SCRIPT MOD1_2D: PARTITIONED REGRESSION

Set basic R-options upfront and load all required R packages:

1. Data Description

The data for this exercise flow from a door-to-door fundraising campaign conducted in Pitt County, North Carolina, during the fall of 2005. The details of this field experiment are described in Landry et al. (2006). Forty-three solicitors interacted with an average of 39 households for a total sample size of 1682 observations. All observations are based on actual interactions, i.e. the "door didn't open" cases are not considered in this data set. The focus of this research was on the effect of lottery designs and solicitor attributes on donation outcomes. There are four possible treatments (or "institutions"), as shown in the table below. Each solicitor administered a single treatment, i.e. the treatment remained the same for a given solicitor across all interactions. Each respondent (or "household") is visited only once, so there are no multiple observations per household in the data.

```
R> data<- read.table('c:/Klaus/AAEC5126/R/data/fundraising.txt',
 sep="\t", header=TRUE)#this time load in variable names
R> save(data, file = "c:/Klaus/AAEC5126/R/data/fundraising.rda")
```

The variables in data set "fundraising" are defines as follows:

Table 1. Variable description

pos.	variable	description
1	solid	original solicitor ID
2	age	age of solicitor
3	race	solicitor race (1 = white)
4	gender	solicitor gender (1 = male)
5	height	solicitor height (inches)
6	weight	solicitor weight (lbs)
7	BMI	body mass index (19 - 25 = "normal weight")
8	beauty	mean beauty index (z-score)
9	spunk	overall personality index (-40 to 40)
10	amount	contribution per household (dollars)
11	genresp	observed gender of respondent (1=male)
12	raceresp	observed race of respondent (1=white)
13	ageresp	estimated age of respondent
14	T1	no lottery mechanism, simple voluntary donation (administered by 7 solicitors)
15	T2	no lottery mechanism, voluntary donation with seed money (12 solicitors)
16	T3	lottery mechanism = lottery with single prize (9 solicitors)
17	T4	lottery mechanism = lottery with multiple prizes (15 solicitors)

2. Data Preparation

For this next part we will only use observation associated with one of the lottery treatments (T3 or T4). We will need to filter these variables out of the overall data. Then, we will focus on the variables age, race, gender, BMI, beauty, and spunk. The dependent variable will be the log of "amount", after adding an increment to the (many...) zero cases.

```
R> sel<-data$T3==1 | data$T4==1 #define selection criteria
R> data1<-data[sel, ]#select corresponding rows, use all columns
R> n<-nrow(data1)
R> attach(data1)
R> logamount<-log(amount+0.01)
R> tt<-data.frame(col1=c("age","race","gender","BMI","beauty","spunk","amount"),
 col2=c(mean(age),mean(race),mean(gender),mean(BMI),mean(beauty),mean(spunk),mean(amount)),
 col3=c(sd(age),sd(race),sd(gender),sd(BMI),sd(beauty),sd(spunk),sd(amount)),
 col4=c(min(age),min(race),min(gender),min(BMI),min(beauty),min(spunk),min(amount)),
 col5=c(max(age),max(race),max(gender),max(BMI),max(beauty),max(spunk),max(amount)))
R> colnames(tt)<-c("variable","mean","std","min","max")

R> ttx<- xtable(tt,caption="Sample statistics")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

Table 2. Sample statistics

variable	mean	std	min	max
age	20.344	2.111	18.000	25.000
race	0.643	0.479	0.000	1.000
gender	0.432	0.496	0.000	1.000
BMI	26.434	6.608	20.918	44.476
beauty	-0.042	0.568	-1.066	1.201
spunk	3.805	6.186	-7.000	22.000
amount	1.622	3.634	0.000	50.000

The new sample size is 815.

Let's take a look at the distribution of "amount" and "logamount"

Not exactly an ideal case for a CLRM, but we will ignore these data shortcomings for now. Let's first run a regression on the full model.

3. OLS on Full Model

```
R> y<-logamount
R> X<-cbind(rep(1,n),age,race,gender,BMI,beauty,spunk)
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y);# compute OLS estimator
R> e<-y-X%*%bols # Get residuals.
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
```

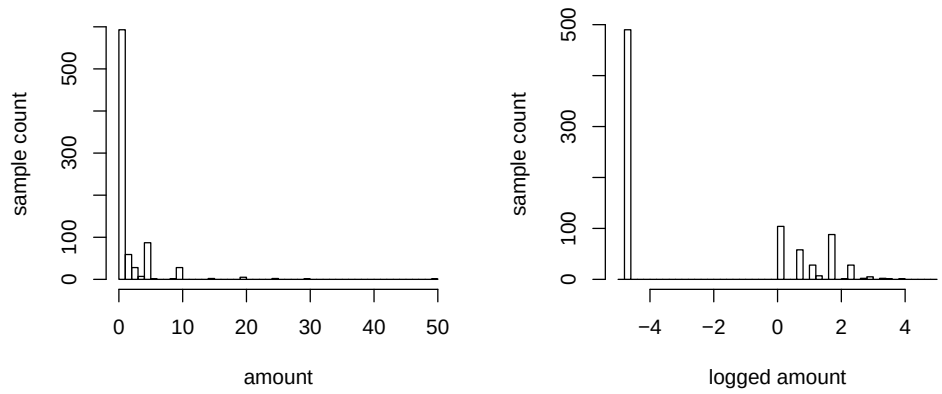

FIGURE 1. Overview of donation outcomes

```
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated variance-covariance matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
R> tt<-data.frame(col1=c("constant","age","race","gender","BMI","beauty","spunk"),
                  col2=bols,
                  col3=se,
                  col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="FULL MODEL")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

TABLE 3. FULL MODEL

variable	estimate	s.e.	t
constant	-2.683	1.015	-2.643
age	0.021	0.056	0.384
race	-0.327	0.248	-1.320
gender	0.235	0.204	1.153
BMI	-0.004	0.019	-0.211
beauty	0.853	0.202	4.227
spunk	0.034	0.016	2.090

The estimated variance of the error term for this model is 7.636.

4. Partitioned Regression, Version 1

Assume we're interested in the isolated effect of "beauty" and "spunk",i.e the last two columns of X. Let's use partitioned regression using the "residual-maker" matrix $M1$.

```
R> X1<-cbind(rep(1,n),age,race,gender,BMI)
R> X2<-cbind(beauty,spunk)
R> I<-diag(n)
R> M1<-I-X1 %*% solve(t(X1) %*% X1) %*% t(X1)  #compute residual-maker matrix
R> X2star<-M1 %*% X2 #purge X2 of any influence from X1
R> ystar<-M1 %*% y #same for y
R> k<-ncol(X2star)
R> b2<-solve((t(X2star)) %*% X2star) %*% (t(X2star) %*% ystar);# compute OLS estimator
R> e<-ystar-X2star%*%b2 # Get residuals.
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X2star))%*%X2star)
R> # get the estimated variance-covariance matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=b2/se # get your t-values.
R> tt<-data.frame(col1=c("beauty","spunk"),
                  col2=b2,
                  col3=se,
                  col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="PARTITIONED RESULTS, V1")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

TABLE 4. PARTITIONED RESULTS, V1

variable	estimate	s.e.	t
beauty	0.853	0.201	4.240
spunk	0.034	0.016	2.096

5. Partitioned Regression, Version 2

Let's use partitioned regression using the "step-wise" approach.

```
R> int1<-solve((t(X1)) %*%X1) %*% (t(X1) %*% X2)
R> #run regression of (every column of) X2 on X1.
R> X2star<-X2-X1 %*% int1
R> #collect residuals - same interpretation as before: X2, purged from any effects due to X1
R>
R> int2<-solve((t(X1)) %*%X1) %*% (t(X1) %*% y) #repeat the above steps for y
R> ystar<-y-X1 %*% int2
R> # The last part is the same as before: OLS on the transformed variables
R> b2<-solve((t(X2star)) %*% X2star) %*% (t(X2star) %*% ystar);# compute OLS estimator
R> e<-ystar-X2star%*%b2 # Get residuals.
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X2star))%*%X2star)
R> # get the estimated variance-covariance matrix of bols
```

```
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=b2/se # get your t-values.
R> tt<-data.frame(col1=c("beauty","spunk"),
                  col2=b2,
                  col3=se,
                  col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="PARTITIONED RESULTS, V2")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

TABLE 5. PARTITIONED RESULTS, V2

variable	estimate	s.e.	t
beauty	0.853	0.201	4.240
spunk	0.034	0.016	2.096

6. ANOVA AND R^2

```
R> I<-diag(n)
R> i<-rep(1,n)
R> Mo=I-i %*% solve(t(i) %*% i) %*% t(i) #create deviation-from-mean matrix
R> b<-bols #just to simplify notation
R> R2<-(t(b) %*% t(X) %*% Mo %*% X %*% b)/(t(y) %*% Mo %*% y)
R> adjR2=1-((n-1)/(n-k))*(1-R2)
```

The original R^2 is 0.033. The adjusted R^2 is 0.032

Now lets add a few nonsensical variable to the model, re-run OLS and see what happens to these two measures.

```
R> add1<-rnorm(n,2,1.7)
R> add2<-rnorm(n,0,4)
R> add3<-sample(1:10,n,replace=TRUE)
R> X<-cbind(X,add1,add2,add3)
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y);# compute OLS estimator
R> I<-diag(n)
R> i<-rep(1,n)
R> Mo=I-i %*% solve(t(i) %*% i) %*% t(i) #create deviation-from-mean matrix
R> b<-bols #just to simplify notation
R> R2<-(t(b) %*% t(X) %*% Mo %*% X %*% b)/(t(y) %*% Mo %*% y)
R> adjR2=1-((n-1)/(n-k))*(1-R2)
```

The new R^2 is 0.035. The new adjusted R^2 is 0.024

```
R> proc.time()-tic
```

```
user  system elapsed
2.12    0.21    2.50
```