total.bpb3, col=c(1:3), lty=c(1:3), mark.time=F)

plot(fit.total.bpb3, col=c(1:3), lty=c(1:3), mark.time=F, ylim=c(0.6,1.0))

title(main="Survival curve in relation to blood lead levels", xlab="Time (months)", ylab="Survival probability")

legend(30,0.7, legend=c("Q1","Q2","Q3"), lty=c(1:3), col=c(1:3))

#crude

cox.bpb3<-coxph(Surv(pmon\_mec, d\_total)~bpb3)

summary(cox.bpb3)

#adjusted

cox.bpb3.adj<-coxph(Surv(pmon\_mec,d\_total)~bpb3+age+factor(sex) +factor(race)+factor(educ)+factor(smk)+factor(alc))

summary(cox.bpb3.adj)

1. **Test for the proportional hazards assumption**: The **cox.zph()** function tests proportionality of all the predictors in the model by creating interactions with time using the transformation of time specified in the transform option. The column ‘rho’ is the Pearson product-moment correlation between the scaled Schoenfeld residuals and time for each covariate. The last row contains the global test for all the interactions tested at once. A p-value less than 0.05 indicates a violation of the proportionality assumption.

test.prop<-cox.zph(cox.bpb3.adj)

test.prop

## Display a graph of the scaled Schoenfeld residuals, along with a smooth curve

plot(test.prop) # for all variables

plot(test.prop, var=1)

plot(test.prop, var=2)

abline(h=0, lty=3, col=2)

1. Sex has a significant P-value suggesting that the assumption is violated. A possible solution is to do a stratified analysis on sex.

cox.bpb3.adj1<-coxph(Surv(pmon\_mec, d\_total)~bpb3+age+strata(sex) +factor(race)+factor(educ)+factor(smk)+factor(alc))

summary(cox.bpb3.adj1)

test.prop1<-cox.zph(cox.bpb3.adj1)

test.prop1

**6.7. Writing your own functions (macros)**: Like macros in SAS, multiple procedures can be executed repeatedly in R. This is especially useful and efficient when you examine “***high-throughput***” data (e.g., multiple single nucleotide polymorphisms (SNPs)) and repeat the same regression models by switching hundreds to millions of independent variables. Functions can be created using the **function(<arguments>)**.

1. Let’s make a simple macro that calculates the mean and standard deviation at the same time. You can define your own function within the operator ***brace*** “**{}**”. Note that to order R to print the multiple results, we need to add **print()** to the function.

mystats<-function(x)

{

print(mean(x, na.rm=T))

print(sd(x, na.rm=T))

}

mystats(age)

mystats(bpb)

# get the results in vector form

mystats<-function(x)

{

mymean<-mean(x, na.rm=T)

mysd<-sd(x, na.rm=T)

c(mean=mymean, sd=mysd)

}

mystats(age)

mystats(bpb)

1. Assume that you are examining age-adjusted associations of SBP with from the 13th variables (**bmi**) to 25th variables (**packyrs**) (n=13). You want to run 13 linear regression models and save beta's and p-values.

summ(nhanes3)

test.var<-nhanes3[,c(13:25)]

head(test.var)

# example

mod<-lm(sbp~bmi+age, data=nhanes3, na.action=na.omit)

summary(mod)

# how to extract beta for bmi?

summary(mod)$coef[2,1]

# how to extract p-value for bmi?

summary(mod)$coef[2,4]

Test<-function(data, y, cov){

nvar<-ncol(data)

newdata<-data.frame(cbind(data, y, cov))

tmatrix<-data.frame(matrix(NA,2,nvar)) # 2 rows

colnames(tmatrix)<-colnames(data) # create a row for each var

rownames(tmatrix)<-c("beta","p")

for(i in 1:nvar){

ind<-data[,i]

model<-lm(y~ind+cov, data=newdata, na.action=na.omit)

tmatrix[1,i]<- summary(model)$coef[2,1]

tmatrix[2,i]<- summary(model)$coef[2,4]

}

return(tmatrix)

write.csv(tmatrix, file='sbp.results.csv')

}

Test(test.var, sbp, age)

**6.8. Exercises**

Problem 1. Using the NHANES3 dataset, examine the association between blood lead and cardiovascular mortality (**d\_cvd**). Use the same covariates (confounders) considered above. Check which covariates violate the proportional hazard assumption. Compute hazard ratio (HR) (95% CI) per for a 1-SD increase in blood lead.

Problem 2. Also, compare the lowest vs. the highest tertiles of blood lead and compute HR and 95% CI.