Paper 2380-2016

Analyzing Hospital Episode Statistics Dataset: How Much Does SAS® Help?

Violeta Balinskaite, Imperial College London; Paul Aylin, Imperial College London

ABSTRACT

Hospital Episode Statistics (HES) is a data set containing records of all admissions, outpatient appointments and accident and emergency (A&E) attendances at National Health Service (NHS) hospitals in England. Each year over 125 million admitted patient, outpatient and A&E records are processed. Such a large data set enables rich research opportunities for researchers and health care professionals. However, patient care data is complex and can be difficult to manage. This paper demonstrates the flexibility and power of SAS programming tools such as DATA step, PROC SQL and Macros to help to analyze HES.

INTRODUCTION

The Health and Social Care Information Center (HSCIC) is national provider of information, data and IT systems for commissioners, analysts and clinicians in a health and social care. It was set up as an executive non-departmental public body in 2013. It is mainly responsible for:

- collecting, analyzing and presenting national health and social care data;
- publishing a register of all the information collected and produced;
- setting standards and guidelines in the field of data collection and reporting;
- creating indicators that can be used to measure the quality of health and care service etc.

The Hospital Episode Statistics dataset (http://www.hscic.gov.uk/hes) contains information on all patients treated in NHS hospitals including private patients treated in NHS hospitals, patients resident outside of England and care delivered by treatment centres (including those in the independent sector) funded by the NHS. Admitted patient care data collection began from 1989, outpatient attendance data from 2003 and A&E data from 2007.

In HES, each record in the inpatient dataset contains data on patient demographics (for example, age, ethnicity, and socioeconomic deprivation based on postcode of residence), the episode of care (for example, hospital name, date of admission and discharge) and clinical information (1, 2). Diagnoses for each patient are recorded using the International Classification of Diseases, 10th edition (ICD-10). Procedures performed during an episode are coded using the Office of Population, Censuses and Surveys Classification of Surgical Operations and Procedures, 4th revision (OPCS4). Each record represents the continuous period of time during which patient is under the care of a consultant or allied health professional and is called an ‘episode’. Episodes can be linked into ‘spells’ (admissions to one provider) and into ‘superspells’ – combining any interhospital transfers.

In addition, each episode related to the delivery of a baby contains details about the labour and delivery (for example, parity, mode of delivery, gestational age, birth weight) in supplementary data fields known as the HES ‘maternity tail’ (see HES dictionary http://www.hscic.gov.uk/hesdatadictionary).

This paper will use examples from a study designed to estimatethe risk of adverse birth outcomes in pregnant women undergoing non-obstetric surgery to demonstrate the power of SAS in analysing HES data.

DATA PREPARATION

Data extraction and cleaning is a necessary step before any actual data analysis. To extract all admissions associated with pregnancy from the hospital inpatient database for a 10 year period, a SAS MACRO was created:

```
%macro deliveries (dat1, dat2)/store;
data &dat1;
set &dat2;
    where oper_01 in:
('R17','R18','R19','R20','R21','R22','R23','R24','R25') or
          oper_02 in:
('R17','R18','R19','R20','R21','R22','R23','R24','R25') or
              ...

          oper_18 in:
('R17','R18','R19','R20','R21','R22','R23','R24','R25') or
          delmeth_1 in ('0','1','2','3','4','5','6','7','8','9','X');
run;
%mend;
```

This allows us to minimize the amount of SAS code to be used. After extraction of data of interest, data cleaning is our next step. Even if HSCIC clean common and obvious data quality errors[1], some more errors, for example duplicates, may occur. To identify duplicate records, we use a three step approach:

STEP 1

First, we use PROC SQL to select only those records which have duplicates according to five variables: ID number (ID), admission date (admidate), episode start date (epistart), provider code (procode) and consultant ID (consult).

```
proc sql;
     create table del_dup as
     select*
     from (select*, count(*) as tmp from deliveries group by
        extract_id, admidate,epistart, procode, consult)
     where tpm>1;  /*select admissions with duplicates*/
     create table del_without_dup as
     select*
     from (select*, count(*) as tmp from deliveries group by
        extract_id, admidate,epistart, procode, consult)
     where tpm=1; /*select admissions without duplicates*/
quit;
```

STEP 2

Second, we use MACRO and PROC SQL to separate records which have same operation and diagnoses codes:

```
%macro dupl_sql_main (dat1, dat2, dat3)/store;
proc sql;
create table &dat2 as
select*
from (select*, count(*) as tmp from &dat1 group by extract_id,
admidate,episatrt, procode, consult, diag_01,
diag_02,diag_03,diag_04,diag_05,diag_06,diag_07,oper_01,oper_02,
oper_03,oper_04,oper_05,oper_06,oper_07)
where tpm>1;
create table &dat3 as
select*
from (select*, count(*) as tmp from &dat1 group by extract_id,
admidate,episatrt, procode, consult, diag_01,
diag_02,diag_03,diag_04,diag_05,diag_06,diag_07,oper_01,oper_02,
oper_03,oper_04,oper_05,oper_06,oper_07)
where tpm=1;
quit;
%mend;
```

STEP 3

In the last step we use PROC SORT or a simple statement to exclude records with duplicates. During the second step, we created two datasets: the first dataset contains observations which had identical diagnoses and procedures code; the second dataset contained observations which had some difference in diagnosis or procedures fields. To delete duplicates from the first dataset we use PROC SORT procedure with NODUPKEY option:

```
proc sort data=del_ident nodupkey;
      by extract_id admidate;
run;
```

To delete duplicates from the second dataset we first checked which of the observations had more information in diagnoses and procedure fields using LENGTHN function:

```
length=LENGTHN(CATS(diag_01,diag_02,diag_03,diag_04,diag_05,diag_06,
diag_07,oper_01,oper_02, oper_03,oper_04,oper_05,oper_06,oper_07));
```

DATA ANALYSES

SAS gives a lot of options when we come to data analyses, starting from simple descriptive statistics and finishing with bootstrapping. Whenever you work with an administrative dataset, you want to know the characteristics of your study population and/or create new variables for analysis. However, when data are collected as counts require a specific kind of data analysis and it does not make sense to calculate means and standard deviations on categorical data. In our case, we wanted to carry out a descriptive analysis of the data, describing total number and rates of risk factors, outcomes and missing data. Using PROC FREQ, we are able to obtain:

- Counts and percentages of women who had operation and who did not.

```
proc freq data=delivelies;
     table operation;
run;
```

- Counts and percentages of operations by maternal age group.

```
proc freq data=delivelies;
     table operation*age;
run;
```

- Counts and percentages of operations by maternal age group where delivery occurred preterm

```
proc freq data=delivelies;
     table operation*age;
     where preterm=1;
run;
```

Despite the fact that the HES dataset is rich, it may happen that not all necessary variables for analysis are presented in the dataset. In medical research it is common to use historical medical information and the use of TABLE LOOK-UP and MACRO are very useful in such situations.

```
proc sort data=test(keep=extract_id) out=test3 nodupkey;
by extract_id;
run;

data ptlookup;
set test3;
start=extract_id;
label='KEEP';
fmtname='$ptlookup';
proc format cntlin=ptlookup;
run;

%macro temp(yr);
data women_pts_adms&yr;
set impusr.hes_apc_&yr(keep= extract_id admiage disage
numpreg admidate  admimeth  oper: diag: delmeth_1 epistart
procode consult);
where put(extract_id,$ptlookup.)='KEEP';run;
%mend;
%temp(2011);
%temp(2010);
.
.
.
%temp(1997);
%temp(1996);
```

In our analysis, we needed various historical information: for example, if a woman had emergency admissions prior to pregnancy or had an operation on amniotic cavity during pregnancy or had previous caesarean sections. In the code above, firstly we used table look-up to create a dataset with ID of the women in our population and then we used MACRO to extract historical information from 1996 to 2011.

In the medical and public health research the odds ratio (ORs) and relative risks (RRs) are the most used measures, specifically, when one wants to evaluate the effect of treatment or exposure on an outcome of interest[2]. There are various statistical methods to estimate these measures depending on the type of outcome variable. In our case, the dependent variables were dichotomous (for example, spontaneous abortion associated with hospitalization (yes or no), preterm delivery (yes or no) and etc.). We used four different statistical approaches:

- **Logistic regression**. It is the most common method to estimate adjusted ORs/RRs in the medical literature. The box below presents basic logistic regression code used in our analysis:

```
proc logistic data=pregnancies desc;
class operation carstairs_quintile (ref='1') age (ref='3')
mult_gestation(ref='0') r10_1(ref='0') emergency(ref='0')
parity(ref='0') charlson_6max(ref='0') charlson_6max_p(ref='0')
d_pr(ref='0') hp_pr(ref='0') cd_pr(ref='0')
ob_oper(ref='0')/param=ref ref=first;
model abor= operation carstairs_quintile age mult_gestation
r10_1 emergency parity charlson_6max charlson_6max_p year
d_pr hp_pr cd_pr ob_oper;
run;
```

- **Log-binomial regression**. As a logistic regression, it models the probability of the outcome and assumes that the error terms have a binominal distribution. The only difference is that in the log-binomial model the log function is used (instead the logit). The box below presents basic log-binomial regression code used in our analysis:

```
proc genmod descending data=pregnancies;
class operation/param=ref ref=first;
model abor= operation carstairs_quintile age mult_gestation
r10_1 emergency parity charlson_6max_new charlson_6max_new_p
year d_pr_new hp_pr cd_pr_new/dist=bin link=log;
Estimate 'RR operation vs. Non-operation' operation 1/exp;
run;
```

- **Poisson regression.** It is usually used for the studies of rare outcomes. This statistical approach provides a correct estimate of the adjusted RRs if the model decently fits the data. The box below presents basic log-binomial regression code used in our analysis:

```
proc genmod descending data=pregnancies;
class operation/param=ref ref=first;
model abor= operation carstairs_quintile age mult_gestation
r10_1 emergency parity charlson_6max_new charlson_6max_new_p
year d_pr_new hp_pr cd_pr_new/dist=poisson link=log;
Estimate 'RR operation vs. Non-operation' operation 1/exp;
run;
```

- **Austin's method[3].** This method derives the adjusted RR from a logistic regression model. It involves determining the probability of the outcome if a patient was treated and if the same patient was not treated. Then it computes the mean probability of success in the sample if all

patients were treated, and the mean probability that of success in the sample if all patients were untreated. Then the RR can be estimated as the ratio of the mean probabilities. The box below presents the code of Austin's method used in our analysis:

```
data population;
    set pregnancies (in=a) pregnancies (in=b);
    if a then operation=1;
    if b then operation=0;
run;
proc logistic data=pregnancies desc;
   class operation /param=ref ref=first;
   model abor= operation carstairs_quintile age mult_gestation
   r10_1 emergency parity charlson_6max charlson_6max_p year
   d_pr hp_pr cd_pr ob_oper;
   Score data=population out=pred_risk;
run;
proc means data=pred_risk nway;
   class operation;
   var p_1;
   output out=pop_risk mean=pop_risk;
run;
proc transpose data=pop_risk out=pop_risk prefix=operation_;
   id operation;
   var pop_risk;
run;
data pop_risk;
   set pop_risk;
   adjusted_rr=operation_1/operation_0;
run;
proc print data=pop_risk;
   var adjusted_rr;
run;
```

The methods described above have their own advantages and disadvantages. The logistic regression directly does not provide the adjusted RRs, however, it is a simple method that allows to approximate a RR from the adjusted odds ratio and to derive an estimate of an association or treatment effect that better represents the true RR. The log-binomial and Poisson regression directly produces an unbiased estimate of the adjusted RR. Nonetheless, the log-binomial model may not converge (this happened in our case) and the Poisson model may overestimate of binomial errors when the outcome is common (in our case it would be cesarean section outcome)[4]. The Austin's method allows to compare outcomes between two populations whose only difference was the exposure. Furthermore, it gives more precise estimates when the outcome is common. However, the main disadvantage of this method is the computation of the confidence intervals, which can be estimated using bootstrap methods and having large dataset may take several days of computing time to run. We created 1000 bootstrap samples and estimated the quantity of interest in each of the bootstrap samples. The endpoints of the nonparametric 95% CIs would be the 2.5th and 97.5th percentiles of that quantity across the bootstrap samples[5] (code presented in the Appendix).

CONCLUSION

The Hospital Episode Statistics is large and rich administrative dataset. However, it is one of the most difficult and challenging datasets to work with: complex coding of data items, missing data, duplicates and other data issues may become a challenge for a researcher. In this paper, it was showed that there are a variety of options in SAS to help the researcher to overcome these issues.

REFERENCES

1. Team HDQ. 24th February 2014. “Methodology for identifying and removing duplicate records from the HES dataset”.
2. Schechtman E. 2002. “Odds ratio, relative risk, absolute risk reduction, and the number needed to treat—which of these should we use? “ *Value in Health*, 5(5):431-6.
3. Austin PC. 2010. “Absolute risk reductions, relative risks, relative risk reductions, and numbers needed to treat can be obtained from a logistic regression model”. *Journal of Clinical Epidemiology*, 63(1):2-6.
4. NcMutt LA, Wu C, Xue X and Hafner JP. 2003. “Estimating the relative risk in cohort studies and clinical trials of common outcomes”. *American Journal of Epidemiology,* 157: 940-943.
5. Efron B, Tibshirani RJ. 1994. *An introduction to the bootstrap*. CRC press.

ACKNOWLEDGMENTS

This study was supported by a grant from the National Institute for Health Research- Health Services and Delivery Research programme (Reference12/209HS&DR). The views expressed are those of the authors and not necessarily those of the NIHR. The funder had no role in the design and conduct of the study; in the collection, analysis, and the interpretation of the data; or in the preparation, review, or approval of the manuscript.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Violeta Balinskaite
Imperial College London, London, UK
v.balinskaite@imperial.ac.uk
http://www.imperial.ac.uk/people/v.balinskaite

APPENDIX

```
%macro attrib(dataset,var,nboot,dyads);
%do h=1 %to &nboot;
data bootsamp;
sampid=&h;
do i=1 to &dyads;
x=int(ranuni(-1)*&dyads)+1;
set &dataset
nobs=nobs
point=x;
output;
end;
stop;
run;


data population;
set bootsamp (in=a)
    bootsamp (in=b);
if a then operation=1;
if b then operation=0;
run;

proc logistic data=bootsamp desc;
class operation/param=ref ref=first;
model &var= operation carstairs_quintile age mult_gestation
 r10_1 emergency parity charlson_6max charlson_6max_p year
 d_pr hp_pr cd_pr ob_oper /rl;
Score data=population out=pred_risk;
run;

proc means data=pred_risk nway;
class operation;
var p_1;
output out=pop_risk_&h mean=pop_risk_&h;
run;

proc transpose data=pop_risk_&h out=pop_risk_&h prefix=operation_;
id operation;
var pop_risk_&h;
run;

data pop_risk_&h;
set pop_risk_&h;
adjusted_rr=operation_1/operation_0;
run;
%end;
%mend;

%attrib(pregnacies,abor,1000,6486280)
```

```
data abor_ci;
length _NAME_ $14.;
set pop_risk_1 - pop_risk_350 open=defer;
run;

data abor_ci;
set abor_ci;
adjusted_rr=oper_append_1/oper_append_0;
ar=sum(oper_append_0,-oper_append_1);
nnh=1/ar;
run;

proc univariate data= abor_ci;
var adjusted_rr;
output out=pctls_rr_sb pctlpts=2.5 97.5 pctlpre=pwid;
run;

proc univariate data= abor_ci;
var ar;
output out=pctls_ar_sb pctlpts=2.5 97.5 pctlpre=pwid;
run;

proc univariate data= abor_ci;
var nnh;
output out=pctls_nnh_sb pctlpts=2.5 97.5 pctlpre=pwid;
run;


proc print data= abor_ci;
var adjusted_rr ar nnh;
run;
```