Cleaning the Survey Data

In this chapter we process the raw survey data by:

1. Renaming columns
2. Recoding response values for quantitative analysis
3. Dropping duplicates, incomplete responses and test responses

Renaming Columns

Columns referencing survey questions are renamed with the help of a data dictionary to improve readability.

```r
torename <- dd %>%
  dplyr::select(vname, vname_raw) %>%
  tidyr::drop_na()

survey_df <- raw_survey_df %>%
  dplyr::rename_at(vars(torename$vname_raw), ~torename$vname)
```

Recoding Survey Responses

The next step is recoding survey responses. Each group of survey questions comes with its own set of valid inputs that must be recoded separately. For example, “N/A”’s are options for some survey questions and not for others, and some survey questions allow for manual text inputs.

The below code chunk separates all survey questions into their respective categories before further separating each category into numeric, text or NA inputs.

NA questions here refer to “Check here if not applicable to your organization” questions in the survey, where a “C” indicates that the respondent has checked the N/A box.

```r
# Function to get survey questions by group and data type from data dictionary
get_survey_questions <- function(data_dict, qgroup, qtype){
  qns <- data_dict %>%
    dplyr::filter(group == qgroup,
                  type == qtype) %>%
    dplyr::pull("vname")
  return(qns)
}
```

```r
# Changes to Programs and Services
program_change_qns_bool <- get_survey_questions(dd, "ProgChanges", "boolean")
program_change_qns_txt <- get_survey_questions(dd, "ProgChanges", "text")

# People Served
people_served_qns_int <- get_survey_questions(dd, "PeopleServed#1", "integer")
people_served_qns_bool <- get_survey_questions(dd, "PeopleServed#2", "boolean")

# Demand for future programs and services
demand_fct_qns <- get_survey_questions(dd, "Demand", "factor")

# Staff and Volunteers
staff_qns_int <- c(
  get_survey_questions(dd, "Staff#1", "integer"),
  get_survey_questions(dd, "Staff#2", "integer")
)
staff_qns_bool <- get_survey_questions(dd, "Staff#3", "boolean")
staff_qns_text <- c(
  get_survey_questions(dd, "Staff#1", "text"),
  get_survey_questions(dd, "Staff#2", "text"),
  get_survey_questions(dd, "Staff#3", "text")
)

# Importance of Volunteers and Donors
volimportance_qns_fct <- get_survey_questions(dd, "VolImportance", "factor")
donimportance_qns_fct <- get_survey_questions(dd, "DonImportance", "factor")

# Fundraising
fundraise_qns_bool <- get_survey_questions(dd, "FRchanges", "boolean")
fundraise_qns_text <- get_survey_questions(dd, "FRchanges", "text")

# Amount for major gifts
majorgift_qn_num <- get_survey_questions(dd, "Frmajgift", "numeric")

# Fundraising Changes
fundraise_change_qns_fct <- get_survey_questions(dd, "FRchanges#1", "factor")

# Number of donors
fundraise_donor_qns_int <- get_survey_questions(dd, "FRnumberdonors", "integer")

# Fundraising Sought and Received
```

```
fundraise_skrcv_qns_bool <- c(
  get_survey_questions(dd, "Funding1#1", "boolean"),
  get_survey_questions(dd, "Funding1#2", "boolean")
)

# Revenue breakdown
finance_revenue_qns_num <- get_survey_questions(dd, "Finances", "numeric")
finance_revenue_qns_text <- get_survey_questions(dd, "Finances", "text")

# Financial reserves
reserve_qns_num <- get_survey_questions(dd, "Reserves", "numeric")
reserve_qns_bool <- get_survey_questions(dd, "Reserves", "boolean")

# CARES Funding
cares_qns_num <- get_survey_questions(dd, "CARES", "numeric")
cares_qns_bool <- get_survey_questions(dd, "CARES", "boolean")

# Changes to finances
finance_chng_qns_bool <- get_survey_questions(dd, "FinanceChanges", "boolean")
finance_chng_qns_text <- get_survey_questions(dd, "FinanceChanges", "text")

# Leadership Changes
leadership_chng_qns_bool <- get_survey_questions(dd, "LeadershipChanges", "boolean")
leadership_chng_qns_text <- get_survey_questions(dd, "LeadershipChanges", "text")

# Race and Gender Qns
race_gender_qns_bool <- c(
  get_survey_questions(dd, "CEOrace", "boolean"),
  get_survey_questions(dd, "CEOgender", "boolean"),
  get_survey_questions(dd, "BCrace", "boolean"),
  get_survey_questions(dd, "BCgender", "boolean")
)

race_gender_qns_text <- c(
  get_survey_questions(dd, "CEOrace", "text"),
  get_survey_questions(dd, "CEOgender", "text"),
  get_survey_questions(dd, "BCrace", "text"),
  get_survey_questions(dd, "BCgender", "text")
)

# External affairs questions
```

```r
extaffairs_qns_fct <- get_survey_questions(dd, "ExternalAffairs", "factor")
# Primary concern question
primary_cncrn_qn_text <- get_survey_questions(dd, "PrimaryConcern", "text")
```

Survey Variable Breakdown

- **15** questions about changes to programs and services
- **4** questions about the number of people each organization serves
- **1** question about overall program demand
- **27** questions about staff numbers
- **2** questions about donor and volunteer importance
- **11** questions about changes to leadership
- **26** questions about the race and gender of CEOs and Board Chairs.
- **8** questions about changes to organizational finances
- **2** questions about CARES Funding
- **2** questions about financial reserves
- **9** questions about revenue sources
- **26** questions about fundraising sources
- **2** questions about donor types in fundraising
- **7** questions about fundraising changes
- **1** questions about major gift amounts
- **1** questions about future concerns

Recode and Relabel Variables

There are several factor and boolean variables in this survey data set with inconsistent coding. We will recode each category of question individually and document our decision criteria.

Using the memisc package, different types of missingness (e.g. Unsure, N/A, -99 etc.) are coded as missing.

```r
# Function to convert survey variable to survey.item with missingness and variable labels
create_survey_item <- function(survey_data, qns, recode_vals, recode_labs, missing_vals){
  survey_data <- survey_data |>
    purrr::modify_at(
      .at = qns,
      .f = memisc::as.item,
      labels = structure(
        .Data = recode_vals,
        names = recode_labs
      ),
      missing.values = missing_vals
    )
  return(survey_data)
}
```

Yes/No Questions

Original Value	Description	Recode Value	Missing ?
Yes	Yes	1	No
No	No	0	No
Unsure	Unsure	97	Yes
N/A	Not Applicable	98	Yes
-99	Incomplete	99	Yes
NA	Unanswered	NA	Yes

```
bool_qns <- c(program_change_qns_bool, fundraise_qns_bool, cares_qns_bool, finance_chng_qn

survey_recode_df <- survey_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(bool_qns),
      ~ dplyr::case_match(.,
                          "Yes" ~ 1, "No" ~ 0, "Unsure" ~ 97, "N/A" ~ 98, "-99" ~ 99,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  bool_qns,
  recode_vals = c(0, 1, 97, 98, 99),
  recode_labs = c("No", "Yes", "Unsure", "Not Applicable", "N/A"),
  missing_vals = c(97, 98, 99)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
```

```r
      tidyselect::all_of(bool_qns[1])
    )
)
```

```
================================================================================

   PrgSrvc_IncrNum

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 97, 98, 99

   Values and labels         N Valid Total

    0   'No'              305  40.6  37.1
    1   'Yes'             446  59.4  54.3
   97 M 'Unsure'            1         0.1
   98 M 'Not Applicable'    7         0.9
   99 M 'N/A'              11         1.3
   NA M                    52         6.3
```

Single Checkboxes

These questions are presented as a checkbox to the respondent. They indicate an affirmative answer to the question.

Seek or Receive Fundraising Questions

These checkboxes are ticked by the respondent to indicate if they have sought or received funding from a specific source.

Original Value	Description	Recode Label	Recode Value	Missing ?
(select all that apply)	Checkbox Checked	Yes	1	No
-99	Checkbox Unchecked	No	0	No
NA	Unanswered	NA	NA	Yes

```r
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(fundraise_skrcv_qns_bool),
      ~ dplyr::case_match(.,
                          "(select all that apply)" ~ 1, "-99" ~ 0,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  fundraise_skrcv_qns_bool,
  recode_vals = c(0, 1),
  recode_labs = c("No", "Yes"),
  missing_vals = c()
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(fundraise_skrcv_qns_bool[1])
    )
)
```

```
================================================================================

   FndRaise_LocGvtGrnt_Seek

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal

   Values and labels      N Valid Total

    0   'No'            317  46.8  38.6
    1   'Yes'           361  53.2  43.9
   NA  M                144        17.5
```

Race and Gender Checkboxes

These checkboxes are ticked by the respondent to indicate if their CEO or board chair belong to a specified race or gender identity.

Original Value	Description	Recode Label	Recode Value	Missing ?
Asian/Pacific Islander	Checkbox Checked	Yes	1	No
Black/African American	Checkbox Checked	Yes	1	No
Latinx/Hispanic	Checkbox Checked	Yes	1	No
Native American/American Indian	Checkbox Checked	Yes	1	No
White	Checkbox Checked	Yes	1	No
Man	Checkbox Checked	Yes	1	No
Woman	Checkbox Checked	Yes	1	No
Trans	Checkbox Checked	Yes	1	No
Gender non-conforming/Non-Binary	Checkbox Checked	Yes	1	No
Other (please specify)	Checkbox Checked	Yes	1	No
0	Checkbox Unchecked	No	0	No
-99	Incomplete	Incomplete	99	Yes
NA	Unanswered	NA	NA	Yes

```r
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(race_gender_qns_bool),
      ~ dplyr::case_match(.,
                          "Asian/Pacific Islander" ~ 1,
                          "Black/African American" ~ 1,
                          "Latinx/Hispanic" ~ 1,
                          "Native American/American Indian" ~ 1,
                          "White" ~ 1,
                          "Man" ~ 1,
                          "Woman" ~ 1,
                          "Trans" ~ 1,
                          "Gender non-conforming/Non-Binary" ~ 1,
                          "Other (please specify):" ~ 1,
                          "-99" ~ 99,
                          "0" ~ 0,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  race_gender_qns_bool,
  recode_vals = c(0, 1, 99),
  recode_labs = c("No", "Yes", "Incomplete"),
  missing_vals = c(99)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(race_gender_qns_bool[1])
    )
)
```

```
================================================================================

   CEOrace_AAPI

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 99

   Values and labels    N Valid Total

    0   'No'           48  98.0   5.8
    1   'Yes'           1   2.0   0.1
   99 M 'Incomplete'    2         0.2
   NA M               771        93.8
```

N/A Checkboxes

These questions are presented as a checkbox to the user to indicate that a question is not applicable. “Yes” here means “Yes, this question is not applicable”.

Original Value	Description	Recode Label	Recode Value	Missing ?
C	Yes, this question is not applicable	Yes	1	No
N/A	Yes, this question is not applicable	Yes	1	No
-99	Incomplete	No	0	No
NA	Unanswered	NA	NA	Yes

```r
na_bool_qns <- c(staff_qns_bool, reserve_qns_bool, people_served_qns_bool)

survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(na_bool_qns),
      ~ dplyr::case_match(.,
                          "C" ~ 1, "-99" ~ 0, "N/A" ~ 1,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  na_bool_qns,
  recode_vals = c(0, 1),
  recode_labs = c("No", "Yes"),
  missing_vals = c()
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(na_bool_qns[1])
    )
)
```

```
================================================================================

   Staff_Fulltime_NA

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal

   Values and labels     N Valid Total

    0   'No'           666  92.2  81.0
    1   'Yes'           56   7.8   6.8
   NA  M               100        12.2
```

Multi-selection Inputs

These questions offer the user with multiple options to select one from. Since the options are ordered categories, they are coded on an ordinal scale.

Increase - Decrease Questions

There 2 questions that ask respondents to define changes via an increase or decrease relative to previous years. They are recoded on an ordinal scale.

Changes in Demand Questions

Original Value	Description	Recode Label	Recode Value	Missing ?
Increase	Increase	Increase	2	No
Stay the same	Stay the same	Unchanged	1	No
Decrease	Decrease	Decrease	0	No
-99	Incomplete	No	99	Yes
NA	Unanswered	NA	NA	Yes

```r
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(demand_fct_qns),
      ~ dplyr::case_match(.,
                          "Increase" ~ 2, "Stay the same" ~ 1, "Decrease" ~ 0, "-99" ~ 99,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  demand_fct_qns,
  recode_vals = c(2, 1, 0, 99),
  recode_labs = c("Increase", "Unchanged", "Decrease", "Incomplete"),
  missing_vals = c(99)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(demand_fct_qns[1])
    )
)
```

```
================================================================================

   Dmnd_NxtYear

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 99

   Values and labels     N Valid Total

    0   'Decrease'      19   2.6   2.3
    1   'Unchanged'    154  21.3  18.7
    2   'Increase'     551  76.1  67.0
   99 M 'Incomplete'    10         1.2
   NA M                 88        10.7
```

Changes in Fundraising Questions

Original Value	Description	Recode Label	Recode Value	Missing ?
Increased significantly (by more than 10%)	Largest Increase	Increase Significantly	5	No
Increased moderately (by less than 10%)	Second Largest Increase	Increase Moderately	4	No
Stayed more or less the same	Third Largest Increase	Unchanged	3	No
Decreased moderately (by less than 10%)	Fourth Largest Increase	Decrease Moderately	2	No
Decreased significantly (by more than 10%)	Fifth Largest Increase	Decrease Significantly	1	No
Unsure	Unsure	Unsure	99	Missing
-99	Incomplete	Incomplete	98	Yes
N/A	Not Applicable	Not Applicable	97	Yes
NA	Unanswered	NA	NA	Yes

```r
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(fundraise_change_qns_fct),
      ~ dplyr::case_match(.,
                          "Increased significantly (by more than 10%)" ~ 5,
                          "Increased moderately (by less than 10%)" ~ 4,
                          "Stayed more or less the same" ~ 3,
                          "Decreased moderately (by less than 10%)" ~ 2,
                          "Decreased significantly (by more than 10%)" ~ 1,
                          "Unsure" ~ 99,
                          "-99" ~ 98,
                          "N/A" ~ 97,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  fundraise_change_qns_fct,
  recode_vals = c(5, 4, 3, 2, 1, 0, 99, 98, 97),
  recode_labs = c("Increase Significantly", "Increase Moderately", "Unchanged", "Decrease
  missing_vals = c(99, 98, 97)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(fundraise_change_qns_fct[1])
    )
)
```

```
================================================================================

   FndRaise_Overall_Chng

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 99, 98, 97

   Values and labels                 N Valid Total

    0   'Unsure'                     0   0.0   0.0
    1   'Decrease Significantly'    80  12.9   9.7
    2   'Decrease Moderately'       77  12.4   9.4
    3   'Unchanged'                161  26.0  19.6
    4   'Increase Moderately'      169  27.3  20.6
    5   'Increase Significantly'   133  21.5  16.2
   97 M 'NA'                         2         0.2
   98 M 'Not Applicable'            33         4.0
   99 M 'Incomplete'                12         1.5
   NA M                            155        18.9
```

```
dplyr::across(
  tidyselect::all_of(fundraise_change_qns),
  ~ dplyr::recode_factor(.,
                      "Increased significantly (by more than 10%)" = 1,
                      "Increased moderately (by less than 10%)" = 2,
                      "Stayed more or less the same" = 3,
                      "Decreased moderately (by less than 10%)" = 4,
                      "Decreased significantly (by more than 10%)" = 5,
                      "Unsure" = 99,
                      "-99" = -99,
                      "N/A" = 1)

)
```

Level of Importance Questions

There are 2 questions that ask respondents to rank the importance of volunteers and donors respectively. However, both sets of options' are not identical. Hence, they are recoded to common values for reproducibility.

Volunteer Importance

Original Value	Description	Recode Label	Recode Value	Missing ?
Essential - we depend entirely on volunteers to carry out our mission and goals	Maximum Importance	Essential	5	No
Very important - we depend on volunteers for a wide range of tasks, but not all	Second Most Important	Very Important	4	No
Somewhat important - we depend on volunteers for several key tasks	Third Most Important	Somewhat Important	3	No
Not very important - we depend on volunteers for only non-essential tasks	Fourth Most Important	Not Very Important	2	No

Original Value	Description	Recode Label	Recode Value	Missing ?
Not at all important - we could carry out our mission and goals without using volunteers	Fifth Most Important	Not At All Important	1	No
We do not use volunteers	Sixth Most Important	Not Used	0	No
-99	Incomplete	Incomplete	99	Yes
NA	Unanswered	NA	NA	Yes

```
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(volimportance_qns_fct),
      ~ dplyr::case_match(.,
                          "Essential - we depend entirely on volunteers to carry out our m
                          "Very important - we depend on volunteers for a wide range of ta
                          "Somewhat important - we depend on volunteers for several key ta
                          "Not very important - we depend on volunteers for only non-essen
                          "Not at all important - we could carry out our mission and goals
                          "We do not use volunteers" ~ 0,
                          "-99" ~ 99,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  volimportance_qns_fct,
  recode_vals = c(5, 4, 3, 2, 1, 0, 99),
  recode_labs = c("Essential", "Very Important", "Somewhat Important", "Not Very Important
  missing_vals = c(99)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(volimportance_qns_fct[1])
    )
)
```

```
================================================================================

   VolImportance

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 99

   Values and labels               N Valid Total

    0   'Not Used'                46   6.4   5.6
    1   'Not At All Important'    41   5.7   5.0
    2   'Not Very Important'      80  11.2   9.7
    3   'Somewhat Important'     196  27.5  23.8
    4   'Very Important'         209  29.3  25.4
    5   'Essential'              142  19.9  17.3
   99 M 'Incomplete'               6         0.7
   NA M                          102        12.4
```

Donor Importance

Original Value	Description	Recode Label	Recode Value	Missing ?
Essential, we depend entirely on individual donations to carry out our mission and goals	Maximum Importance	Essential	5	No

Original Value	Description	Recode Label	Recode Value	Missing ?
Very important, we depend on individual donations for a wide range of activities, but not all	Very Important	4	No	
Important, we depend on individual donations for several key activities	Third Most Important	Somewhat Important	3	No
Not very important, we depend on individual donations for only non-essential activities	Fourth Most Important	Not Very Important	2	No
Not at all important, we could carry out our mission and goals without donations from individuals	Fifth Most Important	Not At All Important	1	No
We do not receive donations from individuals	Sixth Most Important	Not Used	0	No
-99	Incomplete	Incomplete	99	Yes
NA	Unanswered	NA	NA	Yes

```
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(donimportance_qns_fct),
      ~ dplyr::case_match(.,
                          "Essential, we depend entirely on individual donations to carry
                          "Very important, we depend on individual donations for a wide ra
                          "Important, we depend on individual donations for several key ac
                          "Not very important, we depend on individual donations for only
                          "Not at all important, we could carry out our mission and goals
                          "We do not receive donations from individuals" ~ 0,
                          "-99" ~ 99,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  donimportance_qns_fct,
  recode_vals = c(5, 4, 3, 2, 1, 0, 99),
  recode_labs = c("Essential", "Very Important", "Somewhat Important", "Not Very Important
  missing_vals = c(99)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(donimportance_qns_fct[1])
    )
)
```

```
================================================================================

   DonImportance

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 99

   Values and labels                N Valid Total

    0   'Not Used'                 18   2.5   2.2
    1   'Not At All Important'     24   3.4   2.9
    2   'Not Very Important'       59   8.3   7.2
```

```
 3   'Somewhat Important'     137  19.3  16.7
 4   'Very Important'         298  41.9  36.3
 5   'Essential'              175  24.6  21.3
99 M 'Incomplete'               7         0.9
NA M                          104        12.7
```

Frequency Questions

These questions ask respondents to rank the frequency at which they engage in an activity.

Original Value	Description	Recode Label	Recode Value	Missing ?
Frequently	Most Frequent	Frequently	4	No
Almost all the time	Second Most Frequent	More Often Than Not	3	No
Occasionally	Third Most Frequent	Occasionally	2	No
Rarely	Fourth Most Frequent	Rarely	1	No
Never	Fifth Most Frequent	Occasionally	0	No
-99	Incomplete	Incomplete	99	Yes
NA	Unanswered	NA	NA	Yes

```r
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(extaffairs_qns_fct),
      ~ dplyr::case_match(.,
                          "Frequently" ~ 4,
                          "Almost all the time" ~ 3,
                          "Occasionally" ~ 2,
                          "Rarely" ~ 1,
                          "Never" ~ 0,
                          "-99" ~ 99,
                          .default = NA)
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  extaffairs_qns_fct,
  recode_vals = c(4, 3, 2, 1, 0, 99),
  recode_labs = c("Frequently", "More Often than Not", "Occasionally", "Rarely", "Never",
  missing_vals = c(99)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(extaffairs_qns_fct[1])
    )
)
```

```
================================================================================

   ExtAffairs_GenEd

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: nominal
   Missing values: 99

   Values and labels              N Valid Total

    0   'Never'                 167  29.0  20.3
    1   'Rarely'                 98  17.0  11.9
    2   'Occasionally'          149  25.9  18.1
    3   'More Often than Not'    45   7.8   5.5
    4   'Frequently'            116  20.2  14.1
   99 M 'Incomplete'             10         1.2
   NA M                         237        28.8
```

Integer Inputs

These questions accept an integer input from users to indicate the number of staff they have, people they served, or donors they have.

Original Value	Description	Recode Label	Recode Value	Missing ?
Whole Number	Number of Staff, People or Donors	NA	Integer Value	No
N/A	Not Applicable	NA	-1	Yes
-99	Incomplete	NA	-2	Yes
NA	Unanswered	NA	NA	Yes

```r
int_qns <- c(staff_qns_int, people_served_qns_int, fundraise_donor_qns_int)

survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(int_qns),
      ~ dplyr::case_match(.,
                          "N/A" ~ -1,"-99" ~ -2, NA ~ NA,
                          .default = as.integer(.))
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  int_qns,
  recode_vals = c(),
  recode_labs = c(),
  missing_vals = c(-1, -2)
)
```

```
Warning in structure(.Data = recode_vals, names = recode_labs): Calling 'structure(NULL, *)'
  Consider 'structure(list(), *)' instead.
```

```r
codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(int_qns[1])
    )
)
```

```
================================================================================
   Staff_Fulltime_2021

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: interval
   Missing values: -1, -2

   Values                 N Percent

      M (unlab.mss.)     67     8.2
   NA M                 100    12.2

        Min:    0.000
        Max: 3000.000
       Mean:   16.783
   Std.Dev.:  120.028
```

Numeric Inputs

These questions accept a numeric input from users to denote dollar values. For some variables, additional processing is done to remove commas and "$" signs.

Original Value	Description	Recode Label	Recode Value	Missing ?
Dollar Amount	Dollar Amount	NA	Numeric Value	No
-99	Incomplete	NA	-1	Yes
NA	Unanswered	NA	NA	Yes

```r
numeric_qns <- c(majorgift_qn_num, reserve_qns_num, cares_qns_num, finance_revenue_qns_num

survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    CARES_Rcv_Est = stringr::str_replace_all(CARES_Rcv_Est, ",", ""),
    CARES_Rcv_Est = stringr::str_replace_all(CARES_Rcv_Est, " ", ""),
    CARES_Rcv_Est = stringr::str_replace(CARES_Rcv_Est, "\\$", "")
  ) %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(numeric_qns),
      ~ dplyr::case_match(.,
                          "-99" ~ -1,
                          .default = as.numeric(.))
    )
  )

survey_recode_df <- create_survey_item(
  survey_recode_df,
  numeric_qns,
  recode_vals = c(),
  recode_labs = c(),
  missing_vals = c(-1)
)

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(numeric_qns[1])
    )
)
```

```
================================================================================

   FndRaise_MajGift_Amt

--------------------------------------------------------------------------------

   Storage mode: double
   Measurement: interval
   Missing values: -1

   Values                   N Percent

      M (unlab.mss.)       18     2.2
   NA M                   156    19.0

        Min:       0.010
        Max: 1000000.000
       Mean:    5935.894
   Std.Dev.:   44801.714
```

Text Inputs

These questions allow the user to enter raw text as answers. All non-text values are converted to NAs.

```r
text_qns <- c(staff_qns_text, finance_chng_qns_text, finance_revenue_qns_text, fundraise_q

survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    dplyr::across(
      .cols = tidyselect::all_of(text_qns),
      ~ dplyr::case_match(.,
                          "-99" ~ NA, NA ~ NA,
                          .default = as.character(.))
    )
  )

codebook(
  survey_recode_df %>%
    dplyr::select(
      tidyselect::all_of(text_qns[1])
    )
)
```

```
================================================================================

   Staff_Other_Text_2021

--------------------------------------------------------------------------------

   Storage mode: character

   Valid and missing values       N Percent

   Valid                         42     5.1
   Missing (NA)                 780    94.9
   Total                        822

   Min:                   "Accountant"
   Max: "website and graphic designer"
```

Adding Race and Gender Answers from Year 1 Survey

Questions about the race and gender of CEOs and Board Chairs are only presented to users if they indicate that a leadership change has occurred in Year 2 of the survey. Hence, we will need to impute in race and gender responses for participants who did not indicate a change in leadership.

Identifying cases for imputation

These respondents indicate that there were no changes to their CEO in year 2.

```r
no_ceo_chng_ein <- survey_recode_df %>%
  dplyr::filter(
    ! (LeadershipChng_HireCEO == 1 | LeadershipChng_IntrmCEO == 1)
  ) %>%
  dplyr::pull("EIN")

no_bchair_chng_ein <- survey_recode_df %>%
  dplyr::filter(
    LeadershipChng_ChngBC != 1
  ) %>%
  dplyr::pull("EIN")
```

Processing Year 1 Data

In Year 1, the survey questions for race and gender are encoded in a single factor variable instead of the multiple boolean variables in Year 2. We will have to wrangle the data from Year 1 into a format for Year 2.

```r
# Wrangle Data for Year 1 Responses on CEO race and Gender
year1_CEOchng <- year1_raw %>%
  dplyr::select(EIN, CEOrace, CEOgender) %>%
  dplyr::filter(EIN %in% no_ceo_chng_ein,
                ! CEOrace %in% c(-99, NA),
                ! CEOgender %in% c(-99, NA)) %>%
  dplyr::mutate(
    CEOrace = dplyr::case_match(
      CEOrace,
      1 ~ "AAPI", 2 ~ "Black", 3 ~ "Hisp", 4 ~ "NativeAm", 5 ~ "White", 6 ~ "Bi", 7 ~ "Oth
      .default = "Oth"
    ),
    CEOgender = dplyr::case_match(
      CEOgender,
      1 ~ "Man", 2 ~ "Woman", 3 ~ "Trans", 4 ~ "NB", 5 ~ "Oth",
      .default = "Oth"
    ),
  race_check = 1,
  gender_check = 1
  ) %>%
  tidyr::pivot_wider(
    names_from = CEOrace,
    names_glue = "CEOrace_{CEOrace}",
    values_from = race_check,
    values_fill = 0
  ) %>%
  tidyr::pivot_wider(
    names_from = CEOgender,
    names_glue = "CEOgender_{CEOgender}",
    values_from = gender_check,
    values_fill = 0
  )

# Wrangle Data for Year 1 Responses on Board Chair Race and Gender
year1_BCchng <- year1_raw %>%
  dplyr::select(EIN, BCrace, BCgender) %>%
  dplyr::filter(EIN %in% no_bchair_chng_ein,
                ! BCrace %in% c(-99, NA),
                ! BCgender %in% c(-99, NA)) %>%
  dplyr::mutate(
    BCrace = dplyr::case_match(
      BCrace,
      1 ~ "AAPI", 2 ~ "Black", 3 ~ "Hisp", 4 ~ "NativeAm", 5 ~ "White", 6 ~ "Bi", 7 ~ "Oth
      .default = "Oth"
    ),
    BCgender = dplyr::case_match(
      BCgender,
      1 ~ "Man", 2 ~ "Woman", 3 ~ "Trans", 4 ~ "NB", 5 ~ "Oth",
      .default = "Oth"
    ),
  race_check = 1,
  gender_check = 1
  ) %>%
  tidyr::pivot_wider(
    names_from = BCrace,
    names_glue = "BChairrace_{BCrace}",
    values_from = race_check,
    values_fill = 0
  ) %>%
  tidyr::pivot_wider(
    names_from = BCgender,
    names_glue = "BChairgender_{BCgender}",
    values_from = gender_check,
    values_fill = 0
  )
```

Create Biracial Categories in Year 2

Since the Year 1 data has an indicator for biracial CEOs and Board Chairs, we will have to create a boolean Biracial indicator variable in the Year 2 data that returns a 1 if a CEO or Board Chair belongs to 2 or more racial groups.

```
race_ceo_qns_bool <- race_gender_qns_bool[grepl("CEOrace", race_gender_qns_bool)]
race_bchair_qns_bool <- race_gender_qns_bool[grepl("BChairrace", race_gender_qns_bool)]

survey_recode_df <- survey_recode_df %>%
  dplyr::rowwise() %>%
  dplyr::mutate(
    CEOrace_Bi = ifelse(dplyr::between(
      sum(dplyr::c_across(tidyselect::all_of(race_ceo_qns_bool)), na.rm = TRUE), 2, 6
    ), 1, 0),
   BChairrace_Bi = ifelse(dplyr::between(
      sum(dplyr::c_across(tidyselect::all_of(race_bchair_qns_bool)), na.rm = TRUE), 2, 6
    ), 1, 0)
  )
```

Merge Year 1 Race and Gender Variables with Year 2

```
race_ceo_qns_bool <- c(race_ceo_qns_bool, "CEOrace_Bi")

for (race_var in race_ceo_qns_bool){
  survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    !! race_var := ifelse(
    EIN %in% year1_CEOchng$EIN,
    year1_CEOchng[[race_var]],
    .data[[race_var]]
    )
  )
}

gender_ceo_qns_bool <- race_gender_qns_bool[grepl("CEOgender", race_gender_qns_bool)]

for (gender_var in gender_ceo_qns_bool){
  survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    !! gender_var := ifelse(
    EIN %in% year1_CEOchng$EIN & gender_var %in% names(year1_CEOchng),
    year1_CEOchng[[gender_var]],
    .data[[gender_var]]
    )
  )
}

race_bchair_qns_bool <- c(race_bchair_qns_bool, "BChairrace_Bi")

for (race_var in race_bchair_qns_bool){
  survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    !! race_var := ifelse(
    EIN %in% year1_BCchng$EIN,
    year1_BCchng[[race_var]],
    .data[[race_var]]
    )
  )
}
```

```
gender_bchair_qns_bool <- race_gender_qns_bool[grepl("BChairgender", race_gender_qns_bool)

for (gender_var in gender_bchair_qns_bool){
  survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    !! gender_var := ifelse(
    EIN %in% year1_BCchng$EIN & gender_var %in% names(year1_BCchng),
    year1_BCchng[[gender_var]],
    .data[[gender_var]]
    )
  )
}
```

Recode New Race and Gender Variables

```
survey_recode_df <- create_survey_item(
  survey_recode_df,
  race_ceo_qns_bool,
  recode_vals = c(0, 1, 99),
  recode_labs = c("No", "Yes", "Incomplete"),
  missing_vals = c(99)
)

survey_recode_df <- create_survey_item(
  survey_recode_df,
  gender_ceo_qns_bool,
  recode_vals = c(0, 1, 99),
  recode_labs = c("No", "Yes", "Incomplete"),
  missing_vals = c(99)
)

survey_recode_df <- create_survey_item(
  survey_recode_df,
  race_bchair_qns_bool,
  recode_vals = c(0, 1, 99),
  recode_labs = c("No", "Yes", "Incomplete"),
  missing_vals = c(99)
)

survey_recode_df <- create_survey_item(
  survey_recode_df,
  gender_bchair_qns_bool,
  recode_vals = c(0, 1, 99),
  recode_labs = c("No", "Yes", "Incomplete"),
  missing_vals = c(99)
)
```

Create single Race/Gender column

While our race and gender columns are individual binary columns, this is not the case for year 1 and 3 results. Both those years contain single columns for race and gender variables for CEOs and Board Chairs respectively. For easier comparability, we create a new variable aggregating race and gender values for all our individual binary columns.

Aggregated Race Variable

Original Value	Description	Recode Label	Recode Value	Missing ?
1	Asian/Pacific Islander	AAPI	1	No
1	Black/African American	Black	2	No
1	Latinx/Hispanic	Hisp	3	No
1	Native American/American Indian	NativeAm	4	No
1	White	White	5	No
1	Bi/Multi-racial	Bi	6	No
1	Other (please specify)	Oth	7	No
0	Checkbox Unchecked	NA	Yes	

```
# Create New Race variables
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    CEOrace = dplyr::case_when(
      CEOrace_AAPI == 1 ~ 1,
      CEOrace_Black == 1 ~ 2,
      CEOrace_Hisp == 1 ~ 3,
      CEOrace_NativeAm == 1 ~ 4,
      CEOrace_White == 1 ~ 5,
      CEOrace_Bi == 1 ~ 6,
      CEOrace_Oth == 1 ~ 7,
      .default = NA
    ),
    BChairrace = dplyr::case_when(
      BChairrace_AAPI == 1 ~ 1,
      BChairrace_Black == 1 ~ 2,
      BChairrace_Hisp == 1 ~ 3,
      BChairrace_NativeAm == 1 ~ 4,
      BChairrace_White == 1 ~ 5,
      BChairrace_Bi == 1 ~ 6,
      BChairrace_Oth == 1 ~ 7,
      .default = NA
  ))

# Create Survey Item
survey_recode_df <- create_survey_item(
  survey_recode_df,
  c("CEOrace", "BChairrace"),
  recode_vals = c(1, 2, 3, 4, 5, 6, 7),
  recode_labs = c("AAPI", "Black", "Hisp", "NativeAm", "White", "Bi", "Oth"),
  missing_vals = c()
)
```

Aggregated Gender Variable

Original Value | Description | Recode Label | Recode Value | Missing ? |

|1|Man|Man|1|No| |1|Woman|Woman|2|No| |1|Trans|Trans|3|No| |1|Gender non-conforming/Non-Binary|NB|4|No| |1|Other (please specify)|Oth|5|No| |0|Checkbox Unchecked|NA|Yes|

```
# Create New Race variables
survey_recode_df <- survey_recode_df %>%
  dplyr::mutate(
    CEOgender = dplyr::case_when(
      CEOgender_Man == 1 ~ 1,
      CEOgender_Woman == 1 ~ 2,
      CEOgender_Trans == 1 ~ 3,
      CEOgender_NB == 1 ~ 4,
      CEOgender_Oth == 1 ~ 5,
      .default = NA
    ),
    BChairgender = dplyr::case_when(
      BChairgender_Man == 1 ~ 1,
      BChairgender_Woman == 1 ~ 2,
      BChairgender_Trans == 1 ~ 3,
      BChairgender_NB == 1 ~ 4,
      BChairgender_Oth == 1 ~ 5,
      .default = NA
  ))

# Create Survey Item
survey_recode_df <- create_survey_item(
  survey_recode_df,
  c("CEOgender", "BChairgender"),
  recode_vals = c(1, 2, 3, 4, 5),
  recode_labs = c("Man", "Woman", "Trans", "NB", "Oth"),
  missing_vals = c()
)
```

Validate Results

We can see that the number of valid responses for race and gender questions concerning the CEO and Board Chair have increased.

```
race_gender_qns <- c(race_gender_qns_bool, "CEOrace", "CEOgender", "BChairrace","BChairgen

codebook(
  survey_recode_df %>%
    dplyr::select(tidyselect::all_of(race_gender_qns[1]))
)
```

```
================================================================================

   CEOrace_AAPI

--------------------------------------------------------------------------------

   Storage mode: double
```

```
Measurement: nominal
Missing values: 99

Values and labels     N Valid Total

 0   'No'           467  99.8  56.8
 1   'Yes'            1   0.2   0.1
99 M 'Incomplete'     2         0.2
NA M                352        42.8
```

Evaluating Survey Completion Rates and Summing Section Completion

We next evaluate survey completion rates for each category of questions. We separate our variables into question categories and compute the proportion of valid responses.

```
# Create named list of variables

survey_completion_qns_ls <- list(
  "Program Changes" = program_change_qns_bool,
  "People Served" = people_served_qns_int,
  "Next Year's Demand" = demand_fct_qns,
  "Staff & Volunteers" = staff_qns_int,
  "Volunteer Importance" = volimportance_qns_fct,
  "Donor Importance" = donimportance_qns_fct,
  "Fundraising" = fundraise_qns_bool,
  "Major Gift Amount" = majorgift_qn_num,
  "Fundraising Changes" = fundraise_change_qns_fct,
  "Donor Changes" = fundraise_donor_qns_int,
  "Fundraising Sought/Received" = fundraise_skrcv_qns_bool,
  "Revenue Breakdown" = finance_revenue_qns_num,
  "Reserves" = reserve_qns_num,
  "CARES Funding" = cares_qns_num,
  "Finance Changes" = finance_chng_qns_bool,
  "Leadership Changes" = leadership_chng_qns_bool,
  "Race and Gender" = c(race_gender_qns_bool, "BChairrace_Bi", "CEOrace_Bi"),
  "External Affairs" = extaffairs_qns_fct,
  "Primary Concern" = primary_cncrn_qn_text)
# Include Response IDs into questions
survey_completion_qns <- c("ResponseId", unname(unlist(survey_completion_qns_ls)))
```

```r
# Subset Survey to only include questions to calculate response rates
survey_subset <- survey_recode_df %>%
  dplyr::select(
    tidyselect::all_of(survey_completion_qns)
    )
# Compute response rate
response_rate_ls <- purrr::imap(
  .x = survey_completion_qns_ls,
  .f = function(qns, qn_group) {
    # Compute responses for each question belonging to a group
    sum_ls <- purrr::map(qns,
           .f = function(qn){
             rs <- as.integer(survey_subset[[qn]])
             rs <- ifelse(! is.na(rs), 1, 0)
             return(rs)
           })
    # Perform rowwise sum for questions belonging to a group to get number of valid respon
    sum_qn <- purrr::pmap(sum_ls, sum, na.rm = TRUE)
    survey_subset[[qn_group]] <- unlist(sum_qn)
    # Compute response rate based on number of questions in that group
    output_df <- survey_subset %>%
      dplyr::mutate(!!qn_group := .data[[qn_group]] / length(qns)) %>%
      dplyr::select(tidyselect::all_of(c("ResponseId", qn_group)))
    return(output_df)
  },
  .progress = TRUE
)

response_rate_df <- purrr::reduce(response_rate_ls,
                                  dplyr::left_join,
                                  by = "ResponseId")

# Exclude Qualtrics Test Responses
testcases = c(
  "R_1lB6u4BoFZWFeXI",
  "R_3e3RyQttJeNzqLu",
  "R_3kzeIxsE73IIoq9",
  "R_2BxNQBmrv4Dr9Dn",
  "R_Q4WjxQdBfeZbn4B",
  "R_xFNLuOjPbg6iL6N",
  "R_3McyCOS2Gv7FNcd",
  "R_9YWW0OfssFUBEYh",
  "R_PNiTwNoQSfvuRkB",
  "R_111PYLkD7mpeFui",
  "R_a59rvIufWBZhAhr",
  "R_3MGb8by31yzxYW7",
  "R_1exbPd4NNpfWrFt",
  "R_2PwfUW9idAwPE1w",
  "R_3EunbLVD5Ce8b93",
  "R_2u055NoBulnQTvx",
  "R_Wl2yqs0OHXhhaTv",
  "R_1kSRidkXneLKy4u",
  "R_3j3DCg6oVhQWRe0"
)

response_rate_df <- response_rate_df %>%
  dplyr::filter(! ResponseId %in% weighted_survey_df$ResponseId)

response_rate_df
```

Save outputs

```
setwd("Y:/CNP/Generosity Commission/")
readr::write_csv(survey_recode_df,
                 "DATA-PREP/02-data-intermediate/02-wave-two/wave-02-data-intermediate-rec
```