

RESET Test

The RESET test is a general specification test.¹ Typically, we specify the model as linear, entering the independent variables singly, without higher order polynomial terms. But it could be that logs or polynomials would have been more appropriate ways to specify the effects of the independent variables. A very quick way to test the adequacy of ones specification is the RESET test. The logic of the RESET test is as follows.

- 1) Estimate the parameters in your model and create a fitted value for the dependent variable.
- 2) Create two new variables: the square of the fitted value and the cube of the fitted value.
- 3) Add these two new variables back into your original regression and perform an F-test in which the null hypothesis is that the two new parameters are equal to zero. If the p-value is low enough, then reject the null hypothesis. Rejection implies that respecification would improve the model fit; acceptance implies that the current specification is OK.

Test for normality of residuals

The residuals from a regression should be normally distributed. If they are not, then your model is not appropriately specified. An easy test in R is the Shapiro-Wilk test for normality.² Applied to residuals, the null hypothesis is that the residuals are normally distributed. If the p-value is low enough (0.1 is an appropriate cutoff) then reject the null hypothesis.

Multiple Imputation

The usual procedure for handling missing data is *listwise deletion*. In a regression model of the form

$Y = \mathbf{X}\beta$, one drops any row where data are missing in Y or in one of the columns of \mathbf{X} . However, this method forgoes the information present in the dropped observations, which in turn has the potential to cause sample selection bias, so that the estimated coefficients are misleading. An alternative is *multiple imputation*, which uses Bayesian methods to impute values for the missing observations. This method is appropriate when the extant data provide information that can be used to predict the missing data.

King et al.³ (2001: 50-51) provide an intuitive presentation of the circumstances in which multiple imputation is appropriate. Consider the data matrix $\mathbf{D} = \{Y, \mathbf{X}\}$. Some elements of \mathbf{D} are missing (\mathbf{D}_{miss}) and others are observed (\mathbf{D}_{obs}), so that $\mathbf{D} = \{\mathbf{D}_{\text{miss}}, \mathbf{D}_{\text{obs}}\}$. Think of matrix \mathbf{M} as a “missingness indicator matrix” for \mathbf{D} , where cells take the value of “1” when the corresponding cell in \mathbf{D} is observed, and the value of “0” when the cell in \mathbf{D} is missing. Missing data can be divided into three types:

1. *Missing completely at random* (MCAR). The variable with missing values is uncorrelated with other variables, so that a missing value cannot be imputed. A good example of a variable whose missing values are missing completely at random would be the results of a coin toss. \mathbf{M} is not conditioned by \mathbf{D} : $P(\mathbf{M}|\mathbf{D}) = P(\mathbf{M})$.
2. *Missing at random* (MAR). The variable with missing values is correlated with other variables, *and* the values which happen to be missing are random (once controlled for by

¹ Ramsey, J.B. (1969). "Tests for Specification Errors in Classical Linear Least Squares Regression Analysis." *Journal of the Royal Statistical Society B*. 31(2): 350-371.

² Royston, Patrick. (1982) “An extension of Shapiro and Wilk’s W test for normality to large samples.” *Applied Statistics*, **31**, 115–124

³ King, Gary, James Honaker, Anne Joseph, and Kenneth Scheve. (2001). “Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation.” *American Political Science Review* 95(1): 49-69.

the other variables). In this case, a missing value can be imputed using the observed data: $P(\mathbf{M}|\mathbf{D})=P(\mathbf{M}|\mathbf{D}_{obs})$.

3. *Nonignorable* (NI) The probability that a cell is missing is a function of the missing value itself. That is, not just the observed values, but the missing values are needed to estimate the probability that a cell is missing: $P(\mathbf{M}|\mathbf{D})=P(\mathbf{M}|\mathbf{D})$. Here, missing values cannot be imputed.

It is only the second situation that permits imputation. In practice, though, one can usually bring in additional data (\mathbf{D}_{obs}) so that cases classified as NI become MAR and therefore imputable.

Listwise deletion produces inefficient estimates, which are also biased unless MCAR prevails. Multiple imputation provides estimates that are efficient and unbiased under both MCAR and MAR. Under NI, both listwise deletion and multiple imputation produce biased estimates. In general, then, multiple imputation should produce estimates that are more efficient and have less bias than estimates produced using listwise deletion (King et al. 2001: 50-51).

Multiple imputation uses iterated estimation methods to estimate missing values based on all extant values, in each round substituting newly estimated values for the missing values. Gibbs sampling is used to take m random draws from the probability distribution for the missing values, conditional on the variables used to estimate them. One then has m (typically 5 to 10) duplicates of the original data set, each with different values imputed for the missing data.

The equation $Y=\mathbf{X}\beta$ is estimated once for each of the m imputed data sets. This gives m estimates of the parameters β and their associated variances. These estimates are then combined to give the final estimate

of the parameters β_i and their variances, as shown in Rubin (1986: 76-77).⁴

The final estimate of each parameter β_i is simply the mean of the m estimates:

$$\bar{\beta}_i = \sum_{j=1}^m \hat{\beta}_{i,j} / m \tag{4}$$

To calculate the variances, one must consider both the m estimated variances, and the variation in the estimated parameters β_i across the m estimations. First, the mean of the m estimated variances for each parameter i :

$$\bar{U}_i = \sum_{j=1}^m \hat{u}_{i,j} / m \tag{5}$$

Then, the variance in the m estimated values of β_i :

$$B_i = \sum_{j=1}^m (\hat{\beta}_{i,j} - \bar{\beta}_i)^2 / (m - 1) \tag{6}$$

These are then combined to get the total variance in β_i :

$$T_i = \bar{U}_i + (1 + m^{-1}) B_i \tag{7}$$

The following relationship then gives the p-value for the null hypothesis that the true value of β_i equals β_0 .

$$Pr ob \left(F_{1,v} \leq \frac{(\beta_0 - \bar{\beta}_i)^2}{T_i} \right) \tag{8}$$

where the denominator degree of freedom v_i is given by

⁴ Rubin, Donald B. (1987). *Multiple Imputation for Nonresponse in Surveys*. New York: J. Wiley & Sons.

$$v_i = (m-1)(1+r_i^{-1})^2 \tag{9}$$

and r_i is given by

$$r_i = (1+m^{-1}) \frac{B_i}{\bar{U}_i} \tag{10}$$

The fraction of missing information, for each regression parameter, is

$$\gamma_i = \frac{\left(r_i + \frac{2}{v_i + 3} \right)}{r_i + 1} \tag{11}$$

Test statistics produced for each of the m estimations can also be combined to produce a final statistic, as shown in Rubin (1986: 78-81). If d_1, \dots, d_m are test statistics produced in the m estimations, distributed χ^2 with k degrees of freedom,⁵ then they can be combined as follows:

$$\hat{D}_m = \frac{\bar{d}_m - \frac{m-1}{m+1} r_m}{1+r_m}, \text{ where } \bar{d}_m = \sum_{j=1}^m d_j / m \tag{12}$$

\hat{D}_m is distributed F , with k and $(k+1)v/2$ degrees of freedom. Finding v requires a different method of estimating r :

⁵ If the initial test statistics d_j are distributed F , they can be converted to χ^2 by finding the p-value of the F -statistic (with k and f degrees of freedom) and then finding the χ^2 statistic with k degrees of freedom that has the same p-value (Rubin 1986: 79).

$$\hat{r}_m = \frac{(1+m^{-1})s_d^2}{2\bar{d}_m + [4\bar{d}_m^2 - 2ks_d^2]_+^{1/2}}, \text{ where}$$

$$s_d^2 = \sum_{j=1}^m (d_j - \bar{d}_m)^2 / (m-1) \tag{13}$$

The plus sign on the bottom right of the square bracket indicates that the expression within the bracket is set to zero when it is negative.

Homework

Herbert Barry III, Lili Josephson, Edith Lauer, and Catherine Marshall examined ethnographies for evidence of traits inculcated in childhood (Barry et al. 1976).⁶ The ages are approximately four through 12, though the beginning and end stages are defined emically (Barry et al. 1976: 85-86).

“The inculcated traits were coded, if possible, on the basis of reports of the pressures exerted by the people who train the child. The codes were also based on the behavior of the child and were inferred only with great caution from reports of the customary adult behavior or of adult ideology” (Barry et al. 1976: 91).

Barry et al. examined 13 different traits, arranged in five categories. The category called “sociability” contains the traits “generosity”, “honesty”, and “trust”.

“Generosity... refers to the specific behavior encouraged rather than a general attitude, but a wide range of actions may exemplify generosity. These include giving and sharing of food, possessions, time, or services to others of the community or outsiders, e.g., sharing the product of a hunt among the

⁶ Barry, Herbert,III, Lili Josephson, Edith Lauer, and Catherine Marshall. (1976). “Traits Inculcated in Childhood: Cross Cultural Codes 5.” *Ethnology* 15:83-114.

community members whether or not they were active in it its attainment, or sharing and giving treats or toys. Expressions of kindness and affection are included, especially toward younger children or aged, ill, or infirm people. Reciprocity is not necessarily generosity.” (Barry et al. 1976: 95)

“Trust... or mutual confidence... refers to confidence in social relationships, especially towards community members outside the family, e.g., children are welcome in any home in the village, possessions are left unguarded. Sorcery and witchcraft generally indicate a low rating of trust. The code is omitted where in-group and out-group differ widely.” (Barry et al. 1976: 95)

“Honesty... refers to desire and strong approval for truthfulness under all circumstances. Stealing or other criminal or anti-social behavior by children indicate low honesty. It is possible to have high emphasis on honesty towards one’s own social group along with approval for lying, cheating, and stealing against an out-group. It takes into account societies where the concept of honesty differs from ours, e.g., lying is considered ‘smart’, but stealing is dishonest.” (Barry et al. 1976: 95).

The three variables are coded in the SCCS as:

Generosity inculcated	v334
Trust inculcated	v335
Honesty inculcated	v336

- Pick one of these variables as a dependent variable, and build a model explaining it. Each group explains a different dependent variable.
- Employ your usual modeling procedure: *ex ante* discussion to justify the unrestricted model; examine t-statistics to produce a restricted model; use F-statistic to justify the restricted model; tell the story you extracted from the data in your *ex post* discussion.
- In the *notes* section under your tables for regression output, include the following: number of imputations; number of observations; R-squared; RESET test statistic and p-value; Shapiro-Wilk statistic and p-value; F-test on restrictions statistic and p-value.
- Three things will be new this week. The first is a new diagnostic: the RESET test. Try to find a model specification that allows you to accept the null hypothesis that the model is correctly specified.
- The second new thing is the Shapiro-Wilk test for normality of residuals. Find a model specification allowing you to accept the null hypothesis that the residuals are normally distributed.
- The third new thing this week is multiple imputation. In R program *r04a.R*, place your selected independent variables so that you create 10 imputed data sets with these variables. In R program *r04b.R*, you will estimate your model on each of these 10 estimated data sets, combining results using Rubin’s formulas.

R file: S:\TEFF\662\R\r04a.R

```
#MI--make the imputed datasets
#--change the following path to the directory with your data and program--
setwd("e:/class/662/R/")
rm(list=ls(all=TRUE))
options(echo=TRUE)
#--you need the following two packages--you must install them first--
library(foreign)
library(mice)

#--To find the citation for a package, use this function:---
citation("mice")

#-----
#--Read in data, rearrange----
#-----

#--Read in the SCCS dataset---
load(url("http://frank.mtsu.edu/~eaeff/downloads/sccs200909.Rdata"),.GlobalEnv)
#--rename dataframe--shorter name is easier--
SCCS<-sccs200909
#--Read in auxilliary variables---
load(url("http://frank.mtsu.edu/~eaeff/downloads/vaux.Rdata"),.GlobalEnv)
row.names(vaux)<-NULL

#--look at first 6 rows of vaux--
head(vaux)
#--look at field names of vaux--
names(vaux)
#--check to see that rows are properly aligned in the two datasets--
#--sum should equal 186---
sum((SCCS$socname==vaux$socname)*1)
#--remove the society name field--
vaux<-vaux[,-28]
names(vaux)

#--Two nominal variables: brg and rlg----
#--brg: consolidated Burton Regions-----
#0 = (rest of world) circumpolar, South and Meso-America, west North America
#1 = Subsaharan Africa
#2 = Middle Old World
#3 = Southeast Asia, Insular Pacific, Sahul
#4 = Eastern Americas
#--rlg: Religion---
#'0 (no world religion)'
#'1 (Christianity)'
#'2 (Islam)'
#'3 (Hindu/Buddhist) '

#--check to see number of missing values in vaux,
#--whether variables are numeric,
#--and number of discrete values for each variable---
vvn<-names(vaux)
pp<-NULL
for (i in 1:length(vvn)){
  nmiss<-length(which(is.na(vaux[,vvn[i]])))
  numeric<-is.numeric(vaux[,vvn[i]])
  numDiscrVals<-length(table(vaux[,vvn[i]]))
  pp<-rbind(pp,cbind(data.frame(numeric),nmiss,numDiscrVals))
}
row.names(pp)<-vvn
```

```

pp

#--extract variables to be used from SCCS, put in dataframe fx--
fx<-data.frame(
  socname=SCCS$socname,socID=SCCS$ord,
  valchild=(SCCS$v473+SCCS$v474+SCCS$v475+SCCS$v476),
  cultints=SCCS$v232,roots=(SCCS$v233==5)*1,
  cereals=(SCCS$v233==6)*1,gath=SCCS$v203,hunt=SCCS$v204,
  fish=SCCS$v205,anim=SCCS$v206,femsubs=SCCS$v890,
  pigs=(SCCS$v244==2)*1,milk=(SCCS$v245>1)*1,plow=(SCCS$v243>1)*1,
  bovines=(SCCS$v244==7)*1,tree=(SCCS$v233==4)*1,
  foodtrade=SCCS$v819,foodscarc=SCCS$v1685,
  ecorich=SCCS$v857,popdens=SCCS$v156,pathstress=SCCS$v1260,
  CVrain=SCCS$v1914/SCCS$v1913,rain=SCCS$v854,temp=SCCS$v855,
  AP1=SCCS$v921,AP2=SCCS$v928,ndrymonth=SCCS$v196,
  exogamy=SCCS$v72,ncmallow=SCCS$v227,famsize=SCCS$v80,
  settype=SCCS$v234,localjh=(SCCS$v236-1),superjh=SCCS$v237,
  moralgods=SCCS$v238,fempower=SCCS$v663,
  sexratio=1+(SCCS$v1689>85)+(SCCS$v1689>115),
  war=SCCS$v1648,himilexp=(SCCS$v899==1)*1,
  money=SCCS$v155,wagelabor=SCCS$v1732,
  migr=(SCCS$v677==2)*1,brideprice=(SCCS$v208==1)*1,
  nuclearfam=(SCCS$v210<=3)*1,pctFemPolyg=SCCS$v872
)

#--look at first 6 rows of fx--
head(fx)
sapply(fx,function(x) class(x))

#--check to see number of missing values--
#--also check whether numeric--
vvn<-names(fx)
pp<-NULL
for (i in 1:length(vvn)){
  nmiss<-length(which(is.na(fx[,vvn[i]])))
  numeric<-is.numeric(fx[,vvn[i]])
  pp<-rbind(pp,cbind(nmiss,data.frame(numeric)))
}
row.names(pp)<-vvn
pp

#--identify variables with missing values--
z<-which(pp[,1]>0)
zv1<-vvn[z]
zv1
#--identify variables with non-missing values--
z<-which(pp[,1]==0)
zv2<-vvn[z]
zv2

#-----
#----Multiple imputation-----
#-----

#--number of imputed data sets to create--
nimp<-10
#--one at a time, loop through those variables with missing values--
for (i in 1:length(zv1)){
  #--attach the imputand to the auxiliary data--
  zxx<-data.frame(cbind(vaux,fx[,zv1[i]]))
  names(zxx)[NCOL(zxx)]<-zv1[i]
  #--in the following line, the imputation is done--
  aqq<-complete(mice(zxx,maxit=20,m=nimp),action="long")
}

```

```
##--during first iteration of the loop, create dataframe impdat--
if (i==1){
impdat<-data.frame(aqq[,c(".id",".imp")])
}
head(aqq)
##--the imputand is placed as a field in impdat and named--
impdat<-cbind(impdat,data.frame(aqq[,zv1[i]]))
names(impdat)[NCOL(impdat)]<-zv1[i]
}

##--now the non-missing variables are attached to impdat--
gg<-NULL
for (i in 1:nimp){
gg<-rbind(gg,data.frame(fx[,zv2]))
}
impdat<-cbind(impdat,gg)

##--take a look at the top 6 and bottom 6 rows of impdat--
head(impdat)
tail(impdat)
##--check to see what "class" each variable is---
sapply(impdat,function(x) class(x))

##--change all variables that are class "factor" to "numeric"--
z<-which(sapply(impdat,function(x) class(x))=="factor")
impdat[,z]<-data.frame(sapply(impdat[,z],function(x) as.numeric(as.character(x))))
sapply(impdat,function(x) class(x))

##--impdat is saved as an R-format data file--
save(impdat,file="impdat.Rdata")
```

R file: S:\TEFF\662\R\r04b.R

```

#--MI: estimate model, combine results
rm(list=ls(all=TRUE))
#--Set path to your directory with data and program--
setwd("e:/class/662/R/")
options(echo=TRUE)

#--need these packages for estimation and diagnostics--
library(foreign)
library(AER)

#-----
#--Read in data, rearrange----
#-----

#--Read in original SCCS data---
load(url("http://frank.mtsu.edu/~eaeff/downloads/sccs200909.Rdata"),.GlobalEnv)
#--rename dataframe--shorter name is easier--
SCCS<-sccs200909
#--Read in the imputed dataset---
load("impdat.Rdata",.GlobalEnv)

#--create dep.varb. you wish to use from SCCS data--
#--Here we sum variables measuring how much a society values children--
#--can replace "sum" with "max"
childvalue<-apply(SCCS[,c("v473","v474","v475","v476")],1,sum)
#--find obs. for which dep. varb. is non-missing--
zdv<-which(!is.na(childvalue))
childvalue<-childvalue[zdv]

#--look at frequencies and quartiles for the dep. varb.--
summary(childvalue)
table(childvalue)

indpv<-c("femsub", "foodscarc", "exogamy", "ncmallow", "superjh", "moralgods",
"fempower", "sexratio", "war", "himilexp", "wagelabor", "famsize", "settype",
"localjh", "money", "cultints", "roots", "cereals", "gath", "hunt", "fish",
"anim", "pigs", "milk", "plow", "bovines", "tree", "foodtrade",
"ndrymonth", "ecorich", "popdens", "pathstress", "CVrain", "rain",
"temp", "AP1", "AP2", "migr", "brideprice", "nuclearfam", "pctFemPolyg")

#-----
#---Estimate model on each imputed dataset-----
#-----

#--number of imputed datasets--
nimp<-10

#--will append values to these empty objects--
ss<-NULL
beta<-NULL
dng<-NULL

#--loop through the imputed datasets--
for (i in 1:nimp){

#--select the ith imputed dataset--
m9<-impdat[which(impdat$.imp==i),]
#--retain only obs. for which dep. varb. is nonmissing--
m9<-m9[zdv,]

#--OLS estimate of unrestricted model--

```

```

xUR<-lm(childvalue~cultints+roots+cereals+gath+plow+
hunt+fish+anim+pigs+milk+bovines+tree+foodtrade+foodscarce+
+ecorich+popdens+pathstress+exogamy+ncmallow+famsize+
settype+localjh+superjh+moralgods+fempower+femsubs+
sexratio+war+himilexp+money+wagelabor+
migr+brideprice+nuclearfam+pctFemPolyg
,data=m9)

#--OLS estimate of restricted model--
xR<-lm(childvalue ~ cultints + roots + fish + sexratio +
exogamy + settype + femsubs, data = m9)

#--collect coefficients and their variances--
ov<-summary(xR)
ss<-rbind(ss,diag(ov$cov*ov$sigma^2))
beta<-rbind(beta,coef(xR))

#--variables in UR dropped from R-----
dropt<-names(coef(xUR))[which(is.na(match(names(coef(xUR)),names(coef(xR)))))]

#--collect some model diagnostics--

#--Ramsey RESET test--
p1<-qchisq(resettest(xR,type="fitted")$"p.value",1,lower.tail=FALSE)
#--F test that dropped variables had coefficient equal zero--
o<-linearHypothesis(xUR,dropt,white.adjust=TRUE)$"Pr(>F)"[2]
p2<-qchisq(o,1,lower.tail=FALSE)
#--Shapiro-Wilke normality test (H0: residuals normal)
p3<-qchisq(shapiro.test(residuals(xR))$"p.value",1,lower.tail=FALSE)
#--model R2--
p4<-ov$r.squared
dng<-rbind(dng,cbind(p1,p2,p3,p4))

}

#-----
#--Rubin's formulas for combining estimates--
#-----

#--first find final regr. coeffs. and p-values--
mnb<-apply(beta,2,mean)
vrb<-colSums((beta-t(matrix(mnb,length(mnb),10)))^2)/(nimp-1)
mnv<-apply(ss,2,mean)
vrT<-mnv+vrb*(1-nimp^(-1))
fst<-mnb^2/vrT
r<-(1+nimp^(-1))*vrb/mnv
v<-(nimp-1)*(1+r^(-1))^2
pval<-pf(fst,1,v,lower.tail=FALSE)
bbb<-data.frame(round(cbind(mnb,fst,v,pval),3))
names(bbb)<-c("coef","Fstat","ddf","p-value")

#--Then combine the diagnostics we collected--
dng<-data.frame(dng)
names(dng)<-c("RESET","F on restrs.,""SWnormal","R2")
r2<-mean(dng[,4])
adng<-dng[,1:3]
mdm<-apply(adng,2,mean)
vrd<-colSums((adng-t(matrix(mdm,length(mdm),nimp)))^2)/(nimp-1)
aa<-4*mdm^2-2*vrd
aa[which(aa<0)]<-0
rd<-(1+nimp^(-1))*vrd/(2*mdm+aa^.5)
vd<-(nimp-1)*(1+rd^(-1))^2
Dm<-(mdm-(nimp-1)/(nimp+1)*rd)/(1+rd)

```

```
#-All chi-sq we collected have df=1-----
pvald<-pf(Dm,1,vd,lower.tail=FALSE)
ccc<-data.frame(round(cbind(Dm,vd,pvald),3))
names(ccc)<-c("Fstat","df","p-value")

bbb
r2
ccc

#--write results to csv file for perusal in spreadsheet--
#--create a function that will write object to csv format file--
# a1  is the object to write
# a2  is the name of the file (don't include the ".csv" part)
# a3  is either TRUE or FALSE (should object be appended to file?)

Atf<-function(a1,a2,a3){
print(a1)
write.table(a1,file=paste(a2,".csv",sep=""),quote=TRUE,sep=" ",dec=".",qmethod="double",
col.names = NA,row.names = TRUE,a3)
}

Atf("==OLS model for childvalue==","OLSresults",FALSE)
Atf(bbb,"OLSresults",TRUE)
Atf(r2,"OLSresults",TRUE)
Atf(ccc,"OLSresults",TRUE)
```