Translation Syntax (SPSS, Stata, SAS and R)

The Basics

The following conventions are used in this document:
• Bold font indicates code or other text that should be typed literally.
• Un-bolded font shows code or text that should be replaced with user-supplied values (i.e., your own variables and other environment details).

Calling in a data set

SPSS	**GET FILE='**P:\QAC\qac201\Studies\study name\filename.sav**'.**
STATA	**use "**P:\QAC\qac201\Studies\study name\filename**"**
SAS	**LIBNAME** in **"**P:\QAC\QAC201\study name**;** **DATA new; set** in**.**filename**;**
R	**load ("**filename-including-path.**Rdata")** myData_orig **<-** name-of-object-that-loaded-in-your-workspace If calling in from a text file: myData_orig **<- read.table(file = "**filename-including-path.txt**", sep = "\t",** **header = TRUE**)

Selecting variables you want to examine

SPSS	**/KEEP** VAR1 VAR2 VAR3 VAR4 VAR5 VAR6 VAR7 VAR8. (Must follow the **SAVE OUTFILE='**dataname' command)
STATA	**keep** VAR1 VAR2 VAR3 VAR4 VAR5 VAR6 VAR7 VAR8
SAS	**KEEP** VAR1 VAR2 VAR3 VAR4 VAR5 VAR6 VAR7 VAR8**;**
R	var.keep **<- c("**VAR1**", "**VAR2**", "**VAR3**", "**VAR4**", "**VAR5**", "**VAR6**", "**VAR7**", "**VAR8**")** myData **<-** myData_orig **[,**var.keep**]**

Outputting your abbreviated data set

SPSS	**SAVE OUTFILE=** 'Drive:\folder\folder\title_of_new_data_set'**.**
STATA	**save** filename
SAS	**Data libname**.title_of_new_data_set; **set** dataname; **by** unique_id**;**
R	To open in excel: **write.table(**myData**, file = "**filename.txt**", sep = "\t", row.names = FALSE)** To use in R: **save (file = "**filename.txt**",** myData**)**

Sorting the data

SPSS	**SORT CASES BY** unique_id.
STATA	**sort** unique_id
SAS	**proc sort; by** unique_id**;**
R	myData **<-** myData**[order(**myData**$**unique_id**, decreasing = FALSE),]**

Displaying frequency tables

SPSS	**FREQUENCIES VARIABLES**=VAR1 VAR2 VAR3 **/ORDER=ANALYSIS.**
STATA	**tab1** VAR1 VAR2 VAR3
SAS	**PROC FREQ; tables** VAR1 VAR2 VAR3**;**
R	**library(descr)** **freq(as.ordered(**myData**$**VAR1**))** **freq(as.ordered(**myData**$**VAR2**))** **freq(as.ordered(**myData**$**VAR3**))**

Data Management

Basic operations

SPSS	EQ or =	>= or GE	<= or LE	> or GT	< or LT	NE
STATA	==	>=	<=	>	<	!=
SAS	EQ or =	>= or GE	<= or LE	> or GT	< or LT	NE
R	==	>=	<=	>	<	!=

Examples

1. Need to identify missing data

Often, you must define the response categories that represent missing data. For example, if the number 9 is used to represent a missing value, you must either designate in your program that this value represents missingness or else you must recode the variable into a missing data character that your statistical software recognizes. If you do not, the 9 will be treated as a real/meaningful value and will be included in each of your analyses.

SPSS	**RECODE** VAR1 **(**9=**SYSMIS).**
STATA	**replace** VAR1**=. if** VAR1**==9**
SAS	**if** VAR1=9 **then** VAR1=**.;**
R	myData**$**VAR1**[**myData**$**VAR1 **==** 9 **] <- NA**

2. Need to recode responses to "no" based on skip patterns

There are a number of skip outs in some data sets. For example, if we ask someone whether or not they have ever used marijuana, and they say "no", it would not make sense to ask them more detailed questions about their marijuana use (e.g. quantity, frequency, onset, impairment, etc.). When analyzing more detailed questions regarding marijuana (e.g. have you ever smoked marijuana daily for a month or more?), those individuals that never used the substance may show up as missing data. Since they have never used marijuana, we can assume that their answer is "no", they have never smoked marijuana daily. This would need to be explicitly recoded. Note that we commonly code a no as 0 and a yes as 1.

SPSS	**RECODE** VAR1 **(SYSMIS**=7**).**
STATA	**replace** VAR1**=7 if** VAR1**==.**
SAS	**if** VAR1**=. then** VAR1=7**;**
R	myData**$**VAR1**[is.na(**myData**$**VAR1**)] <-** 7

3. Recoding string variables into numeric

It is important when preparing to run statistical analyses in most software packages, that all variables have response categories that are numeric rather than "string" or "character" (i.e. response categories are actual strings of characters and/or symbols). All variables with string responses must therefore be recoded into numeric values. These numeric values are known as dummy codes in that they carry no direct numeric meaning.

SPSS	**RECODE** TREE **('**Maple**'**=1**) ('**Oak**'**=2**) INTO** TREE_N**.**
STATA	**generate** TREE_N**=.** **replace** TREE_N=1 **if** TREE==**"**Maple**"** **replace** TREE_N=2 if TREE==**"**Oak**"** OR by using the encode command **encode** TREE**, gen(**TREE_N**)**
SAS	**IF** TREE**='**Maple**' then** TREE_N**=**1**;** **else if** TREE**= '**Oak**' then** TREE_N**=**2**;**
R	Not necessary in R.

4. Need to collapse response categories

If a variable has many response categories, it can be difficult to interpret the statistical analyses in which it is used. Alternatively, there may be too few subjects or observations identified by one or more response categories to allow for a successful analysis. In these cases, you would need to collapse across categories. For example, if you have the following categories for geographic region, you may want to collapse some of these categories:

Region: New England=1, Middle Atlantic=2, East North Central=3, West North Central=4, South Atlantic=5, East South Central=6, West South Central=7, Mountain=8, Pacific=9.

New_Region: East=1, West=2.

SPSS	**COMPUTE** new_region**=**2**.** **IF (**region=1| region=2|region=3| region=5|region=6**)** new_region=1**.**
STATA	**generate** new_region =2 **replace** new_region=1 **if** region==1| region==2|region==3| region==5|region==6 OR by using the recode command **recode region** (1/3 5 6=2) **gen(**new_region**)**
SAS	**if r**egion=1 or region=2 or region=3 or region=5 or region=6 **then** new_region=1**;** **else if** region=4 or region=7 or region=8 or region=9 **then** new_region=2**;**
R	myData**$**new_region**[**myData**$**region == 1 | myData$region == 2 \| myData**$**region == 3 | myData**$**region == 5 \| myData**$**region == 6**] <-** 1 myData**$**new_region**[**myData**$**region == 4 | myData**$**region == 7 \| myData**$**region == 8 | myData**$**region == 9**] <-** 2

5. Need to aggregate variables

In many cases, you will want to combine multiple variables into one. For example, while NESARC assesses several individual anxiety disorders, I may be interested in anxiety more generally. In this case I would create a general anxiety variable in which those individuals who received a diagnosis of social phobia, generalized anxiety disorder, specific phobia, panic disorder, agoraphobia, or obsessive compulsive disorder would be coded "yes" and those who were free from all of these diagnoses would be coded "no".

Syntax shown on next page.

5. Need to aggregate variables (continued)

SPSS	**IF (**socphob=1|gad=1|specphob=1| panic=1|agora=1|ocd=1**) anxiety=1.** **RECODE** anxiety **(SYSMIS**=0**).**
STATA	**gen** anxiety=1 if socphob==1|gad==1|specphob==1| panic==1|agora==1|ocd==1 **replace** anxiety=0 if anxiety==.
SAS	**if** socphob=1 or gad=1 or specphob=1 or panic=1 or agora=1 or ocd=1 **then** anxiety=1**; else** anxiety=0**;**
R	myData**$**anxiety **<- rep(**0, **nrow(**myData**))** myData**$**anxiety**[**myData**$**socphob == 1 \| myData**$**gad==1 | myData**$**panic == 1 \| myData**$**agora==1 | myData**$**ocd == 1**] <- 1** myData**$**anxiety**[is.na(**myData**$**socphob**) & is.na(**myData**$**gad**) &** **is.na(**myData**$**panic**) & is.na(**myData**$**agora**) &** **is.na(**myData**$**ocd**)] <- NA**

6. Need to create continuous variables

If you are working with a number of items that represent a single construct, it may be useful to create a composite variable/score. For example, I want to use a list of nicotine dependence symptoms meant to address the presence or absence of nicotine dependence (e.g. tolerance, withdrawal, craving, etc.). Rather than using a dichotomous variable (i.e. nicotine dependence present/absent), I want to examine the construct as a dimensional scale (i.e. number of nicotine dependence symptoms). In this case, I would want to recode each symptom variable so that yes=1 and no=0 and then sum the items so that they represent one composite score.

SPSS	**COMPUTE** nd_sum**=sum(**nd_symptom1 nd_symptom2 nd_symptom3 nd_symptom4**).**
STATA	**egen** nd_sum=**rsum(**nd_symptom1 nd_symptom2 nd_symptom3 nd_symptom4**)**
SAS	nd_sum**=sum (of** nd_symptom1 nd_symptom2 nd_symptom3 nd_symptom4**);**
R	myData**$**nd_sum **<-** myData**$**nd_symptom1 **+** myData**$**nd_symptom2 **+** myData**$**nd_symptom3 **+** myData**$**nd_symptom4

7. Labeling variables

Given the often cryptic names that variables are given, it can sometimes be useful to label them.

SPSS	**VARIABLE LABELS** VAR1 **'**label**'.**
STATA	**label variable** VAR1 **"**label**"**
SAS	**LABEL** VAR1**='**label**';**
R	For frequency tables: **library (**Hmisc**)** **label(**myData**$**VAR1**) <- "**label**"**

8. Renaming variables

Given the often cryptic names that variables are given, it can sometimes be useful to give a variable a new name (something that is easier for you to remember or recognize).

SPSS	**COMPUTE** newvarname**=**VAR1**.**
STATA	**rename** VAR1 newvarname
SAS	**RENAME** VAR1=newvarname**;**
R	**names(**myData**)[names(**myData**)== "**VAR1**"] <- "**newvarname**"**

9. Labeling variable responses/values

Given that nominal and ordinal variables have, or are given numeric response values (i.e. dummy codes), it can be useful to label those values so that the labels are displayed in your output.

Syntax shown on next page.

9. Labeling variable responses/values (continued)

SPSS	**VALUE LABELS** VAR1 0 'value0label' 1 'value1label' 2 'value2label' 3 'value3label'**.**
STATA	**label define** VAR1 0 "value0label" 1 "value1label" 2 "value2label" 3 "value3label" **label values** VAR1 newvarname
SAS	**proc format;** VAR1 0**="**value0label**"** 1**="**value1label**"** 2**="**value2label**"** 3**="**value3label**";**
R	Because the function doesn't name the existing levels, make sure you have them all in the right order. **levels(**myData**$**VAR1**)** **levels(**myData**$**VAR1**) <- c("**value0label**", "**value1label**", "**value2label**", "**value3label**")**

10. Need to further subset the sample

When using large data sets, it is often necessary to subset the data so that you are including only those observations that can assist in answering your particular research question. In these cases, you may want to select your own sample from within the survey's sampling frame. For example, if you are interested in identifying demographic predictors of depression among Type II diabetes patients, you would plan to subset the data to subjects endorsing Type II Diabetes.

SPSS	**/SELECT=**diabetes2 EQ 1 (must be added as a command option)
STATA	**if** diabetes2==1 (put this after the command)
SAS	**if** diabetes2=1**;** (put in the data step before sorting the data)
R	title_of_subsetted_data **<-** myData**[**myData**$**diabetes2 **==** 1**,]**

11. Need to create groups that will be compared to one another

Often, you will need to create groups or sub-samples from the data set for the purpose of making comparisons. It is important to be certain that the groups that you would like to compare are of adequate size and number. For example, if you were interested in comparing complications of depression in parents who had lost a child through miscarriage vs. parents who had lost a child in the first year of life, it would be important to have large enough groups of each. It would not be appropriate to attempt to compare 5000 observations in the miscarriage group to only 9 observations in the first year group.

Refer to other data management syntax examples.

Graphing and Data Visualization

1. Univariate

Code for Univariate Output (Categorical):

SPSS	**FREQUENCIES VARIABLES=**CategVar1 CategVar2 CategVar3 **/ORDER=ANALYSIS.**
STATA	**tab1** CategVar1 CategVar2 CategVar3
SAS	**PROC FREQ; tables** CategVar1 CategVar2 CategVar3**;**
R	**library(descr)** **freq(as.ordered(**myData**$**CategVar1**))** **freq(as.ordered(**myData**$**CategVar2**))** **freq(as.ordered(**myData**$**CategVar3**))**

Code for Univariate Graph (Categorical):

SPSS	Use graphical user interface (GUI)
STATA	**histogram** BinaryVar
SAS	**Proc GCHART; VBAR** CategVar **/ Discrete type=PCT** **Width=**30**;**
R	**library(descr)** **freq(as.ordered(**myData**$**CategVar**))** OR **plot(**myData**$**CategVar**)**

Code for Univariate Output (Quantitative):

SPSS	**DESCRIPTIVES VARIABLES=**QuantVar1 QuantVar2 QuantVar3 **/STATISTICS=MEAN STDDEV.**
STATA	**summarize** QuantVar1 QuantVar2 QuantVar3
SAS	**proc means; var** QuantVar1 QuantVar2 QuantVar3**;**
R	Repeat for each variable. **summary(**myData**$**QuantVar1**)** **mean(**myData**$**QuantVar1**, na.rm = TRUE)** **sd(**myData**$**QuantVar1**, na.rm = TRUE)**

Code for Univariate Graph (Quantitative):

SPSS	Use graphical user interface (GUI)
STATA	**histogram** QuantVar
SAS	**Proc GCHART; VBAR** QuantVar**;**
R	**hist(**myData**$**QuantVar**)**

2. Bivariate

Code for Bivariate Output (Categorical Explanatory Variable and Categorical Response Variable):

SPSS	**CROSSTABS** **/TABLES=**CategResponseVar **by** CategExplanatoryVar **/CELLS=COUNT ROW COLUMN TOTAL.**
STATA	**tab** CategResponseVar CategExplanatoryVar**, row column cell**
SAS	**Proc freq; tables** CategResponseVar*****CategExplanatoryVar**;**
R	tab1 **<- table (**myData**$**CategResponseVar**,** myData**$**CategExplanatoryVar**)** tab1 # to output the table tab1_colProp **<- prop.table(**tab1**, 2)** # column proportions tab1_rowProp **<- prop.table(**tab1**, 1)** tab1_cellProp **<- prop.table(**tab1**)** tab1_colProp tab1_rowProp tab1_cellProp

Code for Bivariate Bar Graph (Categorical Explanatory Variable and Categorical Response Variable):

SPSS	Use graphical user interface (GUI)
STATA	**graph bar (mean)** CategResponseVar**, over(**CategExplanatoryVar**)**
SAS	**Proc GCHART; vbar** CategExplanatoryVar **/discrete type=mean** **sumvar=**CategResponseVar **descending;**
R	tab1 **<- table (**myData**$**CategResponseVar**,** myData**$**CategExplanatoryVar**)** tab1_colProp **<- prop.table(**tab1**, 2)** **barplot (**tab1p**[**index-of-row(s)-to-plot**,])**

Code for Bivariate Output (Categorical Explanatory Variable and Quantitative Response Variable):

SPSS	**MEANS TABLES=** CategExplanatoryVar **by** QuantResponseVar **/CELLS MEAN COUNT STDDEV.**
STATA	**bys** CategExplanatoryVar**: su** QuantResponseVar
SAS	**proc sort; by** CategExplanatoryVar**;** **proc means; var** QuantResponseVar**; by** CategExplanatoryVar**;**
R	**by(**myData**$**QuantResponseVar**,** myData**$**CategExplanatoryVar**, mean, na.rm =** **TRUE)** **by(**myData**$**QuantResponseVar**,** myData**$**CategExplanatoryVar**, sd, na.rm = TRUE)** **by(**myData**$**QuantResponseVar**,** myData**$**CategExplanatoryVar**, length)**

Code for Bivariate Bar Graph (Categorical Explanatory Variable and Quantitative Response Variable):

SPSS	Use graphical user interface (GUI)
STATA	**graph box** QuantResponseVar**, over(**CategExplanatoryVar**)**
SAS	**Proc GCHART; vbar** CategExplanatoryVar **/discrete** **type=mean sumvar=**QuantResponseVar**;**
R	groupMeans1 **<- by(**myData**$**QuantResponseVar**,** myData**$**CategExplanatoryVar**,** **mean, na.rm=T)** **barplot(**groupMeans1**)**

Code for Bivariate Scatterplot (Quantitative Explanatory Variable and Quantitative Response Variable):

SPSS	**GRAPH /scatterplot(bivar)=**QuantExplanatoryVar **with** QuantResponseVar**.**
STATA	**twoway (scatter** QuantResponseVar QuantExplanatoryVar**) (lfit** QuantResponseVar QuantExplanatoryVar**)**
SAS	**Proc GPLOT; Plot** QuantResponseVar ***QuantExplanatoryVar;**
R	**plot(**QuantResponseVar **~** QuantExplanatoryVar**, data = myData)**

3. Multivariate

Code for Multivariate Output (Categorical Explanatory Variable, Quantitative Response Variable, Categorical 3rd VAR):

SPSS	**MEANS TABLES=** QuantResponseVar **BY** CategExplanatoryVar **BY** CategThirdVar **/CELLS MEAN COUNT STDDEV.**
STATA	**bys** CategExplanatoryVar CategThirdVar**: su** QuantResponseVar
SAS	**proc sort; by** CategExplanatoryVar CategThirdVar**;** **proc means; var** QuantResponseVar**; by** CategExplanatoryVar CategThirdVar**;**
R	**ftable(by(**myData**$**QuantResponseVar**,** **list(**myData**$**CategExplanatoryVar**,** myData**$**CategThirdVar**),** **mean, na.rm = TRUE))**

Code for Multivariate Output (Categorical Explanatory Variable and Categorical Response Variable, Categorical 3rd VAR):

SPSS	**CROSSTABS** **/TABLES=**CategResponseVar **BY** CategExplanatoryVar **BY** CategThirdVar**.**
STATA	**bys** CategExplanatoryVar CategThirdVar**: tab** CategResponseVar
SAS	**proc sort; by** CategThirdVar**;** **proc freq; tables** CategResponseVar***CategExplanatoryVar**; by** CategThirdVar**;**
R	tab1 **<- ftable(**myData**$**CategResponseVar**,** myData**$**CategExplanatoryVar**,** myData**$**CategThirdVar**)** tab1 tab1__colProp **<- prop.table(**tab1**, 2)** tab1_colProp

Note: If your 3rd variable is continuous, for graphing purposes, create meaningful categories and then use the code above.

Bivariate Analysis

ANOVA

SPSS	**UNIANOVA** QuantResponseVar **BY** CategExplanatoryVar**.**
STATA	**oneway** QuantResponseVar CategExplanatoryVar**, tabulate**
SAS	**proc anova;** **class** CategExplanatoryVar**;** **model** QuantResponseVar **=** CategExplanatoryVar**;** **means** CategExplanatoryVar**;**
R	myAnovaResults **<- aov(**QuantResponseVar **~** CategExplanatoryVar**, data =** myData**)** **summary(**myAnovaResults**)**

Pearson Correlation

SPSS	**CORRELATIONS** **/VARIABLES=** QuantResponseVar QuantExplanatoryVar **/STATISTICS DESCRIPTIVES.**
STATA	**corr** QuantResponseVar QuantExplanatoryVar r OR **pwcorr** QuantResponseVar QuantExplanatoryVar**, sig**
SAS	**Proc corr; var** QuantResponseVar QuantExplanatoryVar**;**
R	**cor.test(**myData**$**QuantResponseVar, myData**$**QuantExplanatoryVar**)**

Chi-Square Test

SPSS	**CROSSTABS** **/TABLES=** CategResponseVar **by** CategExplanatoryVar **/STATISTICS=CHISQ.**
STATA	**tab** CategResponseVar CategExplanatoryVar **, chi2 row col**
SAS	**Proc freq; tables** CategResponseVar*****CategExplanatoryVar**/ chisq;**
R	myChi **<- chisq.test(**myData**$**CategResponseVar**,** myData**$**CategExplanatoryVar**)** myChi myChi**$observed** # for actual, observed cell counts **prop.table(**myChi**$observed, 2)** # for column percentages **prop.table(**myChi**$observed, 1)** # for row percentages

POST HOC TESTS WITHIN ANOVA

SPSS	**UNIANOVA** QuantResponseVar **BY** CategExplanatoryVar **/POSTHOC=**CategExplanatoryVar **(TUKEY)** **/PRINT=ETASQ DESCRIPTIVE.**
STATA	**oneway** QuantResponseVar CategExplanatoryVar**, sidak**
SAS	**Proc anova;** **class** CategExplanatoryVar**;** **model** QuantResponseVar**=**CategExplanatoryVar**;** **means** CategExplanatoryVar**/duncan;**
R	myAnovaResults **<- aov(**QuantResponseVar **~** CategExplanatoryVar, **data =** myData**)** **TukeyHSD(**myAnovaResults**)**

POST HOC TESTS FOR CHI SQUARE (must subset data in order to conduct 2X2 comparisons)

SPSS	**TEMPORARY.** **SELECT IF** CategExplanatoryVar**=1 OR** CategExplanatoryVar **=3.** **CROSSTABS** **/TABLES=** CategResponseVar CategExplanatoryVar **/STATISTICS=CHISQ.**
STATA	**tab** CategResponseVar CategExplanatoryVar **If** CategExplanatoryVar**==1 |** CategExplanatoryVar**==3, chi2**
SAS	**IF (**CategExplanatoryVar **= 1) AND (**CategExplanatoryVar **= 3);** (in data step) **Proc freq; tables** CategResponseVar*****CategExplanatoryVar**/ chisq;**
R	myDataSubset **<-** myData**[**myData**$**CategExplanatoryVar **== 1 |** myData**$**CategExplanatoryVar **== 3,]** myChi **<- chisq.test(**myDataSubset**$**CategResponseVar**,** myDataSubset**$**CategExplanatoryVar**)** myChi myChi**$observed** # for actual, observed cell counts **prop.table(**myChi**$observed, 2)** # for column percentages **prop.table(**myChi**$observed, 1)** # for row percentages

Statistical Interactions: Testing for Moderation

ANOVA

In these analyses, the third variable must be categorical.

SPSS	**SORT CASES BY** CategThirdVar**.** **SPLIT FILE LAYERED BY** CategThirdVar**.** **ONEWAY** QuantResponseVar **BY** CategExplanatoryVar **/ STATISTICS DESCRIPTIVES** **/ POSTHOC = BONFERRONI ALPHA (0.05).** **SPLIT FILE OFF.**
STATA	**bys** CategThirdVar**: oneway** QuantResponseVar CategExplanatoryVar**, tab**
SAS	**Proc sort; by** CategThirdVar**;** **Proc anova; class** CategExplanatoryVar**;** **model** QuantResponseVar**=**CategExplanatoryVar**;** **means** CategExplanatoryVar**; by** CategThirdVar**;**
R	**by(**myData**,** myData**$**CategThirdVar**, function(x)** **list(** **aov(**QuantResponseVar **~** CategExplanatoryVar**, data = x),** **summary(aov(** QuantResponseVar **~** CategExplanatoryVar**, data = x))))**

PEARSON CORRELATION

In these analyses, the third variable must be categorical.

SPSS	**SORT CASES BY** CategThirdVar**.** **SPLIT FILE LAYERED BY** CategThirdVar**.** **CORRELATIONS** **/VARIABLES=** QuantResponseVar QuantExplanatoryVar **/STATISTICS DESCRIPTIVES.** **SPLIT FILE OFF.**
STATA	**bys** CategThirdVar**: corr** QuantResponseVar QuantExplanatoryVar OR **bys** CategThirdVar**: pwcorr** QuantResponseVar QuantExplanatoryVar**, sig**
SAS	**Proc sort; by** CategThirdVar**;** **Proc corr; var** QuantResponseVar QuantExplanatoryVar**; by** CategThirdVar**;**
R	**by(**myData**,** myData**$**CategThirdVar**, function(x)** **cor.test(x$**QuantResponseVar**,** **x$**QuantExplanatoryVar**))**

CHI-SQUARE TEST

In these analyses, the third variable must be categorical.

SPSS	**CROSSTABS** **/TABLES =** CategResponseVar **by** CategExplanatoryVar **by** CategThirdVar **/CELLS = COUNT ROW** **/STATISTICS = CHISQ.**
STATA	**bys** CategThirdVar**: tab** CategResponseVar CategExplanatoryVar**,** **chi2 row**
SAS	**Proc sort; by** CategThirdVar**;** **Proc freq; tables** CategResponseVar*****CategExplanatoryVar**/chisq; by** CategThirdVar**;**
R	**by(**myData**,** myData**$**CategThirdVar**, function(x)** **list(** **chisq.test(x$**CategResponseVar**, x$**CategExplanatoryVar**),** **chisq.test(x$**CategResponseVar**, x$**CategExplanatoryVar**)$observed,** **prop.table(chisq.test(x$**CategResponseVar**,** **x$**CategExplanatoryVar**)$observed, 2)))** # column %s

MULTIPLE REGRESSION

SPSS	**REGRESSION** **/DEPENDENT** QuantResponseVar **/METHOD ENTER** ExplanatoryVar**.**
STATA	**reg quan** QuantResponseVar ExplanatoryVar
SAS	**Proc reg; model** QuantResponseVar**=**ExplanatoryVar**;**
R	my.lm **<- lm(**QuantResponseVar **~** ExplanatoryVar, **data =** myData**))** **summary(**my.lm**)**

LOGISTIC REGRESSION

SPSS	**LOGISTIC REGRESSION** BinaryResponseVar **with** ExplanatoryVar**.**
STATA	**logistic** BinaryResponseVar ExplanatoryVar //displays odds ratios OR **logit** BinaryResponseVar ExplanatoryVar //displays logit coefficients
SAS	**Proc logistic; class** ExplanatoryVar (if ExplanatoryVar is categorical)**;** **model** BinaryResponseVar**=**ExplanatoryVar**;**
R	my.logreg **<- glm(**BinaryResponseVar **~** ExplanatoryVar**, data=**myData**,** **family="binomial")** **summary(**my.logreg**)** # for p-values **exp(**my.logreg**$coefficients)** # for odds ratios **exp(confint(**my.logreg**$coefficients))** # for confidence intervals on the odds ratios

Multivariate Regression: Testing for Confounding

MULTIPLE REGRESSION

SPSS	**REGRESSION** **/DEPENDENT** QuantResponseVar **/METHOD ENTER** ExplanatoryVar ThirdVar1 ThirdVar2**.**
STATA	**reg** QuantResponseVar ExplanatoryVar ThirdVar1 ThirdVar2
SAS	**Proc glm; class** CategExplanatoryVar (ref="0") CategThirdVar (ref="2"); When these variables are categorical with more than 2 levels, the number after ref tells SAS what category you want to compare the other categories to. Be sure to use values appropriate to your categorical variables. **model** QuantResponseVar**=** CategExplanatoryVar CategThirdVar QuantThirdVar **/ solution;**
R	my.lm **<- lm(**QuantResponseVar **~** ExplanatoryVar **+** ThirdVar1 **+** ThirdVar2**,** **data =** myData**)** **summary(**my.lm**)**

LOGISTIC REGRESSION

SPSS	**LOGISTIC REGRESSION** BinaryResponseVar **with** ExplanatoryVar ThirdVar1 ThirdVar2**.**
STATA	**logistic** BinaryResponseVar ExplanatoryVar ThirdVar1 ThirdVar2
SAS	**Proc logistic descending;** Use 'descending' when variables are coded 0 (no) and 1 (yes). No need if coded 1 (yes) and 2 (no) **class** CategExplanatoryVar CategThirdVar**;** **model** BinaryResponseVar**=**CategExplanatoryVar CategThirdVar QuantThirdVar**;**
R	my.logreg **<- glm(**BinaryResponseVar **~** ExplanatoryVar **+** ThirdVar1 **+** ThirdVar2**, data =** myData**, family = "binomial")** **summary(**my.logreg**)** # for p-values **exp(my.logreg$coefficients)** # for odds ratios **exp(confint(**my.logreg**$coefficients))** # for confidence intervals on the odds ratios