**Data transformations:**

**Example R commands**

**Reverse scorE A variable**

EXAMPLE: Reverse score ITEM4 into a new variable NITEM4

**RECODE ITEM4 (1 = 5) (2 = 4) (3 = 3) (4 = 2) (5 = 1) INTO NITEM4 .**

**library(car)**

***dataframe*$*newvar* <-recode(*dataframe*$*var*,"*old-value* = *new-value*; *old-value* = *new-value*; ...")**

library(car)

df1$NITEM4 <- recode(df1$ITEM4,"1 = 5; 2 = 4; 3 = 3; 4 = 2; 5 = 1")

**ReducE the number of groups in A categorical variable**

EXAMPLE: Reduce the number of levels of GROUP from 5 levels to 2

**RECODE GROUP (1 = 1) (2 = 2) (3 = 2) (4 = 2) (5 = 2) INTO NGROUP .**

**library(car)**

***dataframe*$*newvar* <-recode(*dataframe*$*var*,"*old-value* = *new-value*; *old-value* = *new-value*; ...")**

library(car)

df1$NGROUP <- recode(df1$GROUP,"1 = 1; 2 = 2; 3 = 2; 4 = 2; 5 = 2")

**Create a categorical variable from a continuous variable**

EXAMPLE: Group the values of INCOME into a new variable NINCOME (two groups)

**RECODE INCOME (0 thru 49999 = 1) (50000 thru Hi = 2) INTO NINCOME .**

**Create a numerical categorical variable from a continuous variable**

**library(car)**

***dataframe*$*newvar* <- recode(*dataframe*$*var*,"*min*:*max* = *new-value*; *min*:*max* = *new-value*; ...")**

library(car)

df1$NGROUP <- recode(df1$INCOME,"lo:49999 = 1; 50000:hi = 2")

**Create a string categorical variable from a continuous variable**

**library(car)**

***dataframe*$*groupingvar* <- cut(*dataframe*$*contvar*,breaks=c(*min*, *cutpoint*, *max*), labels=c("*Label for Group1*","*Label for Group2*"))**

library(car)

df1$NINCOME <- cut(df1$INCOME,breaks=c(0, 49999, Inf),labels=c("< $50,000"," $50,000+"))

**Create a variable from other variables**

EXAMPLE: Create the variable MITEM that is the mean of ITEM1 to ITEM4

**COMPUTE MITEM = MEAN (ITEM1, ITEM2, ITEM3, ITEM4) .**

***setofvariables* <- subset(*dataframe*,select=c(*var*, *var*, *var*,…) )**

***dataframe*$*newvar* <- rowMeans(*setofvariables*, na.rm=TRUE )**

items <- subset(df1,select=c(ITEM1,ITEM2,ITEM3,ITEM4) )

df1$MITEM <- rowMeans(items, na.rm=TRUE )

**Create a variable from other variables (minimum number of valid values)**

EXAMPLE: Create the variable MITEM that is the mean of ITEM1 to ITEM4 (at least three valid values)

**COMPUTE MITEM2 = MEAN.3 (ITEM1, ITEM2, ITEM3, ITEM4) .**

# count number of valid responses

***dataframe*$*numvalid* <-apply(*dataframe*[*startcol*:*endcol*],1,function(x) sum(!is.na(x)))**

# calculate score for those w/min # valid values, otherwise assign NA

***setofvariables* <-subset(*dataframe*,select=c(*var, var, var,…*) )**

***dataframe*$*newvar* <- ifelse(*dataframe*$*numvalid* >= *#, dataframe*$*newvar* <- rowMeans(*setofvariables*, na.rm=TRUE),NA)**

# count number of valid responses

df1$item.nvalid <- apply(df1[2:5], 1, function(x) sum(!is.na(x)))

# calculate MITEM2 for those with 3+ valid values, otherwise assign NA

items <- subset(df1,select=c(ITEM1,ITEM2,ITEM3,ITEM4) )

df1$MITEM2 <- ifelse(df1$item.nvalid >= 3,df1$MITEM2 <- rowMeans(items, na.rm=TRUE),NA)

**Create a variable from occurrences of values of other variables**

EXAMPLE: Count the number of vices (i.e., SMOKE and DRINK = 1 (yes))

**COUNT NVICES = SMOKE DRINK (1) .**

***dataframe*$*newvar* <- apply(*dataframe* [*startcol*:*endcol*], 1, function(x) length(which(x==*value*)))**

df1$NVICES <- apply(df1[8:9], 1, function(x) length(which(x==1)))

**Perform data transformations when conditions are met**

EXAMPLE: Create groups based on combinations of SMOKE and DRINK (Yes, No)

**IF (SMOKE eq 1 and DRINK eq 1) SDCOMB = 1 .**

**IF (SMOKE eq 1 and DRINK eq 2) SDCOMB = 2 .**

**IF (SMOKE eq 2 and DRINK eq 1) SDCOMB = 2 .**

**IF (SMOKE eq 2 and DRINK eq 2) SDCOMB = 3 .**

***dataframe*$*newvar* <- ifelse((*expression1*), *outcome\_if\_expression1\_true*, ifelse(*expression2*), *outcome\_if\_expression2\_true*, ... , *outcome\_if\_all\_expressions\_false*)**

df1$SDCOMB <- ifelse((df1$SMOKE == 1) & (df1$DRINK == 1), df1$SDCOMB <- 1, ifelse((df1$SMOKE == 1) & (df1$DRINK == 2), df1$SDCOMB <- 2, ifelse((df1$SMOKE == 2) & (df1$DRINK == 1), df1$SDCOMB <- 2, ifelse((df1$SMOKE == 2) & (df1$DRINK == 2), df1$SDCOMB <- 3, NA))))

**Perform data transformations under specified conditions**

EXAMPLE: Count occurrences across set of variables for those with no missing data

**COUNT #MVICES = SMOKE DRINK (MISSING) .**

**DO IF (#MVICES EQ 0) .**

**COUNT NVICES2 = SMOKE DRINK (1) .**

# count the number of missing responses

***dataframe*$*nummissing* <- apply(*dataframe* [*startcol*:*endcol*], 1, function(x) sum(is.na(x)))**

# calculate new var for those with no missing, otherwise assign NA

***dataframe* $*newvar* <- ifelse((*expression*), *outcomeifexpressiontrue*, *outcomeifexpressionfalse*)**

# count the number of missing responses

df1$MVICES <- apply(df1[8:9], 1, function(x) sum(is.na(x)))

# calculate NVICES2 for those with no missing, otherwise assign NA

df1$NVICES2 <- ifelse((df1$MVICES == 0), df1$NVICES2 <- apply(df1[8:9], 1, function(x) length(which(x==1))), NA)

**Perform data transformations under different specified conditions**

EXAMPLE: Create the variable MITEM3 differently for different incomes

**DO IF (NINCOME EQ 1) .**

**COMPUTE MITEM3 = MITEM x 10 .**

**ELSE IF (NINCOME EQ 2) .**

**COMPUTE MITEM3 = MITEM x 100 .**

**END IF .**

***dataframe*$*newvar* <- ifelse((*expression1*), *outcomeifexpression1true*, ifelse(*expression2*), *outcomeifexpression2true*, ... , *outcomeifexpressionsfalse*)**

df1$MITEM3 <- ifelse((df1$NINCOME == 1), df1$MITEM3 <- df1$MITEM \* 10, ifelse((df1$NINCOME == 2), df1$MITEM3 <- df1$MITEM \* 100, NA))

**Use numeric functions in data transformations**

EXAMPLE: Calculate absolute, rounded, truncated, and square root value of Variable X

**\* Absolute value .**

**COMPUTE ABSX = ABS(X) .**

**\* Round .**

**COMPUTE RNDX = RND(X) .**

**\* Truncate .**

**COMPUTE TRUNCX = TRUNC(X) .**

**\* Square root .**

**COMPUTE SQRTX = SQRT(X) .**

**EXECUTE .**

***dataframe*$*newvar* <- *numericfunction*(*dataframe*$*oldvar*)**

# absolute value

df1$ABSX <- abs(df1$x)

# round to zero decimal places

df1$RNDX <- rnd(df1$x, digits = 0)

# truncate

df1$TRUNCX <- trunc(df1$x)

# square root

df1$SQRTX <- sqrt(df1$x