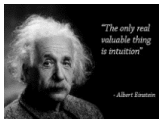
Grouping Variables & Reliability

KOGOD SCHOOL of BUSINESS
AMERICAN UNIVERSITY • WASHINGTON, DC

Grouping Variables

- There are a few **methods** for **grouping** variables into **factors**
- One way is to use the **first few** factor or **principal components** as the new variables (just like **PCA** regression) → good for **accuracy** and to reduce **dimensionality** issues but has **little interpretation**.
- An **alternative** popular method (**preferred**) is simply to **average** the **variables** that **load together** within each **factor** (i.e., high factor loadings within the factor – highly **correlated**) into new **group** variables:
  - If the variables are in the **same scale** (e.g., survey 1 to 7), you can simply **average** them within each of the M factors
  - If the variables are in **different scales** (e.g., weight and size), you can't average apples and oranges, the simplest solution is to **standardize** all variables (see **Transformations** lecture). Once all variables have a **mean** of 0 and **variance** of **1**, they can be averaged
    - Note: standardizing does **NOT** change the factor loadings



Reliability Analysis: Intuition

- Grouping survey items into **factor variables** provides some **assurance** that the **items** included in a factor are **related** (i.e., correlated with each other)
- Some times analysist don't do factor analysis, but even if they do, it is always **good practice** to check the **within-factor average correlation** of the items
- **Cronbach-α** is a measure of the average correlation of a group of items, but it factors in the number of variables included in the group
- The most common rule of thumb is **α > 0.7**
- As the **inter-factor** item **correlation** ↑ and the **number of items** in a factor ↑ the **α** value increases
- Intuitively, this is like having several **judges score** an **Olympic** athlete → as the number of judges ↑ and the correlation of their scores ↑ the reliability of the score ↑

Reliability Analysis: Illustration

```
Reliability analysis
Call: alpha(x = HitEffectiveness)

  raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd
       0.73      0.98       1       0.9  57 0.041  733 646

 Reliability if an item is dropped:
       raw_alpha std.alpha G6(smc) average_r S/N alpha se
CAtBat      0.90      0.98    0.99      0.89  42    0.028
CRuns       0.67      0.98    0.99      0.89  41    0.048
CHits       0.59      0.98    0.99      0.90  43    0.053
CRBI        0.68      0.98    0.98      0.89  41    0.047
CWalks      0.69      0.98    1.00      0.91  54    0.046
CHmRun      0.74      0.99    0.99      0.94  81    0.044

 Item statistics
            n raw.r std.r r.cor r.drop mean   sd
CAtBat    263  1.00  0.98  0.98   0.99 2658 2287
CRuns     263  0.99  0.98  0.98   0.99  361  331
CHits     263  0.99  0.97  0.97   0.99  722  648
CRBI      263  0.96  0.98  0.98   0.96  330  323
CWalks    263  0.92  0.94  0.93   0.91  260  264
CHmRun    263  0.83  0.90  0.89   0.82   69   82
```

- In the example above we grouped 7 items that loaded together in the rotated factor analysis: Hitters, CAtBat, Runs, CHits, CRBI, Cwalks and CHmRun
- The output shows **Cronbach α = 0.98** (std.alpha), which is pretty high, meaning that the 7 items are highly correlated as a group
- raw_alpha is based on covariances and it is not accurate if the variable scales are different. Better use std.alpha
- The r.drop column is very useful because it shows what the **Cronbach α** would be if the item were to be dropped. Notice that we can either drop CAtBat, CRuns or CHits and increase **α** to 0.99

R Tips

`alpha(){psych}` → Function in the {psych} package to compute Cronbach's α reliability statistic ; think of it as an average correlation for a group of variables, adjusted upwards as more variables are included.

`Select(){dplyr}` → A convenient function to create data frame sub-sets from data sets.

`dataSubset=select(dataName, x1, x2, etc.)` → Creates a smaller data subset called dataSubset

`Alpha(dataSubset)` → Provides the reliability statistics shown in the previous slide

`alpha(dataSubset, keys=c(1,-1,1,1,1,1))` → the keys= attribute can be used to reverse one or more variables in the α computation. In this illustration, the second variable (x2) is reversed and the other 5 are not

Aggregating variables into factor averages:

`dataName$NewFactorAvg=(x1+x2+….+x6)/6` → To create a new column in dataName with the average of the 6 variables

KOGOD SCHOOL
of
BUSINESS