# Tidy Data

## 5.2 Overview

Unit Objectives

1. Use data-gathering makefiles to organize the data-gathering process so that the process can be completely reproduced.
2. Apply proper documentation techniques for variables and code.
3. Load data from various sources, both locally from the computer and remotely from the Internet.
4. Define tidy data.
5. Use capabilities in R to create tidy data

Note: Please see the Assignments and Grading Unit for this week's homework

Required Readings

Gandrud, C. (2015), Reproducible Research With R and R-Studio. Boca Ration, FL: CRC Press, chapter 6.

Wickham, H. (2014), "Tidy Data." Journal of Statistical Software, 59, 10.

## 5.3 Importing Locally Stored Data Sets

* Read.csv
* Working directory vs providing explicit fully qualified path

## 5.4 Importing Data from Secure and Non-secure URL’s

* Secure and Unsecure wedsites
* “./” tells OS, ‘current working directory’
* List.files()

## 5.5 Regular Expressions in R

* Regular Expressions typically used with (package = regex):
  + Grep
  + Grepl
  + Sub
  + Gsub
* Combination of literals and metacharacters
  + grep(“string\_being\_searched”, file\_being\_searched)
    - grep(“nuclear”, test\_file)
      * will only locate lower case
      * literal string, nuclear
    - grep(“[Nn]uclear”, test\_file)
      * will locate upper and lower
      * The “[]” are metacharacters
      * “^”, beginning of a line
        + grep(“^I”, test\_file)

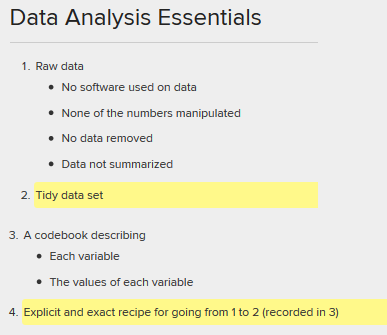
The line must begin with a capitalized “I”

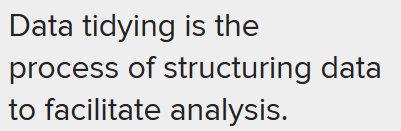
* + - * “$”, end of the line
        + grep(“morning$”, test\_file)

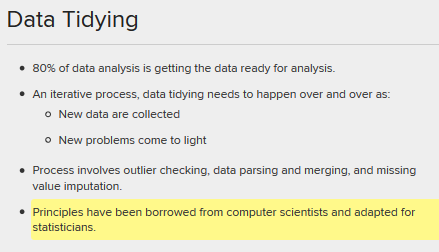
The line must end with, in this case, “morning”

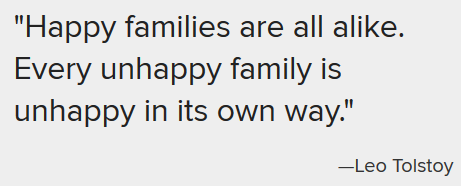
* + - grep(“[0-9]”, test\_file)
      * line must contain any of the metacharacter enclosed values, in this case 0 – 9, at the beginning of the line
    - grep(“9.11”, test\_file)
      * The “.” Is a wildcard, and will get any combination of characters between nine and eleven.
        + 9.11, 9-11, 9/11, 9\_11, 9:11, 9)11, 9(11
    - grep(“flood|fire”, test\_file)
      * The pipe “|” is the OR statement
      * Looking for lines that have flood OR fire, but not both
    - grep(“flood|[Ff]ire|jump|hilltop”, test\_file)
      * Multiple OR search
    - Wildcards
      * “\*”
        + Any number, including none, of the item
      * “+”
        + At least one of the item
      * grep(“[0-9]+ (.\*)[0-9]”, test\_file)

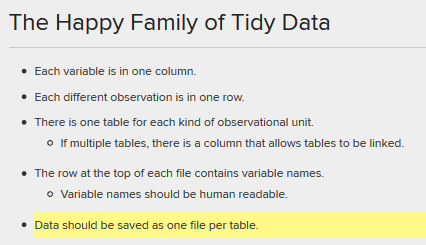
## 5.6 Principles of Tidy Data

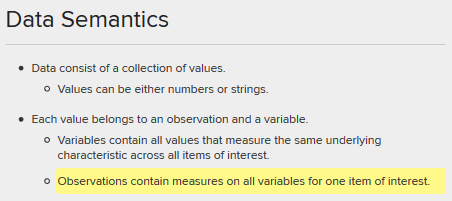


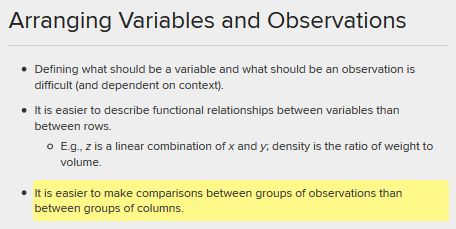


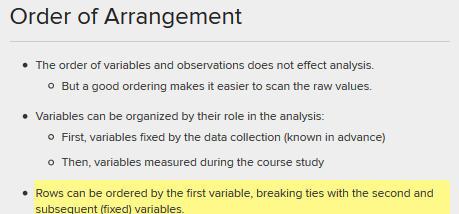


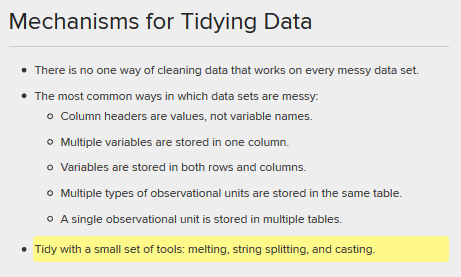


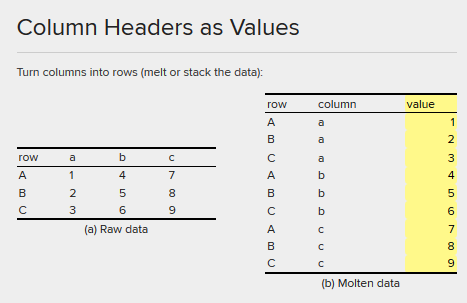










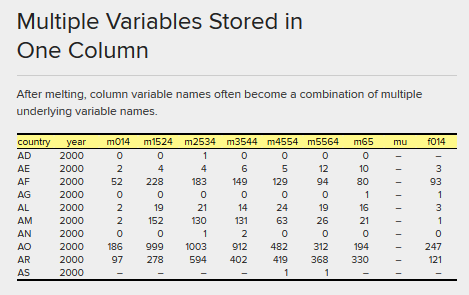


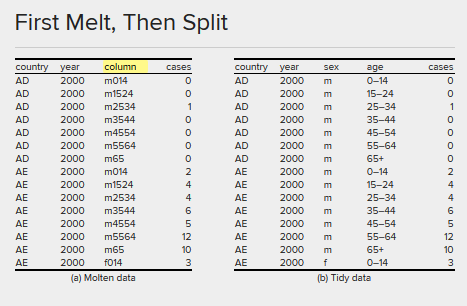
### Multiple Variables Stored in One column

#### Example 1

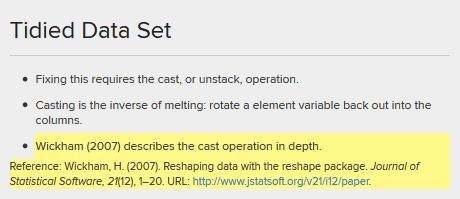
Gender and age in a single column:

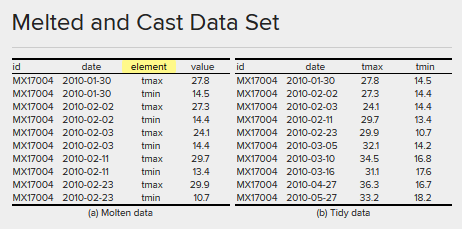
m1524: Male with Age between 15 and 24



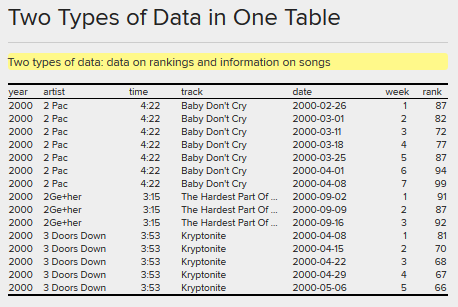


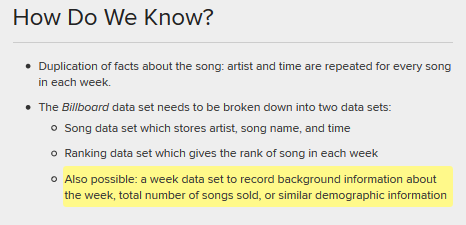
#### Example 2

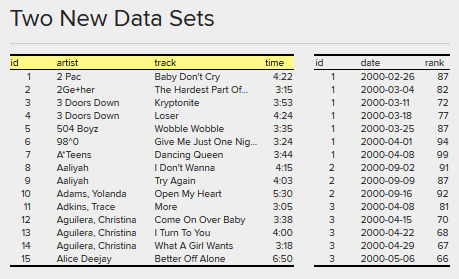


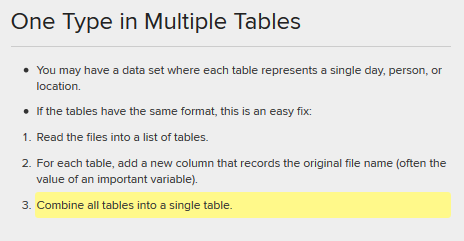


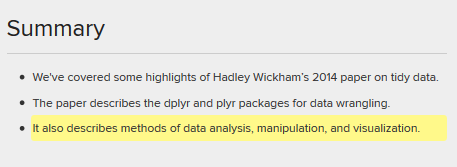
#### Example 3



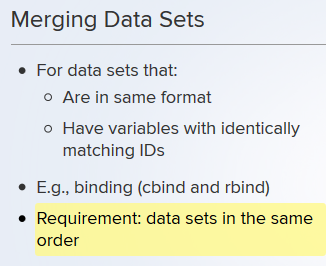


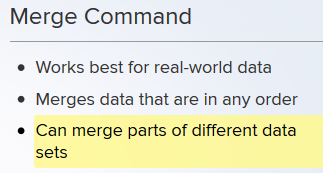


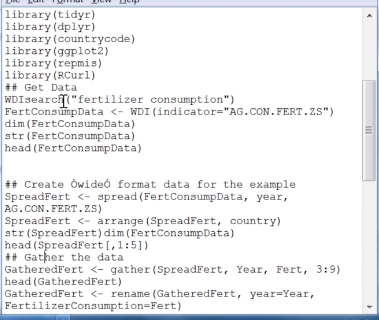


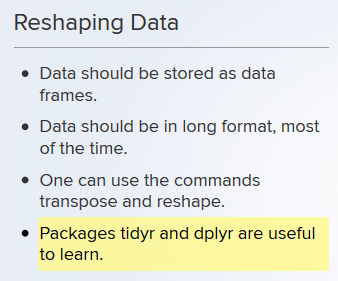


## 5.7 Cleaning Data for Merging



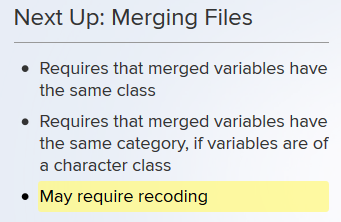






Look to R, Unit\_5\_BLT

## 5.8 Recoding Variables



If merging two datasets based upon a field, say country, all the countries within each dataset need to be the same. Meaning, one dataset can have countries that the other does not and vice versa, but the countries that are in common, spelling must be identical.

### Merging Example

To align spellings, adjust one dataset:





### Transform Variables – Create Variable (Column)

Create a variable that is the log of Fertilizer Consumption



Viewing the data reveals neg infinity:



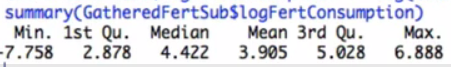
Since it function was log(), this means fertilizer consumption was zero for a country or multiple countries. To fix this, put a small number into the places where fertilizer consumption is zero:



Rerun the log:



View the data again and neg inf is gone:

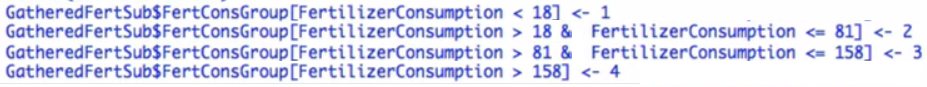


### Creating Factor Variables – ATTACH

Creating a factor variable, consisting of “Low”, “Medium”, and “High”

Performing the attach function inserts the data frame into the search list and can refer to the columns by name without mentioning the data frame:

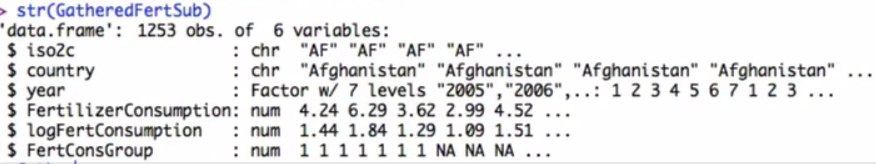




Be sure to evaluate the groups by asking for summary() against the field that was created.

### Creating Variable Labels

#### Factor



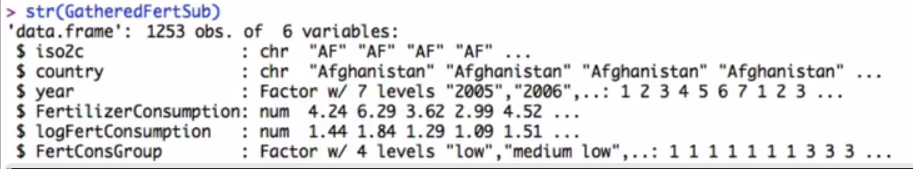
FertConsGroup is currently a numeric, we want to change it to a factor

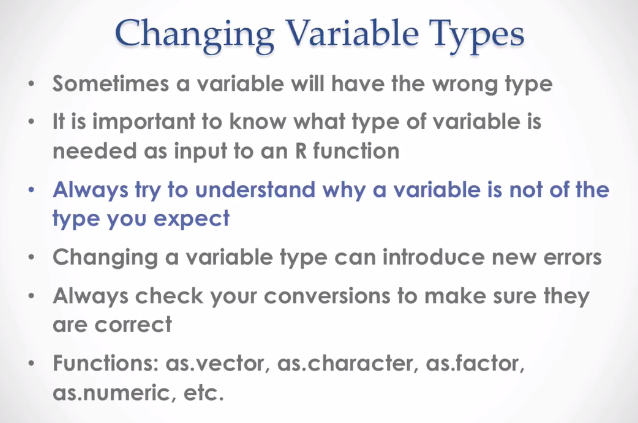




#### Cut







### Variable Coercion

* as.numeric()
* as.character()
* as.factor()
* as.vector()

## 5.9 Merging Data

Objective: merge separate data files

Data file 1:



Data file 2 (requires “RCurl” package):

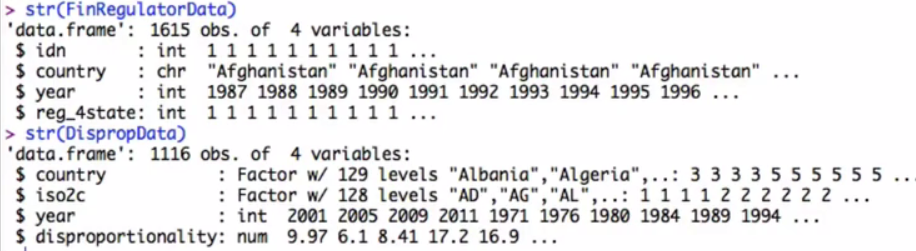


Data file 3:

Fertilizer Consumption Data

Method of Merging:

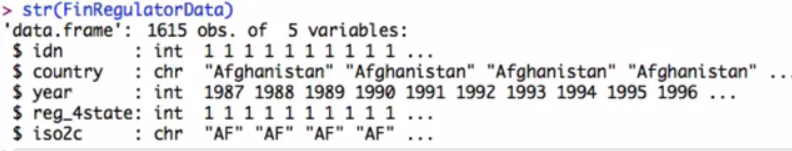
Merge two, clean them up, then merge the third and clean them up. Next, understand the data:



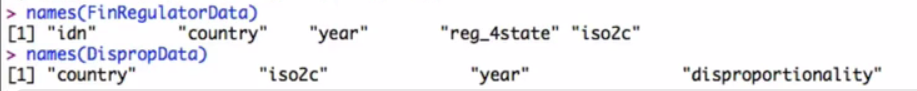
The objective is merge on “CountryCode”, both have “Country” but the “FinRegulatorData” does not have a column called “iso2c” like the “DispropData” does. The countrycode() will be used to create the “iso2c” based upon the “Country” field in the “FinRegulatorData”:



Verify it work as expected:

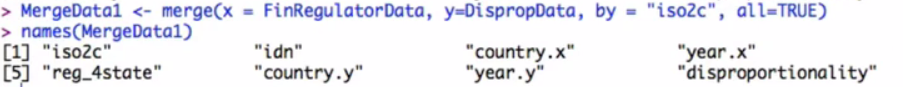


An inspection of the column names of the two datasets:

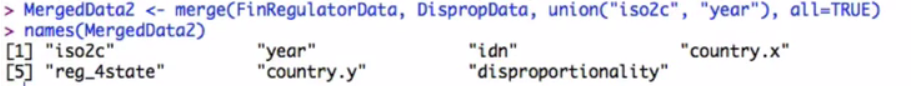


#### Merge

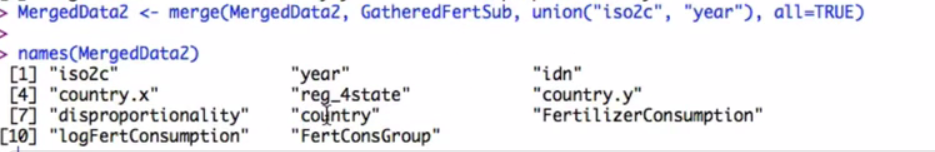
Merge the two datasets based upon the “iso2c” field. The merge process will create a new dataset and will create new columns within the dataset that are common between the two merged datasources, (year, and country) and will be the column name will be subsetted with its source (country.y and country.x):



To merge on multiple columns, use UNION(column\_name1, column\_name2):



Merge the third dataset with MergedData2:



#### duplicated

Getting namesMergedData2) reveals duplicate columns, such as country (country.x, country.y, country, and idn). To understand which columns are duplicated, MergedData2(duplicated(MergedData2[,1:2]),] gets only the dup’ed columns:



This time get only the columns that are NOT duplicated variables:



#### Dyplr::SELECT

Now…

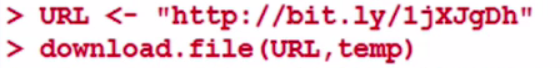


## Week 5 Lecture Assignment

### Importing Compressed Data – Unsecure (http)

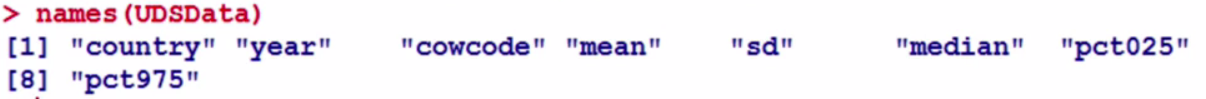
Download and uncompress the Zip or Tar file (Tarball):

#### Download.file()



#### Read.csv()





#### Unlink()

Release the memory address holding the data:

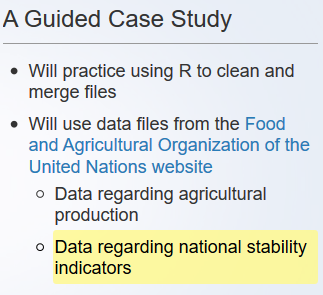


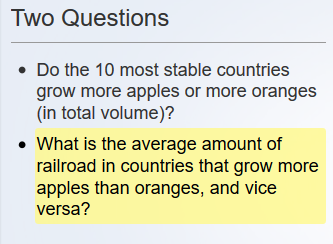
### Importing Compressed Data – Secure (https)

#### RCurl

### Merging Apples and Oranges

<http://faostat3.fao.org>





Refer to “UNIT\_5\_LECTURE\_ASSIGNMENT\_2\_R” for details

## R Functions

* readLines()
* grep(“string\_being\_searched”, file\_being\_searched)
* dev.off()
  + turns of the graphic (GRAPH) device

## Class Notes