Deloitte.

Introduction to 

CAS RPM Seminar
March 19, 2012

Steve Berman, FCAS, MAAA
Jim Guszcza, FCAS, MAAA

R – Data processing

Data loading
QC
Data processing

Data aggregation
Data analysis

Steps in data processing

- Data load
- QC
- Data cleansing
- Data manipulation
- Data aggregation
- Data analysis

- This is not the “fun” part of R, but you can expect for 50-70% of your work to be here

2

Data load

- Most datasets are too large to read in using a data.frame statement
- The simplest way to read in files is to have a version saved as a text file, and then use base statements
- However, there are packages that perform some translations of better known file formats

3

read.table, write.table

- Part of base package
- Reads in text files
- Allows user to set the delimiter, specify if file has header, etc.
- read.csv, write.csv are similar, but defaults for CSV files
- Make sure that data does not have formatting in it – may get read in incorrectly

4

Reading Excel files

- Several packages that read directly from Excel
  - xlsReadWrite
  - xlsx
  - XLConnect
  - RODBC – connects to data using ODBC drivers
- Packages may cover different versions of Excel
- May have trouble loading large Excel files

5

Reading other files

- `foreign` package allows reading from several common formats, including SAS and SPSS
- Many other packages
- Same issues with file type versioning as with Excel
- `sas7bdat` package much better for reading SAS
- `Hmisc` package has functions to read from Microsoft Access (.mdb format only)

6

Exercise 1 – Data load

- You've just gotten data from a small regional auto insurer. The following files exist for years 2003-2006:
  - A single policy file for each year (policy_[yyyy].csv, where [yyyy] is the year)
  - A coverage file for each year (coverage_[yyyy].csv)
  - A single claim transaction file (payments, reserves) (loss.csv)
- Start by loading all of the policy files into a single data frame
  - Check: How many records are in the file?
- We won't need zip code. Remove ins_zipcode from the data frame to save space

Tip: setwd() can be used to change the working directory

7

Data QC

- Common functions to get summary information:
  - summary() – show values at certain percentile values
  - tables() – frequency distribution, other tables
- Individual data cuts to find unusual values

8

Exercise 2 – Data review and cleansing

- Remove “flat cancellations” (where inception date=expiration date)
- For premium file, are there any policies with zero or negative premium? Remove these
- What is the number of policies by state? By coverage?
- Get premium by state, premium by coverage

Tip: for counts, create count field, equal to 1 for all records

Tip: for coverage counts, create indicator fields (ex: bi_ind is 1 if bi_premium>0, and 0 otherwise)

9

More data review and cleansing

- One dimensional tables are good – what about two dimensional? Both `table` and `xtabs` give pivot table like functionality
  - `table` does counts, while `xtabs` sums across dimensions

10

Exercise 3 – review of loss data

- Aggregate incurred loss from transactional level to a claim level
- Get a listing of all claims greater than $50K, sorted in descending order
- Find if any claims have losses less than zero. If so, adjust incurred to zero

11

Combining it all together

- Merge premium and loss data together, by policy number and effective date
  - Note that claims have to be matched via accident date (within effective and expiration dates)
  - Several merge statements required
  - Would be a single SQL step, if only we could do that….
- Not all claims have a loss. Correct missing values to zero
- Calculate loss ratios

12

Predictive variables

- Policy age
- Coverage indicators
- Drivers per vehicle

13

Univariate analysis

- How does loss ratio differ by each of the predictive characteristics?
- Are there correlations between the predictive variables?

14