ManyLabs2 Data Cleaning

ManyLabs2 (Corresponding coder: Fred Hasselman)

24 October 2015

Contents

About this document

This document notes any general and site-specific exclusions or changes of values together with the rationale for doing so (for instance, changes due to a typo or mistranslation that was noticed and corrected for the script deployed at a particular site).

These changes are implemented as specific filters, but affect only a small proportion of the overall participants. The changes are recorded at each step and are reported at the end of this document.

Overall exclusion criteria that may not be noted elsewhere

Exclude responses that were incomplete

- Cases will be removed by applying a filter which selects only cases for which Finished == 1 is true, where Finished is a variable present in each raw data set.

Implementation: The function get.CSVdata(... , finishedOnly = TRUE) merges each raw data file (comma seperated files available here) into a data file per slate. It removes any incomplete cases as described above if finishedOnly = TRUE.

```
# MERGE RAW DATA ------------------------------------------------------------

# Set the working directory to where the raw data files are...
dataDir  <- '~/Dropbox/Manylabs2/Raw Data'
fileList <- list.files(dataDir,".csv")

# Merge Slate 1 & remove incomplete cases
ML2.files.S1 <- fileList[grepl("Slate[_]*1", fileList)]
names(ML2.files.S1) <- ML2.files.S1
ML2.S1 <- tbl_df(ldply(.data = ML2.files.S1, .fun = get.CSVdata, path = dataDir, .inform = T))

# Merge Slate 2 & remove incomplete cases
ML2.files.S2 <- fileList[grepl("Slate[_]*2", fileList)]
names(ML2.files.S2) <- ML2.files.S2
ML2.S2 <- tbl_df(ldply(.data = ML2.files.S2, .fun = get.CSVdata, path = dataDir, .inform = T))

# Start counting cases ans labels at each processing step
CaseCount("1. Merged complete cases (Finished == 1)")
```

Exclude responses of -99

Values of '-99' should be ignored, they indicate some form of "do not wish to respond" or "not applicable" or "other".

- These values will be set to NA

Exclude responses indicating a test run

As indicated by the word test entered into one of the text fields and a list of known test runs by ResponseID.

- These cases will be marked for removal by setting variable Finished to 0
- Cases will be removed by applying a filter which selects only cases that have the value Finished== 1

Implementation: The function clean.ML2fieldsNA() will perform the changes and report wether values were found.

```
# Remove any test trials and '-99'
ML2.S1 <- clean.ML2fieldsNA(ML2.S1)


   ~~~~~~Clean ML2 Test Data - Step 1~~~~~~~
   § Marking known test sessions for removal

   ~~~Clean ML2 Test Data - Step 2~~~~
   § Checking columns:
       age
```

```
    §    sex
    §    hometown
    §    education
    §    comments
    §    raceother
    §    cogref.1
    §    cogref.2
    §    cogref.3
    §    crit1.1_1_TEXT
    §    crit1.1_2_TEXT
    §    crit1.1_3_TEXT
    §    ross.s1.1_1_TEXT
    §    ross.s1.1_2_TEXT
    §    kay1.1
    §    kay1.2_7_TEXT
    §    kay1.2_8_TEXT
    §    kay1.2_9_TEXT
    §    kay1.3
    §    kay2.1
    §    kay2.2_7_TEXT
    §    kay2.2_8_TEXT
    §    kay2.2_9_TEXT
    §    kay2.3
    §    and1.1
    §    and2.1
    § for a variant of pattern: 'test'

    ~~~~~~~~~~~~~~~~~~~~~~Clean Test ML2 Data - Step 3~~~~~~~~~~~~~~~~~~~~~~
    § Checking all columns except "LocationLongitude" for pattern: "-99"
    ~~~~~~~
    § Done!
    ~~~~~~~
ML2.S2 <- clean.ML2fieldsNA(ML2.S2, S1 = FALSE)


    ~~~~~~~Clean ML2 Test Data - Step 1~~~~~~~
    § Marking known test sessions for removal

    ~~~Clean ML2 Test Data - Step 2~~~~
    § Checking columns:
        age
    §    sex
    §    hometown
    §    education
    §    comments
    §    raceother
    §    cogref.1
    §    cogref.2
```

- cogref.3
- ross.s2.1_1_TEXT
- ross.s2.1_2_TEXT
- sava1.2
- sava1.3
- sava1.7
- sava1.8
- sava1.12
- sava1.13
- sava1.18
- sava1.19
- sava1.24
- sava1.25
- sava1.30
- sava1.31
- sava1.36
- sava1.37
- sava1.41
- sava1.42
- sava2.2
- sava2.3
- sava2.7
- sava2.8
- sava2.12
- sava2.13
- sava2.18
- sava2.19
- sava2.24
- sava2.25
- sava2.30
- sava2.31
- sava2.36
- sava2.37
- sava2.41
- sava2.42
- zhon1.1
- zhon2.1
- zav.int.1
- zav1.1
- zav1.2
- zav1.3
- zav1.4
- zav1.5
- zav1.6
- zav1.7
- zav1.8
- zav1.9
- zav1.10

```
§    zav1.11
§    zav1.12
§    zav1.13
§    zav2.1
§    zav2.2
§    zav2.3
§    zav2.4
§    zav2.5
§    zav2.6
§    zav2.7
§    zav2.8
§    zav2.9
§    zav2.10
§    zav2.11
§    zav2.12
§    zav2.13
§ for a variant of pattern: 'test'

~~~~~~~~~~~~~~~~~~~~~~Clean Test ML2 Data - Step 3~~~~~~~~~~~~~~~~~~~~~~
§ Checking all columns except "LocationLongitude" for pattern: "-99"
~~~~~~~
§ Done!
~~~~~~~

CaseCount("2. Removed test and -99")
```

Changes to the *source* variable that identifies the location of data collection

The changes to sourcelabels are conducted by the function clean.Source(), which reads information from a Google Sheet which is a simple lookup table listing all the observed variations of source labels (e.g., due to character encoding or typo's) and the accompanying correct version of the label.

```
# Clean Source labels
ML2.S1$source <- clean.Source(ML2.S1$source, SourceTable)$source.clean
ML2.S2$source <- clean.Source(ML2.S2$source, SourceTable)$source.clean

CaseCount("3.1 Cleaned Source labels (first)")
```

The resulting source labels are compared to a lookup table which lists the R code that should be run to implement the site specific changes.

```
# Apply additional source rules
ML2.S1$source <- as.character(ML2.S1$source)
ML2.S2$source <- as.character(ML2.S2$source)
```

```
for (f in seq_along(FileNameTable$File.name))
{
   if (FileNameTable$Change.Source.ID[[f]] != "")
   {
      ID <- eval(parse(text = paste(FileNameTable$Change.Source.ID[[f]])))
      eval(parse(text = paste(FileNameTable$Change.Source[[f]])))
   }
}

CaseCount("3.2 Clean Source labels (second)")
```

What follows is a listing of the changes along with the rationale. The R code reflects the

Change source = grazvienna

If the data comes from any of the following files source should be grazvienna

- Slate_1_Deutsch__Austria_Revised_Version_2__Kopieren_teamb_r_manuallyrecode_vanp21_text.csv
- Slate_1_Deutsch__Austria_Revised_Version_2__Kopieren_teamc_r_manuallyrecode_vanp21_text.csv
- Slate_1_Deutsch__Austria_Revised_Version_2_fady_r_manuallyrecode_vanp21_text.csv
- Slate_1_Deutsch__Austria_Revised_Version_2_r_manuallyrecode_vanp21_text.csv
- Slate_2_Deutsch__Austria_Revised_Version_2__Kopieren_practikum_r.csv
- Slate_2_Deutsch__Austria_Revised_Version_2__Kopieren_teame_r.csv
- Slate_2_Deutsch__Austria_Revised_Version_2_r.csv

Explanation: The source identifier was omitted from the links to these surveys, but only one site used these survey versions so we can be certain all data in them comes from there.

```
# ID = 76-84 (Example = 76)
FileNameTable$Change.Source.ID[76]
```

```
[1] "(ML2.S1$.id=='Slate_1_Deutsch__Austria_Revised_Version_2__Kopieren_teamb_r_manuallyrecode_vanp21_text.csv')"
```

```
FileNameTable$Change.Source[76]
```

```
[1] "ML2.S1$source[ID] <- 'grazvienna'"
```

Change source = cas

- If data comes from file: ML2_Slate2_Simplified_Chinese_Mainland_r_manually_recode_rosss21.csv
- **AND** the date is between December 8th, 2014 and Dec 18th, 2014.

Explanation: The source identifier was omitted from the link for slate 2 from the “cas” site, but they were the only team running that survey during this time period so we can be sure all data between those dates comes from that site.

Implementation:

```
# ID = 56
FileNameTable$Change.Source.ID[56]
[1] " (ML2.S2$.id=='ML2_Slate2_Simplified_Chinese_Mainland_r_manually_recode_rosss21.csv')&(ML2.S2$EndDate>='12/08/2014')&(
FileNameTable$Change.Source[56]
[1] "ML2.S2$source[ID] <- 'cas'"
```

Change source = moralsense.

- If data comes from this file: ML2_Slate2_USEng_Inlab_DEPLOY_nunziato_r.csv

Explanation: The source identifier was omitted from the link, but this was a custom link for the Moral Sense website so we know all data comes from that sample.

Implementation:

```
# ID = 73
FileNameTable$Change.Source.ID[73]
[1] "(ML2.S2$.id=='ML2_Slate2_USEng_Inlab_DEPLOY_nunziato_r.csv')"
FileNameTable$Change.Source[73]
[1] "ML2.S2$source[ID] <- 'moralsense'"
```

Change source = occid

If source = occid OR occidtab, then recode source variable as follows:

- If meta_4_TEXT = 1280x720, source = occidtab.
- If meta_4_TEXT = , source = occid.

Explanation: Site ran both tablet and PC sessions. Usually, these are identified by using different links for each device, but in this case the links were mixed up. Instead, we determine whether the participant used a tablet or PC based on the resolution used (all tablet sessions were run in 1280x720 resolution, whereas PC sessions varied between other values).

Implementation:

```
# ID = 34
FileNameTable$Change.Source.ID[34]
[1] "(ML2.S1$.id=='ML2_Slate1_UAEEng_Inlab_execution_legal_r.csv')&(ML2.S1$source=='occid'|ML2.S1$source=='occidtab')"
FileNameTable$Change.Source[34]
[1] "ifelse(ML2.S1$meta_1_TEXT[ID]=='MSIE',ML2.S1$source[ID&ML2.S1$meta_1_TEXT=='MSIE']<-'occidtab',ML2.S1$source[ID&M
```

Change source = elte

If data file = ML2_Slate1_Hungarian_Inlab_execution_illegal_DEPLOY_r.csv, recode source variable as follows:

- If meta_1_TEXT = 'MSIE', source = eltetab.

- If meta_1_TEXT = 'Chrome', source = elte.

Explanation: Site ran both tablet and PC sessions. Usually, these are identified by using different links for each device, but in this case the source identifier was sometimes omitted. Instead, we determine whether the participant used a tablet or PC based on the browser used (all tablet sessions were run with Microsoft Internet Explorer; all PC sessions were run with Google Chrome).

Implementation:

```
# ID = 16
FileNameTable$Change.Source.ID[16]
[1] "(ML2.S1$.id=='ML2_Slate1_Hungarian_Inlab_execution_illegal_DEPLOY_r.csv')"
FileNameTable$Change.Source[16]
[1] "ifelse(ML2.S1$meta_1_TEXT[ID]=='MSIE',ML2.S1$source[ID&ML2.S1$meta_1_TEXT=='MSIE']<-'eltetab',ML2.S1$source[ID&ML
```

Change source = mturk

- If data file: 'ML2_Slate1_USEng_mTurk_JC_r.csc **or** ML2_Slate2_USEng_mTurk_JC_r.csv

Explanation: These data for Slate 1 and Slate 2 were collected on MTurk for US participants using a unique link (e.g., no other sites used it). The link was distributed without a built in source identifier.

Implementation:

```
# ID = 38 & 75 (Example = 38)
FileNameTable$Change.Source.ID[38]
[1] "(ML2.S1$.id=='ML2_Slate1_USEng_mTurk_JC_r.csv')"
FileNameTable$Change.Source[38]
[1] "ML2.S1$source[ID] <- 'mturk'"
```

Change source = mturk_india

- If data file = ML2_Slate2_IndiaEng_MTurk_r.csv **or** ML2_Slate1_IndiaEng_execution_legal_MTurk_r.csv

Explanation: These data for Slate 1 and Slate 2 were collected on MTurk for Indian participants using a unique link (e.g., no other sites used it). The link was distributed without a built in source identifier.

Implementation:

```
# ID = 17 & 48 (Example = 17)
FileNameTable$Change.Source.ID[17]
[1] "(ML2.S1$.id=='ML2_Slate1_IndiaEng_execution_legal_MTurk_r.csv')"
FileNameTable$Change.Source[17]
[1] "ML2.S1$source[ID] <- 'mturk_india'"
```

Change source = metu

- If originating file (.id) = ML2_Slate2_Turkish_online_DEPLOY__metu_rmanuallyrenamerosss21_text.csv then source = metu

Explanation: Some sessions did not record the source variable for the "metu" site. We can just re-assign the source variable given that only one site used this study link.

Implementation:

```
# ID = 64
FileNameTable$Change.Source.ID[64]
[1] "(ML2.S2$.id=='ML2_Slate2_Turkish_online_DEPLOY__metu_rmanuallyrenamerosss21_text.csv')"
FileNameTable$Change.Source[64]
[1] "ML2.S2$source[ID] <- 'metu'"
```

Site specific information

Site specific information is added to the cases based on a match between the value of *source* and the information in ML2_SourceInfo.

```
# Add site variables: Language, Population, etc.
ML2.S1 <- get.fieldAdd(ML2.S1, SourceInfoTable)
ML2.S2 <- get.fieldAdd(ML2.S2, SourceInfoTable)

CaseCount("4. Add site variables")
```

Site specific exclusions

Exclusion: All data from source = rio

- All data should be excluded from all analyses (exploratory or otherwise – could effectively remove from dataset if desired; N = 10).

Explanation: This location ran a total N = 10 due to lack of participants, and we've determined it was best to exclude this sample entirely, rather than having figures etc. distracted by such a low-powered test.

Implementation:

```
ML2.S1 <- ML2.S1 %>% filter(source!='rio')

CaseCount("5. rio")
```

Exclusion: Dutch translation of Van Lange

Data from the following files should be excluded from the Van Lange analysis (all other studies unaffected):

- ML2_Slate1_Dutch_execution_illegal_DEPLOY__belgium_r.csv
- ML2_Slate1_Dutch_execution_illegal_DEPLOY__netherlands_r.csv
- ML2_Slate1_Dutch_execution_illegal_DEPLOY__netherlands_tilburgcomm_r.csv

Explanation: The second row from the SVO scale (variables starting with van.p1.2) was accidentally omitted during translation, leaving only 5 of the 6 items that belong on this scale. The data are recoded to reflect the missing row (2). Affects Dutch Slate 1 surveys.

Implentation: All values will be set to NA.

```
ML2.S1[(ML2.S1$.id %in% c('ML2_Slate1_Dutch_execution_illegal_DEPLOY__belgium_r.csv',
                          'ML2_Slate1_Dutch_execution_illegal_DEPLOY__netherlands_r.csv',
                          'ML2_Slate1_Dutch_execution_illegal_DEPLOY__netherlands_tilburgcomm_r.csv')),
       c('van.p1.2_1','van.p1.2_2','van.p1.2_3','van.p1.2_4',
         'van.p1.2_5','van.p1.2_6','van.p2.1_1_TEXT','van.p2.1_2_TEXT')] <- rep(NA,8)
```

Exclusion: Chinese date

- Data collected prior to December 9th, 2014 from file ML2_Slate2_Simplified_Chinese_Mainland_r_manually_recode_rosss21.csv should be excluded from the Slate 2 Ross analysis.

Explanation: On Dec 8th, 2014 12:50pm EST we found and corrected a substantial typo in the Ross Slate 2 text.

Implementation:

```
idRows <- which(
    (ML2.S2$.id == "ML2_Slate2_Simplified_Chinese_Mainland_r_manually_recode_rosss21.csv") &
    (strptime(ML2.S2$EndDate, "%m/%d/%Y") < strptime("12/9/2014", "%m/%d/%Y"))
                )

idCols <- which(grepl("ross.s2", colnames(ML2.S2)))

if(all(length(idRows)>0, length(idCols>0))){
    reNA <- matrix(NA, nrow = length(idRows), ncol = length(idCols))
    ML2.S2[idRows, idCols] <- reNA
}

CaseCount("6. Chinese date")
```

Exclusion: Dutch date

- Data from file: ML2_Slate2_Dutch_Inlab_DEPLOY__netherlands_r.csv collected before November 18th, 2014 should be excluded from *Hsee() analysis

Explanation: Corrected a typo in the scarf condition that mistakenly referred to “coat”.

Implementation:

```
idRows <- which(
    (ML2.S2$.id == "ML2_Slate2_Dutch_Inlab_DEPLOY__netherlands_r.csv") &
    (strptime(ML2.S2$EndDate,"%Y-%m-%d") < strptime("2014-11-18", "%Y-%m-%d"))
            )

idCols <- which(grepl("hsee", colnames(ML2.S2)))

if(all(length(idRows)>0, length(idCols>0))){
    reNA <- matrix(NA, nrow = length(idRows), ncol = length(idCols))
    ML2.S2[idRows, idCols] <- reNA
}

CaseCount("7. Dutch date - Hsee")
```

Exclusion: Uruguay date

- Exclude data from file: ML2_Slate1_Spanish_execution_illegal__Uruguay_r.csv collected before November 13th, 2014 from all Huang analyses.

Explanation: The map graphic was incorrectly implemented and had to be fixed so the coordinates recorded were consistent with measurement at other sites.

Implementation:

```
idRows <- which(
    (ML2.S1$.id == "ML2_Slate1_Spanish_execution_illegal__Uruguay_r.csv") &
    (strptime(ML2.S2$EndDate,"%Y-%m-%d") < strptime("2014-11-13","%Y-%m-%d"))
            )

idCols <- which(grepl("huan", colnames(ML2.S1)))

if(all(length(idRows)>0, length(idCols>0))){
    reNA <- matrix(NA, nrow = length(idRows), ncol = length(idCols))
    ML2.S1[idRows, idCols] <- reNA
}

CaseCount("8. Uruguay date - Huan")
```

Exclusion: French date

- Exclude participants from datafile = ML2_Slate1_French_Inlab_execution_illegal_pencilpaper_r.csv run before November 23rd, 2014 from the Miyamoto analysis.

Explanation: Nov 22, 2014 7:45pm EST changed miyamoto 2.6 from “Sélectionnez le point sur l’échelle suivante qui représente le mieux l’attitude de l’étudiant standard dans votre université.” to “Sélectionnez le point sur l’échelle suivante qui représente le mieux l’attitude de l’étudiant standard dans une université française.” (“votre université” to “une université française”). Miya1.6 already read “une université française”.

Implementation:

```
idRows <- which(
    (ML2.S1$.id == "ML2_Slate1_French_Inlab_execution_illegal_pencilpaper_r.csv") &
    (strptime(ML2.S2$EndDate,"%Y-%m-%d") < strptime("2014-11-23","%Y-%m-%d"))
              )

idCols <- which(grepl("huan", colnames(ML2.S1)))

if(all(length(idRows)>0, length(idCols>0))){
    reNA <- matrix(NA, nrow = length(idRows), ncol = length(idCols))
    ML2.S1[idRows, idCols] <- reNA
}

CaseCount("9. French date - Huan")
```

Exclusion: Mturk USA duplicates

Exclude participants from source = “mturk” if not.mturk.duplicate DOES NOT EQUAL “1” or ip.location DOES NOT EQUAL “USA”. (note: this will have to be incorporated after merging the additional mturk data noted below, and adding the mturk “source” identifier noted above).

Explanation: Ensures participants from the mturk sample took the study only once and are from the USA as desired from this sample.

Implementation:

Get variables from four files (provided privately and not uploaded to OSF due to identifying information). They contain the ID numbers of which it is certain the experiment was completed only once.

```
# Get the IDs that should remain in the data set.
ML2.S1add <- tbl_df(ldply(.data = files.S1, .fun = read.csv, stringsAsFactors = F, .inform = T))
ML2.S2add <- tbl_df(ldply(.data = files.S2, .fun = read.csv, stringsAsFactors = F, .inform = T))

idS1.USA <- (ML2.S1$source == "mturk")
idS2.USA <- (ML2.S2$source == "mturk")

idS1.USA.ip <- ML2.S1add$V1[grepl("USA",ML2.S1add$ip.location)]
idS2.USA.ip <- ML2.S2add$V1[grepl("USA",ML2.S2add$ip.location)]

# Find the cases that should be removed
idS1.USA.remove <- idS1.USA.ip[!(idS1.USA.ip %in% ML2.S1$ResponseID[idS1.USA])]
idS2.USA.remove <- idS2.USA.ip[!(idS2.USA.ip %in% ML2.S2$ResponseID[idS2.USA])]

# Remove them using a filter
```

```
ML2.S1 <- filter(ML2.S1, !(ML2.S1$ResponseID %in% idS1.USA.remove))
ML2.S2 <- filter(ML2.S2, !(ML2.S2$ResponseID %in% idS2.USA.remove))

# Clean up
rm(idS1.USA, idS1.USA.ip, idS1.USA.remove, idS2.USA, idS2.USA.ip, idS2.USA.remove)

CaseCount("10. Filter mturk doubles")
```

Exclusion: Mturk India duplicates

- Exclude participants from source = "mturk_india" if not.mturk.duplicate DOES NOT EQUAL "1" or ip.location DOES NOT EQUAL "India". (note: this will have to be incorporated after merging the additional mturk data noted below, and adding the mturk "source" identifier noted above).

Explanation: Ensures participants from the mturk sample took the study only once and are from India as desired from this sample.

Implementation:

```
# Get the IDs that should remain in the data set.
idS1.India <- (ML2.S1$source == "mturk_india")
idS2.India <- (ML2.S2$source == "mturk_india")

idS1.India.ip <- ML2.S1add$V1[grepl("India",ML2.S1add$ip.location)]
idS2.India.ip <- ML2.S2add$V1[grepl("India",ML2.S2add$ip.location)]

# Find the cases that should be removed
idS1.India.remove <- ML2.S1add$V1[!(idS1.India.ip %in% ML2.S1$ResponseID[idS1.India])]
idS2.India.remove <- ML2.S2add$V1[!(idS2.India.ip %in% ML2.S2$ResponseID[idS2.India])]

# Remove them using a filter
ML2.S1 <- filter(ML2.S1, !(ML2.S1$ResponseID %in% idS1.India.remove))
ML2.S2 <- filter(ML2.S2, !(ML2.S2$ResponseID %in% idS2.India.remove))

# Clean up
rm(idS1.India,idS1.India.ip,idS1.India.remove, idS2.India,idS2.India.ip,idS2.India.remove)

CaseCount("11. Filter mturk_india doubles")
```

Other required changes

Change: Update ID

The following subjects need their Critcher IDs updated. This can be done as follows: (note two variables are involved: "crit1.1_3_TEXT" and "crit2.1_3_TEXT")

Explanation: During testing the critcher IDs manually assigned to participants were incorrectly assigned for a short period, and then corrected. Site leads provided us with experimenter logs to fix the incorrectly assigned

IDs, so that those in the final dataset will now be accurate. (this has no influence on the analysis but keeps the record attaching virtual response -> paper response accurate).

Implementation:

```
ML2.S1$crit1.1_3_TEXT[ML2.S1$ResponseID == "R_0oGc2yQ69dymYlL"] <- NA
ML2.S1$crit1.1_3_TEXT[ML2.S1$ResponseID == "R_1OhWV1L4oLq5UfH"] <- 47
ML2.S1$crit1.1_3_TEXT[ML2.S1$ResponseID == "R_5jcGioO8p9AQyVL"] <- 48
ML2.S1$crit1.1_3_TEXT[ML2.S1$ResponseID == "R_8qRkSy8mRv0Al9D"] <- 49

ML2.S1$crit2.1_3_TEXT[ML2.S1$ResponseID == "R_4VkDXwWluU06qvb"] <- 44
ML2.S1$crit2.1_3_TEXT[ML2.S1$ResponseID == "R_2sGKxwShGfG2tpj"] <- 45
ML2.S1$crit2.1_3_TEXT[ML2.S1$ResponseID == "R_55b6oFggrMj1nNz"] <- 46
ML2.S1$crit2.1_3_TEXT[ML2.S1$ResponseID == "R_1ALpdSyxBcmNwMd"] <- 50

CaseCount("12. IDs")
```

Change: Eindhoventab

- If source = (eindhoven OR eindhoventab) & meta_4_TEXT = 1366x768, then source = eindhoventab.
- If source = (eindhoven OR eindhoventab) & meta_4_TEXT ≠ (NOT equal) 1366x768, then source = eindhoven.

**Explanation:* The Eindhoven site ran participants on both tablet and PC, but must have mixed up the links so that participants were essentially randomly assigned either “eindhoven” or “eindhoventab” as a source identifier, even though the latter should be reserved for only tablet sessions. To correctly identify, we can sort out the tablet sessions because they were the only ones run at 1366x768 resolution.

Implementation:

```
idS1 <- which(ML2.S1$source == "eindhoven" | ML2.S1$source == "eindhoventab")
idS2 <- which(ML2.S2$source == "eindhoven" | ML2.S2$source == "eindhoventab")

ML2.S1$source[idS1][ML2.S1$meta_4_TEXT[idS1] == "1366x768"] <- "eindhoventab"
ML2.S1$source[idS1][ML2.S1$meta_4_TEXT[idS1] != "1366x768"] <- "eindhoven"

CaseCount("13. eindhoventab")
```

Change: Occidtab

- If source = (occid OR occidtab) & meta_4_TEXT = 1280x720, then source = occidtab.
- If source = (occid OR occidtab) & meta_4_TEXT ≠ (NOT equal) 1280x720, then source = occid.

**Explanation:*

Implementation:

```
idS1 <- which(ML2.S1$source == "occid" | ML2.S1$source == "occidtab")
idS2 <- which(ML2.S2$source == "occid" | ML2.S2$source == "occidtab")

ML2.S1$source[idS1][ML2.S1$meta_4_TEXT[idS1] == "1280x720"] <- "occidtab"
ML2.S1$source[idS1][ML2.S1$meta_4_TEXT[idS1] != "1280x720"] <- "occid"

CaseCount("14. occidtab")
```

Exclusion: Small N

These cases contain individual testruns or have small N

Implementation:

```
ML2.S1 <- filter(ML2.S1, ML2.S1$source != "lund")
ML2.S1 <- filter(ML2.S1, !(ML2.S1$.id == "ML2_Slate1_Dutch_execution_illegal_DEPLOY__netherlands_r.csv" & ML2.S1$source =="tilb

ML2.S2 <- filter(ML2.S2, ML2.S2$source != "avans")
ML2.S2 <- filter(ML2.S2, !(ML2.S2$.id == "ML2_Slate2_SpanishCosta_Rica_r_manually_recode_rosss21.csv" & ML2.S2$source == "puc'
ML2.S2 <- filter(ML2.S2, !(ML2.S2$.id == "ML2_Slate2_USEng_Inlab_DEPLOY_r.csv" & ML2.S2$source == "queensland2"))

CaseCount("15. Small N")
```

Exclusion: Empty *source* labels

Filter out cases that have no *source* label.

Implementation:

```
# Get source file of empty labels
emptyS1 <- as.data.frame(table(ML2.S1$.id[ML2.S1$source == ""]))
emptyS2 <- as.data.frame(table(ML2.S2$.id[ML2.S2$source == ""]))

# Remove source fields that still remain empty
ML2.S1 <- ML2.S1 %>% filter(nchar(source) != 0)
ML2.S2 <- ML2.S2 %>% filter(nchar(source) != 0)

CaseCount("16. Remove empty source labels")
```

Exlusion: NA on conversion from text to numeric.

Some studies required a numeric response that was input via the keyboard as text.

van.Lange.1 The number of siblings entered for the van Lange study has to be converted to an (arabic) number. A few cases remain for which this is not possible, these will be set to NA. **Implementation:**

```
# Find text which turns to NA on `as.numeric`

# Older siblings
id1    <- is.na(as.numeric(ML2.S1$van.p2.1_1_TEXT))
(data.frame(table(ML2.S1$van.p2.1_1_TEXT[id1])))
```

```
      Var1 Freq
1            50
2         -    4
3        □     4
4       0□     5
5        □     5
6       1/2    1
7        1`    1
8        1t    1
9      □□      3
10      1□     2
11  1□ □ □      1
12      1□     1
13       □     2
14      2□     2
15      2□     1
16  3 brata    1
17    hayır    1
18    jedan    1
19       no    2
20     none    11
21     None    1
22        O    1
23      one    1
24      Yok    1
25   □ □ □     1
26     □ □     1
27 □ □ □ □     2
28     □ □     1
29       □     1
30       □     1
31     □ □     3
32       □     3
```

```
# Younger siblings
id2    <- is.na(as.numeric(ML2.S1$van.p2.1_2_TEXT))
(data.frame(table(ML2.S1$van.p2.1_2_TEXT[id2])))
```

```
            Var1 Freq
1                  64
2              -    4
```

```
3              .   |
4             `0   |
5             |0   |
6             □   |0
7            0□    4
8            □□    |
9            0□    |
10           0□    |
11            □    |
12      | deceased   |
13            |t   |
14            |□   |
15         2 brata   |
16           2□    2
17           2□    |
18            □    |
19           four   |
20           hayır  |
21           Jag    |
22          nijedan  |
23            no   |
24           none   4
25             o   |
26           one   3
27             P   |
28           two   |
29           yes   |
30           Yok   |
31          □□□    2
32          □□    |
33        □□□□     |
34 □□□□□□□□□     |
35           □□    2
36           □□    |
37            □    3
38           □□    2
39           □□    |
```

```
# Change values
sibs1 <- gsub("([[:space:]])*","",ML2.S1$van.p2.1_1_TEXT)
sibs1 <- gsub("([Nn]one)|(no)|(geen)|([oO□0])","0",sibs1)
sibs1 <- gsub("[(one)□ 1]","1",sibs1)
sibs1 <- gsub("(□)","2",sibs1)
id1   <- is.na(as.numeric(sibs1))

# These will be set to NA by calling as.numeric()
(data.frame(table(ML2.S1$van.p2.1_1_TEXT[id1])))
```

```
  Var1 Freq
```

```
1            50
2         -   4
3        0□    5
4        1/2   1
5         1`   1
6         1t   1
7       □□    3
8        1□    2
9    1□□□     1
10       1□    1
11       2□    2
12       2□    1
13   3 brata   1
14     hayır   1
15     jedan   1
16       Yok   1
17    □□□     1
18     □□     1
19 □□□□      2
20     □□     1
21      □     1
22      □     1
23     □□     3
24      □     3
```

```
ML2.S1$van.p2.1_1_TEXT <- as.numeric(sibs1)

# Change values
sibs2 <- gsub("([[:space:]])*","",ML2.S1$van.p2.1_2_TEXT)
sibs2 <- gsub("([Nn]one)|(no)|(geen)|([oO□0])","0",sibs2)
sibs2 <- gsub("[(one)□  1(1 deceased)]","1",sibs2)
sibs2 <- gsub("[(two)□(2 brata)]","2",sibs2)
sibs2 <- gsub("(□)","3",sibs2)
sibs2 <- gsub("[(four)]","4",sibs2)
id2    <- is.na(as.numeric(sibs2))

# These will be set to NA by calling as.numeric()
(data.frame(table(ML2.S1$van.p2.1_2_TEXT[id2])))
```

```
            Var1  Freq
1                   64
2                 -   4
3                 .   1
4                 `0   1
5                 |0   1
6                 0□    4
7                 □□    1
8                 0□    1
9                 0□    1
```

```
10            1□    1
11            2□    2
12            2□    1
13          hayır   1
14           Jag    1
15         nijedan   1
16             P     1
17           yes     1
18           Yok     1
19          □□□      2
20           □□      1
21         □□□□      1
22 □□□□□□□□□      1
23           □□      2
24           □□      1
25            □      3
26           □□      2
27           □□      1
```

```
ML2.S1$van.p2.1_2_TEXT <- as.numeric(sibs2)

CaseCount("17. van.Lange.1 numbers as text.")
```

Ross.1 and Ross.2 The percentage entered for the Ross studies has to be converted to an (arabic) number. A few cases remain for which this is not possible, these will be set to NA. **Implementation:**

```
# Convert to decimal point and remove % sign.
ML2.S1$ross.s1.1_1_TEXT <- gsub("[.,]",".", ML2.S1$ross.s1.1_1_TEXT,perl = TRUE)
ML2.S1$ross.s1.1_1_TEXT <- gsub("([%□]|[`]|(percent)|(procent)|\\s)*","", ML2.S1$ross.s1.1_1_TEXT,perl = TRUE)

id1    <- is.na(as.numeric(ML2.S1$ross.s1.1_1_TEXT))|(!between(as.numeric(ML2.S1$ross.s1.1_1_TEXT),0,100))
(data.frame(table(ML2.S1$ross.s1.1_1_TEXT[id1])))
```

```
      Var1 Freq
1            51
2       □□    1
3       150   1
4       □□    2
5       □□    1
6       □□    3
7     95-100  1
8  cinquante  1
9       fele  1
10      nose  1
11         t  1
```

```
# Change ross1
ross1 <- gsub("[□1]","1", ML2.S1$ross.s1.1_1_TEXT)
ross1 <- gsub("[□3]","3", ross1)
ross1 <- gsub("[□]","4", ross1)
ross1 <- gsub("[□]","5", ross1)
ross1 <- gsub("([oO□0])","0", ross1)

# These willl be set to NA by calling as.numeric()
id1    <- is.na(as.numeric(ross1))|(!between(as.numeric(ross1),0,100))
(data.frame(table(ML2.S1$ross.s1.1_1_TEXT[id1])))
```

```
      Var1 Freq
1             51
2      150     1
3    95-100    1
4 cinquante     1
5      fele     1
6      nose     1
7         t     1
```

```
ML2.S1$ross.s1.1_1_TEXT <- as.numeric(ross1)

# Change ross2
ML2.S2$ross.s2.1_1_TEXT <- gsub("[.,]",".", ML2.S2$ross.s2.1_1_TEXT,perl = TRUE)
ML2.S2$ross.s2.1_1_TEXT <- gsub("([%□]|[`]|(percent)|(procent)|\\s)*","", ML2.S2$ross.s2.1_1_TEXT,perl = TRUE)

id2    <- is.na(as.numeric(ML2.S2$ross.s2.1_1_TEXT))|(!between(as.numeric(ML2.S2$ross.s2.1_1_TEXT),0,100))
(data.frame(table(ML2.S2$ross.s2.1_1_TEXT[id2])))
```

```
                  Var1 Freq
1                  175
2                    -    1
3               ??????    1
4               □□□     1
5                 1000    5
6                  105    1
7                 1200    5
8                120000    1
9                 1250    1
10                13000    1
11               130000    2
12              1300000    1
13                  137    1
14                 1500    5
15          15000shillings    1
16                  155    1
17            15percent    1
```

```
18                          185    1
19                         2000    1
20                      2000ths    1
21                          250    1
22                          300    1
23                          420    1
24                         50/0    1
25                          500    1
26                         5000    2
27                       555550    1
28                          610    1
29                          700    1
30                          850    1
31                           □□    1
32                    99???????    1
33                          all    1
34                         bsbd    1
35                        dunno    1
36                   finebymail    1
37                    idontknow    1
38                   loremipsum    1
39 Mislimdanikonebiplatiokaznu.    1
40                          n/a    1
41                       niewiem    1
42                           no    2
43                       noidea    1
44                          non    1
45                          okay    1
46                       pay1500    1
47                         super    1
48                         Todos    1
49                       vsichni    1
50                       všichni    1
51                          yeah    1
52                           yes    3
53                   Yourchances    1
```

```
ross2 <- gsub("[□1]","1", ML2.S2$ross.s2.1_1_TEXT)
ross2 <- gsub("[□3]","3", ross2)
ross2 <- gsub("[□]","4", ross2)
ross2 <- gsub("[□]","5", ross2)
ross2 <- gsub("([oO□0])","0", ross2)
ross2 <- gsub("([□])","9", ross2)

# These willl be set to NA by calling as.numeric()
id2    <- is.na(as.numeric(ross2))|(!between(as.numeric(ross2),0,100))
(data.frame(table(ML2.S2$ross.s2.1_1_TEXT[id2])))
```

```
                    Var1 Freq
```

1	175	
2	-	1
3	??????	1
4	1000	5
5	105	1
6	1200	5
7	120000	1
8	1250	1
9	13000	1
10	130000	2
11	1300000	1
12	137	1
13	1500	5
14	15000shillings	1
15	155	1
16	15percent	1
17	185	1
18	2000	1
19	2000ths	1
20	250	1
21	300	1
22	420	1
23	50/0	1
24	500	1
25	5000	2
26	555550	1
27	610	1
28	700	1
29	850	1
30	99???????	1
31	all	1
32	bsbd	1
33	dunno	1
34	finebymail	1
35	idontknow	1
36	loremipsum	1
37	Mislimdanikonebiplatiokaznu.	1
38	n/a	1
39	niewiem	1
40	no	2
41	noidea	1
42	non	1
43	okay	1
44	pay1500	1
45	super	1
46	Todos	1
47	vsichni	1
48	všichni	1
49	yeah	1

```
50                  yes  3
51          Yourchances  1
```

```
ML2.S2$ross.s2.1_1_TEXT <- as.numeric(ross2)

CaseCount("18. Ross.1 and Ross.2 percentages as text.")
```

Exclusion: Errors in Zaval unscrambled sentences

For the Zaval et al. (2014) replication study errors in the sentence unscrambling taks indicate a failure to prime the participant. Performance on the task has been coded for each language and *all the unique responses* are available here: GoogleSheet.

The coding scheme:

- 0 = incorrect response, gibberish, blank, omits priming concept (if a priming trial), or left all the words in in the same order (copy/pasting)
- 1 = correct response (meaningful sentence using four of the five words, containing the priming word if a priming trial), tolerant of misspelling/typos
- 2 = incorrect response (wrong number of words, sentence grammatically incorrect, incomplete, not a full thought, etc.) that probably still primes the concept (if a priming trial, contains the priming word or a similar word). If not a priming trial, this is just a more relaxed standard for a "correct" response. The one exception is if it contains all the words from the task in the same order (in which case it would be marked 0, because that indicates copy/pasting).

Implementation Function get.zavCode() will check wether the answer is correct for each sentence. The googlesheet serves as a lookup table for each participant. Only participants with correct responses (code = 1) will be entered in the primary analyses.

The following variables will be added to the dataset ML2.S2:

- zav.code - Contains codes for the 13 sentences.
- zav.include.strict - Indicator variable for cases with code = 1 for all 13 sentences.
- zav.include.primed - Indicator variable for cases with with code > 0 for all 13 sentences.

Here is an example of the variables that were added:

```
ML2.S2[5,c('zav2.1', 'zav2.10')]
```

```
        zav2.1        zav2.10
5 she boils the egg the old man sweats
```

```
ML2.S2[5,c('zav.code1', 'zav.code10')]
```

```
  zav.code1 zav.code10
5         1          1
```

```
ML2.S2[5,c('zav.include.strict','zav.include.primed')]
```

```
  zav.include.strict zav.include.primed
5               TRUE               TRUE
```

Exclusion: Copy less than half of article Zhong

For the Zong et al. study participants who copy less than half of the target article need to be excluded from the analyses.

Implementation Add variable nCopied, a count of the characters that were copied (spaces and punctuation removed).

```
ML2.S2$nCopied.zhon1 <- nchar(gsub("[[:space:]]|[[:punct:]]","",ML2.S2$zhon1.1))
ML2.S2$nCopied.zhon2 <- nchar(gsub("[[:space:]]|[[:punct:]]","",ML2.S2$zhon2.1))
```

Add a variable nTextChar: If nCopied is smaller than half of the characters in the target text (nTextChar), the case will be excluded.
The target texts for each language are available in a Google spreadsheet.

```
(chartable <- get.GoogleSheet(url="https://docs.google.com/spreadsheets/d/1J9I1JVTQqCrC7x5gz3TzA7sUCvjKGfz0nh8EgfVAVs/pub?g
```

```
Source: local data frame [12 x 3]

                language nCharText.zhon1 nCharText.zhon2
                   (chr)           (int)           (int)
1               English             547             573
2               Turkish            1036            1055
3  Chinese (traditional)             327             337
4               Spanish            1145            1187
5   Chinese (simplified)             286             299
6               Serbian            1079            1103
7                Polish            1126            1126
8               Italian            1281            1312
9                German            1207            1226
10               French            1139            1181
11                Dutch            1137            1179
12                Czech             999            1007
```

```
# Add the number of characters for each language.
ML2.S2 <- cbind(ML2.S2, ldply(seq_along(ML2.S2[,1]), function(l) data.frame(chartable[ML2.S2$Language[l]==chartable$language, c("nC
```

Exclusion: Ages < 18

For some sites a value of age < 18 must be considered invalid.

Implementation Remove cases for those sites

```
excludeAge <- c(
"armstrong",
"aus",
"aus2",
"badaniaon",
"bilgi",
"bogota",
"brookcuny",
"cas",
"csunorthonline",
"hkusttab",
"kenyon",
"kwazulu",
"marian",
"mcdanielonline",
"moralsense",
"munichon",
"pascal",
"plu",
"psumain",
"rotterdam",
"swps",
"tamu",
"tanzania",
"tufts",
"ubc",
"ubctab",
"unswonline",
"vcu",
"vuamsterdam",
"williampat",
"yale")

ML2.S1 <- ML2.S1 %>% filter(!(source%in%excludeAge & age<18))
ML2.S2 <- ML2.S2 %>% filter(!(source%in%excludeAge & age<18))

CaseCount("19. Exclude age < 18.")
```

Other notes about typos or corrections

Changes to the script that were deemed minor enough to NOT warrant exclusion (so, no changes were made in the analysis due to the below notes)

Sample: Any data from ML2_Slate1_Chinese_Mainland_execution_legal_DEPLOY_r

(relevant source variables = zhejiang, zhejiangcomp, hunancomp, henancomp, henan)

Note: Miyamoto study: Changed the name from "□□" to "□□" on Oct 22, 2014 to be consistent with author's suggestion. Deemed minor enough to use all data (note: the vast majority of participants were run before the change was made) but if we notice differences between this site and the other Chinese samples on this study, this could be why. Also made a small wording adjustment from "□□" to "□□□" on that date.

Sample: Datafile = ML2_Slate1_Czech_execution_illegal_r.csv

Note: kay2.5: The "2" is repeated on this scale erroneously (1,2,2,3,4,5,6,7). Does not merit exclusion from Kay altogether, unless we see big differences between this site and similar sites. (affected N = ~143, source = purkyne)

Sample: Datafile = ML2_Slate1_Inlab_execution_illegal_DEPLOY_UK_r.csv

Note: Country references were updated on Oct. 5, 2014 to be consistent with the other samples. Affects ~21/142 participants from this site.

Sample: Datafile = ML2_Slate1_Chinese_Inlab_execution_illegal_DEPLOY__Hong_Kong_r_For_Andrew_Tang

Note: Local site lead requested minor wording changes from "ML2 Slate1 Chinese In-lab (execution illegal) DEPLOY - Hong Kong" survey used in another Chinese location. Wording changes in the translation were made to the following studies: Miyamoto (Miya1.1 and Miya2.1), Hauser (haus2.1), Cirtcher (Crit1.1, crit2.1), Alter (alt1.8, alt2.8), and demographics (IMC1, born.par)

Case Counts

Below is a list of the impact of each filter step on the number of cases (.N), on the unique number of *source* labels (.Nsrc) and on the relative change in NA values expressed as a percentage (.NApct).

CC

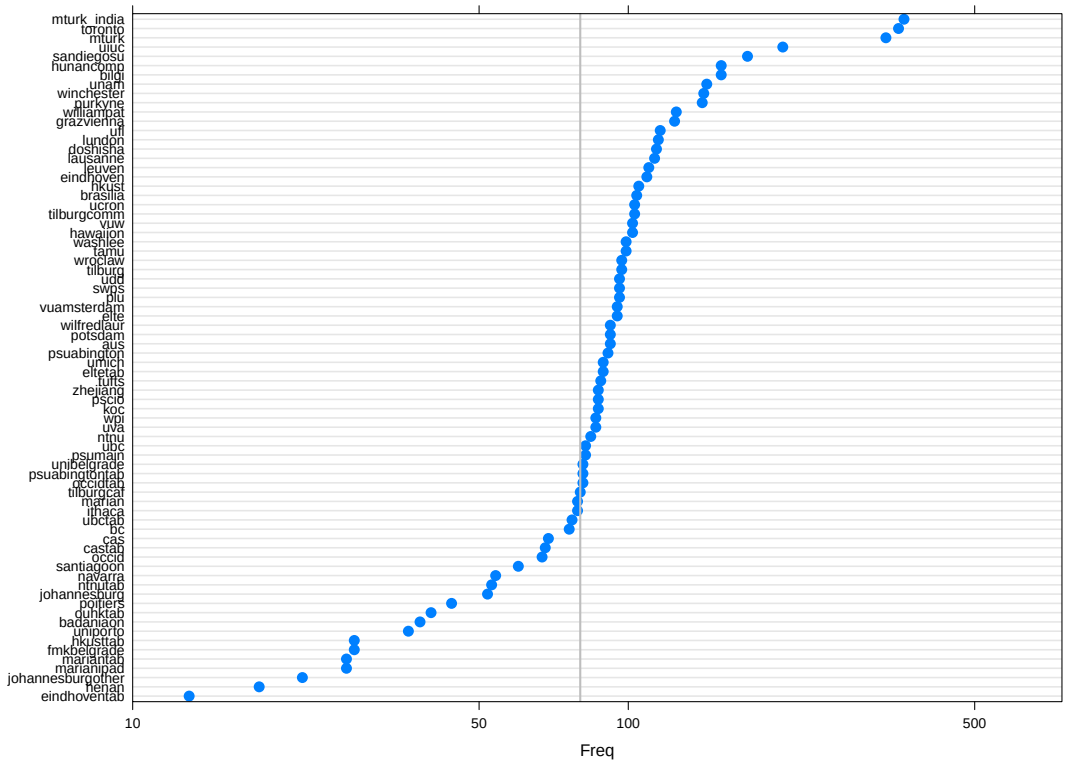
	exclusionRule	S1.N	S2.N	TOT.N	S1.Nsrc	S2.Nsrc	TOT.Nsrc	S1.NApct	S2.NApct
1	1. Merged complete cases (Finished == 1)	7884	8822	16706	92	66	145	Inf	Inf
2	2. Removed test and -99	7678	8687	16365	91	66	144	-3.207	-1.68
3	3.1 Cleaned Source labels (first)	7678	8687	16365	76	60	76	0	0
4	3.2 Clean Source labels (second)	7678	8687	16365	77	62	125	-0.007	-0.03
5	4. Add site variables	7678	8687	16365	77	62	125	0.108	0.241

```
6                                5. rio 7663 8687 16350      76      62      124   -0.206         0
7                            6. Chinese date 7663 8687 16350      76      62      124    0.233     0.005
8                          7. Dutch date - Hsee 7663 8687 16350      76      62      124         0    0.003
9                        8. Uruguay date - Huan 7663 8687 16350      76      62      124         0         0
10                       9. French date - Huan 7663 8687 16350      76      62      124    0.014         0
11                     10. Filter mturk doubles 7663 8687 16350      76      62      124         0         0
12               11. Filter mturk_india doubles 7425 8425 15850      76      62      124   -3.071        -3
13                                   12. IDs 7425 8425 15850      76      62      124         0         0
14                         13. eindhoventab 7425 8425 15850      76      62      124         0         0
15                           14. occidtab 7425 8425 15850      76      62      124         0         0
16                           15. Small N 7421 8425 15846      75      62      123   -0.055         0
17             16. Remove empty source labels 7298 8367 15665      74      61      122    -2.11    -0.766
18           17. van.Lange.1 numbers as text. 7298 8367 15665      74      61      122    0.011         0
19 18. Ross.1 and Ross.2 percentages as text. 7298 8367 15665      74      61      122    0.003     0.005
20                       19. Exclude age < 18. 7263 8056 15319      74      61      122    -0.52    -3.795
```

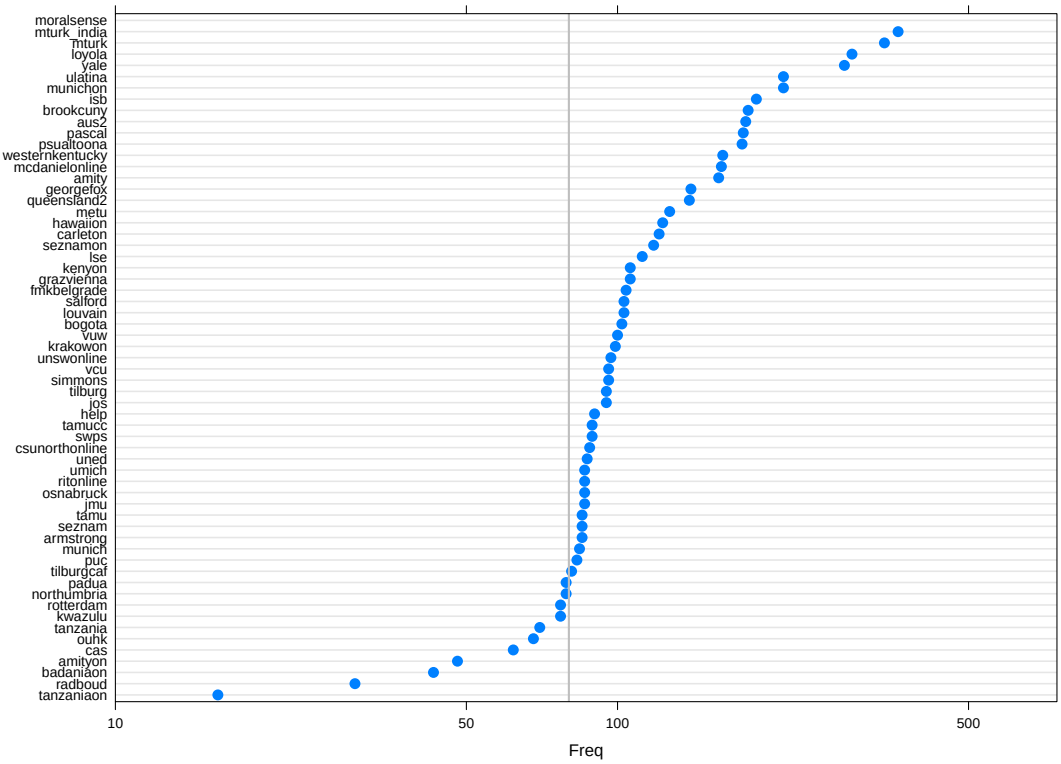
Graphs of N per *source* label.

```
# Some info
require(lattice)
sourceN1 <- as.data.frame(table(ML2.S1$source))
sourceN2 <- as.data.frame(table(ML2.S2$source))

S1fr <- data.frame(ftable(ML2.S1$source))
S1fr <- S1fr[order(S1fr$Freq), ]
dotplot(reorder(S1fr$Var1,S1fr$Freq) ~ Freq,
        scales = list(x = list(log = 10,at = c(10,50,100,500), limits = c(10,750))),
        data = S1fr,
        panel = function (x, y) {
                panel.dotplot(x,y, cex=1.3, lwd=1.5)
                panel.abline(v = log10(80), col = "gray", lwd = 2)
                }
        )
```




```
S2fr <- data.frame(ftable(ML2.S2$source))
S2fr <- S2fr[order(S2fr$Freq), ]
dotplot(reorder(S2fr$Var1,S2fr$Freq) ~ Freq,
        scales = list(x = list(log = 10,at = c(10,50,100,500), limits = c(10,750))),
        data = S2fr,
        panel = function (x, y) {
                panel.dotplot(x,y, cex=1.3, lwd=1.5)
                panel.abline(v = log10(80), col = "gray", lwd = 2)
                }
        )
```


moralsense
mturk_india
mturk
loyola
yale
ulatina
munichon
isb
brookcuny
aus2
pascal
psualtoona
westernkentucky
mcdanielonline
amity
georgefox
queensland2
metu
hawaiion
carleton
seznamon
lse
kenyon
grazvienna
fmkbelgrade
salford
louvain
bogota
vuw
krakowon
unswonline
vcu
simmons
tilburg
jos
help
tamucc
swps
csunorthonline
uned
umich
ritonline
osnabruck
jmu
tamu
seznam
armstrong
munich
puc
tilburgcaf
padua
northumbria
rotterdam
kwazulu
tanzania
ouhk
cas
amityon
badaniaon
radboud
tanzaniaon
10
50
100
500
Freq