Extension Commands and Rim Weighting with IBM® SPSS® Statistics: Theory and Practice
Agenda

- Introduction to IBM SPSS extension commands
  - Installation and set-up of extension commands and working with network shares

- Statistical discussion of raking
  - The problem
  - Assumptions for raking
  - The model

- The SPSSINC RAKE command and dialog box interface

- Examples

- Issues
  - Control totals
  - Missing data
  - Unrakeable problems
  - Extreme weights

- Using external mode or Statisticsb for better performance

- Bounding the weights

- Using the weighted data in SPSS Statistics procedures

- Summary
Extension commands add functionality to IBM SPSS Statistics

- Provide standard Statistics syntax for Python or R programs
  - Most extension command names start with SPSSINC or STATS
- Work like built-in commands
- Usually include a dialog box built with the Custom Dialog Builder
- By convention, `COMMAND-NAME /HELP` displays the syntax help
- Require Python and/or R Essentials
  - Extension Commands collection or Statistical Tools, Graphical Tools or Utilities collections
  - 48 currently available – some installed with Essentials
    - Free
    - Anyone can create them
- Many installable as spe packages ([Utilities] > [Extension bundles] > [Install Extension Bundles])
- Consider using environment variables to specify where to install (See SPSS Community FAQ)
  - Permission problems, especially on Win7 or Vista
  - Sharing
  - Updating to new Statistics versions
  - My settings
    - `SPSS_CDIALOGS_PATH=C:\dlgcommon`
    - `SPSS_EXTENSIONS_PATH=C:\extcommon; C:\extcommon18`
Why do we weight data?

- To match aggregate sums from a random sample (scale weighting)
  - Census 1/1000

- For computational efficiency
  - A case represents n identical cases (replication weight)

- To represent a complex sampling scheme based on one or more categorical variables
  - Stratified, clustered, multi-stage, etc
  - Requires different algorithms as well as weights when calculating variances

- To correct sample nonrepresentativeness
  - Deliberate oversampling for higher accuracy of target groups
  - Differential nonresponse (unit nonresponse)
  - samples representing wrong population (noncoverage or overcoverage)
  - convenience samples

- NOT good for
  - measurement or coding errors
    • E.g., underreporting income from capital gains
  - missing values for respondents (item nonresponse)
    • Imputation techniques address missing values
The appropriate technique depends on what information you have

- Assume a 2-dimensional set of cells defined by crossing categorical variables A and B with categories $A_i$ and $B_j$. The cells are $C_{ij}$. This extends to any number of dimensions.

- If you know the joint distribution in the target population of A and B
  - Weight each cell, $C_{ij}$ as $\frac{N_{ij}}{n_{ij}} \times K$
    - where N's are population counts and n's are sample counts, and K normalizes the weight to sum to the actual sample size
    - Number of cells grows very rapidly – demands a lot of information
    - Small cell counts vulnerable to sampling variability

- If you know only the marginal distributions
  - Consider raking aka rim weighting
    - Assumes probabilities and weights are function of marginals only. No interaction

- If you have partial information about the joint distribution
  - Convert joint distribution to one dimension and rake with that and the remaining marginals
    - May need to further collapse categories
There are other approaches to weighting besides raking

- **Inverse Propensity Score**
  - Estimate the propensity/probability of being in the sample using sample and population data
  - How to estimate propensity?
    - E.g., regress cell counts on population counts in k dimensions
    - Regression on totals
      \[
      \log (n_{ij}) = \beta_1 X_{1i} + \beta_2 X_{2j} + ... 
      \]
  - Weight by inverse of propensity
  - Need to bound weights as \( p \to 0 \)
  - Scale sum of weights to actual sample size

- **Maximum entropy**
  - Use the known marginals and other information and make weights that are consistent with the information but are otherwise as uniform as possible (maximize entropy)
    - Requires solving nonlinear constrained maximization problem
  - If information is only marginal distributions, converges to raking
  - If information is joint distribution, converges to joint probability weighting
Raking adjusts cell counts so that marginal totals match control totals

Originally due to W. Edwards Deming and Frederick F. Stephan, *On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known*, Annals of Mathematical Statistics, Vol 11, Number 4, 1940

"There are situations in sampling wherein the data furnished by the sample must be adjusted for consistency with data obtained from other sources or with deductions from established theory."

\[
\begin{align*}
(1) \quad \bar{N}^{(1)}_{ij} &= \frac{n_{ij}N_{i+}}{n_{i+}} \quad \text{(row)} \\
(2) \quad \bar{N}^{(2)}_{ij} &= \frac{\bar{N}^{(1)}_{ij}N_{+j}/\bar{N}^{(1)}_{+j}} {\bar{N}^{(1)}_{ij}N_{+j}/\bar{N}^{(1)}_{+j}} \quad \text{(column)}
\end{align*}
\]

where + indicates summation over that index:

\[
N_{i+} = \sum_j N_{ij} \text{ etc}
\]

Lowercase symbols refer to the sample
Uppercase symbols refer to the population
Superscripts refer to the iteration
The iterations repeat in a raking pattern in 1 to M dimensions

- Iteration is repeated until convergence is achieved
Raking is equivalent to fitting a loglinear model

Loglinear model:

\[ N_{ij} = a_i b_i n_{ij} \]

written as probabilities, \( \pi_{ij} = a_i b_i p_{ij} \) where

\( \pi_{ij} \) and \( p_{ij} \) are population and sample probabilities, respectively

\[ \log \left( \frac{\pi_{ij}}{p_{ij}} \right) = \log(a_i) + \log(b_j) + \epsilon_{ij} \]

hence SPSSINC RAKE uses GENLOG procedure to fit the model

- Loglinear models are fit by Iterative Proportional Fitting (IPF)
- Observed counts are assumed to be independent Poisson variables
- Fit by MLE using Newton-Raphson algorithm
- You can expose details of GENLOG portion of output in SPSSINC RAKE if necessary and adjust fitting parameters
  - Don't do this for really big problems!
The sample balance can be measured by the **rim weighting efficiency**

\[
E = 100 \times \frac{\sum_i W_i R_i^2}{\sum_i W_i \sum_i W_i R_i^2}
\]

- If no input weights, this reduces to

\[
E = 100 \times \frac{\sum_i R_i^2}{N \sum_i R_i^2}
\]

- **N** is the total number of cases
- The square of the sum of the weights / N * sum of squared weights
- Equals 100 if sample is perfectly balanced
- There is no magic cutoff value

**Compute efficiency in Statistics using MATRIX**

```plaintext
matrix.
get M /variables = W R.
compute W = M(:,1).
compute R = M(:,2).
compute num = csum(W)*R)**2.
compute denom = csum(W) * csum(W)*R*R.
compute eff = 100 * num / denom.
print eff /title="Weighting Efficiency".
end matrix.
```

**If no input weight**

```plaintext
matrix.
get R /variables = R.
compute N = nrow(R).
compute eff = 100 * csum(R)**2 / (N * csum(R)*R).
print eff /title="Weighting Efficiency".
end matrix.
```

csum(R) = N if weights are normalized
SPSSINC RAKE is the extension command for raking

- It requires the Advanced Statistics option because it uses GENLOG

- SPSSINC RAKE /HELP displays the syntax help in the Viewer
  - Extension commands are not included in the Command Syntax Reference

- Variable names in extension commands are Case Sensitive

- Dialog box appears on the Data menu

```spssinc rake /help.
SPSSINC RAKE DIM1=control spec
  [DIM2=control spec]
  [DIM3=control spec]
  [DIM4=control spec]
  [DIM5=control spec]
  [DIM6=control spec]
  [DIM7=control spec]
  [DIM8=control spec]
  [DIM9=control spec]
  [DIM10=control spec]
FINALWEIGHT=varname
[FOPTOTAL=value]
[/OPTIONS [DELTAs=value] [ITERATIONS=value][CONVERGENCE=value [CHECKEMPTY={YES|NO}]]
[SHOW]
```

Example:
SPSSINC RAKE DIM1 = jobcat 1 500 2 300 3 200.

Up to 10 control total dimensions can be specified using DIM1, ..., DIM10. Each DIM specification consists of a variable name followed by a list of category values and the total or fraction for that category. Only numeric variables can be used for control totals.

Note: with many dimensions this command might run out of memory on 32-bit systems and take a long time to complete.

FINALWEIGHT names a new variable to contain the weight.

FOPTOTAL can be specified as a number that the weights should sum to.

The DELTA, ITERATIONS, AND CONVERGENCE settings correspond to the Genlog parameters. The defaults are DELTA=.5, ITERATIONS=20, CONVERGENCE=.0001. If the generated weights do not produce the required sample proportions, try setting DELTA=0.

By default, the table is checked for empty cells. CHECKEMPTY=NO skips that check, which can save time and memory.
Raking example: one dimension: bankloan.sav (in .../samples/english)

- Adjust by age
- Dataset has 850 cases
- Age reported by year
- Suppose we have known percentages by decade of age
- Recode age using Visual Binner – Transform > Visual Binning

<table>
<thead>
<tr>
<th>Age in years</th>
<th>Frequency</th>
<th>Percent</th>
<th>Valid Percent</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid</td>
<td>20</td>
<td>.2</td>
<td>.2</td>
<td>.2</td>
</tr>
<tr>
<td>21</td>
<td>12</td>
<td>1.4</td>
<td>1.4</td>
<td>1.6</td>
</tr>
<tr>
<td>22</td>
<td>14</td>
<td>1.6</td>
<td>1.6</td>
<td>3.3</td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>2.5</td>
<td>2.5</td>
<td>5.8</td>
</tr>
<tr>
<td>24</td>
<td>30</td>
<td>3.5</td>
<td>3.5</td>
<td>9.3</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>2.9</td>
<td>2.9</td>
<td>12.2</td>
</tr>
<tr>
<td>26</td>
<td>30</td>
<td>3.5</td>
<td>3.5</td>
<td>15.8</td>
</tr>
<tr>
<td>27</td>
<td>33</td>
<td>3.9</td>
<td>3.9</td>
<td>19.6</td>
</tr>
<tr>
<td>28</td>
<td>38</td>
<td>4.5</td>
<td>4.5</td>
<td>24.1</td>
</tr>
<tr>
<td>29</td>
<td>51</td>
<td>6.0</td>
<td>6.0</td>
<td>30.1</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>3.5</td>
<td>3.5</td>
<td>33.6</td>
</tr>
<tr>
<td>31</td>
<td>42</td>
<td>4.9</td>
<td>4.9</td>
<td>38.6</td>
</tr>
<tr>
<td>32</td>
<td>30</td>
<td>3.5</td>
<td>3.5</td>
<td>42.1</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>3.6</td>
<td>3.6</td>
<td>45.8</td>
</tr>
<tr>
<td>34</td>
<td>38</td>
<td>4.5</td>
<td>4.5</td>
<td>50.2</td>
</tr>
<tr>
<td>35</td>
<td>40</td>
<td>4.7</td>
<td>4.7</td>
<td>54.9</td>
</tr>
<tr>
<td>36</td>
<td>33</td>
<td>3.9</td>
<td>3.9</td>
<td>58.8</td>
</tr>
<tr>
<td>37</td>
<td>31</td>
<td>3.6</td>
<td>3.6</td>
<td>62.5</td>
</tr>
<tr>
<td>38</td>
<td>30</td>
<td>3.5</td>
<td>3.5</td>
<td>66.0</td>
</tr>
<tr>
<td>39</td>
<td>41</td>
<td>4.0</td>
<td>4.0</td>
<td>70.0</td>
</tr>
<tr>
<td>40</td>
<td>32</td>
<td>3.8</td>
<td>3.8</td>
<td>74.6</td>
</tr>
<tr>
<td>41</td>
<td>36</td>
<td>4.2</td>
<td>4.2</td>
<td>78.8</td>
</tr>
<tr>
<td>42</td>
<td>16</td>
<td>1.9</td>
<td>1.9</td>
<td>80.7</td>
</tr>
<tr>
<td>43</td>
<td>22</td>
<td>2.6</td>
<td>2.6</td>
<td>83.3</td>
</tr>
<tr>
<td>44</td>
<td>15</td>
<td>1.8</td>
<td>1.8</td>
<td>85.1</td>
</tr>
<tr>
<td>45</td>
<td>21</td>
<td>2.5</td>
<td>2.5</td>
<td>87.5</td>
</tr>
</tbody>
</table>
Recoding age with the Visual Binner (Transform > Visual Binning)

- **Make Cutpoints** can find the breaks
- Binner can do automatic labels
- Generates RECODE and metadata syntax.
Rake to adjust proportions

<table>
<thead>
<tr>
<th>Age Group</th>
<th>Age Code</th>
<th>Control Percent age</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 20</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>20-29</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>30-39</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>40-49</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>50+</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

Percentages will be normalized (these could be counts)

Case count unchanged

You could do this without fancy technology!

5 equations in 5 unknowns
Raking example: two dimensions (*schoolsurvey.sav*)

- High school survey
  - simulated but realistic data
  - 4 equal grade sizes
  - N of males = N of females
  - 2000 respondents

- Marginal control totals are known but *not* the full joint distribution
  - If joint distribution known, just scale each cell

- Response rates vary by gender and grade
  - Gender
    - 60 % female
    - 50 % males
  - Grade
    - 90 % freshman
    - 80 % soph
    - 70 % junior
    - 60 % senior

- No response rate interaction between gender and grade

*school size = 3636*
AUTORECODE converts gender to a numeric variable

- RAKE does not accept string variables
- *Transform > Automatic Recode...*
- Generates appropriate value labels
Raking adjusts for the unit nonresponse

- **Data > Rake Weights…**

- Name a weight variable (must not already exist)

- For each control variable, enter pairs of category value and proportion or percentage

- Need not add to 100

- Omitted categories get weight of SYSMIS

- Only numeric variables can be used

- Optionally scale to a fixed population total

- Generates SPSSINC RAKE syntax
  - Up to 10 dimensions in syntax

- SPSSINC RAKE /HELP shows syntax chart
Raking results in exact adjustment

- RAKE automatically turns on the calculated weight
- Six equations (constraints) in 8 unknowns
- In general: – N1 + N2 equations in N1 * N2 unknowns
Exercise 1: Let's do one

- Using the schools dataset, *schoolsurvey.sav*, or, better, a dataset of your own, run a raking adjustment in two or more dimensions
- Make up your own control totals
- If your dataset already has a weight variable, set it on before raking
- Note that SPSSINC RAKE automatically turns on the output weight variable
- Run RAKE a second time naming a new output weight variable but same control totals
- Compare the two weight variables
## Unweighted and weighted crosstabs

### Unweighted

By default, CROSSTABS counts are rounded.
Joint distribution is different after weighting (Spearman correlations)

- The total sample size is different in the two tables
- **NONPAR CORR Algorithms**
  - If a WEIGHT variable is specified, it is used to replicate a case as many times as indicated by the weight value rounded to the nearest integer. If the workspace requirements are exceeded and sampling has been selected, a random sample of cases is chosen for analysis using the algorithm described in SAMPLE.
- We will look at this again later
The weight distribution shows only eight distinct values

- Weighted case count = unweighted count
- This is important when degrees of freedom matter
- IQR = 1.2086 - .8122 = .396
- Max weight = median \( wt + 1.25 \times IQR \)

### Custom Tables

```
[schoolSurvey] C:\aarp\schoolsurvey.sav
```

<table>
<thead>
<tr>
<th>gender</th>
<th>1 Female</th>
<th>2 Male</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>grade</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>.70</td>
<td>.00</td>
</tr>
<tr>
<td>10</td>
<td>.81</td>
<td>.00</td>
</tr>
<tr>
<td>11</td>
<td>.88</td>
<td>.00</td>
</tr>
<tr>
<td>12</td>
<td>1.01</td>
<td>.00</td>
</tr>
</tbody>
</table>

You will always see a decreasing pattern ->
Exercise 2: Study the weight distribution of your output weights

- Turn off weights first
- Frequencies on the weight
- Histogram
- Custom table of mean weight using raking variables in rows, columns, layer
Raking can scale weights up to population totals

- Convenient for scaling tabulations
- DO NOT USE for inference
Using REGRESSION with upscaled weights overstates degrees of freedom

- REGRESSION treats weights as replication weights
Output reports more degrees of freedom than there are cases

- t and F d.f are wrong
- Coefficients are okay

---

There are only 2000 cases
Empty categories stay empty

- Using …\samples\english\employee data.sav
- There are no female custodians in this dataset.
- Rake proportions (hypothetical)
  - jobcat .67 .10 .23
  - gender: .45 .55
- Weight distribution still achieves marginals
- Completely empty categories are ignored
Consider what happens if very small count

- Change one case to jobcat=4
- Rake proportions
  - jobcat .57 .10 .23 .10
  - gender: .45 .55
- The raker meets the goal, but the weight is extreme
  - Median: .9795
  - IQR: .4494
  - Max : median + 109 * IQR !
- This increases variability of results
- Solutions
  - Bound the weights by iteratively reraking
  - Collapse rare categories
    - Clustering?
SPSSINC RAKE allows up to ten dimensions

- The higher the number of dimensions the sparser and larger is the table
  - May produce more extreme weights
  - May not converge
  - May take a long time or run out of memory on 32-bit systems

- Ten dimensions with 4 categories each implies a table of size
  \[4^{10} = 1,048,576\] cells
  - Likely to exceed your sample size resulting in many empty cells

- Run time will depend on
  - Number of dimensions
  - Number of categories
  - Number of cases
  - How far the input data marginals are from the control totals

- A problem with 10 dimensions, each with 4 categories, total of 2000 cases, simulated data
  - GENLOG ran out of memory on a 32-bit system
  - Completed on a Win7 64-bit system in 1 hour
Performance statistics show the benefit of *external mode*

- External mode runs from a command prompt with no user interface like Statisticsb via Python
  - Statisticsb is part of SPSS Statistics Server

- External mode or Statisticsb can use much less memory and time

- A job can be converted to external mode with just a few keystrokes and run from Python
  - Cannot use scripting apis
  - Can capture Viewer output with OMS
The rake job in traditional and external mode forms

**Internal mode – syntax window**

```plaintext
get file="c:/aarp/tendimensions.sav".
* five rake vars each with 4 categories.
SPSSINC RAKE DIM1 = V1 0 15 1 35 2 45 3 20
DIM2=V2 0 15 1 35 2 45 3 20
DIM3=V3 0 15 1 35 2 45 3 20
DIM4=V4 0 15 1 35 2 45 3 20
DIM5=V5 0 15 1 35 2 45 3 20
FINALWEIGHT=weight.
```

**External mode – run from Python**

```python
import spss
spss.Submit("insert file="c:/aarp/rake5.sps".")
```

- In external mode there is no user interface, Viewer, or Data Editor present, but Viewer (spv) files can be produced via OMS
- Run the Python job as
  - `python myjob.py`
  - Plain text output can be captured or suppressed
- Statisticsb operates similarly but without the need for Python
Exercise 3: Run rake in external mode

- Create `myjob.py` in any text editor such as Notepad or using the syntax window
  - Use quotes around the name when saving to force the extension to be `.py`

- Add SAVE command, e.g.,
  - `SAVE OUTFILE="c:\aarp\tendimensionsraked.sav"`. 

- Run the Python job as
  - `python myjob.py`
  - May need path: `c:\python27\python myjob.py` or `c:\python26\python myjob.py`

- Open Statistics and examine the weight

- Set up OMS output capture and run again

```
oms /destination outfile="c:/aarp/results.spv" format=spv.
<your syntax>
omsend.
```
External mode can be much faster and use much less memory

- Capture Viewer output in spv format with OMS
- Viewer output by default appears as text in a console window, but that can be suppressed
- External mode requires Python Essentials – So does SPSSINC RAKE
- External mode applications can also be built with .NET plugin and provide a user interface
Partial knowledge of the joint distribution can be used along with the marginal distributions

- School example revisited

- Suppose known population information includes
  - gender totals
  - grade totals
  - gender by grade totals
  - totals of students from single parent households
    - But not known by gender grade

- Solution
  - Compute a variable representing both gender and grade
    \[
    \text{gendergrade} = 10 \times \text{ngender} + \text{grade}.
    \]
    \[19 = \text{ngender} = 1 \text{ and grade} = 9, \text{ i.e. female frosh}\]
  - Rake by gendergrade and singleparent using joint distribution of gender and grade and marginal distribution of singleparent

- With self-reported data, possible underreporting bias of single parent status (or gender or grade) is not fully resolved
  - differential nonresponse is corrected assuming accurate status
Issues: control totals

- Where do these come from?
- Algorithm treats them as exact
  - Use a source with little or no sampling variability such as census data
  - Could use a prior identical survey to control period to period variability
    - Alternative is to use the raking dimensions as controls
- Case-control matching might be useful if there is a treatment
  - FUZZY extension command can select these samples
- What if some categories of a raking variable are not available for other variables?
  - state by ethnicity
    - American Indian by state
    - No data on Indians except in a few states
    - Combined with white elsewhere
  - solution: create a nested rake dimension combining State and Ethnicity in one dimension using the partial knowledge approach
- A nested rake dimension removes the independence assumption otherwise used
Issue: missing data

- Values of a raking variable recorded as missing for some cases (item nonresponse)
- Example: age group sometimes not reported
- Affects matching national totals
- Solution: Calculate missing percentage and subtract from total age counts
Issues: unrakeable problems

- Can't make bricks without straw – if a category doesn't occur in a sample, raking can't match its total

Before

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>Count</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.00</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>% within x</td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% within y</td>
<td></td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2.00</td>
<td>Count</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>% within x</td>
<td></td>
<td>0.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% within y</td>
<td></td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>% within x</td>
<td></td>
<td>33.3%</td>
<td>66.7%</td>
</tr>
<tr>
<td>% within y</td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

After

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>Count</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.00</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>% within x</td>
<td></td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>% within y</td>
<td></td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2.00</td>
<td>Count</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>% within x</td>
<td></td>
<td>0.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>% within y</td>
<td></td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>% within x</td>
<td></td>
<td>50.0%</td>
<td>50.0%</td>
</tr>
<tr>
<td>% within y</td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

**SPSSINC RAKE**

DIM1 = x 1 50 2 50
DIM2=y 1 75 2 25
FINALWEIGHT=wt.
Impossible problems are - impossible

- Structurally empty cells will stay that way
- Seeding empty cells may result in large weights and, hence larger variance
- Probably better to collapse categories to avoid empty cells except for structurally empty (e.g. pregnant males)
Issues: Bounding the weights

- Sometimes raking can produce a few very large weights

- Examine histogram and quartiles of weight
  - May increase sampling variability
  - Bounding the weight introduces some bias but may reduce MSE

- Ad hoc rules for bounding
  - 5 * mean weight
  - (median weight) + 6 * IQR of weight

- Algorithm
  - Rake
  - Truncate output weights according to bounding rule
  - Rake starting with truncated weights
  - Repeat until all weights within bounds or boredom sets in
  - In truncation after first step use 1 + criterion
  - This can take many iterations
Python program run in Statistics or in external mode can bound weights
Part 1 - run a RAKE command in a program from the syntax window

- rakeTrimmer.sps
- Python code placed between *begin program* and *end program*

```
begin program.
# rake iteration example - straw man
import spss, spssaux

cmd = r"*****SPSSINC RAKE DIM1 = jobcat 1 .57 2 .10 3 .23 4 .10
DIM2=ngender 1 .45 2 .55 FINALWEIGHT=weight."*****
spss.Submit(cmd)
```
Part 2 - Check the weight distribution and, if necessary, trim large weights and run RAKE again

```python
for i in range(10):
    spss.Submit("WEIGHT OFF")
    tag, err = spssaux.createXmlOutput(r"FREQUENCIES weight /format notable/ntiles=4/statistics=max", visible=True)
    values = spssaux.getValuesFromXmlWorkspace(tag, "Statistics", cellAttrib="number")
    median = float(values[-2])
    iqr = float(values[-1]) - float(values[-3])
    themax = values[-4]
    criterion = median + 6 * iqr
    if i > 0:
        criterion += 1
    if themax <= criterion:
        break
    spss.Submit("compute weight2 = min(weight, %s). execute.
    delete variable weight.
    weight by weight2." % criterion)
    spss.Submit(cmd)
```
Part 3 – Report the outcome

```plaintext
print """"Trimming Iterations: %s\n
Final Criterion: %s\n
Final Maximum Weight: %s"""

(i+1, criterion, themax)

end program.
```

Trimming Iterations: 10

Final Criterion: 2.49540909795

Final Maximum Weight: 20.342139354167

- A real implementation would parameterize the rake specification and maximum iteration limit, and stabilize the count
SPSS procedures vary in how weights are treated

- Procedures such as Frequencies, Crosstabs, and Ctables will use fractional weights
- Most other procedures treat weight as a replication weight and round
  - NPTESTS

<table>
<thead>
<tr>
<th>Warnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency weight data values must be positive integers. Positive non-integer frequency weight data values are encountered. These values are rounded to the nearest integers for the analysis.</td>
</tr>
</tbody>
</table>

- Some procedures ignore weights
- Degrees of freedom will be wrong in procedures such as regression if sum of weights does not match number of cases and weights do not represent replication
- Complex Samples procedures treat weights as representing probability of selection
  - Computations reflect sampling design
  - CSCOXREG vs COXREG for survival problems
  - Gentle raking could be used for post stratification
- Weight status reported in Notes table and on Data Editor status bar
CROSSTABS weight options determine how fractional weights are treated

- "if the data file is currently weighted by a weight variable with fractional values (for example, 1.25), cell counts can also be fractional values. You can truncate or round either before or after calculating the cell counts or use fractional cell counts for both table display and statistical calculations."

- CROSSTABS options
  - use weights as is
  - round
  - truncate
  - round or truncate accumulated cell counts

- Statistics such as Chi-square are calculated based on the table
  - Set to No adjustments to handle non-integer weights
Summary

- Raking can adjust for nonrepresentativeness of a sample based on knowledge of only the marginal distributions of one or more relevant classifying variables.
- This may increase variability of results, reducing bias while increasing variance.
- The SPSSINC RAKE extension command constructs the weight.
- SPSSINC RAKE is one of many extension commands available free from the SPSS Community website (www.ibm.com/developerworks/spssdevcentral).
- Good practice is to examine the distribution of the constructed weights and consider adjustment of extremely large values by truncation or iteration.
- External mode can conserve computing resources.
- Implications of using fractional weights should be considered for each statistical procedure used.
- Weighted/rewighted datasets can give a better picture of the population when used in moderation.
Questions
My contact information

- Jon Peck
- peck@us.ibm.com