

# How I use look-up tables to conduct large volumes of analyses

By

Craig Hansen, PhD

Extra Time:

And if there is time I will show you how to create a guitar chord book  
using look-up tables in SAS

# WHY USE A LOOK UP TABLE

- When there are lots of analyses
- Create 'structure' to the list of analyses (transparency)
- Avoid copying and pasting lots of code
- Data requests being updated
- Select particular analyses from the long list

Why not just do something like this

```
ODS OUTPUT CrossTabs=FREQS_CURRENT;  
PROC FREQ DATA=mydata;  
TABLE (sex age education)*(asthma cvd ptsd);  
RUN;  
ODS OUTPUT CLOSE;
```

What if there are 100s of variables?

One big messy output that needs re-shaping

```
ODS OUTPUT CrossTabs=FREQS_CURRENT;  
PROC FREQ DATA=mydata;  
TABLE (&stravarList.)*(&OutcomeList.);  
RUN;  
ODS OUTPUT CLOSE;
```

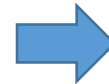
Get the list of variables into a macro var

# ANALYSIS PLAN: set up look-up tables

Variables of interest

Stratification variables	Outcome variables
Sex	Asthma
Age	CVD
Education	PTSD

Create all combinations



Simplified look up table

Strat_Var	Outcome	StratVarID	OutVarID
Sex	Asthma	1	1
Sex	CVD	1	2
Sex	PTSD	1	3
Age	Asthma	2	1
Age	CVD	2	2
Age	PTSD	2	3
Education	Asthma	3	1
Education	CVD	3	2
Education	PTSD	3	3

Import csv/excel file

Get list from data dictionaries

Use data step to create look ups

# ANALYSIS PLAN: set up look-up tables

Example look up table – over 1000 records

STRAT_VAR	COMPARISON	SUBPOPULATION	QUESTION	OUTCOME	TYPE	FRMT	STRATVAR_ID	OUTCOME_ID
ALL	Transition only	adf_all_hours	Q1	adf_all_hours_helpful	FREQS	\$xxxxx.	1	1
ALL	Transition only	adf_website	Q1	adf_website_helpful	FREQS	\$xxxxx.	1	2
ALL	Transition only	at_ease_website	Q1	at_ease_website_helpful	FREQS	\$xxxxx.	1	3
ALL	Transition only	beyond_blue_website	Q1	beyond_blue_website_helpful	FREQS	\$xxxxx.	1	4
ALL	Transition only	black_dog_website	Q1	black_dog_website_helpful	FREQS	\$xxxxx.	1	5
ALL	Transition only	blogs	Q1	blogs_helpful	FREQS	\$xxxxx.	1	6
ALL	Transition only	defence_family	Q1	defence_family_helpful	FREQS	\$xxxxx.	1	7
ALL	Transition only	dva_website	Q1	dva_website_helpful	FREQS	\$xxxxx.	1	8
ALL	Transition only	ecouch	Q1	ecouch_helpful	FREQS	\$xxxxx.	1	9

# ANATOMY OF THE LOOPING MACRO

Base dataset for appending (empty)

Outer loop (outcome variables)

Inner loop (stratification variables)

PROC SQL – get vars from look up table

PROC FREQ – ODS OUTPUT

PROC SQL – CLEANING

PROC APPEND – ADD TO BASE

PROC DATASETS – DELETE UNWANTED

END Inner loop

END Outer loop

```
DATA R2LEVEL2_FREQS
LENGTH STRATVAR $50.
[more code in here]
RUN;

PROC SQL NOPRINT;
SELECT DISTINCT MAX(OUTCOME_ID), MAX(STRATVAR_ID)
INTO :MAX_OUTV, :MAX_STRATV
FROM TEMPL2.CT_LEVEL2_ANALYSES;
QUIT;

%DO j = 1 %TO &MAX_OUTV.;
  %DO k = 1 %TO &MAX_STRATV.;
    PROC SQL NOPRINT;
    SELECT DISTINCT
    OUTCOME, STRAT_VAR, TYPE INTO :OUTV TRIMMED, :STRATV TRIMMED, :TYP TRIMMED
    FROM TEMPL2.&COMP._LEVEL2_ANALYSES
    WHERE OUTCOME_ID=&j. AND STRATVAR_ID=&k.;
    QUIT;

    ODS OUTPUT CrossTabs=FREQS;
    PROC FREQ DATA=mydata;
    TABLE (&STRATV.)*(&OUTV.);
    RUN;
    ODS OUTPUT CLOSE;

    PROC SQL;
    CREATE TABLE FREQS_NEW AS
    SELECT DISTINCT
    [more code in here]
    FROM FREQS;
    QUIT;
    PROC APPEND DATA=FREQS_NEW BASE=R2LEVEL2_FREQS FORCE;
    RUN;
    PROC DATASETS LIB=WORK NOLIST;
    DELETE FREQS_NEW FREQS;
    QUIT; RUN;
  %END;
%END;
```

# REPORT/SHARE RESULTS

## • PROCS USED

- ODS
  - HTML/Tagsets
- REPORT
- SGPLOT

## Simplified example of PROC REPORT

```
ODS tagsets.Excelxp FILE="&LOCATION.\R2_OR_S_&COMP..XLS" STYLE=MINIMAL
OPTIONS (AUTOFIT_HEIGHT='yes'
SHEET_NAME="R2_OR_S_&COMP."
FROZEN_HEADERS='yes'
AUTOFILTER='all');
```

```
PROC REPORT DATA=REP2.R2_OR_S_&COMP. HEADLINE HEADSKIP NOWD MISSING
STYLE (HEADER)=[FONT_FACE=Arial FONT_WEIGHT=BOLD FONT_SIZE=8PT]
STYLE (COLUMN)=[FONT_FACE=Arial FONT_SIZE=8PT];
COLUMN STRATVAR STRATVAR_VALUE OUTCOMEVAR OUTCOME_VALUE Frequency;
DEFINE STRATVAR / DISPLAY 'BY Variable' STYLE (COLUMN)= [cellwidth=3cm];
DEFINE STRATVAR_VALUE / DISPLAY 'By Value' STYLE (COLUMN)= [cellwidth=3cm];
DEFINE OUTCOMEVAR / DISPLAY 'Outcome' STYLE (COLUMN)= [cellwidth=3cm];
DEFINE OUTCOME_VALUE / DISPLAY 'Value' STYLE (COLUMN)= [cellwidth=3cm];
DEFINE Frequency / DISPLAY 'n' STYLE (COLUMN)= [cellwidth=3cm];
RUN;
TITLE;
```

```
ODS tagsets.Excelxp CLOSE;
```

## Filter results

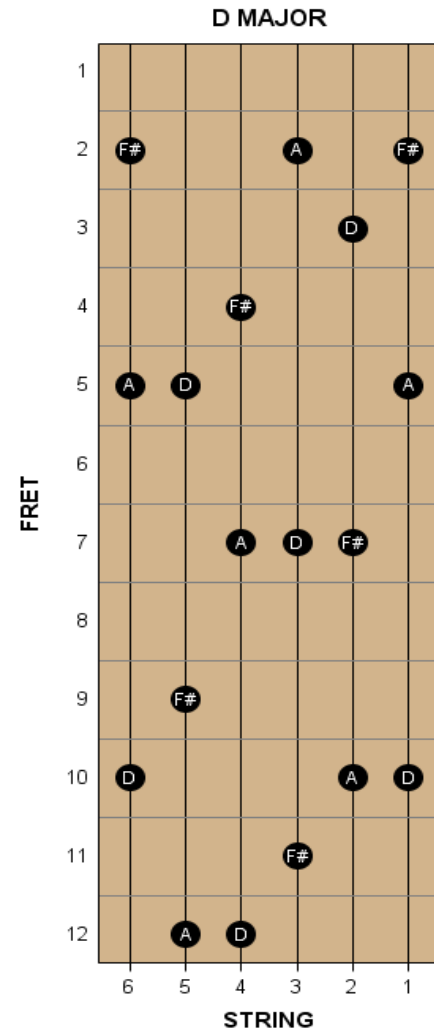
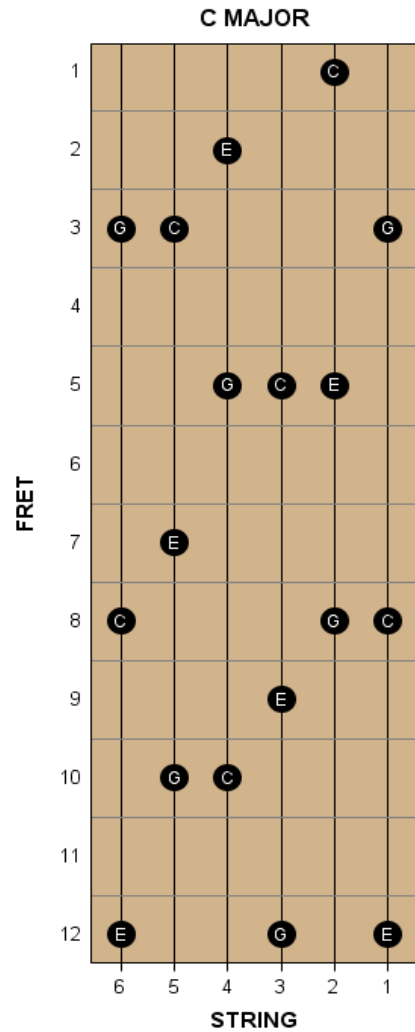
The screenshot shows an Excel spreadsheet with the following data table:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Question	TWRP_GRP	Sub-Population	Sub-Pop. Value	Analyses	BY Variable	By Value	Outcome	Outcome Value	n	Weighted n	StdDev	StdErr	%	LowerCL	UpperCL
14	Q1	Current	adf_all_hours	1	Table 2 of ANY_PCL_K10 * adf_all_hours_helpful	ANY_PCL_K10	0	adf_all_hours_helpful								
15	Q1	Current	adf_all_hours	1	Table 2 of ANY_PCL_K10 * adf_all_hours_helpful	ANY_PCL_K10	0	adf_all_hours_helpful	0							
16	Q1	Current	adf_all_hours	1	Table 2 of ANY_PCL_K10 * adf_all_hours_helpful	ANY_PCL_K10	0	adf_all_hours_helpful	1							

The Filter task pane on the right shows the 'Outcome' column selected for filtering. The 'Filter by Color' section is expanded, and a search for 'Outcome' has been performed. The following items are listed in the search results:

- (Select All)
- accessibility
- accessibility\_GRP
- adf\_all\_hours\_helpful
- adf\_website\_helpful
- afraid\_to\_ask
- afraid\_to\_ask\_GRP
- anything\_could\_help
- anything\_could\_help\_GRP
- assist\_seek\_help
- at\_ease\_website\_helpful
- beyond\_blue\_website\_helpful
- black\_dog\_website\_helpful
- blogs\_helpful
- can\_function

# Create a Guitar Chord Book



Who would have thought you could create a guitar chord book using SAS

# Dataset of all the music notes

```
* SET UP FORMATS FOR THE NOTES OF MUSIC;
PROC FORMAT;
VALUE S_NOTES          1='A' 2='A#' 3='B' 4='C' 5='C#' 6='D' 7='D#' 8='E' 9='F' 10='F#' 11='G' 12='G#';
VALUE F_NOTES          1='A' 2='Bb' 3='B' 4='C' 5='Db' 6='D' 7='Eb' 8='E' 9='F' 10='Gb' 11='G' 12='Ab';

VALUE C_MAJOR          1='C' 2='D' 3='E' 4='F' 5='G' 6='A' 7='B';
VALUE G_MAJOR          1='G' 2='A' 3='B' 4='C' 5='D' 6='E' 7='F#';
VALUE D_MAJOR          1='D' 2='E' 3='F#' 4='G' 5='A' 6='B' 7='C#';
VALUE A_MAJOR          1='A' 2='B' 3='C#' 4='D' 5='E' 6='F#' 7='G#';
VALUE E_MAJOR          1='E' 2='F#' 3='G#' 4='A' 5='B' 6='C#' 7='D#';
VALUE B_MAJOR          1='B' 2='C#' 3='D#' 4='E' 5='F#' 6='G#' 7='A#';
VALUE FS_MAJOR         1='F#' 2='G#' 3='A#' 4='B' 5='C#' 6='D#' 7='E#';
RUN;

* SET UP THE 12 NOTES OF MUSIC;
DATA KEYS;
DO NUM = 1 TO 12;
  S_NOTES=PUT(NUM,S_NOTES.);
  F_NOTES=PUT(NUM,F_NOTES.);
  OUTPUT;
END;
RUN;
```

NUM	S_NOTES	F_NOTES
1	A	A
2	A#	Bb
3	B	B
4	C	C
5	C#	Db
6	D	D
7	D#	Eb
8	E	E
9	F	F
10	F#	Gb
11	G	G
12	G#	Ab



# Dataset of chords (major)

```
* IDENTIFY THE NOTES THAT BELONG TO THE MAJOR KEYS;
```

```
DATA KEYS;
```

```
DO NOTES = 1 TO 7;
```

```
  C_MAJOR=PUT (NOTES, C_MAJOR.);
```

```
  G_MAJOR=PUT (NOTES, G_MAJOR.);
```

```
  D_MAJOR=PUT (NOTES, D_MAJOR.);
```

```
  A_MAJOR=PUT (NOTES, A_MAJOR.);
```

```
  E_MAJOR=PUT (NOTES, E_MAJOR.);
```

```
  B_MAJOR=PUT (NOTES, B_MAJOR.);
```

```
  FS_MAJOR=PUT (NOTES, FS_MAJOR.);
```

```
OUTPUT;
```

```
END;
```

```
RUN;
```

```
DATA KEYS; SET KEYS;
```

```
IF NOTES IN (1,3,5) THEN MAJ=1;
```

```
IF NOTES IN (1,3,5,7) THEN MAJ7=1;
```

```
IF NOTES IN (1,3,5,7,9) THEN MAJ9=1;
```

```
IF NOTES IN (1,3,6) THEN MAJ6=1;
```

```
RUN;
```

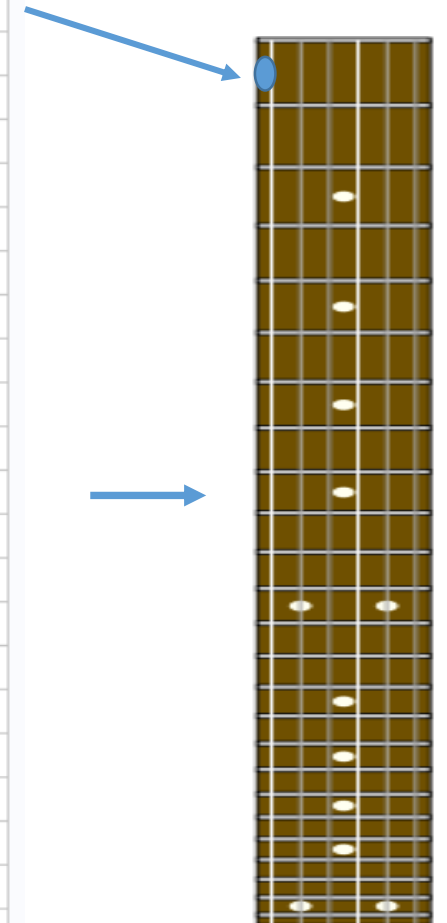
NOTES	C_MAJOR	G_MAJOR	D_MAJOR	A_MAJOR	E_MAJOR	B_MAJOR	FS_MAJOR	MAJ	MAJ7	MAJ9	MAJ6
1	C	G	D	A	E	B	F#	1	1	1	1
2	D	A	E	B	F#	C#	G#	.	.	.	.
3	E	B	F#	C#	G#	D#	A#	1	1	1	1
4	F	C	G	D	A	E	B	.	.	.	.
5	G	D	A	E	B	F#	C#	1	1	1	.
6	A	E	B	F#	C#	G#	D#	.	.	.	1
7	B	F#	C#	G#	D#	A#	E#	.	1	1	.

# Dataset of the notes on guitar

```
* SET UP THE NOTES ON THE FRETBOARD;  
  
DATA FRETBOARD_NOTES;  
/* 6TH STRING */  
STRING=6;  
    DO _NOTE=9 TO 12;      OUTPUT; END;  
    DO _NOTE=1 TO 8;      OUTPUT; END;  
/* 5TH STRING */  
STRING=5;  
    DO _NOTE=2 TO 12;     OUTPUT; END;  
    DO _NOTE=1 TO 1;     OUTPUT; END;  
/* 4TH STRING */  
STRING=4;  
    DO _NOTE=7 TO 12;     OUTPUT; END;  
    DO _NOTE=1 TO 6;     OUTPUT; END;  
/* 3RD STRING */  
STRING=3;  
    DO _NOTE=12 TO 12;    OUTPUT; END;  
    DO _NOTE=1 TO 11;    OUTPUT; END;  
/* 2ND STRING */  
STRING=2;  
    DO _NOTE=4 TO 12;     OUTPUT; END;  
    DO _NOTE=1 TO 3;     OUTPUT; END;  
/* 1ST STRING */  
STRING=1;  
    DO _NOTE=9 TO 12;     OUTPUT; END;  
    DO _NOTE=1 TO 8;     OUTPUT; END;  
  
RUN;  
DATA FRETBOARD_NOTES; SET FRETBOARD_NOTES;  
BY STRING NOTSORTED;  
IF FIRST.STRING THEN FRET=1;  
ELSE FRET+1;  
NOTE=PUT(_NOTE,S_NOTES.);  
F_NOTE=PUT(_NOTE,F_NOTES.);  
RUN;
```

STRING	_NOTE	FRET	NOTE	F_NOTE
6	9	1	F	F
6	10	2	F#	Gb
6	11	3	G	G
6	12	4	G#	Ab
6	1	5	A	A
6	2	6	A#	Bb
6	3	7	B	B
6	4	8	C	C
6	5	9	C#	Db
6	6	10	D	D
6	7	11	D#	Eb
6	8	12	E	E
5	2	1	A#	Bb
5	3	2	B	B
5	4	3	C	C
5	5	4	C#	Db
5	6	5	D	D
5	7	6	D#	Eb
5	8	7	E	E
5	9	8	F	F
5	10	9	F#	Gb
5	11	10	G	G
5	12	11	G#	Ab
5	1	12	A	A

Guitar Fretboard



# Select chord and create plot

```
* PLOT THE NOTES OF THE SELECTED CHORD;
PROC SQL;
CREATE TABLE PLOTME AS
SELECT DISTINCT
    STRING
    , FRET
    , NOTE
FROM FRETBOARD_NOTES
WHERE NOTE IN (SELECT DISTINCT C_MAJOR FROM KEYS WHERE MAJ NE .);
QUIT;

ODS LISTING GPATH="C:\Users" STYLE=GUITSTYLE;
ODS GRAPHICS / RESET=ALL IMAGEFMT=PNG IMAGENAME="C MAJOR" NOBORDER WIDTH=8CM HEIGHT=20CM;
TITLE 'C MAJOR';
PROC SGPLOT DATA=PLOTME NOAUTOLEGEND;
    REFLINE 6 5 4 3 2 1 / AXIS=X LINEATTRS=(COLOR=BLACK);
    REFLINE 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 / AXIS=Y;

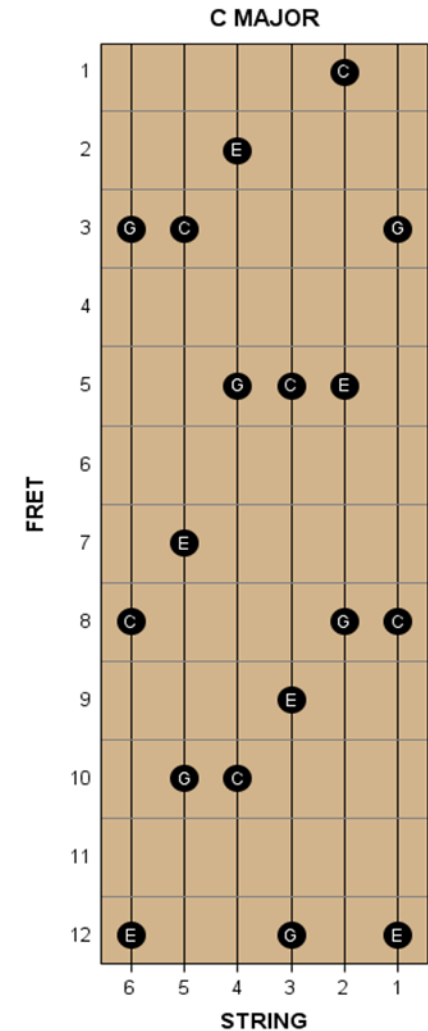
    SCATTER X=STRING Y=FRET /
    MARKERATTRS=(SYMBOL=CIRCLEFILLED COLOR=BLACK SIZE=18);

    SCATTER X=STRING Y=FRET /
    MARKERCHAR=NOTE
    MARKERCHARATTRS=(COLOR=WHITE SIZE=8);

    YAXIS REVERSE VALUES=(1 TO 12 BY 1) DISPLAY=(NOTICKS NOLINE);
    XAXIS REVERSE GRID;

RUN;
ODS LISTING CLOSE;
```

C MAJOR		
STRING	FRET	NOTE
1	3	G
1	8	C
1	12	E
2	1	C
2	5	E
2	8	G
3	5	C
3	9	E
3	12	G
4	2	E
4	5	G
4	10	C
5	3	C
5	7	E
5	10	G
6	3	G
6	8	C
6	12	E



Thankyou – questions?