Age (Covid CNS)

Abigail ter Kuile

13/01/2022

This R markdown: 1) Cleans the variable self-reported age at sign-up 2) Cleans date of birth (DOB) variables (age, month, year) 3) Creates DOB as a variable and calculates the variable age at sign-up from DOB 4) Cleans the variable age at sign-up calculated from DOB

Set up

Delete everything in your global environment

```
remove(list = ls())
```

Add the add_numeric function - used to convert character variables into numeric variables. Add the sumscores function - used to generate sumscores Add the package_check function - used to install and load dependencies

```
source(file = "../functions/add_numeric_1.R")
source(file = "../functions/remove_duplicates.R")
source(file = "../functions/sumscores.R")
source(file = "../functions/package_check.R")
source(file = "../functions/imp_check.R")
```

Note: always load tidyverse last

```
packages = c(
  "summarytools",
  "sjlabelled",
  "data.table",
  "Amelia",
  "lubridate",
  "tidyverse"
  )
package_check(packages)
```

```
Loading required package: summarytools
```

```
Warning: package 'summarytools' was built under R version 4.0.5
```

```
Registered S3 method overwritten by 'pryr':
  method      from
  print.bytes Rcpp

Loading required package: sjlabelled

Attaching package: 'sjlabelled'

The following object is masked from 'package:summarytools':

    unlabel

Loading required package: data.table

Warning: package 'data.table' was built under R version 4.0.5

Loading required package: Amelia

Warning: package 'Amelia' was built under R version 4.0.5

Loading required package: Rcpp

Warning: package 'Rcpp' was built under R version 4.0.5

##
## Amelia II: Multiple Imputation
## (Version 1.8.0, built: 2021-05-26)
## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## Refer to http://gking.harvard.edu/amelia/ for more information
##

Loading required package: lubridate

Warning: package 'lubridate' was built under R version 4.0.5

Attaching package: 'lubridate'

The following objects are masked from 'package:data.table':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:base':

    date, intersect, setdiff, union

Loading required package: tidyverse

Warning: package 'tidyverse' was built under R version 4.0.5
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.1 --

v ggplot2 3.3.5     v purrr   0.3.4
v tibble  3.1.5     v dplyr   1.0.7
v tidyr   1.1.4     v stringr 1.4.0
v readr   2.0.2     v forcats 0.5.1

Warning: package 'ggplot2' was built under R version 4.0.5

Warning: package 'tibble' was built under R version 4.0.5

Warning: package 'tidyr' was built under R version 4.0.5

Warning: package 'readr' was built under R version 4.0.5

Warning: package 'purrr' was built under R version 4.0.5

Warning: package 'dplyr' was built under R version 4.0.5

Warning: package 'stringr' was built under R version 4.0.5

Warning: package 'forcats' was built under R version 4.0.5

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x lubridate::as.difftime() masks base::as.difftime()
x forcats::as_factor()     masks sjlabelled::as_factor()
x dplyr::as_label()        masks ggplot2::as_label(), sjlabelled::as_label()
x dplyr::between()         masks data.table::between()
x lubridate::date()        masks base::date()
x dplyr::filter()          masks stats::filter()
x dplyr::first()           masks data.table::first()
x lubridate::hour()        masks data.table::hour()
x lubridate::intersect()   masks base::intersect()
x lubridate::isoweek()     masks data.table::isoweek()
x dplyr::lag()             masks stats::lag()
x dplyr::last()            masks data.table::last()
x lubridate::mday()        masks data.table::mday()
x lubridate::minute()      masks data.table::minute()
x lubridate::month()       masks data.table::month()
x lubridate::quarter()     masks data.table::quarter()
x lubridate::second()      masks data.table::second()
x lubridate::setdiff()     masks base::setdiff()
x purrr::transpose()       masks data.table::transpose()
x lubridate::union()       masks base::union()
x tibble::view()           masks summarytools::view()
x lubridate::wday()        masks data.table::wday()
x lubridate::week()        masks data.table::week()
x lubridate::yday()        masks data.table::yday()
x lubridate::year()        masks data.table::year()
```

Retrieve the recent date

We are using the recent date to save files with paste0() as an extension to not overwrite old versions

```
date = Sys.Date()
date
```

```
[1] "2022-02-22"
```

Read in file with path to ilovedata channel on Teams

```
source(file = "../credentials/paths.R")
```

Read in the data: Demographics

COVID CNS data

```
dat <- read_rds(
  file = paste0(ilovedata, "/data_raw/latest_freeze/covid_cns/baseline/dem_covid_cns.rds")
  )

# check variable names in dataframe
dat %>%
  colnames()
```

```
  [1] "externalDataReference"
  [2] "startDate"
  [3] "endDate"
  [4] "dem.day"
  [5] "dem.month"
  [6] "dem.year"
  [7] "dem.required_question_eligibility_criteria.txt"
  [8] "dem.what_gender_do_you_identify_with"
  [9] "dem.what_gender_do_you_identify_with.txt"
 [10] "dem.do_you_consider_yourself_to_be_transgender"
 [11] "dem.have_you_ever_been_pregnant"
 [12] "dem.what_is_your_sexual_orientation"
 [13] "dem.what_is_your_sexual_orientation.txt"
 [14] "dem.what_is_your_current_maritalrelationship_status"
 [15] "dem.what_is_your_current_maritalrelationship_status.txt"
 [16] "dem.how_would_you_describe_your_vision"
 [17] "dem.how_would_you_describe_your_hearing"
 [18] "dem.which_hand_do_you_usually_write_with"
 [19] "dem.college_or_university_degree"
 [20] "dem.a_levelsas_levels_or_equivalent"
 [21] "dem.o_levelsgcses_or_equivalent"
 [22] "dem.cses_or_equivalent"
 [23] "dem.nvq_or_hnd_or_hnc_or_equivalent"
```

```
[24] "dem.other_professional_qualifications_"
[25] "dem.other_professional_qualifications_text.txt"
[26] "dem.none_of_the_above"
[27] "dem.prefer_not_to_say"
[28] "dem.british_mixed_british"
[29] "dem.irish"
[30] "dem.northern_irish"
[31] "dem.any_other_white_background"
[32] "dem.white_and_black_caribbean"
[33] "dem.white_and_black_africa"
[34] "dem.white_and_asian"
[35] "dem.any_other_mixed_background"
[36] "dem.indian_or_british_indian"
[37] "dem.pakistani_or_british_pakistani"
[38] "dem.bangladeshi_or_british_bangladeshi"
[39] "dem.any_other_asian_background"
[40] "dem.caribbean"
[41] "dem.african"
[42] "dem.any_other_black_background"
[43] "dem.chinese"
[44] "dem.any_other_ethnic_group"
[45] "dem.other"
[46] "dem.othertext.txt"
[47] "dem.english"
[48] "dem.scottish"
[49] "dem.welsh"
[50] "dem.cornish"
[51] "dem.cypriot_"
[52] "dem.greek"
[53] "dem.greek_cypriot"
[54] "dem.italian"
[55] "dem.irish_traveller"
[56] "dem.traveller"
[57] "dem.gypsyromany"
[58] "dem.polish"
[59] "dem.republics_made_ussr"
[60] "dem.kosovan"
[61] "dem.albanian"
[62] "dem.bosnian"
[63] "dem.croatian"
[64] "dem.serbian"
[65] "dem.republics_made_yugoslavia"
[66] "dem.mixed_white"
[67] "dem.other_white_european_european_unspecified_european_mix"
[68] "dem.black_and_asian"
[69] "dem.black_and_chinese"
[70] "dem.black_and_white"
[71] "dem.chinese_and_white"
[72] "dem.asian_and_chinese"
[73] "dem.other_mixed_mixed_unspecified"
[74] "dem.other_mixed_mixed_unspecifiedtext.txt"
[75] "dem.mixed_asian"
[76] "dem.punjabi"
[77] "dem.kashmiri"
```

```
 [78] "dem.east_african_asian"
 [79] "dem.tamil"
 [80] "dem.sinhalese"
 [81] "dem.british_asian"
 [82] "dem.caribbean_asian"
 [83] "dem.other_asian_asian_unspecified"
 [84] "dem.other_asian_asian_unspecifiedtext.txt"
 [85] "dem.somali"
 [86] "dem.mixed_black"
 [87] "dem.nigerian"
 [88] "dem.black_british"
 [89] "dem.other_black_black_unspecified"
 [90] "dem.other_black_black_unspecifiedtext.txt"
 [91] "dem.is_english_your_first_language"
 [92] "dem.what_is_your_first_language"
 [93] "dem.what_is_your_first_language.txt"
 [94] "dem.please_select_your_preferred_units_of_measurement"
 [95] "dem.what_is_your_current_height"
 [96] "dem.what_is_your_current_height.1"
 [97] "dem.what_is_your_current_height.2"
 [98] "dem.pregnant_weigh_weight_provide"
 [99] "dem.pregnant_weigh_weight_provide.1"
[100] "dem.pregnant_weigh_weight_provide.2"
[101] "dem.pregnant_weighed_weight_provide"
[102] "dem.pregnant_weighed_weight_provide.1"
[103] "dem.pregnant_weighed_weight_provide.2"
[104] "dem.highest_weight"
[105] "dem.stopped_growing_adult_height"
[106] "dem.stopped_growing_adult_height.1"
[107] "dem.stopped_growing_adult_height.2"
[108] "dem.body_suffered_injury_involving"
[109] "dem.middle_wake_night_covid19"
[110] "dem.middle_wake_night_covid19.1"
[111] "dem.medical_history_birth_relevant"
[112] "dem.affects_concerned_live_memory"
[113] "dem.memory_problem_worse_year"
[114] "dem.based_confirm_living_question"
[115] "dem.diagnosed_required_question_covid19"
[116] "dem.long_ago_diagnosed_required"
[117] "dem.long_ago_diagnosed_required.1"
[118] "dem.diagnosed_covid19_experienced_similar"
[119] "dem.quality_rate_life"
[120] "dem.energy_everyday_life"
[121] "dem.opportunity_leisure_activities"
[122] "dem.money_day"
[123] "dem.middle_wake_night_trouble"
[124] "dem.affects_concerned_live_memory.1"
[125] "dem.affects_concerned_live_memory.2"
[126] "dem.has_your_memory_got_progressively_worse"
[127] "dem.vietnamese"
[128] "dem.filipino"
[129] "dem.malaysian"
[130] "dem.any_other_group"
[131] "dem.any_other_grouptext.txt"
```

```
[132] "dem.lowest_weight_adult_height"
[133] "dem.happy_general_health"
```

```
# Inspect dimensions of dataframe (number of rows and columns)
dat %>%
  dim()
```

```
[1] 235 133
```

Select & rename relevant columns

```
dat_id <- dat %>% # new dataset with ID
  drop_na(externalDataReference) %>% # Drop participants with no ID
  distinct(externalDataReference, .keep_all = TRUE) %>% # Changed to distinct due to NA coercion
  add_column(sample = "COVIDCNS",
             .after = "externalDataReference") %>% # Create new sample column
  select(
         ID = externalDataReference,# ID
         sample,
         startDate,
         endDate,
         dem.how_old_are_you_now.txt = dem.required_question_eligibility_criteria.txt, #self-reported a
         dem.day, # day of birth
         dem.month, # month of birth
         dem.year # year of birth
         )  %>%

  rename_with( ~ paste(.x, "unc", sep = "_"), starts_with("dem"))

# Inspect colnames
dat_id %>%
  colnames()
```

```
[1] "ID"                              "sample"
[3] "startDate"                       "endDate"
[5] "dem.how_old_are_you_now.txt_unc" "dem.day_unc"
[7] "dem.month_unc"                   "dem.year_unc"
```

Look at number of people excluded

```
# Inspect dimensions of new data set
dat_id %>%
  dim()
```

```
[1] 228   8
```

```
# Inspect number of rows dropped
excluded <- dim(dat_id)[1]-dim(dat)[1]
excluded
```

```
[1] -7
```

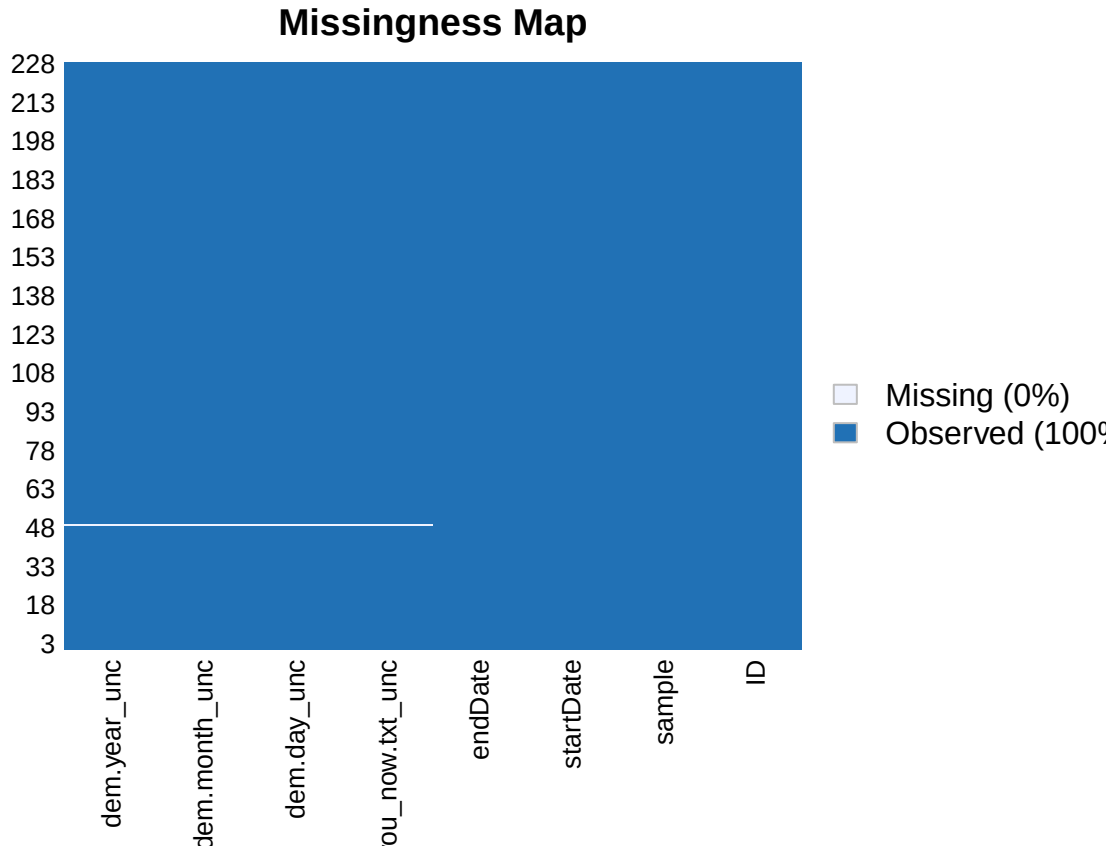
Check missingness by missmap

```
miss_map <- dat_id %>%
  missmap()
```

```
Warning: Unknown or uninitialised column: 'arguments'.

Warning: Unknown or uninitialised column: 'arguments'.

Warning: Unknown or uninitialised column: 'imputations'.
```




```
miss_map
```

```
NULL
```

Clean COVID CNS age variables

Age at sign-up (self-reported)

Inspect age variable

```
dat_id %>%
  freq(dem.how_old_are_you_now.txt_unc)
```

```
Frequencies
dat_id$dem.how_old_are_you_now.txt_unc
Label: How old are you now? *This question is required for eligibility criteria.
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
17	1	0.44	0.44	0.44	0.44
18	1	0.44	0.88	0.44	0.88
20	1	0.44	1.32	0.44	1.32
21	1	0.44	1.76	0.44	1.75
24	1	0.44	2.20	0.44	2.19
25	1	0.44	2.64	0.44	2.63
27	1	0.44	3.08	0.44	3.07
28	2	0.88	3.96	0.88	3.95
29	2	0.88	4.85	0.88	4.82
31	2	0.88	5.73	0.88	5.70
32	3	1.32	7.05	1.32	7.02
33	3	1.32	8.37	1.32	8.33
34	5	2.20	10.57	2.19	10.53
35	2	0.88	11.45	0.88	11.40
36	2	0.88	12.33	0.88	12.28
37	2	0.88	13.22	0.88	13.16
38	2	0.88	14.10	0.88	14.04
39	5	2.20	16.30	2.19	16.23
40	7	3.08	19.38	3.07	19.30
41	4	1.76	21.15	1.75	21.05
42	3	1.32	22.47	1.32	22.37
43	2	0.88	23.35	0.88	23.25
44	8	3.52	26.87	3.51	26.75
45	1	0.44	27.31	0.44	27.19
46	5	2.20	29.52	2.19	29.39
47	7	3.08	32.60	3.07	32.46
48	3	1.32	33.92	1.32	33.77
49	4	1.76	35.68	1.75	35.53
50	3	1.32	37.00	1.32	36.84
51	7	3.08	40.09	3.07	39.91
52	6	2.64	42.73	2.63	42.54
53	4	1.76	44.49	1.75	44.30
54	7	3.08	47.58	3.07	47.37
55	3	1.32	48.90	1.32	48.68
56	5	2.20	51.10	2.19	50.88
57	5	2.20	53.30	2.19	53.07
58	9	3.96	57.27	3.95	57.02
59	4	1.76	59.03	1.75	58.77
60	6	2.64	61.67	2.63	61.40
61	7	3.08	64.76	3.07	64.47
62	5	2.20	66.96	2.19	66.67
63	12	5.29	72.25	5.26	71.93
64	4	1.76	74.01	1.75	73.68

65	9	3.96	77.97	3.95	77.63
66	4	1.76	79.74	1.75	79.39
67	7	3.08	82.82	3.07	82.46
68	7	3.08	85.90	3.07	85.53
69	2	0.88	86.78	0.88	86.40
70	1	0.44	87.22	0.44	86.84
71	2	0.88	88.11	0.88	87.72
72	4	1.76	89.87	1.75	89.47
73	2	0.88	90.75	0.88	90.35
74	2	0.88	91.63	0.88	91.23
75	6	2.64	94.27	2.63	93.86
76	2	0.88	95.15	0.88	94.74
77	1	0.44	95.59	0.44	95.18
78	1	0.44	96.04	0.44	95.61
79	1	0.44	96.48	0.44	96.05
80	1	0.44	96.92	0.44	96.49
81	1	0.44	97.36	0.44	96.93
82	2	0.88	98.24	0.88	97.81
83	1	0.44	98.68	0.44	98.25
84	1	0.44	99.12	0.44	98.68
85	1	0.44	99.56	0.44	99.12
88	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

###Convert negative values of age to positive values and convert to numeric

```
dat_id <- dat_id %>%
  mutate(
    dem.how_old_are_you_now.txt_unc =
      abs(
        as.numeric(
        dem.how_old_are_you_now.txt_unc)
      )
  )
#Check
dat_id %>%

  freq(dem.how_old_are_you_now.txt_unc)
```

```
Frequencies
dat_id$dem.how_old_are_you_now.txt_unc
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
17	1	0.44	0.44	0.44	0.44
18	1	0.44	0.88	0.44	0.88
20	1	0.44	1.32	0.44	1.32
21	1	0.44	1.76	0.44	1.75
24	1	0.44	2.20	0.44	2.19
25	1	0.44	2.64	0.44	2.63
27	1	0.44	3.08	0.44	3.07

28	2	0.88	3.96	0.88	3.95
29	2	0.88	4.85	0.88	4.82
31	2	0.88	5.73	0.88	5.70
32	3	1.32	7.05	1.32	7.02
33	3	1.32	8.37	1.32	8.33
34	5	2.20	10.57	2.19	10.53
35	2	0.88	11.45	0.88	11.40
36	2	0.88	12.33	0.88	12.28
37	2	0.88	13.22	0.88	13.16
38	2	0.88	14.10	0.88	14.04
39	5	2.20	16.30	2.19	16.23
40	7	3.08	19.38	3.07	19.30
41	4	1.76	21.15	1.75	21.05
42	3	1.32	22.47	1.32	22.37
43	2	0.88	23.35	0.88	23.25
44	8	3.52	26.87	3.51	26.75
45	1	0.44	27.31	0.44	27.19
46	5	2.20	29.52	2.19	29.39
47	7	3.08	32.60	3.07	32.46
48	3	1.32	33.92	1.32	33.77
49	4	1.76	35.68	1.75	35.53
50	3	1.32	37.00	1.32	36.84
51	7	3.08	40.09	3.07	39.91
52	6	2.64	42.73	2.63	42.54
53	4	1.76	44.49	1.75	44.30
54	7	3.08	47.58	3.07	47.37
55	3	1.32	48.90	1.32	48.68
56	5	2.20	51.10	2.19	50.88
57	5	2.20	53.30	2.19	53.07
58	9	3.96	57.27	3.95	57.02
59	4	1.76	59.03	1.75	58.77
60	6	2.64	61.67	2.63	61.40
61	7	3.08	64.76	3.07	64.47
62	5	2.20	66.96	2.19	66.67
63	12	5.29	72.25	5.26	71.93
64	4	1.76	74.01	1.75	73.68
65	9	3.96	77.97	3.95	77.63
66	4	1.76	79.74	1.75	79.39
67	7	3.08	82.82	3.07	82.46
68	7	3.08	85.90	3.07	85.53
69	2	0.88	86.78	0.88	86.40
70	1	0.44	87.22	0.44	86.84
71	2	0.88	88.11	0.88	87.72
72	4	1.76	89.87	1.75	89.47
73	2	0.88	90.75	0.88	90.35
74	2	0.88	91.63	0.88	91.23
75	6	2.64	94.27	2.63	93.86
76	2	0.88	95.15	0.88	94.74
77	1	0.44	95.59	0.44	95.18
78	1	0.44	96.04	0.44	95.61
79	1	0.44	96.48	0.44	96.05
80	1	0.44	96.92	0.44	96.49
81	1	0.44	97.36	0.44	96.93
82	2	0.88	98.24	0.88	97.81

```
          83      1      0.44          98.68      0.44          98.25
          84      1      0.44          99.12      0.44          98.68
          85      1      0.44          99.56      0.44          99.12
          88      1      0.44         100.00      0.44          99.56
        <NA>      1                                0.44         100.00
       Total    228    100.00         100.00    100.00         100.00
```

Age outliers: - Lower: The data set should not have individuals younger than 16 - Upper: The oldest person in the world is 117 years.

Set age limits

```
age_lower_limit = 16
age_upper_limit = 117
```

Check for number of outliers in age variable

```
dat_id %>%
  filter(
    dem.how_old_are_you_now.txt_unc > age_upper_limit | # older than the age limit
      dem.how_old_are_you_now.txt_unc < age_lower_limit # younger than the age limit
    ) %>%
  nrow()
```

```
[1] 0
```

Recode age outliers to -666

```
dat_id <- dat_id%>%
    mutate(
      dem.how_old_are_you_now.txt = # remove "_unc" from the end to show that it is now cleaned
        if_else(
          dem.how_old_are_you_now.txt_unc > age_upper_limit |
            dem.how_old_are_you_now.txt_unc < age_lower_limit,
          true = -666,
          false = dem.how_old_are_you_now.txt_unc,
          missing = NA_real_
        )
    )
```

Inspect clean age variable

```
dat_id %>%
  freq(dem.how_old_are_you_now.txt)
```

Frequencies
dat_id$dem.how_old_are_you_now.txt
Type: Numeric

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
17	1	0.44	0.44	0.44	0.44
18	1	0.44	0.88	0.44	0.88
20	1	0.44	1.32	0.44	1.32
21	1	0.44	1.76	0.44	1.75
24	1	0.44	2.20	0.44	2.19
25	1	0.44	2.64	0.44	2.63
27	1	0.44	3.08	0.44	3.07
28	2	0.88	3.96	0.88	3.95
29	2	0.88	4.85	0.88	4.82
31	2	0.88	5.73	0.88	5.70
32	3	1.32	7.05	1.32	7.02
33	3	1.32	8.37	1.32	8.33
34	5	2.20	10.57	2.19	10.53
35	2	0.88	11.45	0.88	11.40
36	2	0.88	12.33	0.88	12.28
37	2	0.88	13.22	0.88	13.16
38	2	0.88	14.10	0.88	14.04
39	5	2.20	16.30	2.19	16.23
40	7	3.08	19.38	3.07	19.30
41	4	1.76	21.15	1.75	21.05
42	3	1.32	22.47	1.32	22.37
43	2	0.88	23.35	0.88	23.25
44	8	3.52	26.87	3.51	26.75
45	1	0.44	27.31	0.44	27.19
46	5	2.20	29.52	2.19	29.39
47	7	3.08	32.60	3.07	32.46
48	3	1.32	33.92	1.32	33.77
49	4	1.76	35.68	1.75	35.53
50	3	1.32	37.00	1.32	36.84
51	7	3.08	40.09	3.07	39.91
52	6	2.64	42.73	2.63	42.54
53	4	1.76	44.49	1.75	44.30
54	7	3.08	47.58	3.07	47.37
55	3	1.32	48.90	1.32	48.68
56	5	2.20	51.10	2.19	50.88
57	5	2.20	53.30	2.19	53.07
58	9	3.96	57.27	3.95	57.02
59	4	1.76	59.03	1.75	58.77
60	6	2.64	61.67	2.63	61.40
61	7	3.08	64.76	3.07	64.47
62	5	2.20	66.96	2.19	66.67
63	12	5.29	72.25	5.26	71.93
64	4	1.76	74.01	1.75	73.68
65	9	3.96	77.97	3.95	77.63
66	4	1.76	79.74	1.75	79.39
67	7	3.08	82.82	3.07	82.46
68	7	3.08	85.90	3.07	85.53
69	2	0.88	86.78	0.88	86.40

70	1	0.44	87.22	0.44	86.84
71	2	0.88	88.11	0.88	87.72
72	4	1.76	89.87	1.75	89.47
73	2	0.88	90.75	0.88	90.35
74	2	0.88	91.63	0.88	91.23
75	6	2.64	94.27	2.63	93.86
76	2	0.88	95.15	0.88	94.74
77	1	0.44	95.59	0.44	95.18
78	1	0.44	96.04	0.44	95.61
79	1	0.44	96.48	0.44	96.05
80	1	0.44	96.92	0.44	96.49
81	1	0.44	97.36	0.44	96.93
82	2	0.88	98.24	0.88	97.81
83	1	0.44	98.68	0.44	98.25
84	1	0.44	99.12	0.44	98.68
85	1	0.44	99.56	0.44	99.12
88	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Age at sign up (based on DOB)

Inspect DOB variables

```
dat_id %>%
  freq(dem.day_unc)
```

```
Frequencies
dat_id$dem.day_unc
Label: What is your date of birth? * This question is required. Day
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	6	2.64	2.64	2.63	2.63
2	8	3.52	6.17	3.51	6.14
3	6	2.64	8.81	2.63	8.77
4	7	3.08	11.89	3.07	11.84
5	11	4.85	16.74	4.82	16.67
6	8	3.52	20.26	3.51	20.18
7	8	3.52	23.79	3.51	23.68
8	9	3.96	27.75	3.95	27.63
9	9	3.96	31.72	3.95	31.58
10	6	2.64	34.36	2.63	34.21
11	9	3.96	38.33	3.95	38.16
12	9	3.96	42.29	3.95	42.11
13	6	2.64	44.93	2.63	44.74
14	11	4.85	49.78	4.82	49.56
15	4	1.76	51.54	1.75	51.32
16	7	3.08	54.63	3.07	54.39
17	10	4.41	59.03	4.39	58.77
18	3	1.32	60.35	1.32	60.09

19	7	3.08	63.44	3.07	63.16
20	6	2.64	66.08	2.63	65.79
21	7	3.08	69.16	3.07	68.86
22	7	3.08	72.25	3.07	71.93
23	11	4.85	77.09	4.82	76.75
24	6	2.64	79.74	2.63	79.39
25	6	2.64	82.38	2.63	82.02
26	8	3.52	85.90	3.51	85.53
27	8	3.52	89.43	3.51	89.04
28	8	3.52	92.95	3.51	92.54
29	4	1.76	94.71	1.75	94.30
30	10	4.41	99.12	4.39	98.68
31	2	0.88	100.00	0.88	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

```
dat_id %>%
  freq(dem.month_unc)
```

```
Frequencies
dat_id$dem.month_unc
Label: What is your date of birth? * This question is required. Month
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	23	10.13	10.13	10.09	10.09
2	19	8.37	18.50	8.33	18.42
3	15	6.61	25.11	6.58	25.00
4	22	9.69	34.80	9.65	34.65
5	24	10.57	45.37	10.53	45.18
6	17	7.49	52.86	7.46	52.63
7	25	11.01	63.88	10.96	63.60
8	23	10.13	74.01	10.09	73.68
9	14	6.17	80.18	6.14	79.82
10	13	5.73	85.90	5.70	85.53
11	9	3.96	89.87	3.95	89.47
12	23	10.13	100.00	10.09	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

```
dat_id %>%
  freq(dem.year_unc)
```

```
Frequencies
dat_id$dem.year_unc
Label: What is your date of birth? * This question is required. Year
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
14	1	0.44	0.44	0.44	0.44
17	1	0.44	0.88	0.44	0.88

18	1	0.44	1.32	0.44	1.32
19	1	0.44	1.76	0.44	1.75
20	3	1.32	3.08	1.32	3.07
22	2	0.88	3.96	0.88	3.95
24	1	0.44	4.41	0.44	4.39
25	1	0.44	4.85	0.44	4.82
26	3	1.32	6.17	1.32	6.14
27	6	2.64	8.81	2.63	8.77
29	4	1.76	10.57	1.75	10.53
30	3	1.32	11.89	1.32	11.84
31	1	0.44	12.33	0.44	12.28
32	2	0.88	13.22	0.88	13.16
33	3	1.32	14.54	1.32	14.47
34	7	3.08	17.62	3.07	17.54
35	8	3.52	21.15	3.51	21.05
36	5	2.20	23.35	2.19	23.25
37	5	2.20	25.55	2.19	25.44
38	7	3.08	28.63	3.07	28.51
39	13	5.73	34.36	5.70	34.21
40	3	1.32	35.68	1.32	35.53
41	10	4.41	40.09	4.39	39.91
42	3	1.32	41.41	1.32	41.23
43	6	2.64	44.05	2.63	43.86
44	9	3.96	48.02	3.95	47.81
45	2	0.88	48.90	0.88	48.68
46	5	2.20	51.10	2.19	50.88
47	5	2.20	53.30	2.19	53.07
48	5	2.20	55.51	2.19	55.26
49	4	1.76	57.27	1.75	57.02
50	7	3.08	60.35	3.07	60.09
51	7	3.08	63.44	3.07	63.16
52	3	1.32	64.76	1.32	64.47
53	3	1.32	66.08	1.32	65.79
54	3	1.32	67.40	1.32	67.11
55	8	3.52	70.93	3.51	70.61
56	4	1.76	72.69	1.75	72.37
57	1	0.44	73.13	0.44	72.81
58	9	3.96	77.09	3.95	76.75
59	3	1.32	78.41	1.32	78.07
60	2	0.88	79.30	0.88	78.95
61	4	1.76	81.06	1.75	80.70
62	7	3.08	84.14	3.07	83.77
63	4	1.76	85.90	1.75	85.53
64	3	1.32	87.22	1.32	86.84
65	1	0.44	87.67	0.44	87.28
66	3	1.32	88.99	1.32	88.60
67	2	0.88	89.87	0.88	89.47
68	4	1.76	91.63	1.75	91.23
69	3	1.32	92.95	1.32	92.54
70	3	1.32	94.27	1.32	93.86
71	2	0.88	95.15	0.88	94.74
73	2	0.88	96.04	0.88	95.61
74	2	0.88	96.92	0.88	96.49
75	1	0.44	97.36	0.44	96.93

77	1	0.44	97.80	0.44	97.37
78	1	0.44	98.24	0.44	97.81
81	1	0.44	98.68	0.44	98.25
82	1	0.44	99.12	0.44	98.68
84	1	0.44	99.56	0.44	99.12
85	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

###Convert day, month, year to numeric and negative values to positive values Day conversion numeric and positive values

```
dat_id <- dat_id %>%
  mutate(
    dem.day_unc =
      abs(
        as.numeric(
        dem.day_unc)
      )
  )
#Check
dat_id %>%
  freq(dem.day_unc)
```

```
Frequencies
dat_id$dem.day_unc
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	6	2.64	2.64	2.63	2.63
2	8	3.52	6.17	3.51	6.14
3	6	2.64	8.81	2.63	8.77
4	7	3.08	11.89	3.07	11.84
5	11	4.85	16.74	4.82	16.67
6	8	3.52	20.26	3.51	20.18
7	8	3.52	23.79	3.51	23.68
8	9	3.96	27.75	3.95	27.63
9	9	3.96	31.72	3.95	31.58
10	6	2.64	34.36	2.63	34.21
11	9	3.96	38.33	3.95	38.16
12	9	3.96	42.29	3.95	42.11
13	6	2.64	44.93	2.63	44.74
14	11	4.85	49.78	4.82	49.56
15	4	1.76	51.54	1.75	51.32
16	7	3.08	54.63	3.07	54.39
17	10	4.41	59.03	4.39	58.77
18	3	1.32	60.35	1.32	60.09
19	7	3.08	63.44	3.07	63.16
20	6	2.64	66.08	2.63	65.79
21	7	3.08	69.16	3.07	68.86
22	7	3.08	72.25	3.07	71.93
23	11	4.85	77.09	4.82	76.75

24	6	2.64	79.74	2.63	79.39
25	6	2.64	82.38	2.63	82.02
26	8	3.52	85.90	3.51	85.53
27	8	3.52	89.43	3.51	89.04
28	8	3.52	92.95	3.51	92.54
29	4	1.76	94.71	1.75	94.30
30	10	4.41	99.12	4.39	98.68
31	2	0.88	100.00	0.88	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Month conversion numeric and positive values

```
dat_id <- dat_id %>%
  mutate(
    dem.month_unc =
      abs(
        as.numeric(
        dem.month_unc)
      )
  )
#Check
dat_id %>%
  freq(dem.month_unc)
```

```
Frequencies
dat_id$dem.month_unc
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	23	10.13	10.13	10.09	10.09
2	19	8.37	18.50	8.33	18.42
3	15	6.61	25.11	6.58	25.00
4	22	9.69	34.80	9.65	34.65
5	24	10.57	45.37	10.53	45.18
6	17	7.49	52.86	7.46	52.63
7	25	11.01	63.88	10.96	63.60
8	23	10.13	74.01	10.09	73.68
9	14	6.17	80.18	6.14	79.82
10	13	5.73	85.90	5.70	85.53
11	9	3.96	89.87	3.95	89.47
12	23	10.13	100.00	10.09	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Year conversion numeric and positive values

```
dat_id <- dat_id %>%
  mutate(
    dem.year_unc =
      abs(
        as.numeric(
```

```
        dem.year_unc)
      )
  )
#Check
dat_id %>%
  freq(dem.year_unc)
```

```
Frequencies
dat_id$dem.year_unc
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
14	1	0.44	0.44	0.44	0.44
17	1	0.44	0.88	0.44	0.88
18	1	0.44	1.32	0.44	1.32
19	1	0.44	1.76	0.44	1.75
20	3	1.32	3.08	1.32	3.07
22	2	0.88	3.96	0.88	3.95
24	1	0.44	4.41	0.44	4.39
25	1	0.44	4.85	0.44	4.82
26	3	1.32	6.17	1.32	6.14
27	6	2.64	8.81	2.63	8.77
29	4	1.76	10.57	1.75	10.53
30	3	1.32	11.89	1.32	11.84
31	1	0.44	12.33	0.44	12.28
32	2	0.88	13.22	0.88	13.16
33	3	1.32	14.54	1.32	14.47
34	7	3.08	17.62	3.07	17.54
35	8	3.52	21.15	3.51	21.05
36	5	2.20	23.35	2.19	23.25
37	5	2.20	25.55	2.19	25.44
38	7	3.08	28.63	3.07	28.51
39	13	5.73	34.36	5.70	34.21
40	3	1.32	35.68	1.32	35.53
41	10	4.41	40.09	4.39	39.91
42	3	1.32	41.41	1.32	41.23
43	6	2.64	44.05	2.63	43.86
44	9	3.96	48.02	3.95	47.81
45	2	0.88	48.90	0.88	48.68
46	5	2.20	51.10	2.19	50.88
47	5	2.20	53.30	2.19	53.07
48	5	2.20	55.51	2.19	55.26
49	4	1.76	57.27	1.75	57.02
50	7	3.08	60.35	3.07	60.09
51	7	3.08	63.44	3.07	63.16
52	3	1.32	64.76	1.32	64.47
53	3	1.32	66.08	1.32	65.79
54	3	1.32	67.40	1.32	67.11
55	8	3.52	70.93	3.51	70.61
56	4	1.76	72.69	1.75	72.37
57	1	0.44	73.13	0.44	72.81
58	9	3.96	77.09	3.95	76.75

59	3	1.32	78.41	1.32	78.07
60	2	0.88	79.30	0.88	78.95
61	4	1.76	81.06	1.75	80.70
62	7	3.08	84.14	3.07	83.77
63	4	1.76	85.90	1.75	85.53
64	3	1.32	87.22	1.32	86.84
65	1	0.44	87.67	0.44	87.28
66	3	1.32	88.99	1.32	88.60
67	2	0.88	89.87	0.88	89.47
68	4	1.76	91.63	1.75	91.23
69	3	1.32	92.95	1.32	92.54
70	3	1.32	94.27	1.32	93.86
71	2	0.88	95.15	0.88	94.74
73	2	0.88	96.04	0.88	95.61
74	2	0.88	96.92	0.88	96.49
75	1	0.44	97.36	0.44	96.93
77	1	0.44	97.80	0.44	97.37
78	1	0.44	98.24	0.44	97.81
81	1	0.44	98.68	0.44	98.25
82	1	0.44	99.12	0.44	98.68
84	1	0.44	99.56	0.44	99.12
85	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Add values to birthyear

Birth year coding: Oldest person self-reported age is 88 (born in 1933, calculated using participant startdate) - coded values start from 14 (participant with age 88) which needs to be converted to 1933 (add 1919 years)

```
#add values to birth year
dat_id <- dat_id %>%
  mutate(dem.year_unc = dem.year_unc + 1919) ##add 1919 years

dat_id %>%
  freq(dem.year_unc)
```

```
Frequencies
dat_id$dem.year_unc
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1933	1	0.44	0.44	0.44	0.44
1936	1	0.44	0.88	0.44	0.88
1937	1	0.44	1.32	0.44	1.32
1938	1	0.44	1.76	0.44	1.75
1939	3	1.32	3.08	1.32	3.07
1941	2	0.88	3.96	0.88	3.95
1943	1	0.44	4.41	0.44	4.39
1944	1	0.44	4.85	0.44	4.82
1945	3	1.32	6.17	1.32	6.14
1946	6	2.64	8.81	2.63	8.77

1948	4	1.76	10.57	1.75	10.53
1949	3	1.32	11.89	1.32	11.84
1950	1	0.44	12.33	0.44	12.28
1951	2	0.88	13.22	0.88	13.16
1952	3	1.32	14.54	1.32	14.47
1953	7	3.08	17.62	3.07	17.54
1954	8	3.52	21.15	3.51	21.05
1955	5	2.20	23.35	2.19	23.25
1956	5	2.20	25.55	2.19	25.44
1957	7	3.08	28.63	3.07	28.51
1958	13	5.73	34.36	5.70	34.21
1959	3	1.32	35.68	1.32	35.53
1960	10	4.41	40.09	4.39	39.91
1961	3	1.32	41.41	1.32	41.23
1962	6	2.64	44.05	2.63	43.86
1963	9	3.96	48.02	3.95	47.81
1964	2	0.88	48.90	0.88	48.68
1965	5	2.20	51.10	2.19	50.88
1966	5	2.20	53.30	2.19	53.07
1967	5	2.20	55.51	2.19	55.26
1968	4	1.76	57.27	1.75	57.02
1969	7	3.08	60.35	3.07	60.09
1970	7	3.08	63.44	3.07	63.16
1971	3	1.32	64.76	1.32	64.47
1972	3	1.32	66.08	1.32	65.79
1973	3	1.32	67.40	1.32	67.11
1974	8	3.52	70.93	3.51	70.61
1975	4	1.76	72.69	1.75	72.37
1976	1	0.44	73.13	0.44	72.81
1977	9	3.96	77.09	3.95	76.75
1978	3	1.32	78.41	1.32	78.07
1979	2	0.88	79.30	0.88	78.95
1980	4	1.76	81.06	1.75	80.70
1981	7	3.08	84.14	3.07	83.77
1982	4	1.76	85.90	1.75	85.53
1983	3	1.32	87.22	1.32	86.84
1984	1	0.44	87.67	0.44	87.28
1985	3	1.32	88.99	1.32	88.60
1986	2	0.88	89.87	0.88	89.47
1987	4	1.76	91.63	1.75	91.23
1988	3	1.32	92.95	1.32	92.54
1989	3	1.32	94.27	1.32	93.86
1990	2	0.88	95.15	0.88	94.74
1992	2	0.88	96.04	0.88	95.61
1993	2	0.88	96.92	0.88	96.49
1994	1	0.44	97.36	0.44	96.93
1996	1	0.44	97.80	0.44	97.37
1997	1	0.44	98.24	0.44	97.81
2000	1	0.44	98.68	0.44	98.25
2001	1	0.44	99.12	0.44	98.68
2003	1	0.44	99.56	0.44	99.12
2004	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Set minimum and maximum values

```
# Day
day.min.scale = 1
day.max.scale = 31

# Month
month.min.scale = 1
month.max.scale = 12

# Year
year.min.scale = 1899
year.max.scale = 2021 # note max age set later in age_upper_limit
```

Clean dem.day

```
dat_id <- dat_id %>%
  mutate(dem.day_unc_clean =
           case_when(dem.day_unc < day.min.scale | dem.day_unc > day.max.scale ~ -666, # implausible va
                     TRUE ~ dem.day_unc)
         ) # leave as is


# Check for implausible values
dat_day_imp_n <- dat_id %>%
  filter(dem.day_unc_clean == -666) %>%
  nrow()

# Check
dat_day_imp_n
```

```
[1] 0
```

```
# If statement
if (dat_day_imp_n == 0) {
  print(paste0("The number of implausible values in the COVID CNS day of birth variable is ", dat_day_i
  setnames(dat_id,
         old = "dem.day_unc_clean",
         new = "dem.day")
} else {
  print(paste0("The number of implausible values in the COVID CNS day of birth variable is ", dat_day_i
  setnames(dat_id,
         old = "dem.day_unc_clean",
         new = "dem.day")
}
```

```
[1] "The number of implausible values in the COVID CNS day of birth variable is 0. This is fine."
```

```r
# Check
colnames(dat_id)
```

```
 [1] "ID"                                "sample"
 [3] "startDate"                         "endDate"
 [5] "dem.how_old_are_you_now.txt_unc" "dem.day_unc"
 [7] "dem.month_unc"                     "dem.year_unc"
 [9] "dem.how_old_are_you_now.txt"     "dem.day"
```

```r
# Check cleaned variable
dat_id %>%
  freq(dem.day)
```

```
Frequencies
dat_id$dem.day
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	6	2.64	2.64	2.63	2.63
2	8	3.52	6.17	3.51	6.14
3	6	2.64	8.81	2.63	8.77
4	7	3.08	11.89	3.07	11.84
5	11	4.85	16.74	4.82	16.67
6	8	3.52	20.26	3.51	20.18
7	8	3.52	23.79	3.51	23.68
8	9	3.96	27.75	3.95	27.63
9	9	3.96	31.72	3.95	31.58
10	6	2.64	34.36	2.63	34.21
11	9	3.96	38.33	3.95	38.16
12	9	3.96	42.29	3.95	42.11
13	6	2.64	44.93	2.63	44.74
14	11	4.85	49.78	4.82	49.56
15	4	1.76	51.54	1.75	51.32
16	7	3.08	54.63	3.07	54.39
17	10	4.41	59.03	4.39	58.77
18	3	1.32	60.35	1.32	60.09
19	7	3.08	63.44	3.07	63.16
20	6	2.64	66.08	2.63	65.79
21	7	3.08	69.16	3.07	68.86
22	7	3.08	72.25	3.07	71.93
23	11	4.85	77.09	4.82	76.75
24	6	2.64	79.74	2.63	79.39
25	6	2.64	82.38	2.63	82.02
26	8	3.52	85.90	3.51	85.53
27	8	3.52	89.43	3.51	89.04
28	8	3.52	92.95	3.51	92.54
29	4	1.76	94.71	1.75	94.30
30	10	4.41	99.12	4.39	98.68
31	2	0.88	100.00	0.88	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Clean dem.month

```
dat_id <- dat_id %>%
  mutate(dem.month_unc_clean =
           case_when(dem.month_unc < month.min.scale | dem.month_unc > month.max.scale ~ -666, # implau
                     TRUE ~ dem.month_unc)
         ) # leave as is

# Check for implausible values
dat_month_imp_n <- dat_id %>%
  filter(dem.month_unc_clean == -666) %>%
  nrow()

# Check
dat_month_imp_n
```

```
[1] 0
```

```
# If statement
if (dat_month_imp_n == 0) {
  print(paste0("The number of implausible values in the COVID CNS month of birth variable is ", dat_mon
  setnames(dat_id,
           old = "dem.month_unc_clean",
           new = "dem.month")
} else {
  print(paste0("The number of implausible values in the COVID CNS month of birth variable is ", dat_mon
  setnames(dat_id,
           old = "dem.month_unc_clean",
           new = "dem.month")
}
```

```
[1] "The number of implausible values in the COVID CNS month of birth variable is 0. This is fine."
```

```
# Check
colnames(dat_id)
```

```
 [1] "ID"                                "sample"
 [3] "startDate"                         "endDate"
 [5] "dem.how_old_are_you_now.txt_unc" "dem.day_unc"
 [7] "dem.month_unc"                     "dem.year_unc"
 [9] "dem.how_old_are_you_now.txt"     "dem.day"
[11] "dem.month"
```

```
# Check clean variable
dat_id %>%
  freq(dem.month)
```

```
Frequencies
dat_id$dem.month
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1	23	10.13	10.13	10.09	10.09
2	19	8.37	18.50	8.33	18.42
3	15	6.61	25.11	6.58	25.00
4	22	9.69	34.80	9.65	34.65
5	24	10.57	45.37	10.53	45.18
6	17	7.49	52.86	7.46	52.63
7	25	11.01	63.88	10.96	63.60
8	23	10.13	74.01	10.09	73.68
9	14	6.17	80.18	6.14	79.82
10	13	5.73	85.90	5.70	85.53
11	9	3.96	89.87	3.95	89.47
12	23	10.13	100.00	10.09	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

The month variable should now be clean.

Clean dem.year

```
dat_id <- dat_id %>%
  mutate(dem.year_unc_clean =
           case_when(dem.year_unc < year.min.scale | dem.year_unc > year.max.scale ~ -666,  # implausib
                     TRUE ~ dem.year_unc)
         )

# Check for implausible values
year_imp_n <- dat_id %>%
  filter(dem.year_unc_clean == -666) %>%
  nrow()

# Check
year_imp_n
```

```
[1] 0
```

```
# If statement
if (year_imp_n == 0) {
  print(paste0("The number of implausible values in the COVID CNS year of birth variable is ", year_imp
  setnames(dat_id,
           old = "dem.year_unc_clean",
           new = "dem.year")
} else {
  print(paste0("The number of implausible values in the COVID CNS year of birth variable is ", year_imp
  setnames(dat_id,
           old = "dem.year_unc_clean",
           new = "dem.year")
}
```

```
[1] "The number of implausible values in the COVID CNS year of birth variable is 0. This is fine."
```

```
# Check
dat_id %>%
  freq(dem.year)
```

```
Frequencies
dat_id$dem.year
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
1933	1	0.44	0.44	0.44	0.44
1936	1	0.44	0.88	0.44	0.88
1937	1	0.44	1.32	0.44	1.32
1938	1	0.44	1.76	0.44	1.75
1939	3	1.32	3.08	1.32	3.07
1941	2	0.88	3.96	0.88	3.95
1943	1	0.44	4.41	0.44	4.39
1944	1	0.44	4.85	0.44	4.82
1945	3	1.32	6.17	1.32	6.14
1946	6	2.64	8.81	2.63	8.77
1948	4	1.76	10.57	1.75	10.53
1949	3	1.32	11.89	1.32	11.84
1950	1	0.44	12.33	0.44	12.28
1951	2	0.88	13.22	0.88	13.16
1952	3	1.32	14.54	1.32	14.47
1953	7	3.08	17.62	3.07	17.54
1954	8	3.52	21.15	3.51	21.05
1955	5	2.20	23.35	2.19	23.25
1956	5	2.20	25.55	2.19	25.44
1957	7	3.08	28.63	3.07	28.51
1958	13	5.73	34.36	5.70	34.21
1959	3	1.32	35.68	1.32	35.53
1960	10	4.41	40.09	4.39	39.91
1961	3	1.32	41.41	1.32	41.23
1962	6	2.64	44.05	2.63	43.86
1963	9	3.96	48.02	3.95	47.81
1964	2	0.88	48.90	0.88	48.68
1965	5	2.20	51.10	2.19	50.88
1966	5	2.20	53.30	2.19	53.07
1967	5	2.20	55.51	2.19	55.26
1968	4	1.76	57.27	1.75	57.02
1969	7	3.08	60.35	3.07	60.09
1970	7	3.08	63.44	3.07	63.16
1971	3	1.32	64.76	1.32	64.47
1972	3	1.32	66.08	1.32	65.79
1973	3	1.32	67.40	1.32	67.11
1974	8	3.52	70.93	3.51	70.61
1975	4	1.76	72.69	1.75	72.37
1976	1	0.44	73.13	0.44	72.81
1977	9	3.96	77.09	3.95	76.75
1978	3	1.32	78.41	1.32	78.07
1979	2	0.88	79.30	0.88	78.95
1980	4	1.76	81.06	1.75	80.70

1981	7	3.08	84.14	3.07	83.77
1982	4	1.76	85.90	1.75	85.53
1983	3	1.32	87.22	1.32	86.84
1984	1	0.44	87.67	0.44	87.28
1985	3	1.32	88.99	1.32	88.60
1986	2	0.88	89.87	0.88	89.47
1987	4	1.76	91.63	1.75	91.23
1988	3	1.32	92.95	1.32	92.54
1989	3	1.32	94.27	1.32	93.86
1990	2	0.88	95.15	0.88	94.74
1992	2	0.88	96.04	0.88	95.61
1993	2	0.88	96.92	0.88	96.49
1994	1	0.44	97.36	0.44	96.93
1996	1	0.44	97.80	0.44	97.37
1997	1	0.44	98.24	0.44	97.81
2000	1	0.44	98.68	0.44	98.25
2001	1	0.44	99.12	0.44	98.68
2003	1	0.44	99.56	0.44	99.12
2004	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

dat_id = clean data set, all implausible values are set to -666 However, we need to drop these values to NA in order to calculate DOB (needed for the next steps)

Create a new data set where implausible values are dropped to NA

Drop all -666 to NA

```
dat_no_imps <- dat_id %>%
  mutate_if(is.numeric, ~na_if(., -666)) # Implausible value
```

Drop all variables with "_unc" on the end

```
dat_clean <- dat_no_imps %>%
  select(!contains("_unc")) # selects ID, sample and drops all uncleaned variables

# Check (there should be no variables with "_unc" in the name now)
colnames(dat_clean)
```

```
[1] "ID"                          "sample"
[3] "startDate"                   "endDate"
[5] "dem.how_old_are_you_now.txt" "dem.day"
[7] "dem.month"                   "dem.year"
```

Create age variable from date of birth

Use lubridate for this:

```
dat_clean <- dat_clean %>%
  mutate(dem.dob = make_date(dem.year, dem.month, dem.day))

# check
dat_clean %>%
  select(dem.day,
         dem.month,
         dem.year,
         dem.dob) %>%
  head()
```

```
# A tibble: 6 x 4
  dem.day dem.month dem.year dem.dob
    <dbl>     <dbl>    <dbl> <date>
1      12        11     1962 1962-11-12
2      11         4     1954 1954-04-11
3       8         2     1956 1956-02-08
4      27         3     1985 1985-03-27
5      26         9     1963 1963-09-26
6      16         2     1960 1960-02-16
```

Calculate age from birth date and startdate

note: using startdate instead of enddate and this increases N (some participants did not reach the end of the questionaire)

```
dat_clean$dem.dob_age <- interval(
    start = dat_clean$dem.dob,
    end = dat_clean$startDate) %/% # use modulo to round down by %/%
        duration(num = 1, units = "years")


# check COVID CNS age variables
dat_clean %>%
  select(dem.dob,
         dem.dob_age,
         dem.how_old_are_you_now.txt) %>%
  head()
```

```
# A tibble: 6 x 3
  dem.dob    dem.dob_age dem.how_old_are_you_now.txt
  <date>           <dbl>                      <dbl>
1 1962-11-12          58                         58
2 1954-04-11          66                         66
3 1956-02-08          65                         65
4 1985-03-27          35                         35
5 1963-09-26          57                         57
6 1960-02-16          61                         61
```

Inspect difference self-report age and DOB age

```
# check COVID CNS
diff_age_variabl_n <- dat_clean %>%
  filter(
    dem.dob_age != dem.how_old_are_you_now.txt) %>%
  select(ID,
         dem.dob_age,
         dem.how_old_are_you_now.txt,
         dem.dob)

diff_age_variabl_n
```

```
# A tibble: 17 x 4
   ID       dem.dob_age dem.how_old_are_you_now.txt dem.dob
   <chr>          <dbl>                       <dbl> <date>
 1 CNS01013          67                          68 1953-09-12
 2 CNS01018          61                          62 1959-07-07
 3 CNS02020          64                          65 1956-12-04
 4 CNS02024          74                          75 1946-10-06
 5 CNS01036          52                          51 1969-06-14
 6 CNS01044          76                          75 1945-09-06
 7 CNS07001          53                          54 1967-10-08
 8 CNS07011          40                          41 1980-12-08
 9 CNS01102          59                          58 1962-04-25
10 CNS01110          84                          85 1936-12-05
11 CNS06034          51                          50 1970-01-17
12 CNS01140          58                          59 1963-08-01
13 CNS02064          73                          74 1948-07-21
14 CNS07023          63                          64 1958-07-20
15 CNS07025          70                          71 1951-07-16
16 CNS05017          64                          65 1957-09-02
17 CNS06021          45                          44 1977-01-21
```

`diff_age_variabl_n` COVID CNS participants self-report a different age to their age calculated from date of birth.

Convert dem.dob_age to numeric and negative values to positive values

```
dat_clean <- dat_clean %>%
  mutate(
    dem.dob_age =
      abs(
        as.numeric(
        dem.dob_age)
      )
  )
#Check
dat_clean %>%
  freq(dem.dob_age)
```

Frequencies
dat_clean$dem.dob_age
Type: Numeric

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
17	1	0.44	0.44	0.44	0.44
18	1	0.44	0.88	0.44	0.88
20	1	0.44	1.32	0.44	1.32
21	1	0.44	1.76	0.44	1.75
24	1	0.44	2.20	0.44	2.19
25	1	0.44	2.64	0.44	2.63
27	1	0.44	3.08	0.44	3.07
28	2	0.88	3.96	0.88	3.95
29	2	0.88	4.85	0.88	4.82
31	2	0.88	5.73	0.88	5.70
32	3	1.32	7.05	1.32	7.02
33	3	1.32	8.37	1.32	8.33
34	5	2.20	10.57	2.19	10.53
35	2	0.88	11.45	0.88	11.40
36	2	0.88	12.33	0.88	12.28
37	2	0.88	13.22	0.88	13.16
38	2	0.88	14.10	0.88	14.04
39	5	2.20	16.30	2.19	16.23
40	8	3.52	19.82	3.51	19.74
41	3	1.32	21.15	1.32	21.05
42	3	1.32	22.47	1.32	22.37
43	2	0.88	23.35	0.88	23.25
44	7	3.08	26.43	3.07	26.32
45	2	0.88	27.31	0.88	27.19
46	5	2.20	29.52	2.19	29.39
47	7	3.08	32.60	3.07	32.46
48	3	1.32	33.92	1.32	33.77
49	4	1.76	35.68	1.75	35.53
50	2	0.88	36.56	0.88	36.40
51	7	3.08	39.65	3.07	39.47
52	7	3.08	42.73	3.07	42.54
53	5	2.20	44.93	2.19	44.74
54	6	2.64	47.58	2.63	47.37
55	3	1.32	48.90	1.32	48.68
56	5	2.20	51.10	2.19	50.88
57	5	2.20	53.30	2.19	53.07
58	9	3.96	57.27	3.95	57.02
59	4	1.76	59.03	1.75	58.77
60	6	2.64	61.67	2.63	61.40
61	8	3.52	65.20	3.51	64.91
62	4	1.76	66.96	1.75	66.67
63	13	5.73	72.69	5.70	72.37
64	5	2.20	74.89	2.19	74.56
65	7	3.08	77.97	3.07	77.63
66	4	1.76	79.74	1.75	79.39
67	8	3.52	83.26	3.51	82.89
68	6	2.64	85.90	2.63	85.53
69	2	0.88	86.78	0.88	86.40

70	2	0.88	87.67	0.88	87.28
71	1	0.44	88.11	0.44	87.72
72	4	1.76	89.87	1.75	89.47
73	3	1.32	91.19	1.32	90.79
74	2	0.88	92.07	0.88	91.67
75	4	1.76	93.83	1.75	93.42
76	3	1.32	95.15	1.32	94.74
77	1	0.44	95.59	0.44	95.18
78	1	0.44	96.04	0.44	95.61
79	1	0.44	96.48	0.44	96.05
80	1	0.44	96.92	0.44	96.49
81	1	0.44	97.36	0.44	96.93
82	2	0.88	98.24	0.88	97.81
83	1	0.44	98.68	0.44	98.25
84	2	0.88	99.56	0.88	99.12
88	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Check for number of outliers in DOB age at sign up variable using age limits

Same age limit as used in earlier chunk for self-reported age: age_lower_limit = 16 age_upper_limit = 117

```
dat_clean %>%
  filter(
    dem.dob_age > age_upper_limit | # older than the age limit
      dem.dob_age < age_lower_limit # younger than the age limit
    ) %>%
  nrow()
```

```
[1] 0
```

Recode DOB age at sign up at sign up outliers to -666

```
dat_clean <- dat_clean %>%
    mutate(
      dem.dob_age =
        if_else(
          dem.dob_age > age_upper_limit |
            dem.dob_age < age_lower_limit,
          true = -666,
          false = dem.dob_age,
          missing = NA_real_
        )
    )
```

Inspect clean DOB age at sign up at sign up variable

```
dat_clean %>%
  freq(dem.dob_age)
```

```
Frequencies
dat_clean$dem.dob_age
Type: Numeric
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
17	1	0.44	0.44	0.44	0.44
18	1	0.44	0.88	0.44	0.88
20	1	0.44	1.32	0.44	1.32
21	1	0.44	1.76	0.44	1.75
24	1	0.44	2.20	0.44	2.19
25	1	0.44	2.64	0.44	2.63
27	1	0.44	3.08	0.44	3.07
28	2	0.88	3.96	0.88	3.95
29	2	0.88	4.85	0.88	4.82
31	2	0.88	5.73	0.88	5.70
32	3	1.32	7.05	1.32	7.02
33	3	1.32	8.37	1.32	8.33
34	5	2.20	10.57	2.19	10.53
35	2	0.88	11.45	0.88	11.40
36	2	0.88	12.33	0.88	12.28
37	2	0.88	13.22	0.88	13.16
38	2	0.88	14.10	0.88	14.04
39	5	2.20	16.30	2.19	16.23
40	8	3.52	19.82	3.51	19.74
41	3	1.32	21.15	1.32	21.05
42	3	1.32	22.47	1.32	22.37
43	2	0.88	23.35	0.88	23.25
44	7	3.08	26.43	3.07	26.32
45	2	0.88	27.31	0.88	27.19
46	5	2.20	29.52	2.19	29.39
47	7	3.08	32.60	3.07	32.46
48	3	1.32	33.92	1.32	33.77
49	4	1.76	35.68	1.75	35.53
50	2	0.88	36.56	0.88	36.40
51	7	3.08	39.65	3.07	39.47
52	7	3.08	42.73	3.07	42.54
53	5	2.20	44.93	2.19	44.74
54	6	2.64	47.58	2.63	47.37
55	3	1.32	48.90	1.32	48.68
56	5	2.20	51.10	2.19	50.88
57	5	2.20	53.30	2.19	53.07
58	9	3.96	57.27	3.95	57.02
59	4	1.76	59.03	1.75	58.77
60	6	2.64	61.67	2.63	61.40
61	8	3.52	65.20	3.51	64.91
62	4	1.76	66.96	1.75	66.67
63	13	5.73	72.69	5.70	72.37
64	5	2.20	74.89	2.19	74.56
65	7	3.08	77.97	3.07	77.63

66	4	1.76	79.74	1.75	79.39
67	8	3.52	83.26	3.51	82.89
68	6	2.64	85.90	2.63	85.53
69	2	0.88	86.78	0.88	86.40
70	2	0.88	87.67	0.88	87.28
71	1	0.44	88.11	0.44	87.72
72	4	1.76	89.87	1.75	89.47
73	3	1.32	91.19	1.32	90.79
74	2	0.88	92.07	0.88	91.67
75	4	1.76	93.83	1.75	93.42
76	3	1.32	95.15	1.32	94.74
77	1	0.44	95.59	0.44	95.18
78	1	0.44	96.04	0.44	95.61
79	1	0.44	96.48	0.44	96.05
80	1	0.44	96.92	0.44	96.49
81	1	0.44	97.36	0.44	96.93
82	2	0.88	98.24	0.88	97.81
83	1	0.44	98.68	0.44	98.25
84	2	0.88	99.56	0.88	99.12
88	1	0.44	100.00	0.44	99.56
<NA>	1			0.44	100.00
Total	228	100.00	100.00	100.00	100.00

Save cleaned data

Check colnames before exporting final dataset

```
colnames(dat_clean)
```

```
 [1] "ID"                           "sample"
 [3] "startDate"                    "endDate"
 [5] "dem.how_old_are_you_now.txt" "dem.day"
 [7] "dem.month"                    "dem.year"
 [9] "dem.dob"                      "dem.dob_age"
```

Save variables for exporting in clean dataset - note: DOB variables have been excluded as they contain identifiable information

```
export_variables <- dat_clean %>%
  select(ID,
         startDate,
         endDate,
         sample,
         dem.how_old_are_you_now.txt,
         dem.dob_age) %>%
  colnames()
```

```
dat_clean %>%
  select(all_of(export_variables)) %>%
  saveRDS(
    file = paste0(ilovedata, "/data/latest_freeze/covidcns/age_covidcns_clean.rds")
    )
```

SAVED FOR INTERNAL USE ONLY (contains dob)

```
dat_clean %>%
  saveRDS(
    file = paste0(ilovedata, "/data/latest_freeze/covidcns/age_covidcns_clean_INTERNAL_ONLY.rds")
    )
```