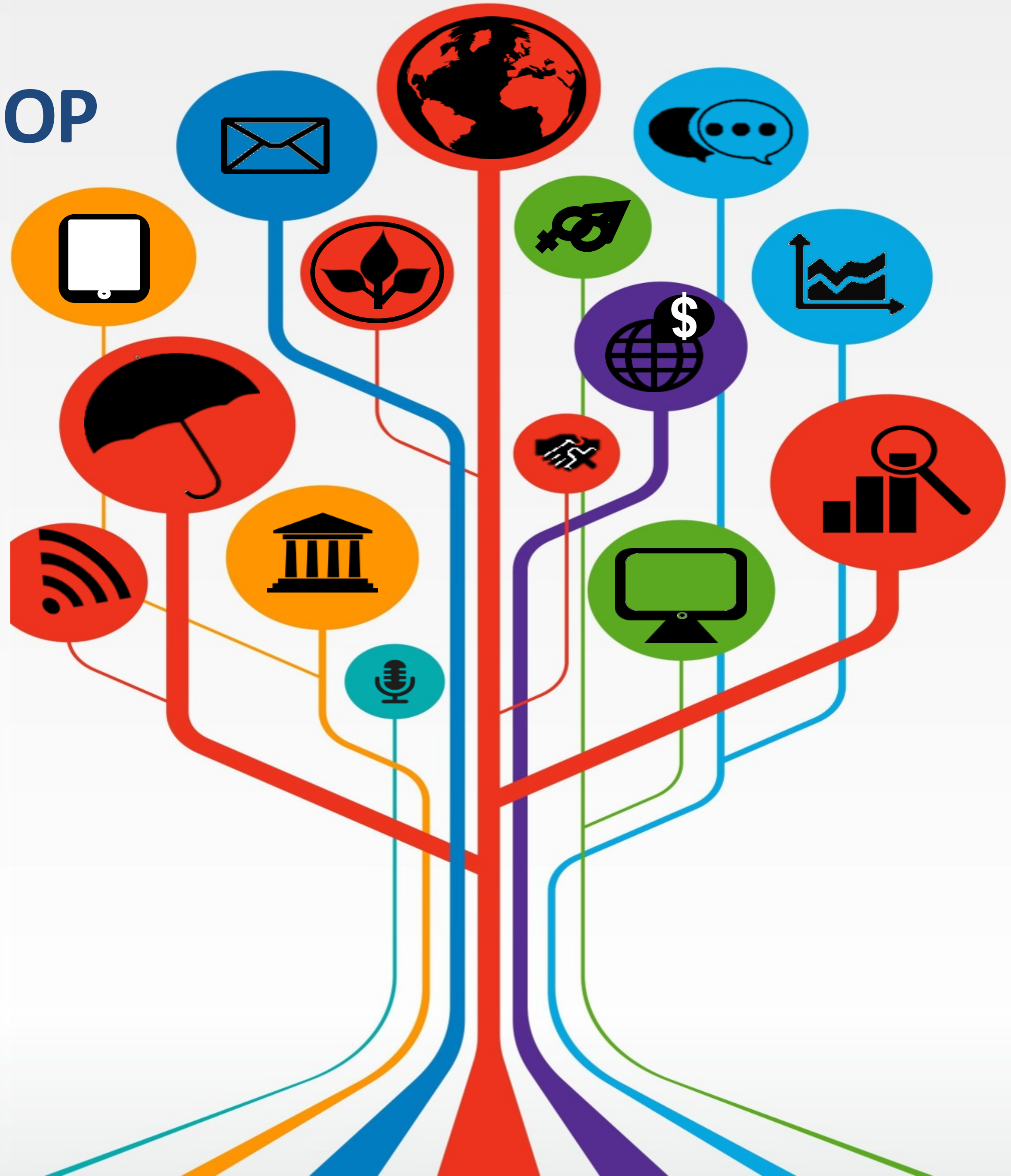
FIELD COORDINATOR WORKSHOP



Manage Successful Impact Evaluations

18 - 22 JUNE 2018
WASHINGTON, DC

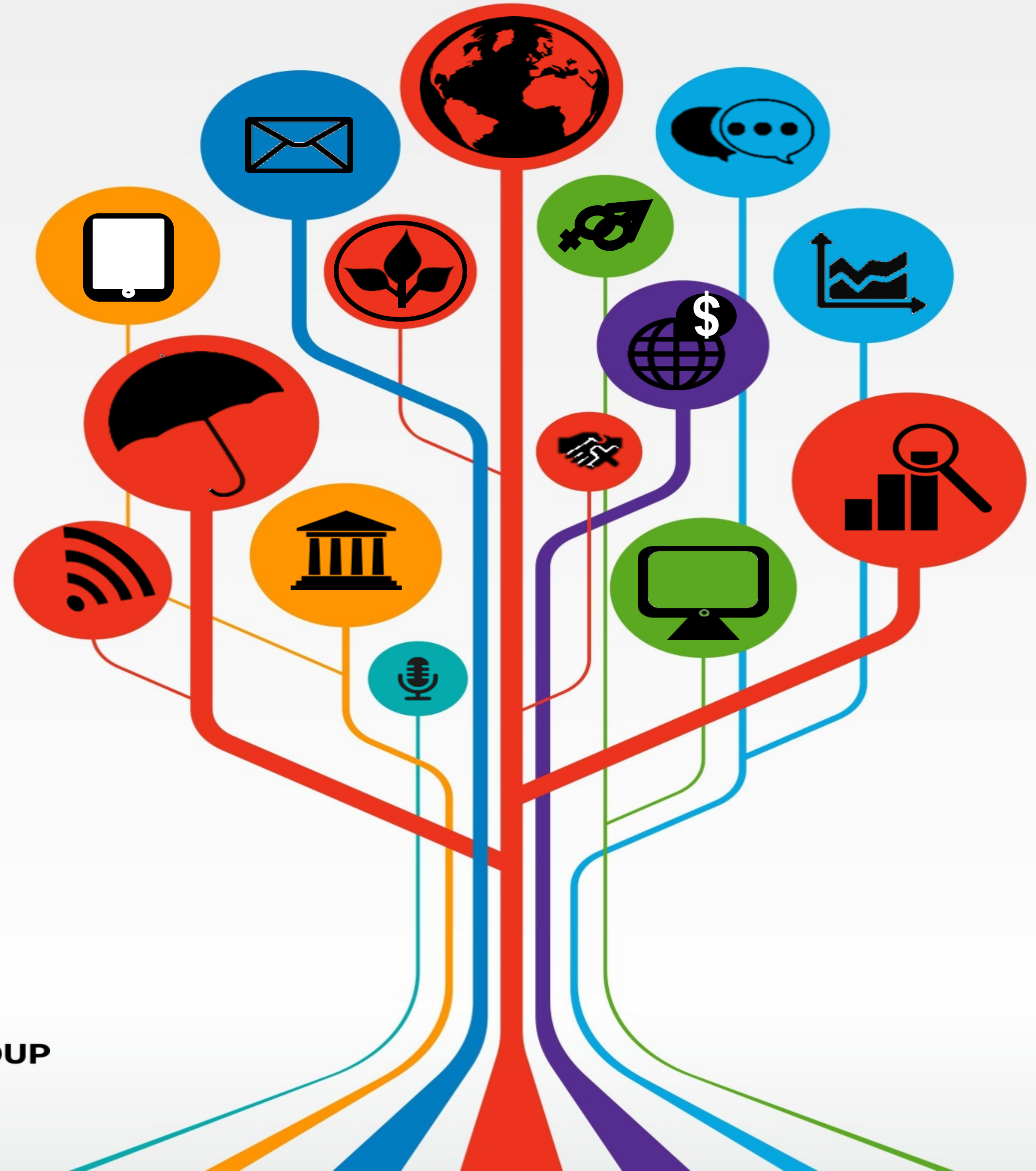
i2i DIME
TRANSFORM DEVELOPMENT

WORLD BANK GROUP

Lab 5 – Track 1 – Randomization

Sakina Shibuya & Kristoffer Bjärkefur

20 June 2018



TRANSFORM DEVELOPMENT

WORLD BANK GROUP

Key Terms

- Treatment and Control
  - Treatment means receiving the project
  - control means not receiving the project

- Treatment Arms
  - If there are multiple versions of the intervention, each group receiving a different version of the project is called a treatment arm.

Types of Randomization

- Simple
  - Each individual person is assigned either T or C
- Cluster
  - Randomization happens on group level (village, school etc.) and all individuals in a group are assigned the same treatment
- Stratified
  - Sub-sets of similar observations (rich/poor, male/female etc.) are determined in advance, and randomization is done separately within each sub-set
  - Guarantees that equal number of similar observations (rich/poor, male/female etc.) are assigned to be each treatment and control
- Pairwise
  - Extreme form of stratification
  - All units are matched to make pairs that are as similar as possible and one unit from each pair is assigned to be T or C

Why do we randomize?

- “Random” does not imply “completely random”, we want a controlled randomization

- We want to assign the intervention of our projects so that the control group is as similar as possible to the treatment group as possible
  - This is called a balance treatment assignment
  - Randomization is the most common tool to achieve that

- Each observation needs to have the same probability to end up in the each treatment arm, and all members in a strata need to be statistically similar

Methods of randomization

- Good:
  - Field Based
  - Stata, R, Python – and other replicable software
- Bad:
  - Excel – and other non-replicable software
  - Excel and many other software has random generators, but they do not allow a controlled randomization

Method: Field Based

- Examples
  - Drawing numbers from a hat, flipping a coin etc.
- Advantages
  - Transparent to participants
  - Allows randomization without exactly knowing treatment population in advance
- Disadvantages
  - Not transparent to anyone not present
  - Not replicable
  - Difficult to manage any a complex randomization with, for example, stratification

Method: Stata

- Advantages
  - Fully replicable
  - Relatively easy to set up complex randomizations
  - We can run a test randomization and analyze the outcome before we draw new random numbers for the actual randomization

- Disadvantages
  - Can seem very mysterious to beneficiaries and project staff

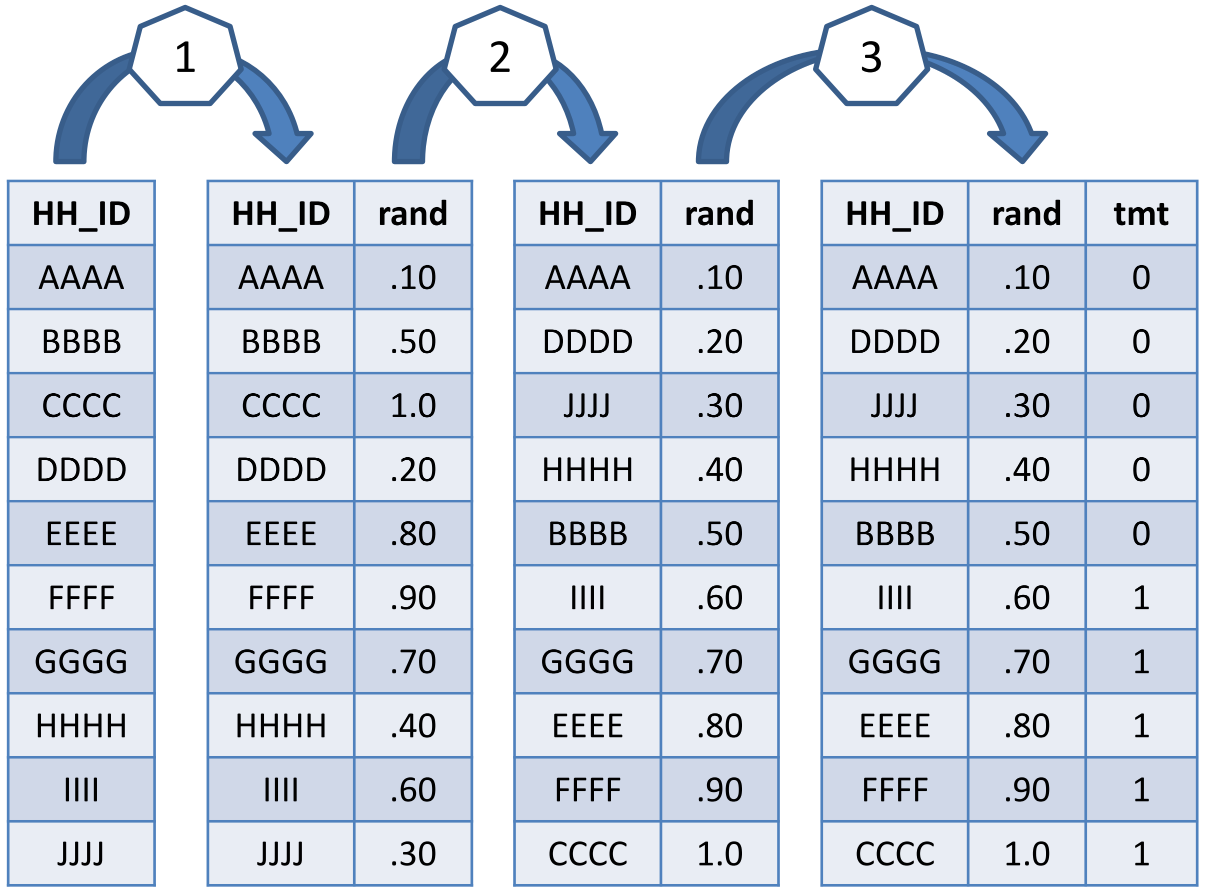
Prepare randomization in Stata

- Obtain a list of observations to be randomized
- Define a randomization rule
  - How many units (people) are in the population?
  - How many treatment arms do you have?
  - How big share of the observations should end up in each group?
  - Are we using stratification?
  - Which variables will we use to test balance?
- Randomize and document using Stata!

Ex 1: Basic Randomization in Stata

- We have 10 students and we want half of them to be treatment and control
- What is our randomization rule?
- How do we do this in Stata so it is random but replicable?

Simplified example of randomization



HH_ID
AAAA
BBBB
CCCC
DDDD
EEEE
FFFF
GGGG
HHHH
IIII
JJJJ

HH_ID	rand
AAAA	.10
BBBB	.50
CCCC	1.0
DDDD	.20
EEEE	.80
FFFF	.90
GGGG	.70
HHHH	.40
IIII	.60
JJJJ	.30

HH_ID	rand
AAAA	.10
DDDD	.20
JJJJ	.30
HHHH	.40
BBBB	.50
IIII	.60
GGGG	.70
EEEE	.80
FFFF	.90
CCCC	1.0

HH_ID	rand	tmt
AAAA	.10	0
DDDD	.20	0
JJJJ	.30	0
HHHH	.40	0
BBBB	.50	0
IIII	.60	1
GGGG	.70	1
EEEE	.80	1
FFFF	.90	1
CCCC	1.0	1

1. Start with a sorted list of observations you want to randomize. Generate a random number.

2. Sort the observations after this random number. The order of the observations are now randomly sorted.

3. Assign 0 (control) to the first half of the observations, and assign 1 (treatment) to the second half.

The 3 rules of replicable randomization

- We want to be able to replicate the randomization and get the same results each time. This is needed for research transparency

- To achieve that in Stata we have three rules:
  1. Set the version of Stata
  2. Set the seed
  3. Stable sort

- The next slides explains the meaning of these rules and why it matters

Rule 1: Set the version of Stata

Stata 11	1st	2nd	3rd	4th	5th	6th	7th	8th
	0.625	0.000	0.125	0.375	0.250	0.875	0.500	0.750

Stata 12	1st	2nd	3rd	4th	5th	6th	7th	8th
	0.625	0.500	0.875	0.000	0.375	0.125	0.250	0.750

Stata 13	1st	2nd	3rd	4th	5th	6th	7th	8th
	0.250	0.625	0.125	0.750	0.875	0.500	0.000	0.375

- Stata has pre-calculated list of random numbers. However, these lists differs between versions of Stata.
- For our purposes, all these lists are equally good, but we need to pick one. You can set Stata to use an older list but not a newer
- In reality these lists are billions of items long, instead of 8 as in the example above

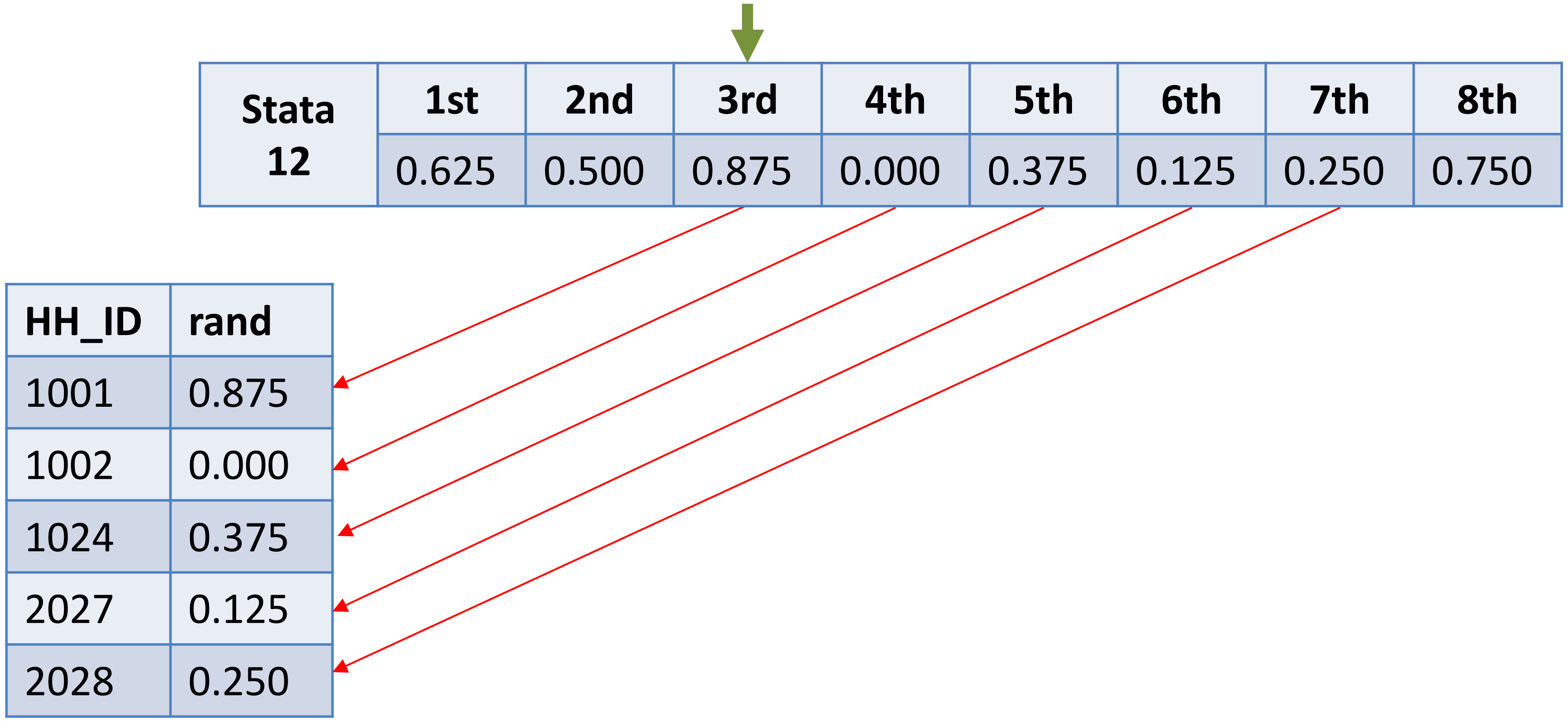
Rule 1: Set the version of Stata

Stata 12	1st	2nd	3rd	4th	5th	6th	7th	8th
	0.625	0.500	0.875	0.000	0.375	0.125	0.250	0.750

HH_ID	rand
1001	0.625
1002	0.500
1024	0.875
2027	0.000
2028	0.375

- Stata goes through the lists and assigns the 1st value to the first observation, 2nd to the second observation, etc.

Rule 2: Set the seed

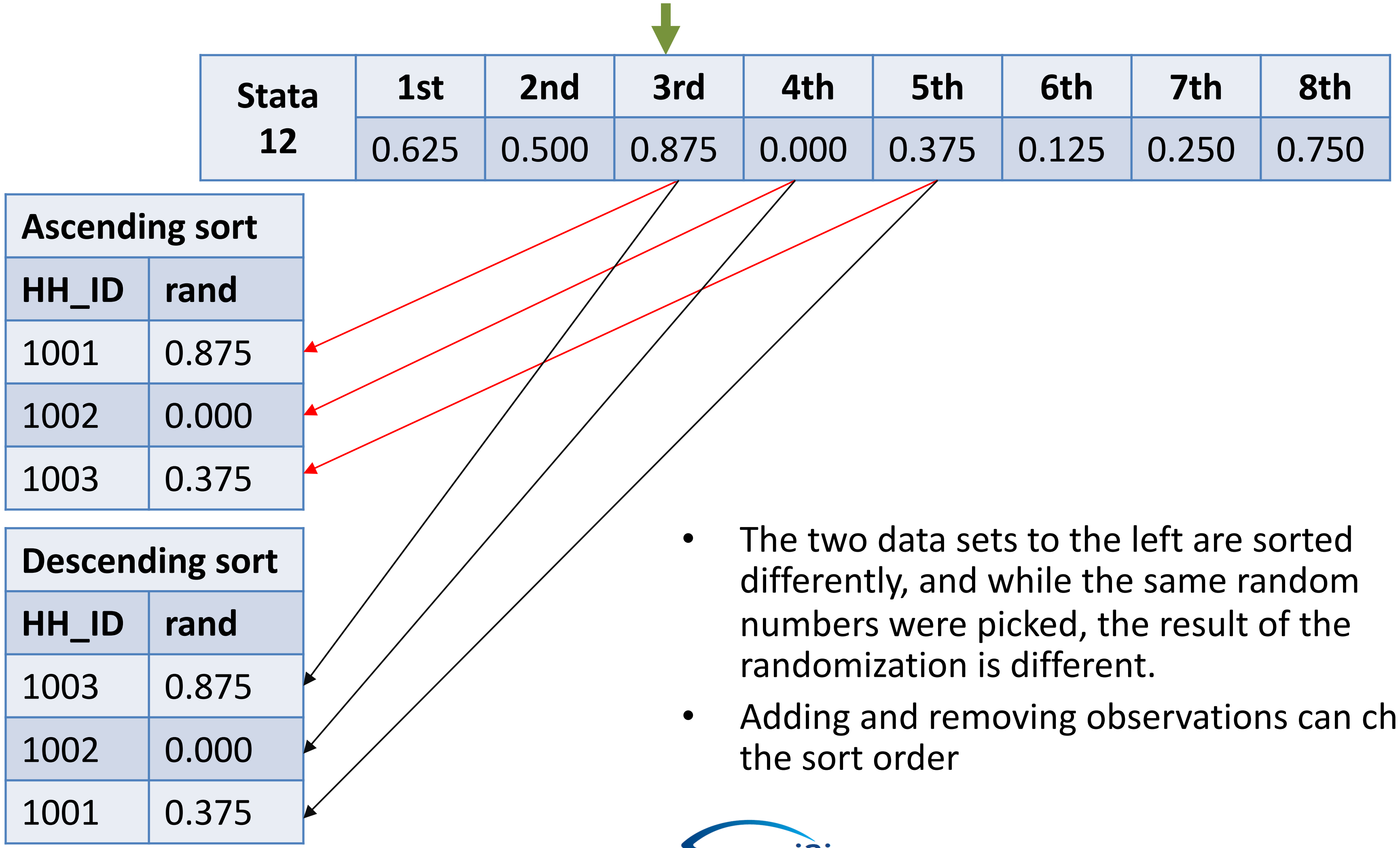


Stata 12	1st	2nd	3rd	4th	5th	6th	7th	8th
	0.625	0.500	0.875	0.000	0.375	0.125	0.250	0.750

HH_ID	rand
1001	0.875
1002	0.000
1024	0.375
2027	0.125
2028	0.250

- Setting the seed change the starting place in the list
- Randomly selecting a seed means randomizing the starting point
- If no seed is set, a seed is randomized each time you run the code, this means random but not replicable

Rule 3: Stable sort



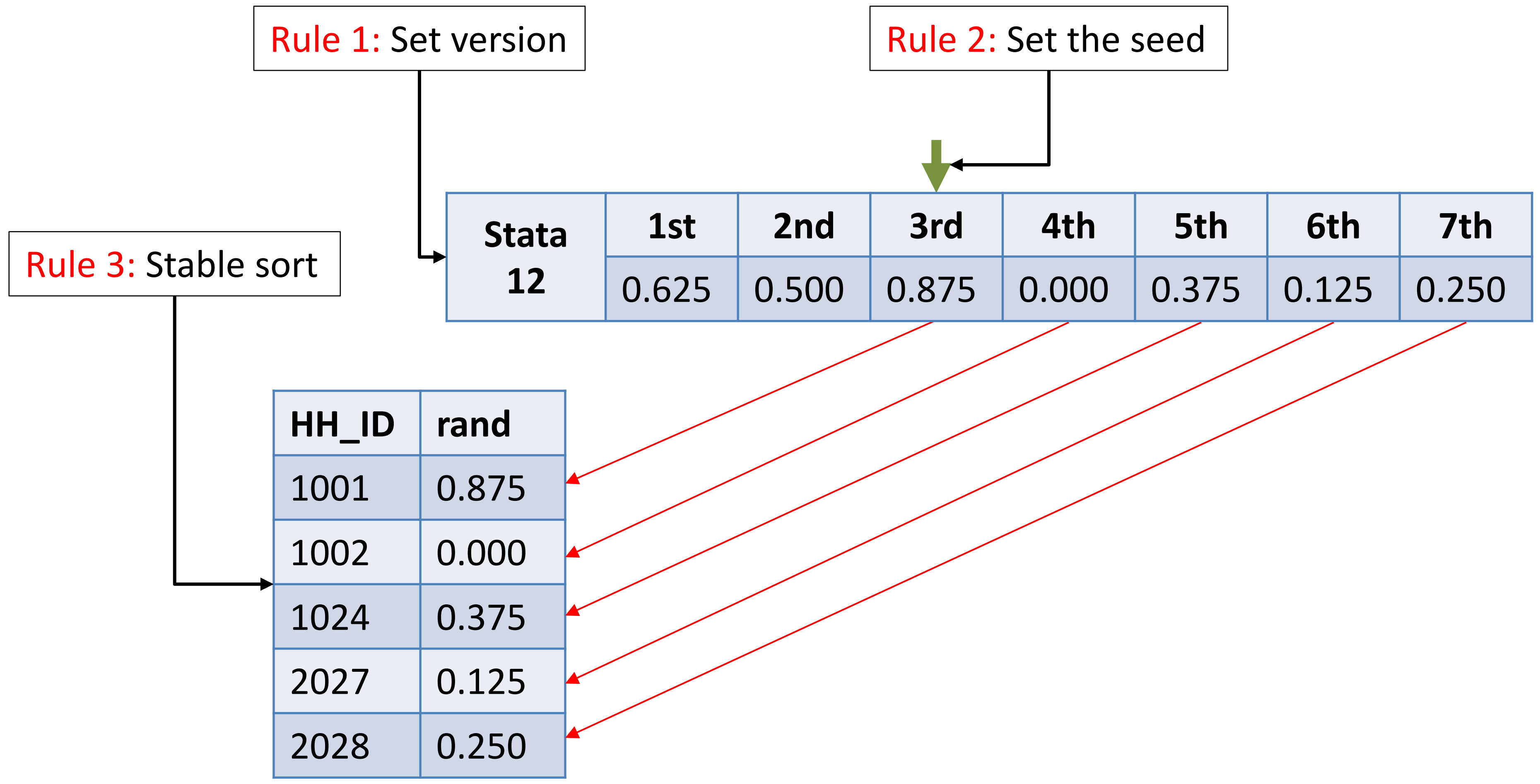
Stata 12	1st	2nd	3rd	4th	5th	6th	7th	8th
	0.625	0.500	0.875	0.000	0.375	0.125	0.250	0.750

Ascending sort	
HH_ID	**rand**
1001	0.875
1002	0.000
1003	0.375

Descending sort	
HH_ID	**rand**
1003	0.875
1002	0.000
1001	0.375

- The two data sets to the left are sorted differently, and while the same random numbers were picked, the result of the randomization is different.
- Adding and removing observations can change the sort order

The 3 rules of replicable randomization



Stata 12	1st	2nd	3rd	4th	5th	6th	7th
	0.625	0.500	0.875	0.000	0.375	0.125	0.250

HH_ID	rand
1001	0.875
1002	0.000
1024	0.375
2027	0.125
2028	0.250

3 rules for replicable randomization

1. Set the version of Stata
   - Guarantees the same list of random numbers
   - See commands *ieboilstart* or *version* in Stata
2. Set the seed
   - Guarantees the same starting point in that list
   - See command *set seed.* Randomize a seed at least 6 digits long (for example at www.random.org)
3. Stable sort
   - Guarantees that the same observation gets the same random number from the list
   - Sort the data in way that will remain constant even if someone else change the sort order of the data set you are using
   - Be aware that changing the data set, adding or removing observations, changes the sort order!

Ex 1: Basic Randomization in Stata

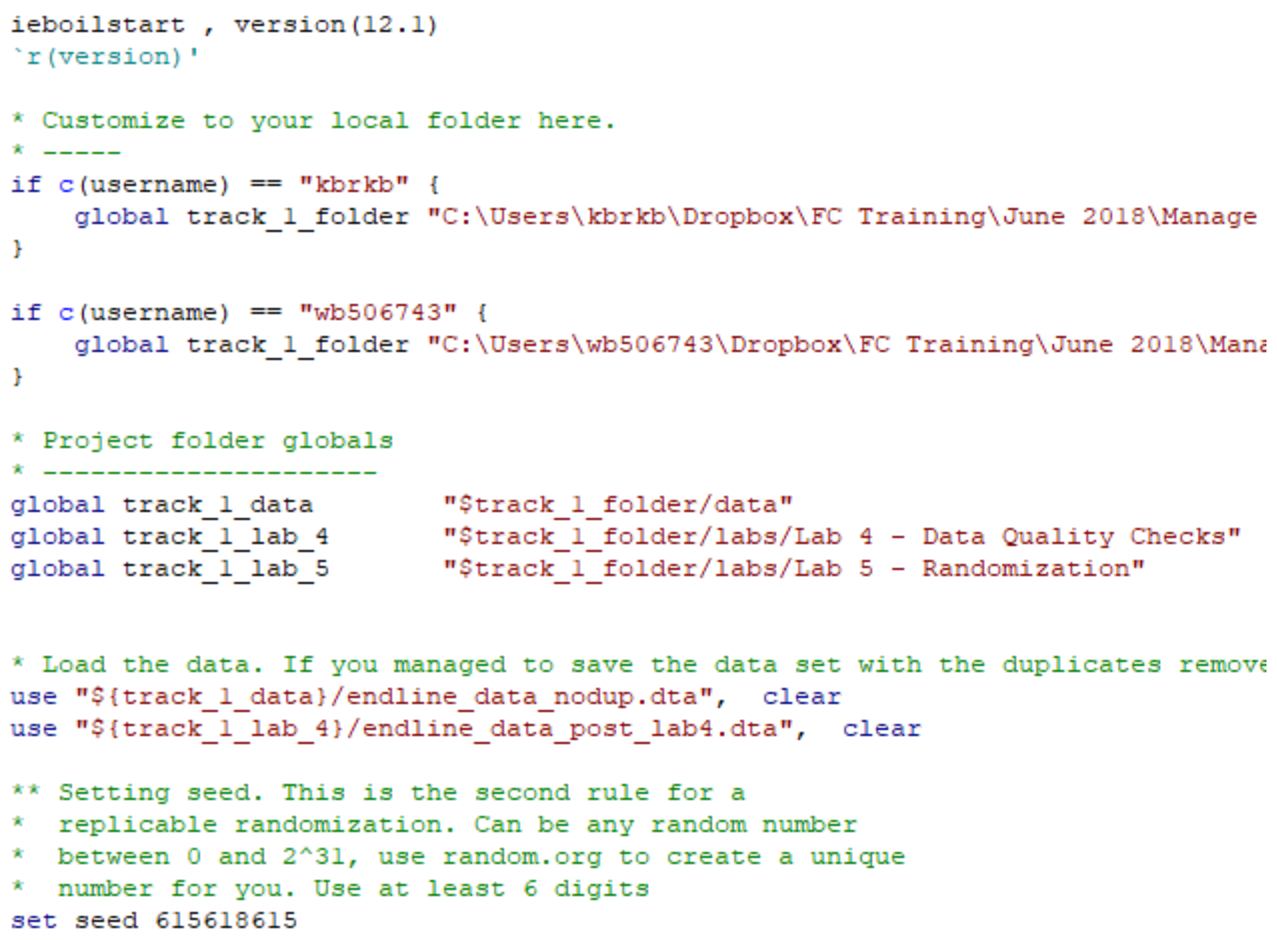
- Open the data set you saved after resolving the duplicates or use *endline_data_nodup.dta*
- We have 1065 households and we want half of them to be treatment and control
  - If you have 1067 observation you are using a data set where the duplicates are not removed yet
- What is our randomization rule?
- Let’s see an example of a replicable randomization of this

Ex 1: Basic Randomization in Stata

Rule 1: Set version

Load data

Rule 2: Set seed



```
ieboilstart , version(12.1)
`r(version)'

* Customize to your local folder here.
* -----
if c(username) == "kbrkb" {
    global track_1_folder "C:\Users\kbrkb\Dropbox\FC Training\June 2018\Manage
}

if c(username) == "wb506743" {
    global track_1_folder "C:\Users\wb506743\Dropbox\FC Training\June 2018\Mana
}

* Project folder globals
* ---------------------
global track_1_data         "$track_1_folder/data"
global track_1_lab_4        "$track_1_folder/labs/Lab 4 - Data Quality Checks"
global track_1_lab_5        "$track_1_folder/labs/Lab 5 - Randomization"


* Load the data. If you managed to save the data set with the duplicates remove
use "${track_1_data}/endline_data_nodup.dta",  clear
use "${track_1_lab_4}/endline_data_post_lab4.dta",  clear

** Setting seed. This is the second rule for a
*  replicable randomization. Can be any random number
*  between 0 and 2^31, use random.org to create a unique
*  number for you. Use at least 6 digits
set seed 615618615
```

Ex 1: Basic Randomization in Stata

Rule 3: Stable sort

```
** Stable sort. This is the third rule for a
*  replicable randomization.
sort id_05
```

Generate random number and sort

```
** Generate a variable with a random number for all
*  observations and sort the observations after that
*  number.
generate     rand = runiform()
sort         rand
```

Variables used in assignment

```
** Create one variable with the rank on the random
*  number. And a varaible with the total number of
*  observations.
generate rank    = _n
generate tot_obs = _N
```

Assign to treatment if rank is less then half

```
** Create the treatment variable. Change the value
*  to 1 if the rank is more than half the number
*  of total observations in the data set.
generate     hh_treatment = 0
replace      hh_treatment = 1 if rank > tot_obs/2
```

Label the variable

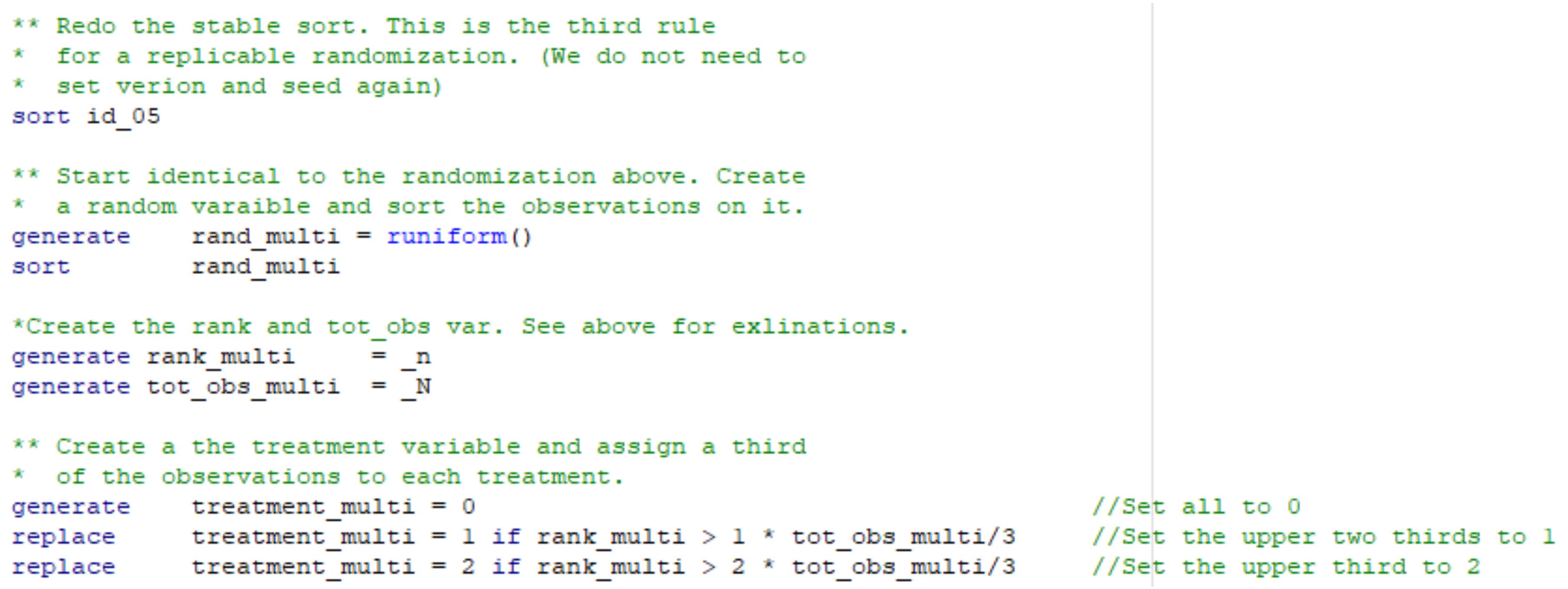
```
*Create a label docuementing the treatment variable
label define                treat_lab 0 "Control" 1 "Treatment"
label values hh_treatment   treat_lab

*Test the randomization
tabulate hh_treatment
```

Ex 2: Randomization in Stata

- Multiple treatment arms
- We have 1065 households and we want one third of them to be control and two treatment arms with one third in each

Ex 2: Randomization in Stata



```
** Redo the stable sort. This is the third rule
*  for a replicable randomization. (We do not need to
*  set verion and seed again)
sort id_05

** Start identical to the randomization above. Create
*  a random varaible and sort the observations on it.
generate      rand_multi = runiform()
sort          rand_multi

*Create the rank and tot_obs var. See above for exlinations.
generate rank_multi      = _n
generate tot_obs_multi   = _N

** Create a the treatment variable and assign a third
*  of the observations to each treatment.
generate      treatment_multi = 0                                    //Set all to 0
replace       treatment_multi = 1 if rank_multi > 1 * tot_obs_multi/3     //Set the upper two thirds to 1
replace       treatment_multi = 2 if rank_multi > 2 * tot_obs_multi/3     //Set the upper third to 2
```

- Similar to Ex 1, but we assign the households to 4 groups

Real life issues

- The example we have covered here is a school book example, but in real life, there are many common issues that makes randomization more complicated

- There are commands that take care of these types of issues. Among them we recommend *randtreat.* To use this in a real life example, see track 2

- But if you do not understand the school book example, you are likely to not know how to use *randtreat* properly

Common Issues

- Differently sized groups. For example 40% is control, and treatment arm 1 and 2 is 30% each – *randtreat* solves this

- Stratification. Split up the observations into groups. Rich/Poor, Male/Female, regions, etc. and then do the randomization in each group – *randtreat* solves this

- Uneven groups (for example, divide 20 observations in 8 groups). The number of observations is not evenly divisible with the number of treatment arms. How to deal with the left overs observation? All to control? All to treatment? – *randtreat* solves this

The 3 rules still apply to *randtreat*

- Even if you are using *randtreat* you need to remember to apply the three rules for the randomization to be fully replicable:
  - *Version*
  - *Seed*
  - *Sort*

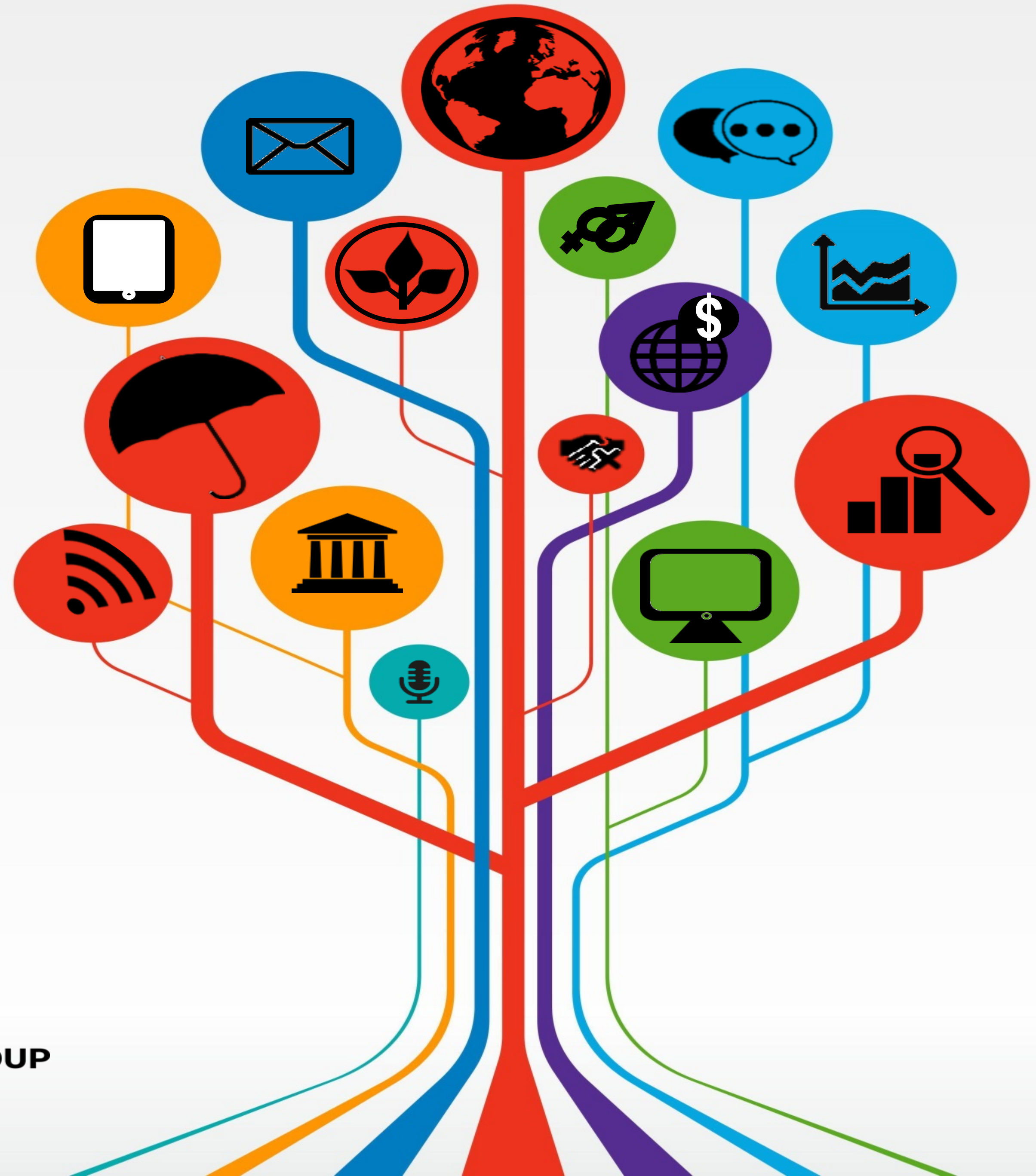
Additional Resources

- Duflo, Glennerster, and Kremer Handbook
  - Useful for understanding types of randomization and reasons for randomizing
  - http://economics.mit.edu/files/806

Thank you!

Sakina Shibuya & Kristoffer Bjärkefur

20 June 2018



i2i DIME TRANSFORM DEVELOPMENT

WORLD BANK GROUP