The Variable Key Overview

Paul Johnson[1]

[1]Center for Research Methods and Data Analysis

2018

Outline

KU

About CRMDA

http://crmda.ku.edu

- Research project planning & support
- Data/stat consulting (fee for service)
- Summer Statistical Institute
- Software Carpentry-ish workshops, training for Git, shell, LaTeX, stat packs

- High performance computing access point
- R package development for various purposes
  - `rockchalk` : regression plots & tables
  - `portableParallelSeeds` : HPC random stream manager
  - `kutils` : data structuring strategies
  - `stationery` : Sweave, Markdown templates & compiler

KU

The kutils package for R

- kutils
  - release version available on CRAN
  - test versions on KRAN: `http://rweb.crmda.ku.edu/kran`
- Vignette, "The Variable Key Data Management Framework"

KU

Outline

KU

Clients, Data Managers and Data Analysts

- Clients give us rectangular data structures
  - Qualtrics (or similar) output, csv, Excel sheets
  - Data sets from SPSS, Stata, etc
  - Large surveys distributed by NORC, ICPSR, etc
- Perhaps there are
  - many files and
  - desire to "normalize" and integrate them (if lucky, "crosswalk file" exists)

KU

What happens next?

- Data Analyst uses a stats program, say R (R Core Team, 2017).
- Analyst gazes at columns of data
  - Renames them for her/his convenience
  - Re-assigns values for her/his convenience
- Analyst sends some model summary tables and graphs
- When we notice something peculiar in the report, we go digging in the archives for the code. What do we find?

KU

Example 1

```
## Read in raw data files in .dat format
w6 <- read.table("w6.dat", header = TRUE)
w7 <- read.table("w7.dat", header = TRUE)
w8 <- read.table("w8.dat", header = TRUE)

##      Merging Data      ##

## Create variable names for merged dataset
mnames <- c("distid", "wave", "region",
   "province", "district", "sec1", "sec2",
   "sec3", "sec4", "sec5","trn1", "trn2",
   "trn3", "trn4", "crp1", "crp2", "crp3",
   "crp4", "crp5","pgv1", "pgv2", "pgv3",
   "pgv4", "pgv5","dgv1", "dgv2", "dgv3",
   "dgv4", "dgv5", "dgv6", "rec1", "rec2")
```

Example 1 ...

```
## Wave 6 data management
w6$wave <- 6
w6$fill <- NA
w6vars <- c("dist", "wave", "m4b", "m7",
   "m5", "q1", "q2", "q3", "q9", "q30",
   "q12", "q14", "q20", "q26", "q13", "q21",
   "q35e", "q36e", "q63", "q35a", "q41a",
   "q41b", "q41c", "q41d", "q36a", "q43a",
   "q43b","q43c", "q43d", "q47c", "q10",
   "q73")
w6 <- w6[w6vars]
colnames(w6) <- mnames

## Wave 7 data management
```

Example 1 ...

```
w7$wave <- 7
w7$fill <- NA
w7vars <- c("dist", "wave", "m4b", "m7",
   "m5","q1", "q2", "q3", "q9", "TB_Strngth",
   "q11", "q12", "q15", "q17", "fill", "q16",
   "q27e", "q28e", "q60", "Prov_Gvnr_Perf",
   "q34a", "q34b", "q34c", "q35",
   "Dist_Gvnr_Perf", "q37a", "q37b", "q37c",
   "q38", "q44c", "q10", "q72")
w7 <- w7[w7vars]
colnames(w7) <- mnames

## Wave 8 data management
w8$wave <- 8
w8$fill <- NA
```

Example 1 ...

```
w8vars <- c("dist", "wave", "m4b", "m7",
   "m5","q1", "q2", "q3", "q5", "TB_Strngth",
   "q13", "q15", "q18", "q20", "q14", "q19",
   "q37e", "q38e", "q64",
   "Prov_Gvnr_Perf","q43a", "q43b", "q43c",
   "q44", "Dist_Gvnr_Perf", "fill",
   "fill","fill", "fill", "fill", "q7", "q76")
w8 <- w8[w8vars]
colnames(w8) <- mnames

## Merge datasets
mdata <- rbind(w6, w7, w8)
mdata[mdata == -999] <- NA

##-------------------##
```

Example 1 ...

```
##      Recoding      ##
##--------------------##

## "sec1"
mdata$sec1[mdata$sec1 == 100] <- 1
mdata$sec1[mdata$sec1 == 101] <- 2
mdata$sec1[mdata$sec1 == 102] <- 3
mdata$sec1[mdata$sec1 > 3] <- NA

## "sec2"
mdata$sec2[mdata$sec2 == 100] <- 1
mdata$sec2[mdata$sec2 == 101] <- 2
mdata$sec2[mdata$sec2 == 102] <- 3
mdata$sec2[mdata$sec2 > 3] <- NA
```

Example 1 ...

```
## "sec3"
mdata$sec3[mdata$sec3 == 100] <- 1
mdata$sec3[mdata$sec3 == 101] <- 2
mdata$sec3[mdata$sec3 == 102] <- 3
mdata$sec3[mdata$sec3 > 3] <- NA

## "sec4"
mdata$sec4[mdata$sec4 > 5] <- NA
mdata$sec4[mdata$sec4 < 1] <- NA

## "sec5"
mdata$sec5[mdata$sec5 > 3] <- NA

## "trn1" Recoded
mdata$trn1[mdata$trn1 == 100] <- 6
```

Example 1 ...

```
mdata$trn1[mdata$trn1 == 101] <- 5
mdata$trn1[mdata$trn1 == 102] <- 4
mdata$trn1[mdata$trn1 == 103] <- 3
mdata$trn1[mdata$trn1 == 104] <- 2
mdata$trn1[mdata$trn1 == 7] <- 6
mdata$trn1[mdata$trn1 == 7] <- 6
mdata$trn1[mdata$trn1 == 105] <- 1
mdata$trn1[mdata$trn1 > 6] <- NA
mdata$trn1 <- mdata$trn1 + 6
mdata$trn1[mdata$trn1 == 7] <- 6
mdata$trn1[mdata$trn1 == 8] <- 5
mdata$trn1[mdata$trn1 == 9] <- 4
mdata$trn1[mdata$trn1 == 10] <- 3
mdata$trn1[mdata$trn1 == 11] <- 2
mdata$trn1[mdata$trn1 == 12] <- 1
```

Example 1 ...

```
## "trn2"
mdata$trn2[mdata$trn2 == 100] <- 1
mdata$trn2[mdata$trn2 == 101] <- 2
mdata$trn2[mdata$trn2 == 102] <- 3
mdata$trn2[mdata$trn2 == 103] <- 4
mdata$trn2[mdata$trn2 == 104] <- 5
mdata$trn2[mdata$trn2 > 5] <- NA
```

Example 2

```
library(car)

## Read in data
dat<-read.csv(file="fulldata.csv", header =
   TRUE, na.string = c("-980", "-981",
   "-982", "-983", "-984", "-985", "-986",
   "-987", "-988", "-989", "-990", "-991",
   "-992", "-993", "-994", "-995", "-996",
   "-997", "-998", "-999"))
##head(dat)
summary(as.factor(dat$w2_Dis12))
summary(as.factor(dat$np1Dis_Recod))
##family predictors, home independence
dat$Rnp1G1e<-recode(dat$np1G1e, "0=2;1=1;2=0")
dat$Rnp1G1h<-recode(dat$np1G1h, "0=2;1=1;2=0")
```

Example 2 ...

```
dat$Rnp1G5a<-recode(dat$np1G5a,
   "4=1;3=2;2=3;1=4")
dat$Rnp1G5b<-recode(dat$np1G5b,
   "4=1;3=2;2=3;1=4")
dat$Rnp1G5c<-recode(dat$np1G5c,
   "4=1;3=2;2=3;1=4")
dat$Rnp1G5d<-recode(dat$np1G5d,
   "4=1;3=2;2=3;1=4")
dat$Rnp1F1d<-recode(dat$np1F1d,
   "1=6;2=5;3=4;4=3;5=2;6=1")

##family predictors, parent perception of
   school exp
dat$Rnp1D12a<-recode(dat$np1D12a,
   "4=1;3=2;2=3;1=4")
```

Example 2 ...

```
dat$Rnp1D12b<-recode(dat$np1D12b,
   "4=1;3=2;2=3;1=4")
dat$Rnp1D12c<-recode(dat$np1D12c,
   "4=1;3=2;2=3;1=4")
dat$Rnp1D12d<-recode(dat$np1D12d,
   "4=1;3=2;2=3;1=4")
dat$Rnp1D12e<-recode(dat$np1D12e,
   "4=1;3=2;2=3;1=4")
dat$Rnp1H4<-recode(dat$np1H4,
   "4=1;3=2;2=3;1=4")

### student predictors, cpmmunication skills
dat$Rnp1B5a<-recode(dat$np1B5a,
   "4=1;3=2;2=3;1=4")
```

Example 2 ...

```
dat$Rnp1B5b<-recode(dat$np1B5b,
   "4=1;3=2;2=3;1=4")
dat$Rnp1B5d<-recode(dat$np1B5d,
   "4=1;3=2;2=3;1=4")
dat$Rnp1B5e<-recode(dat$np1B5e,
   "4=1;3=2;2=3;1=4")

### student predictors, classroom behavior
dat$Rnpr1C4a<-rowMeans(cbind(dat$npr1C4a,
   dat$npr1D18a, dat$nts1C1a), na.rm=T)
dat$Rnpr1C4b<-rowMeans(cbind(dat$npr1C4b,
   dat$npr1D18b, dat$nts1C1b), na.rm=T)
dat$Rnpr1C4c<-rowMeans(cbind(dat$npr1C4c,
   dat$npr1D18c, dat$nts1C1c), na.rm=T)
```

Example 2 ...

```
dat$Rnpr1C4d<-rowMeans(cbind(dat$npr1C4d,
   dat$npr1D18d, dat$nts1C1d), na.rm=T)
dat$Rnpr1C5a<-rowMeans(cbind(dat$npr1C5a,
   dat$npr1D19a, dat$nts1C6a), na.rm=T)
dat$Rnpr1C5b<-rowMeans(cbind(dat$npr1C5b,
   dat$npr1D19b, dat$nts1C6b), na.rm=T)
dat$Rnpr1C5c<-rowMeans(cbind(dat$npr1C5c,
   dat$npr1D19c, dat$nts1C6c), na.rm=T)
dat$npr1C5d2<-recode(dat$npr1C5d,
   "1=4;2=3;3=2;4=1")
dat$npr1D19d2<-recode(dat$npr1D19d,
   "1=4;2=3;3=2;4=1")
dat$nts1C6d2<-recode(dat$nts1C6d,
   "1=4;2=3;3=2;4=1")
```

Example 2 ...

```
dat$Rnpr1C5d<-rowMeans(cbind(dat$npr1C5d2,
   dat$npr1D19d2, dat$nts1C6d2), na.rm=T)
dat$Rnpr1C5e<-rowMeans(cbind(dat$npr1C5e,
   dat$npr1D19e, dat$nts1C6e), na.rm=T)

### student predictors, well-being
dat$Rnp1D12b<-recode(dat$np1D12b,
   "4=1;3=2;2=3;1=4")
dat$Rnts1C2b<-rowMeans(cbind(dat$nts1C2b,
   dat$npr1D7b), na.rm=T)
dat$Rnts1C2f<-rowMeans(cbind(dat$nts1C2f,
   dat$npr1D7f), na.rm=T)
dat$RndaF1friend<-recode(dat$ndaF1_friend,
   "2=1;3=2;1=3")
```

Example 2 ...

```
dat$RndaF2lonely<-recode(dat$ndaF2_lonely,
   "2=1;3=2;1=3")

## youth outcome, rights
## dat$np5P7b_J3b, dat$np5T4l_L4l,
   np5S3k_S4i_S5j_K6j_K7g_K8h, np5T4a_L4o
dat$RTgrpsn <- dat$np5P7b_J3b
dat$RTwdis <- dat$np5T4l_L4l
dat$RTserps <- dat$np5S3k_S4i_S5j_K6j_K7g_K8h
dat$RTseremp <- dat$np5T4o_L4o



## youth outcome, societal inclusion
```

Example 2 ...

```
dat$SISCRCSV <- (dat$np5T10c_C1b_a +
   dat$np5T10c_C1b_b + dat$np5T10c_C1b_c +
   dat$np5T10c_C1b_d + dat$np5T10c_C1b_e +
   dat$np5T10c_C1b_f + dat$np5T10c_C1b_g +
   dat$np5T10c_C1b_h + at$np5T10c_C1b_i +
   dat$np5T10c_C1b_j + dat$np5T10c_C1b_k +
   dat$np5T10c_C1b_l + dat$np5T10c_C1b_m +
   dat$np5T10c_C1b_n + dat$np5T10c_C1b_o +
   dat$np5T10c_C1b_p + dat$np5T10c_C1b_q +
   dat$np5T10c_C1b_r + dat$np5T10c_C1b_s +
   dat$np5T10c_C1b_t + dat$np5T10c_C1b_u +
   dat$np5T10c_C1b_v)
dat$SISCSEMG <- dat$np5C1b_u
dat$SISHLPWK <- dat$np5S3i_S4g_S5h_K6h_K7e_K8f
dat$SICPCOMA <- dat$np5P6_A4h
```

Example 2 ...

```
dat$SICYVCSV <- dat$np5P8_J4
dat$SICYRTVT <- dat$np5U9_J16
dat$SICPGPAC <- dat$np5A4h
```

What's wrong here?

- Hard-to-catch user errors
  - The blending of *project-specific values* with *programming idioms* requires an expert in both to review the work
- Costly/time-consuming to create comprehensive report to client about what was done to their data

KU

Outline

KU

All Involved Parties have a Shared Variable Key

Coordinate by creating a Variable Key file.

name_old	name_new	class_old	class_new	value_old	value_new
V23419	sex	integer	factor	1|2|3	male|female|neither
V32422	education	integer	ordered	1|2|3|4|5	elem|hs|somecoll|ba|post
V54532	income	numeric	numeric	.	.

Each column in an R data frame is characterized by its class. The ones we are focused on are:

`logical`, `character`, `integer`, `double` (aka numeric), `factor`, `ordered`

Example: Johnson County Basic Risk Factors

	A	B	C	D	E	F	G	H	I	J	K
1	name_old	name_new	class_old	class_new	value_old	value_new	missings	recodes			
2	SEQNO	seqno	integer	integer							
3	BPHIGH4	bphigh4	integer	integer	1|2|3|4	1|2|3|4					
4	BPHIGH4	bphigh4f	integer	factor	1|2|3|4	Yes|Preg|No|Borderline					
5	DIABETE3	diabete3f	integer	integer	1|2|3|4	1|2|3|4					
6	DIABETE3	diabete3	integer	factor	1|2|3|4	Yes|Preg|No|Borderline					
7	AGE	age	integer	integer							
8	SEX	sex	integer	integer	1|2	1|2					
9	SEX	sexf	integer	factor	2|1	Female|Male					
10	EXERANY2	exerany2	integer	integer	1|2	1|2					
11	EXERANY2	exerany2f	integer	factor	1|2	Yes|No					
12	PREDIAB1	prediab1	integer	integer							
13	PREDIAB1	prediab1f	integer	factor	1|2|3	Yes|Preg|No					
14	X_STSTR	x_ststr	integer	integer							
15	X_LCPWTV1	x_lcpwtv1	numeric	numeric							
16	HTCM4	htcm4	integer	integer							
17	WTG3	wtg3	integer	integer							
18	FAMILYDIAB	familydiab	character	character							
19	AGE	agecut	integer	factor				cut(x, c(-1, 40, 49, 59, 110), right = FALSE, labels = c("Below 4(			
20	AGE	agept	integer	integer				as.integer(as.character(cut(x, c(-1, 40, 49, 59, 110), right = FAL:			
21	SEX	sexpt	integer	logical	1|2	TRUE|FALSE					
22	DIABETE3	gestdbpt	integer	logical	1|2|3|4	FALSE|TRUE|FALSE|FALSE					
23	FAMILYDIAB	famdiabpt	character	logical				!grepl("5", x, fixed = TRUE)			
24	BPHIGH4	bppt	integer	logical	1|2|3|4	TRUE|FALSE|FALSE|FALSE					
25	EXERANY2	exercpt	integer	logical	1|2	TRUE|FALSE					
26											
27											
28											
29											
30											

Workflow Step 1. Create a key template

The function `keyTemplate` can scan an existing data frame and create a template variable key

name_old	name_new	class_old	class_new	value_old	value_new
V23419	V23419	integer	integer	1|2|3	1|2|3
V32422	V32422	integer	integer	1|2|3|4|5	1|2|3|4|5
V54532	V54532	numeric	numeric	.	.

- Ask the client to choose `name_new`, `values_new` (*hope* they can fill out `class_new`, but less likely to succeed)

Client fills in key

- Discrete variables can have an enumerated list of values
- Numeric variables are usually not enumerated, so we leave "." in values_old and value_new

name_old	name_new	class_old	class_new	value_old	value_new
V23419	sex	integer	factor	1|2|3	male|female|neither
V32422	education	integer	ordered	1|2|3|4|5	elem|hs|somecoll|ba|post
V54532	income	numeric	numeric	.	.

- `kutils` includes a function `keyImport`. Does data integrity checks

And the most important step is. . .

- The analyst "applies" the key to the data with the `kutils` function `keyApply`
- The variables are renamed, the values are re-aligned
- Profuse variable-by-variable output summarizing changes

```
> brf2 <- keyApply(brfss, key)
[1] "Variable seqno has 20 unique values. Too
   large for a table."
       BPHIGH4 (old var)
bphigh4    1    2    3    4
      1 1090    0    0    0
      2    0  307    0    0
      3    0    0  558    0
      4    0    0    0   45
```

And the most important step is. . . ...

```
                BPHIGH4 (old var)
bphigh4f            1    2    3    4
  Yes            1090    0    0    0
  Preg              0  307    0    0
  No                0    0  558    0
  Borderline        0    0    0   45
           DIABETE3 (old var)
diabete3f     1     2     3     4
        1   239     0     0     0
        2     0    16     0     0
        3     0     0  1709     0
        4     0     0     0    36
               DIABETE3 (old var)
diabete3            1    2    3    4
  Yes             239    0    0    0
```

And the most important step is. . . ...

```
  Preg            0    16     0     0
  No              0     0  1709     0
  Borderline      0     0     0    36
[1] "Variable age has 20 unique values. Too
   large for a table."
   SEX (old var)
sex    1    2
  1  974    0
  2    0 1026
        SEX (old var)
sexf          1    2
  Female      0 1026
  Male      974    0
        EXERANY2 (old var)
exerany2      1    2
```

And the most important step is. . . ...

```
        1 1154     0
        2    0   846
          EXERANY2 (old var)
exerany2f    1    2
      Yes 1154    0
      No     0  846
        PREDIAB1 (old var)
prediab1     1    2    3 <NA>
    1      190    0    0    0
    2        0  167    0    0
    3        0    0 1404    0
    <NA>     0    0    0  239
          PREDIAB1 (old var)
prediab1f    1    2    3 <NA>
     Yes   190    0    0    0
```

And the most important step is. . . ...

```
      Preg     0   167     0     0
      No       0     0  1404     0
      <NA>     0     0     0   239
```

- Some details remaining, checking for errors, inspect old and new data comparisons

Outline

Additional key Column: missing

- enumerated variables can insert "." to represent missing

value_old	value_new
1|2|3|99	1|2|3|.

- for numeric variables, we introduce column missing, which allows a limited range of expressions

missing	
c(98,99)	
< 0	
> 99	

Additional Key Column: recode

- Valid R expressions, where x is a symbol for the variable under consideration
- examples

```
kutils::reverse(as.ordered(x))
```

```
cut(x, c(-1, 40, 49, 59, 110), right = FALSE,
    labels = c("Below 40", "40s", "50s",
    "Above 59"), ordered_result = TRUE)
```

The "long" key

- In the Wide Key, editing values is difficult in spreadsheet if values are verbose or many
  - Example: "low|moderate|medium|luke|tepid|warm|hot|boiling"
- The Long Key style has one row per value

name_old	name_new	class_old	class_new	values_old	values_new
V23419	sex	integer	factor	1	male
V23419	sex	integer	factor	2	female
V23419	sex	integer	factor	3	neither
V32422	education	integer	ordered	1	elem
V32422	education	integer	ordered	2	hs
V32422	education	integer	ordered	3	somecoll
V32422	education	integer	ordered	4	ba
V32422	education	integer	ordered	5	post
V54532	income	numeric	numeric	.	.

The "long" key ...

- The `keyImport` function checks on the type and imports the file
- `kutils` provides functions `wide2long` and `long2wide` for conversion

long key snapshot

The American National Election Study (ANES)

long key snapshot ...

V041001	V041001f	integer	factor	0	0. Pre-election interview only		
V041001	V041001f	integer	factor	1	1. Both Pre-election and Post-election interviews		
V041101	V041101f	integer	factor	1	1. One person in HH		
V041101	V041101f	integer	factor	2	2. Two persons in HH		
V041101	V041101f	integer	factor	3	3. Three persons in HH		
V041101	V041101f	integer	factor	4	4. Four persons in HH		
V041101	V041101f	integer	factor	5	5. Five persons in HH		
V041101	V041101f	integer	factor	6	6. Six persons in HH		
V041101	V041101f	integer	factor	7	7. Seven persons in HH		
V041101	V041101f	integer	factor	8	8. Eight persons in HH		
V041101	V041101f	integer	factor	9	9. Nine persons in HH		
V041102	V041102f	integer	factor	1	1. One adult in HH		
V041102	V041102f	integer	factor	2	2. Two adults in HH		
V041102	V041102f	integer	factor	3	3. Three adults in HH		
V041102	V041102f	integer	factor	4	4. Four adults in HH		
V041102	V041102f	integer	factor	5	5. Five adults in HH		
V041102	V041102f	integer	factor	6	6. Six adults in HH		
V041102A	V041102Af	integer	factor	1	1. One eligible adult in HH		
V041102A	V041102Af	integer	factor	2	2. Two eligible adults in HH		
V041102A	V041102Af	integer	factor	3	3. Three eligible adults in HH		
V041102A	V041102Af	integer	factor	4	4. Four eligible adults in HH		
V041102A	V041102Af	integer	factor	5	5. Five eligible adults in HH		
V041102A	V041102Af	integer	factor	6	6. Six eligible adults in HH		
V041102B	V041102Bf	integer	factor	0	0. No ineligible adult in HH		
V041102B	V041102Bf	integer	factor	1	1. One ineligible adult in HH		
V041102B	V041102Bf	integer	factor	2	2. Two ineligible adults in HH		
V041102B	V041102Bf	integer	factor	3	3. Three ineligible adults in HH		
V041102B	V041102Bf	integer	factor	4	4. Four ineligible adults in HH		
V041102C	V041102Cf	integer	factor	0	0. No female adult in HH		
V041102C	V041102Cf	integer	factor	1	1. One female adult in HH		
V041102C	V041102Cf	integer	factor	2	2. Two female adults in HH		
V041102C	V041102Cf	integer	factor	3	3. Three female adults in HH		

Key customized to input format

SPSS and Stata use “values” and “value label” paradigm

Recently, we introduce key format to summarize their value labels.

Key depiction of an SPSS data set

Children of Immigrants (CILS) in wide key format

	A	B	C	D	E	F
1	name_	name_	class_o	class_n	value_old	value_new
2	CASEID	CASEID	numeric	numeric	.	.
3	V1	V1	numeric	numeric	.	.
4	V2	V2f	numeric	factor	1|3|4	Miami|Ft. Lauderdale|San Diego
5	V4	V4	numeric	numeric	.	.
6	V5	V5f	numeric	factor	7|8|9|10	Seventh grade|Eighth grade|Ninth grade|Tenth grade
7	V7	V7f	numeric	factor	1|2|3|.	Yes|No|Dead/unknown|.
8	V8	V8f	numeric	factor	1|2|3|4|5|6|7|	Same city|Another city in Miami smsa|Another city in Florida|Another US state|Abroad|San Diego/neighboring city|
9	V9	V9	numeric	numeric	.	.
10	V10	V10	numeric	numeric	.	.
11	V11	V11f	numeric	factor	1|2|.	Yes|No|.
12	V13	V13f	numeric	factor	1|2|.	Yes|No|.
13	V14	V14f	numeric	factor	1|2|3|4|5|6|7|	Same city|Another city in Miami smsa|Another city in Florida|Another US state|Abroad|San Diego/neighboring city|
14	V15	V15	numeric	numeric	.	.
15	V16	V16	numeric	numeric	.	.
16	V17	V17f	numeric	factor	1|2|.	Yes|No|.
17	V18	V18f	numeric	factor	1|2	Male|Female
18	V19	V19	numeric	numeric	.	.
19	V20	V20	numeric	numeric	.	.
20	V21	V21	numeric	numeric	.	.
21	V21A	V21A	numeric	numeric	.	.
22	V22	V22f	numeric	factor	1|2|3|4|.	All my life|Ten years or more|Five to nine years|Less than five years|.
23	V23	V23f	numeric	factor	1|2|.	Yes|No|.
24	V24	V24f	numeric	factor	1|2|3|4|.	Not at all|Now well|Well|Very well|.
25	V25	V25f	numeric	factor	1|2|3|4|.	Not at all|Now well|Well|Very well|.

Key depiction of an SPSS data set ...

	A	B	C	D	E	F	G	H
24	V24	V24f	numeric	factor	1|2|3|4|.	Not at all|Now well|Well|Very well|.		
25	V25	V25f	numeric	factor	1|2|3|4|.	Not at all|Now well|Well|Very well|.		
26	V26	V26f	numeric	factor	1|2|3|4|.	Not at all|Now well|Well|Very well|.		
27	V27	V27f	numeric	factor	1|2|3|4|.	Not at all|Now well|Well|Very well|.		
28	V28	V28f	numeric	factor	1|2|3|4|5|6|7|	Father and mother|Father and step-mother/other female adult|Mother and step-father/other male adult|Father alone|Mother alone|Alter		
29	V29A	V29Af	numeric	factor	0|1|2|3|4|5|6|	None|One|Two|Three|Four|Five|Six|Seven|Eight or more|.		
30	V29B	V29Bf	numeric	factor	0|1|2|3|4|.	None|One|Two|Three|Four|.		
31	V29C	V29Cf	numeric	factor	0|1|2|3|4|5|6|	None|One|Two|Three|Four|Five|Six|Seven|Eight or more|.		
32	V29D	V29Df	numeric	factor	0|1|2|3|4|5|6|	None|One|Two|Three|Four|Five|Six|Seven|Eight or more|.		
33	V29E	V29Ef	numeric	factor	0|1|2|3|4|5|6|	None|One|Two|Three|Four|Five|Six|Seven|Eight or more|.		
34	V30	V30	numeric	numeric	.	.		
35	V31	V31	numeric	numeric	.	.		
36	V32	V32	numeric	numeric	.	.		
37	V33	V33f	numeric	factor	1|2|.	Yes|No|.		
38	V34A	V34Af	numeric	factor	1|2|3|.	Working at different occupation|Unemployed, looking for work|Unemployed, not looking for work|.		
39	V34B	V34B	numeric	numeric	.	.		
40	V35	V35	numeric	numeric	.	.		
41	V36	V36f	numeric	factor	1|2|3|4|5|6|.	Elementary school or less|Middle school or less|Some high school|High school graduate|Some college/university|College graduate or more		
42	V37	V37	numeric	numeric	.	.		
43	V38	V38f	numeric	factor	1|2|.	Yes|No|.		
44	V39A	V39Af	numeric	factor	1|2|3|.	Working at different occupation|Unemployed, looking for work|Unemployed, not looking for work|.		
45	V39B	V39B	numeric	numeric	.	.		
46	V40	V40	numeric	numeric	.	.		
47	V41	V41f	numeric	factor	1|2|3|4|5|6|.	Elementary school or less|Middle school or less|Some high school|High school graduate|Some college/university|College graduate or more		
48	V42	V42f	numeric	factor	1|2|3|4|5|.	Own|Rent|Lives with relatives|Lives with friends/non-relatives|Other|.		

[. . . snip many rows]

Partial Variable Keys

- `keyApply` argument `drop = c("vars", "vals")`
- If drop = "vars", then variables that are not mentioned in the key are removed from the new data frame
  - Use Case: Client only wants to use 20 variables from data set that includes 1000s of columns
  - Otherwise, all variables remain, but only 20 in key are recoded.
- If drop = "vals", then key omission of scores from "value_old" will cause those observations to be changed to missing
  - Otherwise, values omitted from key pass through to output data unaltered

KU

Outline

KU

Class Conversion Diagnostics

- 6 variable classes, old and new, are accounted for
  - Conversion from any value into a categorical (factor or ordered variable) is easy
    - Integer $\rightarrow$ Factor is easy
    - Character$\rightarrow$Factor is easy
  - Conversion from categorical to numeric or integer requires type-checking
- Unit tests implemented 2017 to validate code updates

KU

Working on Codebook Generator

- We want output that integrates the key information with observed data frequencies
- Existing code can generate nice reports for discrete variables

References

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Session

```
sessionInfo()
```

```
R version 3.4.4 (2018-03-15)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 17.10

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
     LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8
     LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
     LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8
    LC_IDENTIFICATION=C
```

Session ...

```
attached base packages:
[1] stats     graphics  grDevices utils     datasets  base

other attached packages:
[1] stationery_0.79

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.15        quadprog_1.5-5      rprojroot_1.3-2
     digest_0.6.15       plyr_1.8.4
 [6] backports_1.1.2     xtable_1.8-2        magrittr_1.5
     stats4_3.4.4        evaluate_0.10.1
[11] stringi_1.1.6       pbivnorm_0.6.0      openxlsx_4.0.17
    rmarkdown_1.8       tools_3.4.4
[16] stringr_1.2.0       foreign_0.8-69      kutils_1.39
    compiler_3.4.4      mnormt_1.5-5
[21] htmltools_0.3.6     knitr_1.19          lavaan_0.5-23.1097
    methods_3.4.4
```