WEEK 1

Variables into cases = wide to long – **varstocases,** long to wide **casestovars**
Wide = lots of variables (each person represented once), **Long** = person represented on multiple rows)

Keep syntax for variables don't want to change (e.g. age and gender, each person only has one, these aren't 4 variables that happen to have same value)
Index variable will tell us which subscales each row is

*= active data set. Asterix (*) = active dataset
Dataset activate demog = Which one is going into which. Anxiety is going into dem
'break=sex' - will calculate means separately for males and females

nmv = no. of missing values - **compute nmv=nmiss(weight)**
nvv = number of valid values **compute nvv=nvalid(weight)**
Compute does on case by case basis, does for each person not dataset as whole
If compute **$(X_1+X_2+X_3+X_4)/4$** will exclude missing cases
If compute variable **mean(X_1, X_2, X_3, X_4)** will include missing cases

Compute target = f.n(varlist) → the ".n" indicates valid values on at least n amount of variables e.g. mean.1 = valid values on at least 1 variable
Importing data → **readnames on**: tells SPSS excel file top row is var name, not 1st line of data

Recode reverse scored items
Compute item2r = 7 – item2 (subtract original value from max score +1) - correlations become +. E.g., if the scale range is 1-6 then calculate: New = 7 – old
Recode item2 (6=1)(5=2)(4=3(3=4)(2=5)(1=6) into item2rr – less efficient but same
Cronbachs alpha (reliability) >.8 is desirable - changes after recoding!

Compute ngt10 = sum(afees>10, aees>10, dees>10, tees>10).
- Computes score as true/false, if have score > 10 will get a 1 for that score, so if person had value > 10 on every scale they would get score of 4

Need **Temporary** before **select if** otherwise will delete stuff out of dataset.
Recode tees (lo thru 10=0)(10 thru 20=1). person with 10.0 will be in first group not second
Compute tees5 = trunc(tees/10) → Trunc = 9.999 in 1st group then 10.0 in 2nd, in recode takes 10 first time it sees it, whereas here goes up to 9.9999
Random allocation to groups
Compute runi = rv.uniform(0,1) → randomly assign to groups variable 0 or 1
Compute alloc = runi >0.5. →Allocating to two groups. If over .5 they are put in 1

Getting scores of 1-6 to be from 0-5
Do repeat x=item1 to item6
/v = newitem1 to newitem6
compute v=x-1
end repeat.
Exe.

WEEK 2 – SAMPLE SIZE AND POWER

Type I error	Our test is sig (says is effect) but no effect in population	Reject H_0 when its true in pop	Ctrl by choice of sig level (α = sig level)	Pr(type I error) = α
Type II error	Our test is not sig (says no effect) but effect in pop	Accept H_0 when its false in pop	Ctrl by sample size & power. > ss = > power	Pr(type II err) = β

Power = 1- β = probability of not making a type II error / ability to detect effect when there is one). Increased power leads to reduced prob of making type II error
= (1-β) >0.8 is adequate, >0.9 is better
- p-value > α (sig level)- possible power problem