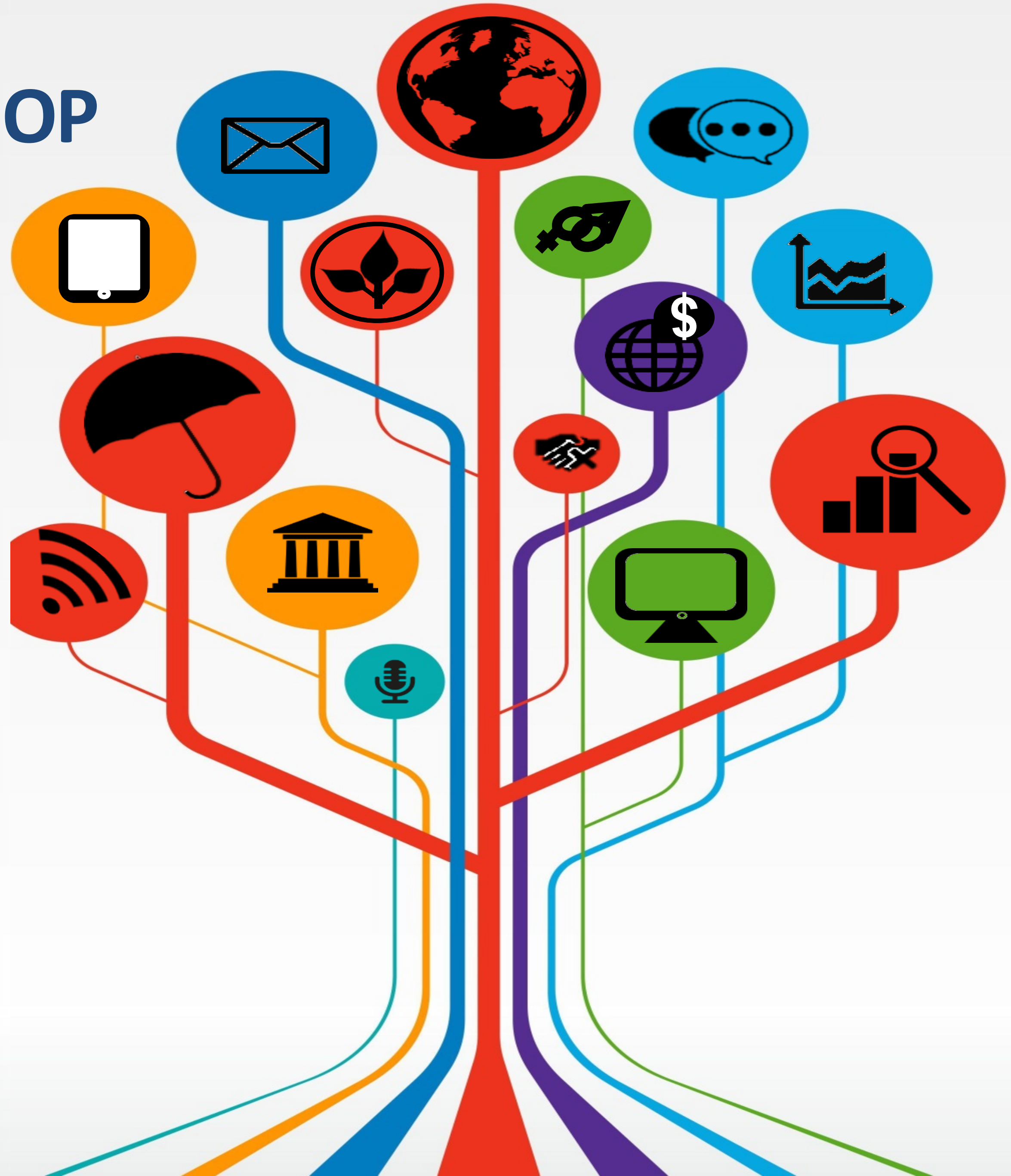
FIELD COORDINATOR WORKSHOP



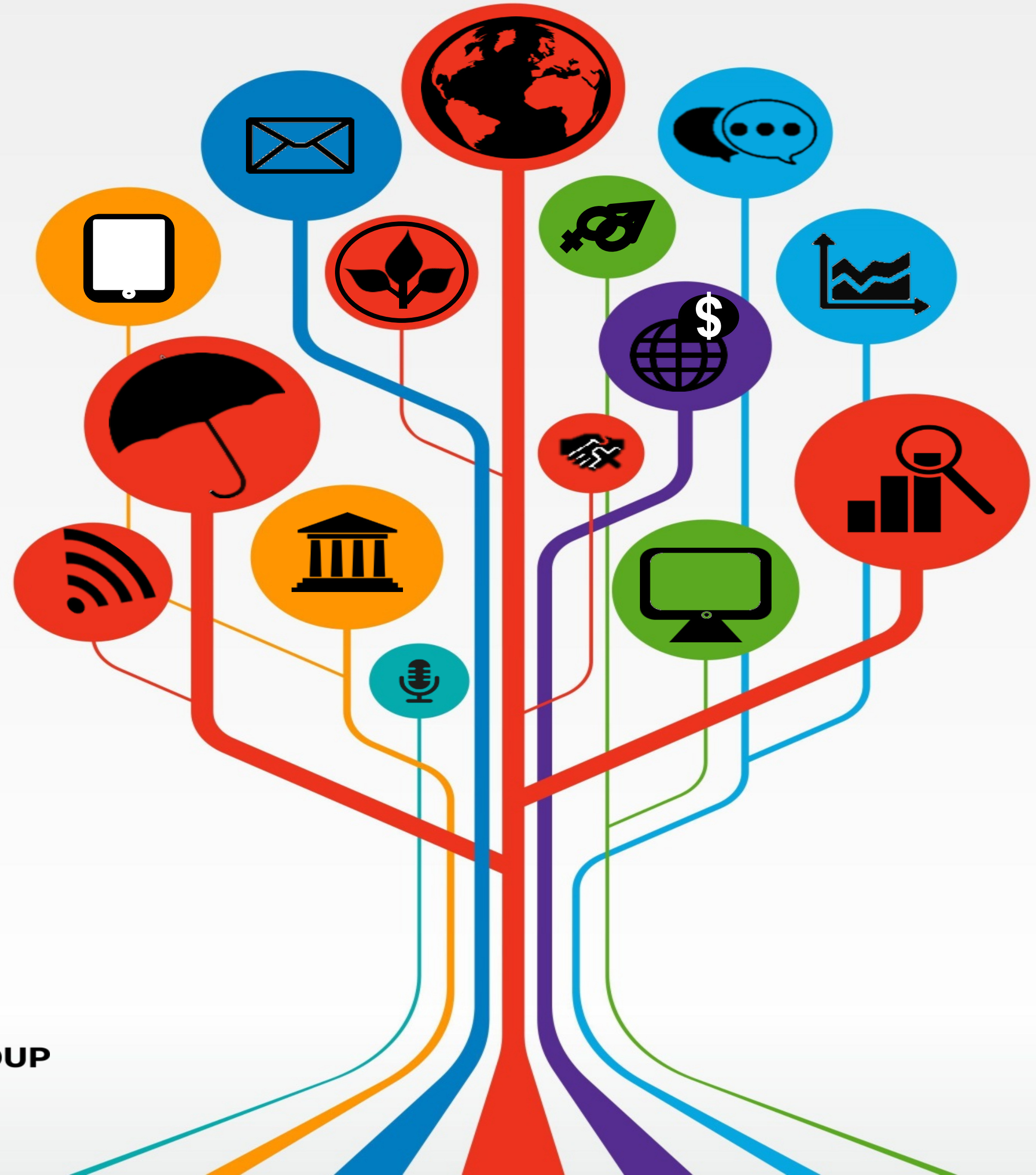
Manage Successful Impact Evaluations

18 - 22 JUNE 2018
WASHINGTON, DC

i2i DIME TRANSFORM DEVELOPMENT

WORLD BANK GROUP

Lab 3 – Track 1 – Data Management and Cleaning



Michael Orevba & Kristoffer Bjärkefur

19 June 2018

i2i DIME TRANSFORM DEVELOPMENT

WORLD BANK GROUP

Think reproducible

Outline

1. Folder organization
2. Master do-files
3. iefolder
4. First overview of a data set
5. Master data sets
6. Basic data cleaning tasks:
   - Unique id,
   - Variable and value labels
   - Missing values
   - Check for outliers

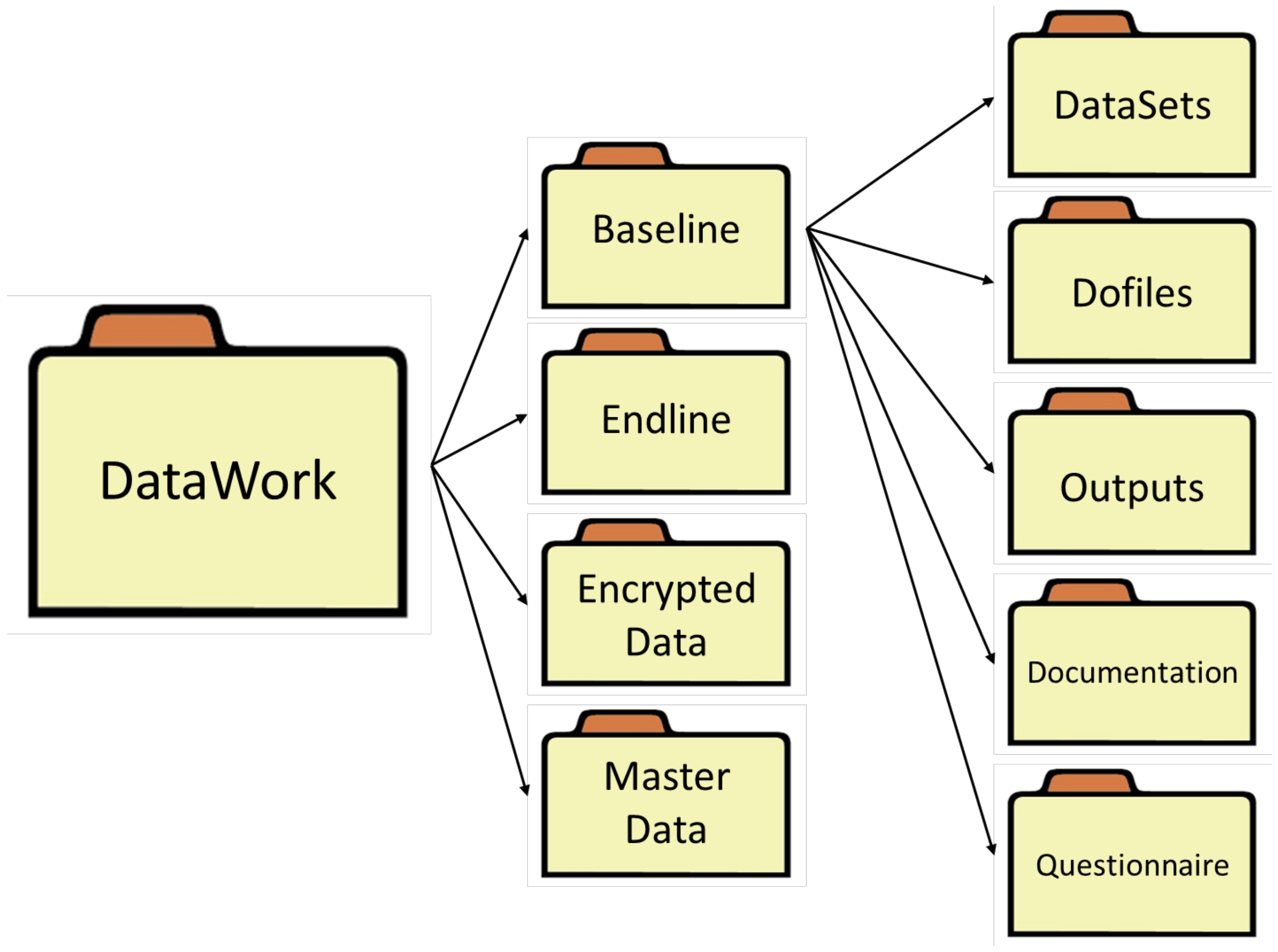
Data Management

- The first part of this presentation will introduce you to some data management practices that might seem very advanced

- Don’t worry about how to set up what we will talk about on the next slides. We have developed easy-to-use tools that will do that for you

Folder Organization

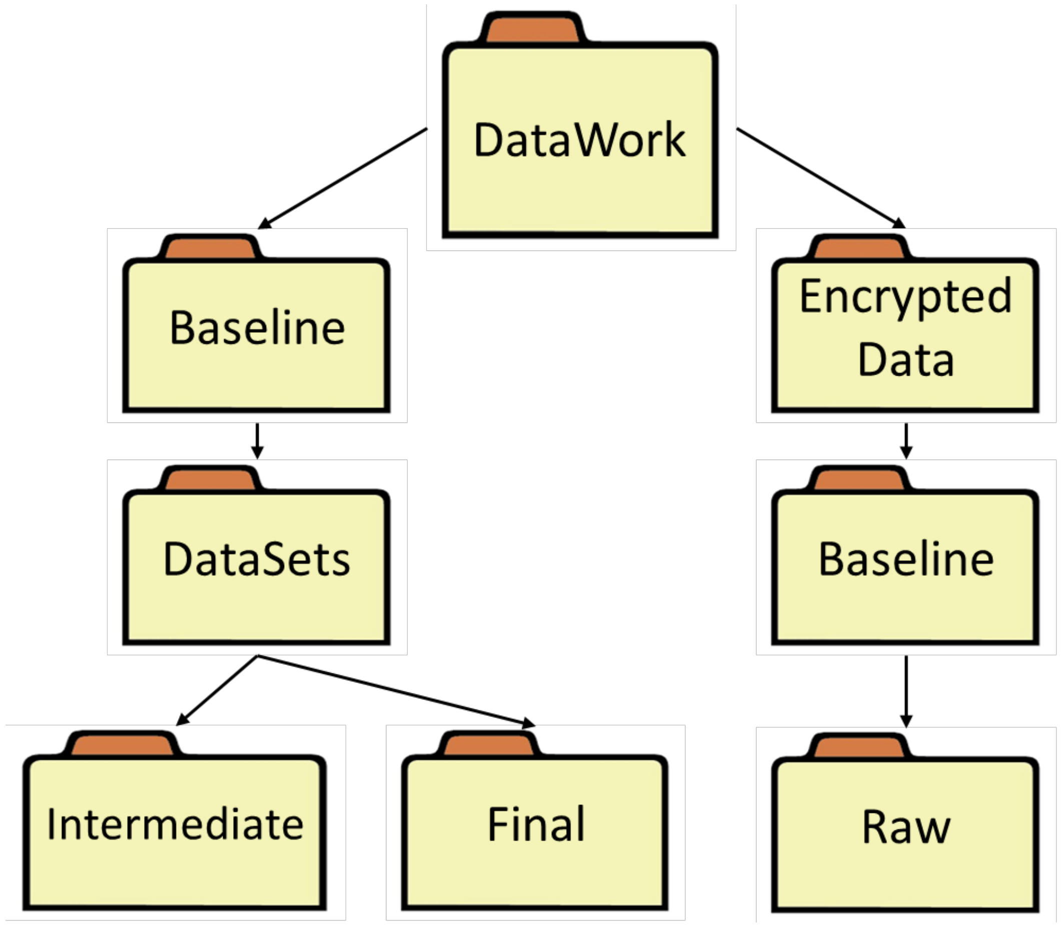
Sample Folder Structure



Folder Organization

Data Sets folders in detail

- Raw data with identifying information should be stored in the *EncryptedData* folder
- All data in the *Intermediate* or the *Final* folder should be de-identified
- Use boxcryptor or truecrypt to encrypt the *EncryptedData* folder



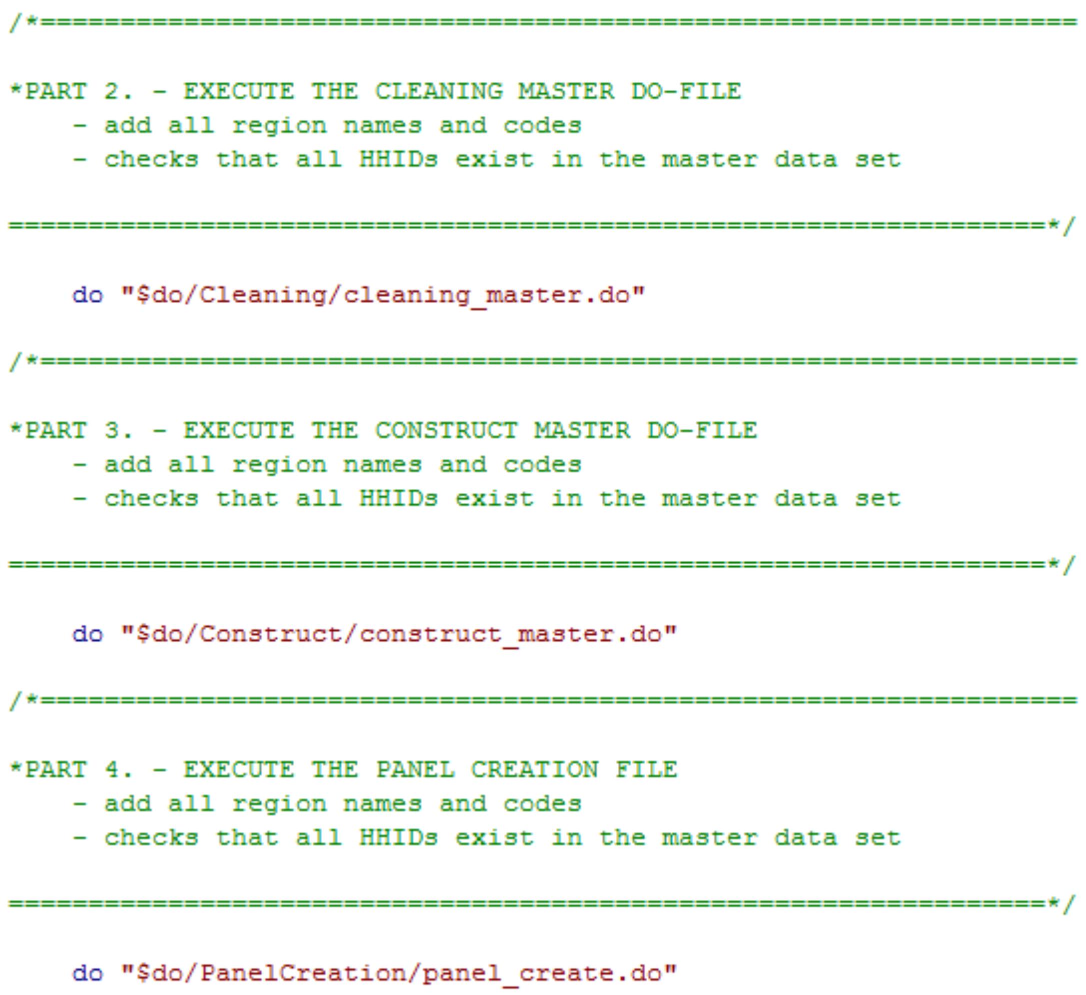
Outline

1. Folder organization
2. Master do-files
3. iefolder
4. First overview of a data set
5. Master data sets
6. Basic data cleaning tasks:
    - Unique id,
    - Variable and value labels
    - Missing values
    - Check for outliers

What is a Master do-file?

- You can make a do-file run other dofiles
  - First, run do-file 1
  - Then, run do-file 2
  - Then, run do-file 3, etc.
- This (and a few secondary but still important things) is what a master do-file does.
- Without a master do-file, you either need one extremely long do-file, or write instructions in which order to run all the do-files

Exempt from a Master do-file



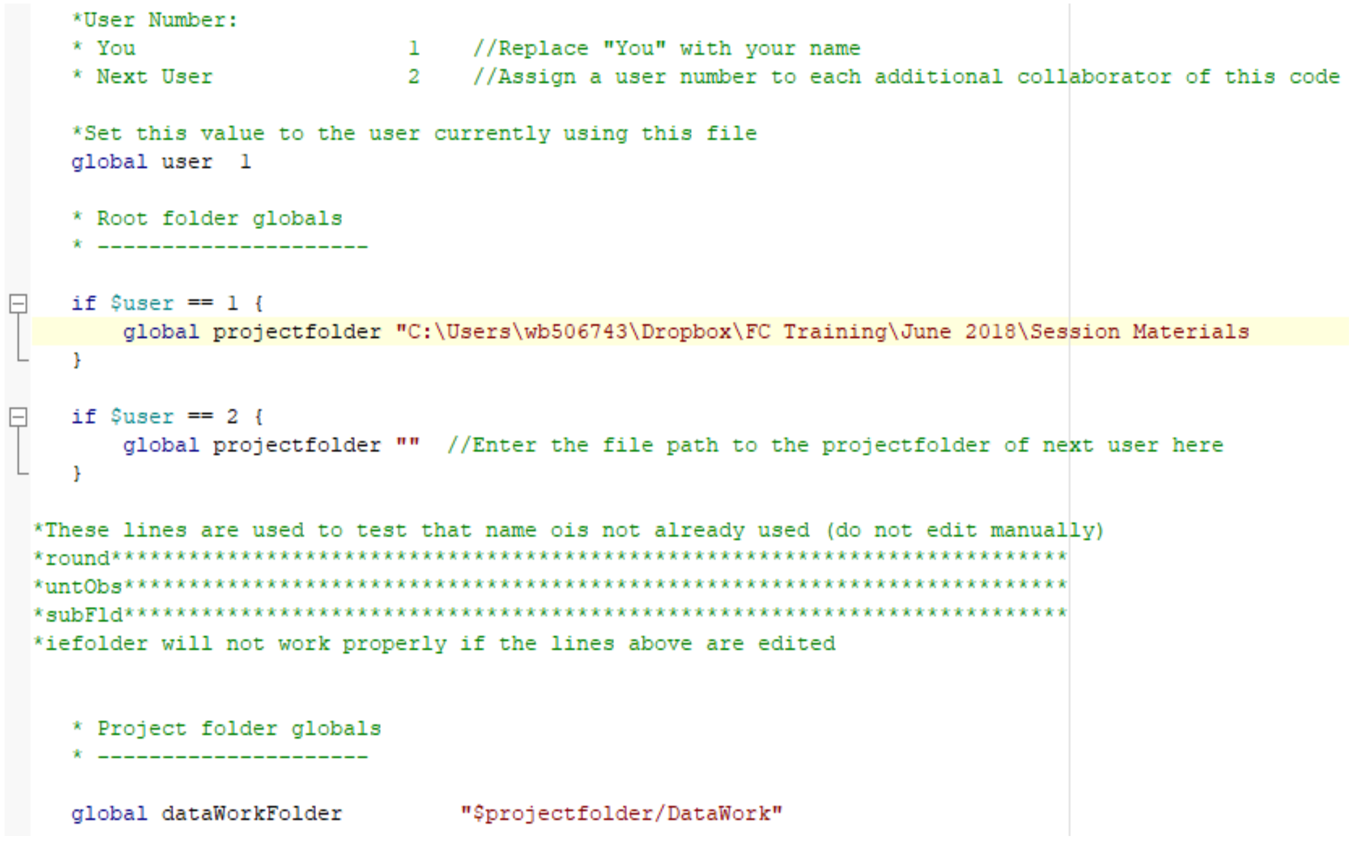
```
/*=================================================================

*PART 2. - EXECUTE THE CLEANING MASTER DO-FILE
    - add all region names and codes
    - checks that all HHIDs exist in the master data set

=================================================================*/

    do "$do/Cleaning/cleaning_master.do"

/*=================================================================

*PART 3. - EXECUTE THE CONSTRUCT MASTER DO-FILE
    - add all region names and codes
    - checks that all HHIDs exist in the master data set

=================================================================*/

    do "$do/Construct/construct_master.do"

/*=================================================================

*PART 4. - EXECUTE THE PANEL CREATION FILE
    - add all region names and codes
    - checks that all HHIDs exist in the master data set

=================================================================*/

    do "$do/PanelCreation/panel_create.do"
```

Master do-file
The map to all data work

- At the end of every round of a project:
  - You should be able to reproduce all your work from raw data to all outputs with one click in this do-file
  - Anyone should be able to follow and to reproduce all your work form raw data to all outputs with one click in this do-file, after adding only their root folder path

Master do-file
Allows easy collaboration

- If we share a project over DropBox or OneDrive all team member have the same folder structure.
- A master do-file allows multiple people to set their own global to the project folder.
- This way anyone sharing the project folder can easily run your do-files.



```
   *User Number:
   * You                    1    //Replace "You" with your name
   * Next User              2    //Assign a user number to each additional collaborator of this code

   *Set this value to the user currently using this file
   global user  1

   * Root folder globals
   * ---------------------

   if $user == 1 {
       global projectfolder "C:\Users\wb506743\Dropbox\FC Training\June 2018\Session Materials
   }

   if $user == 2 {
       global projectfolder ""  //Enter the file path to the projectfolder of next user here
   }

*These lines are used to test that name ois not already used (do not edit manually)
*round**************************************************************************
*untObs*************************************************************************
*subFld*************************************************************************
*iefolder will not work properly if the lines above are edited


   * Project folder globals
   * ---------------------

   global dataWorkFolder         "$projectfolder/DataWork"
```

Outline

1. Folder organization
2. Master do-files
3. iefolder
4. First overview of a data set
5. Master data sets
6. Basic data cleaning tasks:
   - Unique id,
   - Variable and value labels
   - Missing values
   - Check for outliers

iefolder

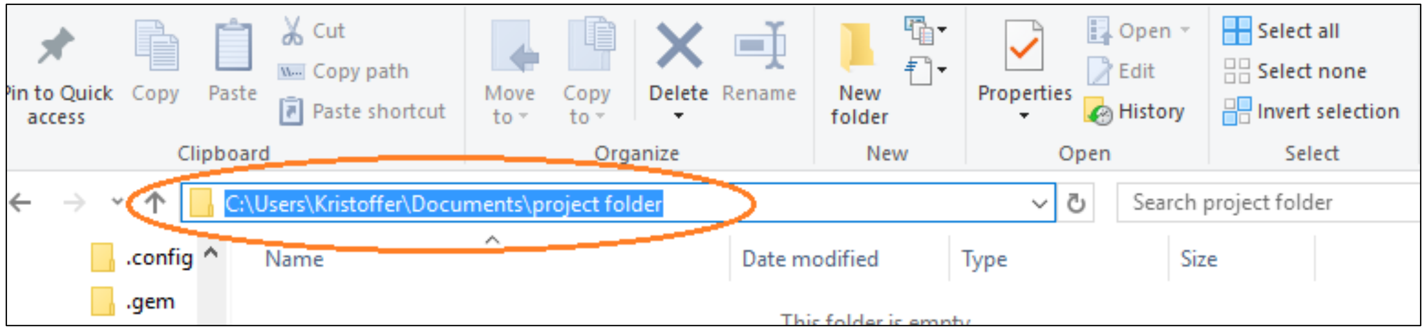
- Sounds complicated to set up? Don’t worry, we have a command that sets that up for you

- It is called *iefolder* and it creates the folders and master do-files we have spoken about for you

- Part of *ietoolkit*. If you have not installed that package yet, then type *ssc install ietoolkit* in Stata

Task 1.

1. Create a folder on your computer and call it the project folder
2. Create a global called *projectfolder* to that folder. A shortcut to get the folder path is by clicking in the navigation bar of that folder



C:\Users\Kristoffer\Documents\project folder

3. Now create a new data work folder using *iefolder* like this:

```
iefolder new project, projectfolder(${projectfolder})
```

iefolder - rounds

- Inside the data work folder you create folders for each round. A round is a data collection exercise
  - Example of round is baseline, endline etc.
  - If you have simultaneous data collections on different unit of observations, for example students and schools, then create separate rounds for them

Task 2.

- Now we want to create a round called *baseline*. We can use the same global *projectfolder.*

1. create the data work folder inside that folder using *iefolder* like this:

```
iefolder new round baseline, projectfolder(${projectfolder})
```

2. Did you do this in a do-file? What happens if you run the same code again?

Task 3.

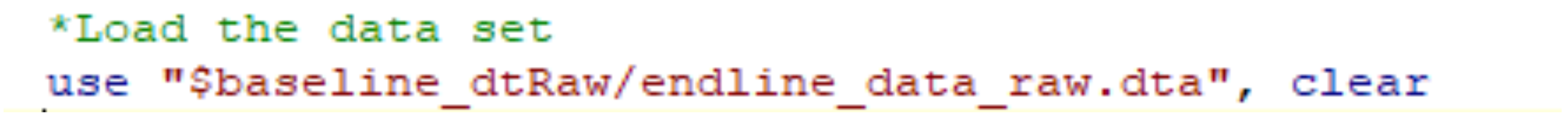
- Now we want to use the folders and the master do-files created by iefolder.

1. Take the data set ***endline_data_raw.dta*** and move it to the *intermediate* folder inside the *DataSets* folder in the round you just created.

2. Run the *Project_MasterDofile.do file.* This sets up the globals in this file. You will have to do this each time you close and open Stata

3. Then run the *baseline_MasterDofile.do* file. You need to run these master do-files in this order as the globals in second do-file depends on the one in the first do-file.

4. Open up a new do-file. In this do-file, open up the data set ***endline_data_raw.dta*** from the intermediate folder using the global *baseline_dtInt* that you generated when you ran *baseline_MasterDofile.do. Like this:*



```
*Load the data set
use "$baseline_dtRaw/endline_data_raw.dta", clear
```

Task 4.

1. Now save the do-file you used in the last task with the name *load_imported_data.do* in this folder: *baseline\Dofiles\Cleaning*

2. Now open the *baseline_cleaning_MasterDofile.do* file in the dofiles folder. Look for the part that says and replace the parts in red:
   - *****do "$baseline_doCln/dofile1.do“
   - *do "$baseline_doCln/load_imported_data.do“*

3. Now run the *baseline_cleaning_MasterDofile.do* file again and see how the data set is opened again

4. Now save all your files and close Stata. Re-open stata and open the do-files *Project_MasterDofile.do* and *baseline_MasterDofile.do*. Run both files in that order and notice that your data set is opened again.

Keep using the folder you created

- If you want to, through the rest of this session you can create a do-file for each task, and run them from the round master do-file.

- That would be similar to how we want you to run a DIME project. More on this in track 2.

- A do-file is instructions for a task, and this system of folders and master do-files is how these task instructions becomes instructions for a full project

Outline

1. Folder organization
2. Master do-files
3. iefolder
4. First overview of a data set
5. Master data sets
6. Basic data cleaning tasks:
    - Unique id,
    - Variable and value labels
    - Missing values
    - Check for outliers

Explore a new data set

- What is the first thing you want to look for every single time you open a new data set for the first time?

  1. Unit of observation
  2. Uniquely and fully identifying ID variable

Explore a raw data set

- Household_data.csv

HHID	Village	District	HH number	HH head	HHH Age
022501	25	2	1	Andrew	52
022502	25	2	2	Patrick	48
023207	32	2	7	Charles	29
023205	32	2	5	Jeffrey	37
012501	25	1	1	Walter	48
011103	11	1	3	Anne	26
011205	12	1	5	Lawrence	61
024502	45	2	2	Dennis	45
024501	45	2	1	Nancy	41

Explore a raw data set

- Clinic_data.csv

Clinic ID	Clinic Number	District	Patient	Age
02452	542	2	Andrew	52
02543	543	2	Patrick	48
02156	156	2	Charles	29
01152	152	1	Jeffrey	37
01152	152	1	Walter	49
01238	238	1	Anne	26
01122	122	1	Lawrence	61
02122	122	2	Dennis	45
02122	122	2	Nancy	41

First thing to check

Unit of observation check list when opening a new data set:

1. What does each row represent?
2. Which variable do you think is the main ID uniquely identifying each row?
3. Test that the variable indeed is uniquely and fully identifying
   - Stata commands: *isid* and *codebook*

Task 5

- Re-run the baseline master do-file that you created in the last task. Make sure that after it runs you have ***endline_data_raw.dta*** open.

1. What is the unit of observation, i.e. what does each row represents?
2. Which variable do you think is the variable that uniquely and fully identifies the data set?
3. Test the variable to make sure that is the unique ID variable.

Outline

What is a Master data sets?

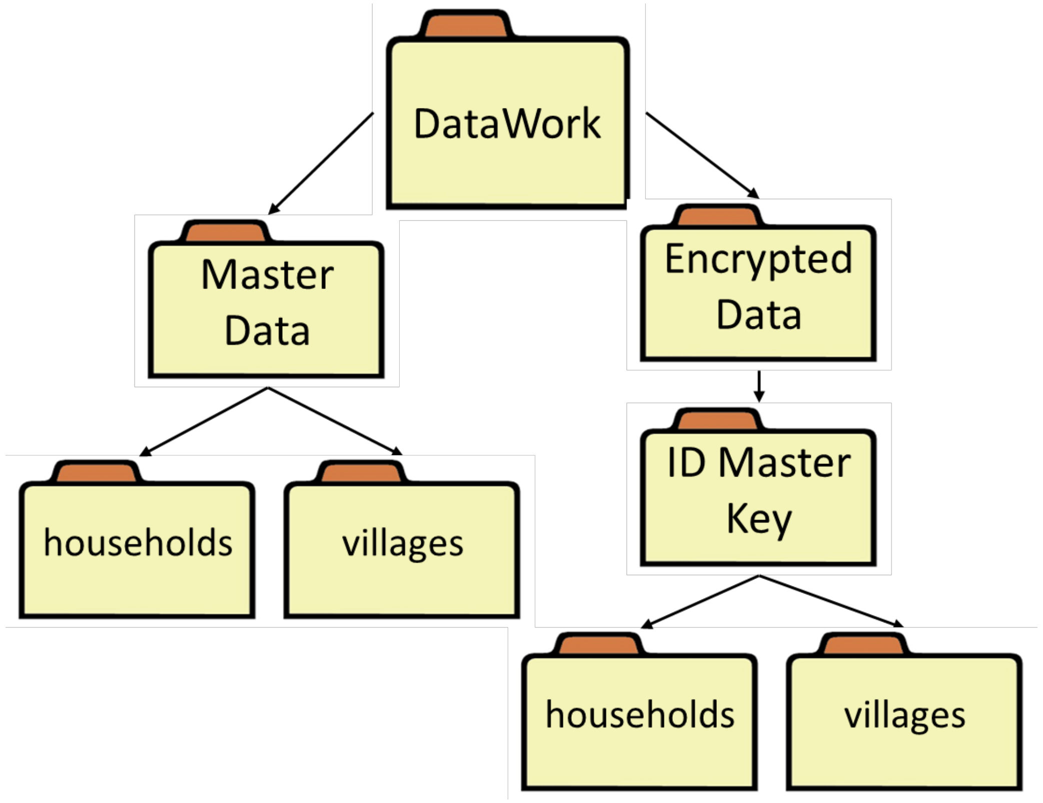
- With multiple data set you need to have a tool to keep track of your observations across different data collections and over time. You guessed it – that tool is master data sets!

- A master data set should include all observations the project has ever encountered. Not just observations assigned to treatment or sampled for data collection.

- A master data set should include information relevant across data set that is constant across the project
  - Uniquely identifying ID
  - Treatment assignment
  - Sampling
  - Monitor status

Data Management
Master Data

- For each unit of observation you have one encrypted master data set, and one available to all
- The encrypted has unique ID and identifying information
- The other one should not include any identifying information



Outline

Role division in cleaning the data

- Field Coordinator / Research Assistants
  - No one will look at the data as much as the FC/RA
  - Irregularities in the data that the FC/RA does not identify will often never be discovered
- Principle Investigators
  - In charge of deciding which irregularities will be corrected and how
    - Never make any decision on data reformatting without talking to PIs
  - PIs completely depend on FCs/RAs to identify irregularities and get the information to make the best call
    - Of course, this all comes up when the PI starts writing up the analysis
      - E.g. we sampled 4,567 HHs and we only have 3,897 in the regressions …

Data Cleaning Tasks

Formatting the data set	Edit the data set
• Make the data easier to work with for yourself, your team and other researchers • Document the knowledge you have about the data now. You will soon forget	• Remove data points that will bias the analysis or in any other way make the analysis invalid
1. Renaming variables 2. Remove strings 3. Variable and value labels	1. Standardize units 2. Missing values 3. Check for outliers

Naming conventions

- Do not change the names of variables coming from a survey
- Keep all variables collected with survey data the same as in the questionnaire. Even if you do not like the naming conventions used in the questionnaire
- Exception: In a loop over, for example, crops where a variable like harvest is asked for each crop and automatically named, harvest_1, harvest_2 etc., then those can be renamed to harvest_crop1, harvest_crop2 etc., or harvest_c1, harvest_c2 etc.

No string variables in the final data

- Exceptions:
  - Proper nouns that are <u>not</u> categories
  - Digits with leading zeros or long IDs (over 15 digits)
- Categorical string variables <u>must</u> be made numeric with value labels
  - See *encode*
  - Best practice using encode includes using both the *label()* and the *noextend* option.

Variable and value labels

We already introduced labels in the first lab. When cleaning a data set you should also make sure that all variables are properly labeled.

1. Check all variables have variable labels (in English)
2. Check all that all categorical variables have value labels (in English)

Useful commands: *label variable , label define, label value, label dir, label list*

Task 6

- The district variable pl_id_09 is a string variable. We need to make this variable numeric. Since it is a categorical variable we should end up with a numeric variable with value labels. Stata has a command called *encode* which do this for us.
  - *encode pl_id_09, gen(pl_id_09_code)*

- See the result. Stata has assigned the codes 1-4 in alphabetic order. What if we already had the following codes? 44 for Burera, 41 for Gicumbi, 54 for Kayonza, 41 for Rulindo, 51 for Rwamagana. Replace your code with this:
  - label define dist 44 "Burera" 41 "Gicumbi" 54 "Kayonza" 41 "Rulindo" 51 "Rwamagana"
  - *encode pl_id_09, gen(pl_id_09_code) label(dist)*

- *Move the new variable next to the old variable. Add this after your code*
  - *order pl_id_09_code, after(pl_id_09)*

Standardize units

- Quantity variables should be converted into standardized units. One unit for weight (usually kg), length/distance (usually meter), etc.

- Often a difficult task due to usage of local non-standardized units. Use field team to collect as much information about local units and then decide on a conversion rate together with your PI

Missing values / Survey codes

- During surveys missing answers are recorded using different codes. Like -88 for "I do not know" or -999 for "Declined to answer"
- We need to turn these survey codes to missing values since -88 or -999 will distort averages and other calculations
- Use any value of .a, .b etc. to represent "I do not know", another one for "Declined to answer", and another for each other missing answer code you have
- Be consistent across your data set. If you pick .a for "Do not know" then use .a to represent that in all variables in your data set
- Use the command *summarize* is efficient to find codes for missing answer as they stand out as negative values (that is why we use negative values as missing answer codes)

Task 7

Variable *ag_08_1_1* is a dummy variable where 1 means that a household used compost, and 0 that they didn't. We want to know the percentage of farmers that used compost. (Tip! The mean of a dummy is the percentage in decimals)

1. Summarize *ag_08_1_1* to get the mean. Is the value between 1 and 0 like you expect a percentage in decimal form to be?
   – Why not? Look at min and max

2. We need to replace the survey code -99 with a missing value. Let's use .a.
   – replace ag_08_1_1 = .a if ag_08_1_1 == -99

3. Summarize the variable again. This looks like we expected.
   – Look what happened to the number of observations. Why did that happen?

Outliers - identifying

- We do not want our results to be driven by a few individuals. For example, the village leaders get all benefits
- No exact rule for what is an outlier. Ask if your PI for preference of specific rule
- Identifying outliers often comes down to common sense.
- Can the outlier be explained by typos? Especially common when selecting units from multiple choice lists

Useful commands: *sum detail, tabulate, inspect, assert, histogram*

Outliers

How to deal with outliers?

- This is ultimately the decision of the researcher.
- The goal is to deal with them while losing as little information as possible.
  - Delete observations
    - Almost never desired as it loses a lot of information in variables that were correct
  - Delete specific data point (i.e. replace to missing)
    - Frequently used but the observation would be dropped in regressions using that variable
  - Winsorization is a method were observations are not dropped, neither from the data set or from regressions

Outliers - Winsorization

- This method assumes that an upper tail outlier is still some large number
- We set all values larger than the 99th percentile to the 99th percentile
- The outliers will still be a large number, but it will no longer greatly bias the estimators
- Can be applied in both upper and lower tail
- Can be set to other limits than the 99th percentile

Task 8

Variables aa_02_1, aa_02_2, aa_02_3, aa_02_4, aa_02_5, aa_02_6, aa_02_7, aa_02_8, aa_02_9, and aa_02_10 lists purchases of common animals. Let's check for outliers

1. Start by using summarize. The two lines of code below are gives you the same result. The question mark tells Stata to include all variables that start on AA_02_ plus exactly one more character
   - summarize aa_02_1 aa_02_2 aa_02_3 aa_02_4 aa_02_5 aa_02_6 aa_02_7 aa_02_8 aa_02_9 aa_02_10, detail
   - summarize aa_02_?, detail

2. Do you see any implausible values? What can we do to fix this? There is no exact answer what to do as these are obvious typos. Discuss what action to take with your PI.

Task 9

- Winsorize the income variable
  - Winsorizing works better at large estimated values like, income, harvest value, land area etc.
  - Winsorizing works less great on small exactly counted number like number of visit at a clinic, animals owned etc. and is never a good idea on a dummy

1. Summrize the **inc_01** variable. Use detail option. Or do a histogram (we will speak about that tomorrow)
2. Winsorize **inc_01** to the 95th percentile. Like this:
   - *ssc install winsor*
   - *winsor inc_01, gen(inc_01_w) high p(.05)*
   - The option *high* means we are only winsorizing the values that are too large. We are not concerned about values close to 0, and a variable like revenue cannot be negative.
3. Compare winsorized vs. non-winsorized means

Saving files – naming conventions

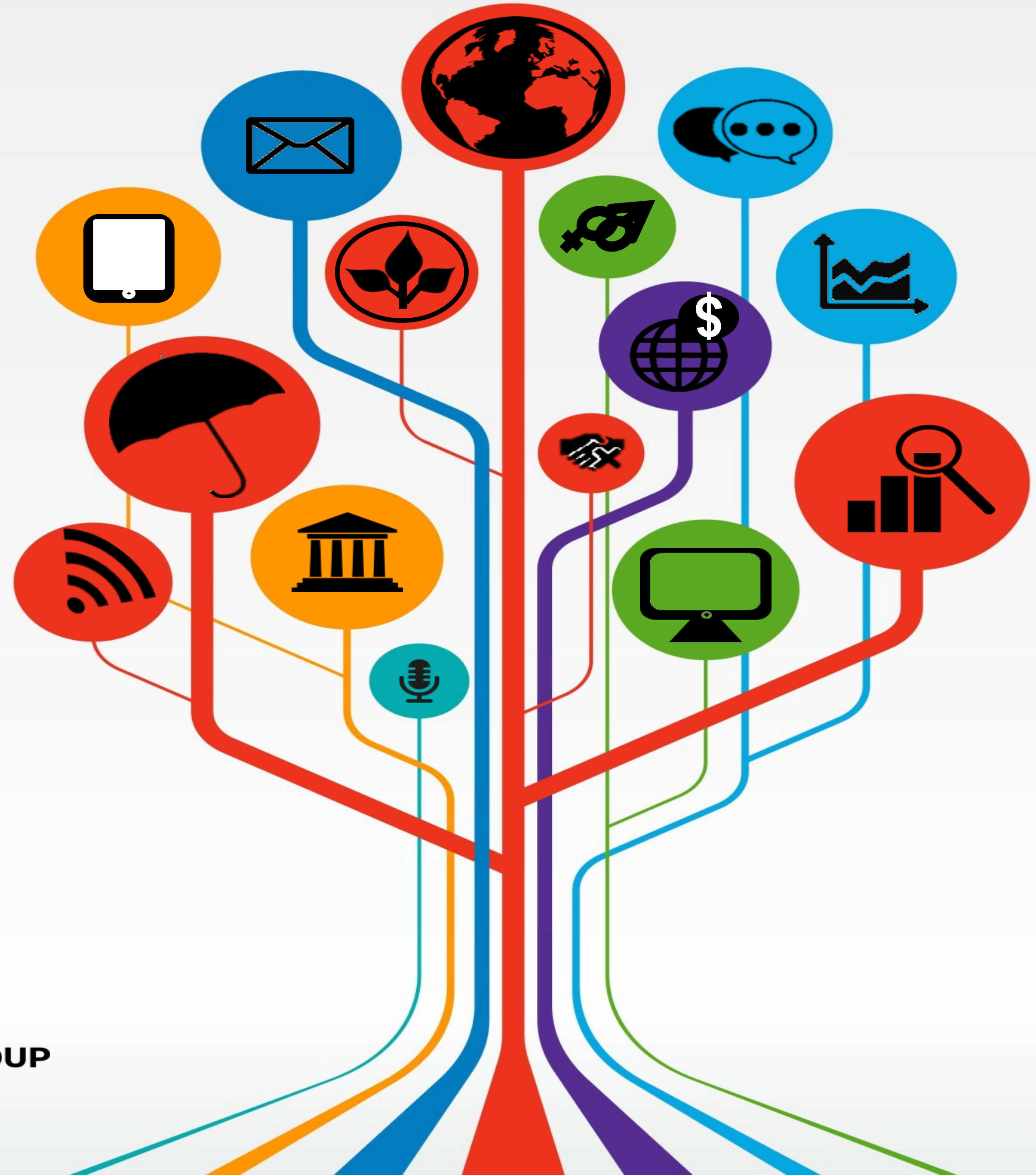
- Static file names
  - Do not use _v1, _v2 etc. for any final files. This leads to bugs in do-files that depend on these files when a new versions is added.
  - OK to use _v1, _v2 etc. for old versions of files
- Make sure all output files, datasets and others are clearly and uniquely labeled

  i.e.: "desc_stats_tmt_only.xls"
  "input_plan_adm_data.dta"

Thank you!

Michael Orevba & Kristoffer Bjärkefur

19 June 2018



i2i DIME TRANSFORM DEVELOPMENT

WORLD BANK GROUP