Candy\_report

Neil

2022-08-02

# Brief Introduction to the Dataset

The data is in three parts, one file for each year 2015 - 2017. Sadly none of the datasets agree on how they are set out - many of the columns are different. It broadly records peoples experience of going trick or treating and their ratings (Despair, Joy or Meh) for various types of candy handed out. Also collected are whether they are going trick or treating, age, gender etc.

# List of Assumptions

1. I have tried where at all possible to leave in as much data as possible. I did wonder if I should trim the data down to the minimum required for the tasks but ended up keeping in every column which existed in at least two of the three datasets. UPDATE - I ened up trimming everything due to performance issues with my PC. But the cleaning script is still written to leave these in up until the last line before the merge which chops out all extra columns.
2. I have made some assumptions around spelling mistakes and amalgamating these with what I thought must have been meant. e.g. boxo\_raisins == box\_o\_raisins, sweetums\_a\_friend\_to\_diabetes = sweetums etc. I have labelled these as I go
3. Theres a myriad of responses in the country column. This should REALLY be a dropdown on the quesntionnaire/survey. I have amalgamted as logically as possible. Outliers were tied to NA values. Could these have been labelled “smart\_answer” or something like that?
4. Task 8 - should possible be filtered there are total counts of 0 and 1 in there which do not accrately report anything at all. Perhaps should be filtered for total\_count > 1 at the least.

# Steps to clean data

# Libraries ---------------------------------------------------------------  
  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.7 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(readxl)  
library(here)

## here() starts at C:/Users/neilp/Documents/CODECLAN/dirty\_data\_codeclan\_project\_neil\_plenderleith/task\_4

library(janitor)

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

# Load in Raw Data --------------------------------------------------------  
  
candy\_2015 <- read\_excel(here("raw\_data/boing-boing-candy-2015.xlsx"))  
candy\_2016 <- read\_excel(here("raw\_data/boing-boing-candy-2016.xlsx"))  
candy\_2017 <- read\_excel(here("raw\_data/boing-boing-candy-2017.xlsx"))

## New names:  
## • `` -> `...114`

# Clean Column Names ------------------------------------------------------  
  
# lets pass through clean\_names to get rid of punctuation and bad column names  
candy\_2015\_clean <- candy\_2015 %>%   
 clean\_names()  
candy\_2016\_clean <- candy\_2016 %>%   
 clean\_names()  
candy\_2017\_clean <- candy\_2017 %>%   
 clean\_names()  
  
# First Look at Data -----------------------------------------------------  
  
# used glimpse(), summary(), view(), skimr::skim() on each to get an idea  
  
# lets also have a look at column names  
names(candy\_2015\_clean)

## [1] "timestamp"   
## [2] "how\_old\_are\_you"   
## [3] "are\_you\_going\_actually\_going\_trick\_or\_treating\_yourself"   
## [4] "butterfinger"   
## [5] "x100\_grand\_bar"   
## [6] "anonymous\_brown\_globs\_that\_come\_in\_black\_and\_orange\_wrappers"   
## [7] "any\_full\_sized\_candy\_bar"   
## [8] "black\_jacks"   
## [9] "bonkers"   
## [10] "bottle\_caps"   
## [11] "box\_o\_raisins"   
## [12] "brach\_products\_not\_including\_candy\_corn"   
## [13] "bubble\_gum"   
## [14] "cadbury\_creme\_eggs"   
## [15] "candy\_corn"   
## [16] "vials\_of\_pure\_high\_fructose\_corn\_syrup\_for\_main\_lining\_into\_your\_vein"   
## [17] "candy\_that\_is\_clearly\_just\_the\_stuff\_given\_out\_for\_free\_at\_restaurants"   
## [18] "cash\_or\_other\_forms\_of\_legal\_tender"   
## [19] "chiclets"   
## [20] "caramellos"   
## [21] "snickers"   
## [22] "dark\_chocolate\_hershey"   
## [23] "dental\_paraphenalia"   
## [24] "dots"   
## [25] "fuzzy\_peaches"   
## [26] "generic\_brand\_acetaminophen"   
## [27] "glow\_sticks"   
## [28] "broken\_glow\_stick"   
## [29] "goo\_goo\_clusters"   
## [30] "good\_n\_plenty"   
## [31] "gum\_from\_baseball\_cards"   
## [32] "gummy\_bears\_straight\_up"   
## [33] "creepy\_religious\_comics\_chick\_tracts"   
## [34] "healthy\_fruit"   
## [35] "heath\_bar"   
## [36] "hershey\_s\_kissables"   
## [37] "hershey\_s\_milk\_chocolate"   
## [38] "hugs\_actual\_physical\_hugs"   
## [39] "jolly\_rancher\_bad\_flavor"   
## [40] "jolly\_ranchers\_good\_flavor"   
## [41] "kale\_smoothie"   
## [42] "kinder\_happy\_hippo"   
## [43] "kit\_kat"   
## [44] "hard\_candy"   
## [45] "lapel\_pins"   
## [46] "lemon\_heads"   
## [47] "licorice"   
## [48] "licorice\_not\_black"   
## [49] "lindt\_truffle"   
## [50] "lollipops"   
## [51] "mars"   
## [52] "mary\_janes"   
## [53] "maynards"   
## [54] "milk\_duds"   
## [55] "laffy\_taffy"   
## [56] "minibags\_of\_chips"   
## [57] "joy\_joy\_mit\_iodine"   
## [58] "reggie\_jackson\_bar"   
## [59] "pixy\_stix"   
## [60] "nerds"   
## [61] "nestle\_crunch"   
## [62] "nown\_laters"   
## [63] "pencils"   
## [64] "milky\_way"   
## [65] "reese\_s\_peanut\_butter\_cups"   
## [66] "tolberone\_something\_or\_other"   
## [67] "runts"   
## [68] "junior\_mints"   
## [69] "senior\_mints"   
## [70] "mint\_kisses"   
## [71] "mint\_juleps"   
## [72] "mint\_leaves"   
## [73] "peanut\_m\_m\_s"   
## [74] "regular\_m\_ms"   
## [75] "mint\_m\_ms"   
## [76] "ribbon\_candy"   
## [77] "rolos"   
## [78] "skittles"   
## [79] "smarties\_american"   
## [80] "smarties\_commonwealth"   
## [81] "chick\_o\_sticks\_we\_don\_t\_know\_what\_that\_is"   
## [82] "spotted\_dick"   
## [83] "starburst"   
## [84] "swedish\_fish"   
## [85] "sweetums"   
## [86] "those\_odd\_marshmallow\_circus\_peanut\_things"   
## [87] "three\_musketeers"   
## [88] "peterson\_brand\_sidewalk\_chalk"   
## [89] "peanut\_butter\_bars"   
## [90] "peanut\_butter\_jars"   
## [91] "trail\_mix"   
## [92] "twix"   
## [93] "vicodin"   
## [94] "white\_bread"   
## [95] "whole\_wheat\_anything"   
## [96] "york\_peppermint\_patties"   
## [97] "please\_leave\_any\_remarks\_or\_comments\_regarding\_your\_choices"   
## [98] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_joy"   
## [99] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_despair"   
## [100] "guess\_the\_number\_of\_mints\_in\_my\_hand"   
## [101] "betty\_or\_veronica"   
## [102] "check\_all\_that\_apply\_i\_cried\_tears\_of\_sadness\_at\_the\_end\_of"   
## [103] "that\_dress\_that\_went\_viral\_early\_this\_year\_when\_i\_first\_saw\_it\_it\_was"   
## [104] "fill\_in\_the\_blank\_taylor\_swift\_is\_a\_force\_for"   
## [105] "what\_is\_your\_favourite\_font"   
## [106] "if\_you\_squint\_really\_hard\_the\_words\_intelligent\_design\_would\_look\_like"   
## [107] "fill\_in\_the\_blank\_imitation\_is\_a\_form\_of"   
## [108] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jk\_rowling"   
## [109] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jj\_abrams"   
## [110] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_beyonce"   
## [111] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_bieber"   
## [112] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_kevin\_bacon"   
## [113] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_francis\_bacon\_1561\_1626"  
## [114] "sea\_salt\_flavored\_stuff\_probably\_chocolate\_since\_this\_is\_the\_it\_flavor\_of\_the\_year"   
## [115] "necco\_wafers"   
## [116] "which\_day\_do\_you\_prefer\_friday\_or\_sunday"   
## [117] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_bruce\_lee"   
## [118] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_jk\_rowling"   
## [119] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_malala\_yousafzai"   
## [120] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_thom\_yorke"   
## [121] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_jj\_abrams"   
## [122] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_hillary\_clinton"   
## [123] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_donald\_trump"   
## [124] "please\_estimate\_the\_degrees\_of\_separation\_you\_have\_from\_the\_following\_folks\_beyonce\_knowles"

names(candy\_2016\_clean)

## [1] "timestamp"   
## [2] "are\_you\_going\_actually\_going\_trick\_or\_treating\_yourself"   
## [3] "your\_gender"   
## [4] "how\_old\_are\_you"   
## [5] "which\_country\_do\_you\_live\_in"   
## [6] "which\_state\_province\_county\_do\_you\_live\_in"   
## [7] "x100\_grand\_bar"   
## [8] "anonymous\_brown\_globs\_that\_come\_in\_black\_and\_orange\_wrappers"   
## [9] "any\_full\_sized\_candy\_bar"   
## [10] "black\_jacks"   
## [11] "bonkers\_the\_candy"   
## [12] "bonkers\_the\_board\_game"   
## [13] "bottle\_caps"   
## [14] "boxo\_raisins"   
## [15] "broken\_glow\_stick"   
## [16] "butterfinger"   
## [17] "cadbury\_creme\_eggs"   
## [18] "candy\_corn"   
## [19] "candy\_that\_is\_clearly\_just\_the\_stuff\_given\_out\_for\_free\_at\_restaurants"   
## [20] "caramellos"   
## [21] "cash\_or\_other\_forms\_of\_legal\_tender"   
## [22] "chardonnay"   
## [23] "chick\_o\_sticks\_we\_don\_t\_know\_what\_that\_is"   
## [24] "chiclets"   
## [25] "coffee\_crisp"   
## [26] "creepy\_religious\_comics\_chick\_tracts"   
## [27] "dental\_paraphenalia"   
## [28] "dots"   
## [29] "dove\_bars"   
## [30] "fuzzy\_peaches"   
## [31] "generic\_brand\_acetaminophen"   
## [32] "glow\_sticks"   
## [33] "goo\_goo\_clusters"   
## [34] "good\_n\_plenty"   
## [35] "gum\_from\_baseball\_cards"   
## [36] "gummy\_bears\_straight\_up"   
## [37] "hard\_candy"   
## [38] "healthy\_fruit"   
## [39] "heath\_bar"   
## [40] "hersheys\_dark\_chocolate"   
## [41] "hershey\_s\_milk\_chocolate"   
## [42] "hersheys\_kisses"   
## [43] "hugs\_actual\_physical\_hugs"   
## [44] "jolly\_rancher\_bad\_flavor"   
## [45] "jolly\_ranchers\_good\_flavor"   
## [46] "joy\_joy\_mit\_iodine"   
## [47] "junior\_mints"   
## [48] "senior\_mints"   
## [49] "kale\_smoothie"   
## [50] "kinder\_happy\_hippo"   
## [51] "kit\_kat"   
## [52] "laffy\_taffy"   
## [53] "lemon\_heads"   
## [54] "licorice\_not\_black"   
## [55] "licorice\_yes\_black"   
## [56] "lindt\_truffle"   
## [57] "lollipops"   
## [58] "mars"   
## [59] "mary\_janes"   
## [60] "maynards"   
## [61] "mike\_and\_ike"   
## [62] "milk\_duds"   
## [63] "milky\_way"   
## [64] "regular\_m\_ms"   
## [65] "peanut\_m\_m\_s"   
## [66] "blue\_m\_ms"   
## [67] "red\_m\_ms"   
## [68] "third\_party\_m\_ms"   
## [69] "minibags\_of\_chips"   
## [70] "mint\_kisses"   
## [71] "mint\_juleps"   
## [72] "mr\_goodbar"   
## [73] "necco\_wafers"   
## [74] "nerds"   
## [75] "nestle\_crunch"   
## [76] "nown\_laters"   
## [77] "peeps"   
## [78] "pencils"   
## [79] "person\_of\_interest\_season\_3\_dvd\_box\_set\_not\_including\_disc\_4\_with\_hilarious\_outtakes"   
## [80] "pixy\_stix"   
## [81] "reese\_s\_peanut\_butter\_cups"   
## [82] "reeses\_pieces"   
## [83] "reggie\_jackson\_bar"   
## [84] "rolos"   
## [85] "skittles"   
## [86] "smarties\_american"   
## [87] "smarties\_commonwealth"   
## [88] "snickers"   
## [89] "sourpatch\_kids\_i\_e\_abominations\_of\_nature"   
## [90] "spotted\_dick"   
## [91] "starburst"   
## [92] "sweet\_tarts"   
## [93] "swedish\_fish"   
## [94] "sweetums\_a\_friend\_to\_diabetes"   
## [95] "tic\_tacs"   
## [96] "those\_odd\_marshmallow\_circus\_peanut\_things"   
## [97] "three\_musketeers"   
## [98] "tolberone\_something\_or\_other"   
## [99] "trail\_mix"   
## [100] "twix"   
## [101] "vials\_of\_pure\_high\_fructose\_corn\_syrup\_for\_main\_lining\_into\_your\_vein"   
## [102] "vicodin"   
## [103] "whatchamacallit\_bars"   
## [104] "white\_bread"   
## [105] "whole\_wheat\_anything"   
## [106] "york\_peppermint\_patties"   
## [107] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_joy"   
## [108] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_despair"   
## [109] "please\_leave\_any\_witty\_snarky\_or\_thoughtful\_remarks\_or\_comments\_regarding\_your\_choices"   
## [110] "guess\_the\_number\_of\_mints\_in\_my\_hand"   
## [111] "betty\_or\_veronica"   
## [112] "that\_dress\_that\_went\_viral\_a\_few\_years\_back\_when\_i\_first\_saw\_it\_it\_was"   
## [113] "what\_is\_your\_favourite\_font"   
## [114] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jk\_rowling"   
## [115] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jj\_abrams"   
## [116] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_beyonce"   
## [117] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_bieber"   
## [118] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_kevin\_bacon"   
## [119] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_francis\_bacon\_1561\_1626"   
## [120] "which\_day\_do\_you\_prefer\_friday\_or\_sunday"   
## [121] "do\_you\_eat\_apples\_the\_correct\_way\_east\_to\_west\_side\_to\_side\_or\_do\_you\_eat\_them\_like\_a\_freak\_of\_nature\_south\_to\_north\_bottom\_to\_top"  
## [122] "when\_you\_see\_the\_above\_image\_of\_the\_4\_different\_websites\_which\_one\_would\_you\_most\_likely\_check\_out\_please\_be\_honest"   
## [123] "york\_peppermint\_patties\_ignore"

names(candy\_2017\_clean)

## [1] "internal\_id"   
## [2] "q1\_going\_out"   
## [3] "q2\_gender"   
## [4] "q3\_age"   
## [5] "q4\_country"   
## [6] "q5\_state\_province\_county\_etc"   
## [7] "q6\_100\_grand\_bar"   
## [8] "q6\_anonymous\_brown\_globs\_that\_come\_in\_black\_and\_orange\_wrappers\_a\_k\_a\_mary\_janes"  
## [9] "q6\_any\_full\_sized\_candy\_bar"   
## [10] "q6\_black\_jacks"   
## [11] "q6\_bonkers\_the\_candy"   
## [12] "q6\_bonkers\_the\_board\_game"   
## [13] "q6\_bottle\_caps"   
## [14] "q6\_boxo\_raisins"   
## [15] "q6\_broken\_glow\_stick"   
## [16] "q6\_butterfinger"   
## [17] "q6\_cadbury\_creme\_eggs"   
## [18] "q6\_candy\_corn"   
## [19] "q6\_candy\_that\_is\_clearly\_just\_the\_stuff\_given\_out\_for\_free\_at\_restaurants"   
## [20] "q6\_caramellos"   
## [21] "q6\_cash\_or\_other\_forms\_of\_legal\_tender"   
## [22] "q6\_chardonnay"   
## [23] "q6\_chick\_o\_sticks\_we\_don\_t\_know\_what\_that\_is"   
## [24] "q6\_chiclets"   
## [25] "q6\_coffee\_crisp"   
## [26] "q6\_creepy\_religious\_comics\_chick\_tracts"   
## [27] "q6\_dental\_paraphenalia"   
## [28] "q6\_dots"   
## [29] "q6\_dove\_bars"   
## [30] "q6\_fuzzy\_peaches"   
## [31] "q6\_generic\_brand\_acetaminophen"   
## [32] "q6\_glow\_sticks"   
## [33] "q6\_goo\_goo\_clusters"   
## [34] "q6\_good\_n\_plenty"   
## [35] "q6\_gum\_from\_baseball\_cards"   
## [36] "q6\_gummy\_bears\_straight\_up"   
## [37] "q6\_hard\_candy"   
## [38] "q6\_healthy\_fruit"   
## [39] "q6\_heath\_bar"   
## [40] "q6\_hersheys\_dark\_chocolate"   
## [41] "q6\_hershey\_s\_milk\_chocolate"   
## [42] "q6\_hersheys\_kisses"   
## [43] "q6\_hugs\_actual\_physical\_hugs"   
## [44] "q6\_jolly\_rancher\_bad\_flavor"   
## [45] "q6\_jolly\_ranchers\_good\_flavor"   
## [46] "q6\_joy\_joy\_mit\_iodine"   
## [47] "q6\_junior\_mints"   
## [48] "q6\_senior\_mints"   
## [49] "q6\_kale\_smoothie"   
## [50] "q6\_kinder\_happy\_hippo"   
## [51] "q6\_kit\_kat"   
## [52] "q6\_laffy\_taffy"   
## [53] "q6\_lemon\_heads"   
## [54] "q6\_licorice\_not\_black"   
## [55] "q6\_licorice\_yes\_black"   
## [56] "q6\_lindt\_truffle"   
## [57] "q6\_lollipops"   
## [58] "q6\_mars"   
## [59] "q6\_maynards"   
## [60] "q6\_mike\_and\_ike"   
## [61] "q6\_milk\_duds"   
## [62] "q6\_milky\_way"   
## [63] "q6\_regular\_m\_ms"   
## [64] "q6\_peanut\_m\_m\_s"   
## [65] "q6\_blue\_m\_ms"   
## [66] "q6\_red\_m\_ms"   
## [67] "q6\_green\_party\_m\_ms"   
## [68] "q6\_independent\_m\_ms"   
## [69] "q6\_abstained\_from\_m\_ming"   
## [70] "q6\_minibags\_of\_chips"   
## [71] "q6\_mint\_kisses"   
## [72] "q6\_mint\_juleps"   
## [73] "q6\_mr\_goodbar"   
## [74] "q6\_necco\_wafers"   
## [75] "q6\_nerds"   
## [76] "q6\_nestle\_crunch"   
## [77] "q6\_nown\_laters"   
## [78] "q6\_peeps"   
## [79] "q6\_pencils"   
## [80] "q6\_pixy\_stix"   
## [81] "q6\_real\_housewives\_of\_orange\_county\_season\_9\_blue\_ray"   
## [82] "q6\_reese\_s\_peanut\_butter\_cups"   
## [83] "q6\_reeses\_pieces"   
## [84] "q6\_reggie\_jackson\_bar"   
## [85] "q6\_rolos"   
## [86] "q6\_sandwich\_sized\_bags\_filled\_with\_boo\_berry\_crunch"   
## [87] "q6\_skittles"   
## [88] "q6\_smarties\_american"   
## [89] "q6\_smarties\_commonwealth"   
## [90] "q6\_snickers"   
## [91] "q6\_sourpatch\_kids\_i\_e\_abominations\_of\_nature"   
## [92] "q6\_spotted\_dick"   
## [93] "q6\_starburst"   
## [94] "q6\_sweet\_tarts"   
## [95] "q6\_swedish\_fish"   
## [96] "q6\_sweetums\_a\_friend\_to\_diabetes"   
## [97] "q6\_take\_5"   
## [98] "q6\_tic\_tacs"   
## [99] "q6\_those\_odd\_marshmallow\_circus\_peanut\_things"   
## [100] "q6\_three\_musketeers"   
## [101] "q6\_tolberone\_something\_or\_other"   
## [102] "q6\_trail\_mix"   
## [103] "q6\_twix"   
## [104] "q6\_vials\_of\_pure\_high\_fructose\_corn\_syrup\_for\_main\_lining\_into\_your\_vein"   
## [105] "q6\_vicodin"   
## [106] "q6\_whatchamacallit\_bars"   
## [107] "q6\_white\_bread"   
## [108] "q6\_whole\_wheat\_anything"   
## [109] "q6\_york\_peppermint\_patties"   
## [110] "q7\_joy\_other"   
## [111] "q8\_despair\_other"   
## [112] "q9\_other\_comments"   
## [113] "q10\_dress"   
## [114] "x114"   
## [115] "q11\_day"   
## [116] "q12\_media\_daily\_dish"   
## [117] "q12\_media\_science"   
## [118] "q12\_media\_espn"   
## [119] "q12\_media\_yahoo"   
## [120] "click\_coordinates\_x\_y"

# Order of Operations:  
# Get large pivot data the same for all 3 datasets  
# perform pivot on all three  
# then investigate other columns  
  
# let first start with all the columns between "100\_grand\_bar" and   
# "york peppermint butter" these are out pivot to longer columns  
candy\_2015\_clean <- candy\_2015\_clean %>%   
 rename("100\_grand\_bar" = "x100\_grand\_bar") %>%   
 relocate(butterfinger, .after = bubble\_gum) %>%   
 relocate(necco\_wafers, .after = minibags\_of\_chips)  
   
   
candy\_2016\_clean <- candy\_2016\_clean %>%   
 rename("100\_grand\_bar" = "x100\_grand\_bar")  
  
# 2017 data required a q1\_, q2\_ etc removed from the column names  
candy\_2017\_clean <- candy\_2017\_clean %>%   
 rename\_with( ~ str\_remove(., pattern = "q[0-90-9]+\_"))  
  
  
# pivot longer on all three datasets for the candy and rating columns  
  
candy\_2015\_long <- candy\_2015\_clean %>%   
 pivot\_longer(`100\_grand\_bar`:york\_peppermint\_patties,   
 names\_to = "candy\_type",   
 values\_to = "candy\_rating")  
  
candy\_2016\_long <- candy\_2016\_clean %>%   
 pivot\_longer(`100\_grand\_bar`:york\_peppermint\_patties,   
 names\_to = "candy\_type",   
 values\_to = "candy\_rating")  
  
candy\_2017\_long <- candy\_2017\_clean %>%   
 pivot\_longer(`100\_grand\_bar`:york\_peppermint\_patties,   
 names\_to = "candy\_type",   
 values\_to = "candy\_rating")  
  
  
# Ok pivots look good lets look through each dataset and see which columns are   
# required and which can go. In this decision I am thinking of the upcoming join  
# of the three data frames. It looks like this will have to be a bind\_rows function   
# I will hope to keep each column that's in at least 2 of the 3 data frames.  
  
# 2015 Table  
  
  
# lets work out how many NA's are across all columns and sort by highest  
NA\_2015 <- candy\_2015\_long %>%  
 summarise(across(.fns = ~ sum(is.na(.x)))) %>%   
 pivot\_longer(cols = everything(),   
 names\_to = "old\_column\_name",   
 values\_to = "count\_of\_NA") %>%   
 arrange(desc(count\_of\_NA))  
  
# lets drop these 9 full NA columns - no use to us  
candy\_2015\_long <- candy\_2015\_long %>%   
 select(-fill\_in\_the\_blank\_taylor\_swift\_is\_a\_force\_for,   
 -starts\_with("please\_estimate\_the\_degrees\_of\_"))  
  
# 2016  
  
# lets work out how many NA's are across all columns and sort by highest  
NA\_2016 <- candy\_2016\_long %>%  
 summarise(across(.fns = ~ sum(is.na(.x)))) %>%   
 pivot\_longer(cols = everything(),   
 names\_to = "old\_column\_name",   
 values\_to = "count\_of\_NA") %>%   
 arrange(desc(count\_of\_NA))  
  
# lets drop this full NA column - no use to us  
candy\_2016\_long <- candy\_2016\_long %>%   
 select(-york\_peppermint\_patties\_ignore)  
  
#2017  
  
# lets work out how many NA's are across all columns and sort by highest  
NA\_2017 <- candy\_2017\_long %>%  
 summarise(across(.fns = ~ sum(is.na(.x)))) %>%   
 pivot\_longer(cols = everything(),   
 names\_to = "old\_column\_name",   
 values\_to = "count\_of\_NA") %>%   
 arrange(desc(count\_of\_NA))  
  
# lets drop columns x114 which is ambigous and very nearly empty as well as the  
# media columns - these are also nearly empty and are not in the other datasets  
candy\_2017\_long <- candy\_2017\_long %>%   
 select(-x114,   
 -starts\_with("media"))  
  
  
# Ok lets recap. We have removed the easy columns so far - the full NA's and   
# columns with very little data. Candy\_2017\_long has fewer columns than the   
# others - lets use this as a template for the other data frame columns and try to   
# get our column names matching to prepare data for a future join  
  
names(candy\_2015\_long)

## [1] "timestamp"   
## [2] "how\_old\_are\_you"   
## [3] "are\_you\_going\_actually\_going\_trick\_or\_treating\_yourself"   
## [4] "please\_leave\_any\_remarks\_or\_comments\_regarding\_your\_choices"   
## [5] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_joy"   
## [6] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_despair"   
## [7] "guess\_the\_number\_of\_mints\_in\_my\_hand"   
## [8] "betty\_or\_veronica"   
## [9] "check\_all\_that\_apply\_i\_cried\_tears\_of\_sadness\_at\_the\_end\_of"   
## [10] "that\_dress\_that\_went\_viral\_early\_this\_year\_when\_i\_first\_saw\_it\_it\_was"   
## [11] "what\_is\_your\_favourite\_font"   
## [12] "if\_you\_squint\_really\_hard\_the\_words\_intelligent\_design\_would\_look\_like"   
## [13] "fill\_in\_the\_blank\_imitation\_is\_a\_form\_of"   
## [14] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jk\_rowling"   
## [15] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jj\_abrams"   
## [16] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_beyonce"   
## [17] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_bieber"   
## [18] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_kevin\_bacon"   
## [19] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_francis\_bacon\_1561\_1626"  
## [20] "sea\_salt\_flavored\_stuff\_probably\_chocolate\_since\_this\_is\_the\_it\_flavor\_of\_the\_year"   
## [21] "which\_day\_do\_you\_prefer\_friday\_or\_sunday"   
## [22] "candy\_type"   
## [23] "candy\_rating"

names(candy\_2016\_long)

## [1] "timestamp"   
## [2] "are\_you\_going\_actually\_going\_trick\_or\_treating\_yourself"   
## [3] "your\_gender"   
## [4] "how\_old\_are\_you"   
## [5] "which\_country\_do\_you\_live\_in"   
## [6] "which\_state\_province\_county\_do\_you\_live\_in"   
## [7] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_joy"   
## [8] "please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_despair"   
## [9] "please\_leave\_any\_witty\_snarky\_or\_thoughtful\_remarks\_or\_comments\_regarding\_your\_choices"   
## [10] "guess\_the\_number\_of\_mints\_in\_my\_hand"   
## [11] "betty\_or\_veronica"   
## [12] "that\_dress\_that\_went\_viral\_a\_few\_years\_back\_when\_i\_first\_saw\_it\_it\_was"   
## [13] "what\_is\_your\_favourite\_font"   
## [14] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jk\_rowling"   
## [15] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_jj\_abrams"   
## [16] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_beyonce"   
## [17] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_bieber"   
## [18] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_kevin\_bacon"   
## [19] "please\_estimate\_the\_degree\_s\_of\_separation\_you\_have\_from\_the\_following\_celebrities\_francis\_bacon\_1561\_1626"   
## [20] "which\_day\_do\_you\_prefer\_friday\_or\_sunday"   
## [21] "do\_you\_eat\_apples\_the\_correct\_way\_east\_to\_west\_side\_to\_side\_or\_do\_you\_eat\_them\_like\_a\_freak\_of\_nature\_south\_to\_north\_bottom\_to\_top"  
## [22] "when\_you\_see\_the\_above\_image\_of\_the\_4\_different\_websites\_which\_one\_would\_you\_most\_likely\_check\_out\_please\_be\_honest"   
## [23] "candy\_type"   
## [24] "candy\_rating"

names(candy\_2017\_long)

## [1] "internal\_id" "going\_out"   
## [3] "gender" "age"   
## [5] "country" "state\_province\_county\_etc"  
## [7] "joy\_other" "despair\_other"   
## [9] "other\_comments" "dress"   
## [11] "day" "click\_coordinates\_x\_y"   
## [13] "candy\_type" "candy\_rating"

# lets get 2016 into shape - change col names to agree with 2017  
candy\_2016\_long <- candy\_2016\_long %>%   
 rename(  
 id = timestamp,  
 going\_out = are\_you\_going\_actually\_going\_trick\_or\_treating\_yourself,  
 gender = your\_gender,  
 age = how\_old\_are\_you,  
 country = which\_country\_do\_you\_live\_in,  
 state\_province\_county\_etc = which\_state\_province\_county\_do\_you\_live\_in,  
 joy\_other = please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_joy,  
 despair\_other = please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_despair,  
 other\_comments = please\_leave\_any\_witty\_snarky\_or\_thoughtful\_remarks\_or\_comments\_regarding\_your\_choices,  
 day = which\_day\_do\_you\_prefer\_friday\_or\_sunday,  
 dress = that\_dress\_that\_went\_viral\_a\_few\_years\_back\_when\_i\_first\_saw\_it\_it\_was,  
 guess\_mints = guess\_the\_number\_of\_mints\_in\_my\_hand,  
 favourite\_font = what\_is\_your\_favourite\_font)   
  
candy\_2016\_long <- candy\_2016\_long %>%   
 rename\_with( ~ str\_replace(., pattern = "please\_.\*celebrities\_", replacement = "separation\_"))  
   
  
# reorder to agree more with 2017  
candy\_2016\_long <- candy\_2016\_long %>%  
relocate(dress, day, candy\_type, candy\_rating, .after = other\_comments)  
  
# drop columns which dont appear elsewhere, have little values - i have  
# detailed this in the assumptions   
candy\_2016\_long <- candy\_2016\_long %>%  
 select(-(23:24))  
  
names(candy\_2016\_long) # looks good

## [1] "id" "going\_out"   
## [3] "gender" "age"   
## [5] "country" "state\_province\_county\_etc"   
## [7] "joy\_other" "despair\_other"   
## [9] "other\_comments" "dress"   
## [11] "day" "candy\_type"   
## [13] "candy\_rating" "guess\_mints"   
## [15] "betty\_or\_veronica" "favourite\_font"   
## [17] "separation\_jk\_rowling" "separation\_jj\_abrams"   
## [19] "separation\_beyonce" "separation\_bieber"   
## [21] "separation\_kevin\_bacon" "separation\_francis\_bacon\_1561\_1626"

# lets tweak 2017 in line with 2016 above  
  
candy\_2017\_long <- candy\_2017\_long %>%   
 rename(id = internal\_id) %>%   
 select(-click\_coordinates\_x\_y) %>% # only in this 1 data frame  
 mutate(guess\_mints = NA, betty\_or\_veronica = NA, favourite\_font = NA,  
 separation\_jk\_rowling = NA, separation\_jj\_abrams = NA, separation\_beyonce = NA, separation\_bieber = NA, separation\_kevin\_bacon = NA, separation\_francis\_bacon\_1561\_1626 = NA) # add in columns that are in other 2  
  
names(candy\_2016\_long) == names(candy\_2017\_long) # columns all match!

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE

# OK time for 2015 to match the other two years  
candy\_2015\_long <- candy\_2015\_long %>%   
 rename(  
 id = timestamp,  
 age = how\_old\_are\_you,  
 going\_out = are\_you\_going\_actually\_going\_trick\_or\_treating\_yourself,  
 other\_comments = please\_leave\_any\_remarks\_or\_comments\_regarding\_your\_choices,  
 joy\_other = please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_joy,  
 despair\_other = please\_list\_any\_items\_not\_included\_above\_that\_give\_you\_despair,  
 guess\_mints = guess\_the\_number\_of\_mints\_in\_my\_hand,  
 favourite\_font = what\_is\_your\_favourite\_font,  
 day = which\_day\_do\_you\_prefer\_friday\_or\_sunday,  
 dress = that\_dress\_that\_went\_viral\_early\_this\_year\_when\_i\_first\_saw\_it\_it\_was)  
  
# rename large column names  
candy\_2015\_long <- candy\_2015\_long %>%   
 rename\_with( ~ str\_replace(., pattern = "please\_.\*celebrities\_", replacement = "separation\_"))  
  
# reorder to agree more with others  
candy\_2015\_long <- candy\_2015\_long %>%  
 relocate(dress, day, candy\_type, candy\_rating, .after = other\_comments)  
  
# delete columns not in other datasets  
candy\_2015\_long <- candy\_2015\_long %>%  
 select(-if\_you\_squint\_really\_hard\_the\_words\_intelligent\_design\_would\_look\_like,  
 - fill\_in\_the\_blank\_imitation\_is\_a\_form\_of,  
 - sea\_salt\_flavored\_stuff\_probably\_chocolate\_since\_this\_is\_the\_it\_flavor\_of\_the\_year,  
 -check\_all\_that\_apply\_i\_cried\_tears\_of\_sadness\_at\_the\_end\_of)  
  
# add in columns in line with others - these will need to be NA as theres no data  
# only alternative would be to lose all data from other tables   
candy\_2015\_long <- candy\_2015\_long %>%  
 mutate(gender = NA, country = NA, state\_province\_county\_etc = NA)  
  
# final reorder to agree more with others  
candy\_2015\_long <- candy\_2015\_long %>%  
 relocate(going\_out, gender, .after = id) %>%   
relocate(country, state\_province\_county\_etc, joy\_other, despair\_other,   
 other\_comments, .after = age)  
  
names(candy\_2015\_long) == names(candy\_2016\_long)

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE

# So we now have 3 datasets ready to be bound together  
# Then we can look at the columns like country that need cleaned up  
  
candy\_2015\_long <- candy\_2015\_long %>%  
 mutate(id = as.character(id))  
candy\_2016\_long <- candy\_2016\_long %>%  
 mutate(id = as.character(id))  
candy\_2017\_long <- candy\_2017\_long %>%  
 mutate(id = as.character(id))  
  
  
## Performance related data cleanse ----------------------------------------  
  
# Going to delete all columns not strictly needed for the tasks  
# This is purely for performance as my PC is running so slow  
  
candy\_2015\_long <- candy\_2015\_long %>%   
select(-(state\_province\_county\_etc:day)) %>%   
 select(-(guess\_mints:separation\_francis\_bacon\_1561\_1626))  
candy\_2016\_long <- candy\_2016\_long %>%   
 select(-(state\_province\_county\_etc:day)) %>%   
 select(-(guess\_mints:separation\_francis\_bacon\_1561\_1626))  
candy\_2017\_long <- candy\_2017\_long %>%   
 select(-(state\_province\_county\_etc:day)) %>%   
 select(-(guess\_mints:separation\_francis\_bacon\_1561\_1626))  
  
  
# Joining the three tables ------------------------------------------------  
  
# Lets join the 3 sets together  
candy\_combined <- bind\_rows(candy\_2015\_long,   
 candy\_2016\_long,   
 candy\_2017\_long,   
 .id = "year")   
# last line adds a column to let us know original table data is from  
  
  
# Cleaning the column data ------------------------------------------------  
  
# We have our tables combined! Yay! But wait - they are extremely dirty  
# Lets go through column by column and have a look  
  
  
## Cleaning candy\_type column ----------------------------------------------  
  
  
table(candy\_combined["candy\_type"])

## candy\_type  
## 100\_grand\_bar   
## 9349   
## abstained\_from\_m\_ming   
## 2460   
## anonymous\_brown\_globs\_that\_come\_in\_black\_and\_orange\_wrappers   
## 6889   
## anonymous\_brown\_globs\_that\_come\_in\_black\_and\_orange\_wrappers\_a\_k\_a\_mary\_janes   
## 2460   
## any\_full\_sized\_candy\_bar   
## 9349   
## black\_jacks   
## 9349   
## blue\_m\_ms   
## 3719   
## bonkers   
## 5630   
## bonkers\_the\_board\_game   
## 3719   
## bonkers\_the\_candy   
## 3719   
## bottle\_caps   
## 9349   
## box\_o\_raisins   
## 5630   
## boxo\_raisins   
## 3719   
## brach\_products\_not\_including\_candy\_corn   
## 5630   
## broken\_glow\_stick   
## 9349   
## bubble\_gum   
## 5630   
## butterfinger   
## 9349   
## cadbury\_creme\_eggs   
## 9349   
## candy\_corn   
## 9349   
## candy\_that\_is\_clearly\_just\_the\_stuff\_given\_out\_for\_free\_at\_restaurants   
## 9349   
## caramellos   
## 9349   
## cash\_or\_other\_forms\_of\_legal\_tender   
## 9349   
## chardonnay   
## 3719   
## chick\_o\_sticks\_we\_don\_t\_know\_what\_that\_is   
## 9349   
## chiclets   
## 9349   
## coffee\_crisp   
## 3719   
## creepy\_religious\_comics\_chick\_tracts   
## 9349   
## dark\_chocolate\_hershey   
## 5630   
## dental\_paraphenalia   
## 9349   
## dots   
## 9349   
## dove\_bars   
## 3719   
## fuzzy\_peaches   
## 9349   
## generic\_brand\_acetaminophen   
## 9349   
## glow\_sticks   
## 9349   
## goo\_goo\_clusters   
## 9349   
## good\_n\_plenty   
## 9349   
## green\_party\_m\_ms   
## 2460   
## gum\_from\_baseball\_cards   
## 9349   
## gummy\_bears\_straight\_up   
## 9349   
## hard\_candy   
## 9349   
## healthy\_fruit   
## 9349   
## heath\_bar   
## 9349   
## hershey\_s\_kissables   
## 5630   
## hershey\_s\_milk\_chocolate   
## 9349   
## hersheys\_dark\_chocolate   
## 3719   
## hersheys\_kisses   
## 3719   
## hugs\_actual\_physical\_hugs   
## 9349   
## independent\_m\_ms   
## 2460   
## jolly\_rancher\_bad\_flavor   
## 9349   
## jolly\_ranchers\_good\_flavor   
## 9349   
## joy\_joy\_mit\_iodine   
## 9349   
## junior\_mints   
## 9349   
## kale\_smoothie   
## 9349   
## kinder\_happy\_hippo   
## 9349   
## kit\_kat   
## 9349   
## laffy\_taffy   
## 9349   
## lapel\_pins   
## 5630   
## lemon\_heads   
## 9349   
## licorice   
## 5630   
## licorice\_not\_black   
## 9349   
## licorice\_yes\_black   
## 3719   
## lindt\_truffle   
## 9349   
## lollipops   
## 9349   
## mars   
## 9349   
## mary\_janes   
## 6889   
## maynards   
## 9349   
## mike\_and\_ike   
## 3719   
## milk\_duds   
## 9349   
## milky\_way   
## 9349   
## minibags\_of\_chips   
## 9349   
## mint\_juleps   
## 9349   
## mint\_kisses   
## 9349   
## mint\_leaves   
## 5630   
## mint\_m\_ms   
## 5630   
## mr\_goodbar   
## 3719   
## necco\_wafers   
## 9349   
## nerds   
## 9349   
## nestle\_crunch   
## 9349   
## nown\_laters   
## 9349   
## peanut\_butter\_bars   
## 5630   
## peanut\_butter\_jars   
## 5630   
## peanut\_m\_m\_s   
## 9349   
## peeps   
## 3719   
## pencils   
## 9349   
## person\_of\_interest\_season\_3\_dvd\_box\_set\_not\_including\_disc\_4\_with\_hilarious\_outtakes   
## 1259   
## peterson\_brand\_sidewalk\_chalk   
## 5630   
## pixy\_stix   
## 9349   
## real\_housewives\_of\_orange\_county\_season\_9\_blue\_ray   
## 2460   
## red\_m\_ms   
## 3719   
## reese\_s\_peanut\_butter\_cups   
## 9349   
## reeses\_pieces   
## 3719   
## reggie\_jackson\_bar   
## 9349   
## regular\_m\_ms   
## 9349   
## ribbon\_candy   
## 5630   
## rolos   
## 9349   
## runts   
## 5630   
## sandwich\_sized\_bags\_filled\_with\_boo\_berry\_crunch   
## 2460   
## senior\_mints   
## 9349   
## skittles   
## 9349   
## smarties\_american   
## 9349   
## smarties\_commonwealth   
## 9349   
## snickers   
## 9349   
## sourpatch\_kids\_i\_e\_abominations\_of\_nature   
## 3719   
## spotted\_dick   
## 9349   
## starburst   
## 9349   
## swedish\_fish   
## 9349   
## sweet\_tarts   
## 3719   
## sweetums   
## 5630   
## sweetums\_a\_friend\_to\_diabetes   
## 3719   
## take\_5   
## 2460   
## third\_party\_m\_ms   
## 1259   
## those\_odd\_marshmallow\_circus\_peanut\_things   
## 9349   
## three\_musketeers   
## 9349   
## tic\_tacs   
## 3719   
## tolberone\_something\_or\_other   
## 9349   
## trail\_mix   
## 9349   
## twix   
## 9349   
## vials\_of\_pure\_high\_fructose\_corn\_syrup\_for\_main\_lining\_into\_your\_vein   
## 9349   
## vicodin   
## 9349   
## whatchamacallit\_bars   
## 3719   
## white\_bread   
## 9349   
## whole\_wheat\_anything   
## 9349   
## york\_peppermint\_patties   
## 9349

# OK so there are a few obvious duplicates with slightly different names  
# There are some slightly the same but ambiguous - these will be left  
  
candy\_combined <- candy\_combined %>%   
 mutate(candy\_type = recode(candy\_type,  
 "anonymous\_brown\_globs\_that\_come\_in\_black\_and\_orange\_wrappers\_a\_k\_a\_mary\_janes" = "mary\_janes"),  
 candy\_type = recode(candy\_type,"bonkers\_the\_candy" = "bonkers"),  
 candy\_type = recode(candy\_type,"boxo\_raisins" = "box\_o\_raisins"),  
 candy\_type = recode(candy\_type,"licorice\_yes\_black" = "licorice"),  
 candy\_type = recode(candy\_type,"sweetums\_a\_friend\_to\_diabetes" = "sweetums"))  
  
  
## Cleaning country column -------------------------------------------------  
  
  
# count(candy\_combined, country)  
# table(candy\_combined["country"]) # look at dirty column 'country' eek  
  
# get a few easy ones with regex  
candy\_combined <- candy\_combined %>%   
 mutate(country = if\_else(grepl("(?i)usa+", country),"USA",country)) %>%   
 mutate(country = if\_else(grepl("(?i)united s+", country),"USA",country)) %>%   
 mutate(country = if\_else(grepl("(?i)amer", country),"USA",country)) %>%   
 mutate(country = if\_else(grepl("(?i)stat", country),"USA",country))  
  
#make vectors of USA outliers and some to change to NA values  
usa\_outliers = c("Alaska", "California", "EUA", "Merica", "Murica", "murrika",  
 "New Jersey", "New York", "North Carolina", "Pittsburgh",   
 "The Yoo Ess of Aaayyyyyy", "Trumpistan", "U S", "u s a", "u.s.",  
 "U.s.", "U.S.", "u.s.a.", "U.S.A.", "UD", "us", "Us", "US", "US of A",  
 "USSA", "'merica")  
change\_to\_NA = c(1, 30.0, 32, 35, 44.0, 45, 45.0, 46, 47.0, 51.0, 54.0)  
change\_to\_NA2 = c("30.0", "44.0", "45.0", "47.0", "51.0", "54.0")  
silly\_values = c(  
 "A tropical island south of the equator", "A", "Atlantis",  
 "Canae", "cascadia ", "Cascadia", "Denial", "Earth", "Fear and Loathing",   
 "god's country", "I don't know anymore", "insanity lately",   
 "there isn't one for old men", "soviet canuckistan", "Narnia", "Neverland",  
 "one of the best ones", "See above", "Somewhere",   
 "Subscribe To Dm4uz3 On Youtube", "The republic of Cascadia", "this one",   
 "Europe", " Cascadia", "Cascadia ", " Subscribe To Dm4uz3 On Youtube",  
 "Subscribe To Dm4uz3 On Youtube ")  
  
# use the vectors above   
candy\_combined <- candy\_combined %>%  
mutate(country = if\_else(country %in% usa\_outliers ,  
 "USA", country))   
  
candy\_combined <- candy\_combined %>%  
 mutate(country = if\_else(country %in% silly\_values|  
 country %in% change\_to\_NA|  
 country %in% change\_to\_NA2,   
 NA\_character\_, country)) %>%   
 mutate(country = str\_to\_title(country)) # change to title case to help  
  
# recode anything else  
candy\_combined <- candy\_combined %>%  
mutate(country = recode(country, "The Netherlands" = "Netherlands"),  
 country = recode(country, "Can" = "Canada"),  
 country = recode(country, "Canada`" = "Canada"),  
 country = recode(country, "Endland" = "United Kingdom"),  
 country = recode(country, "England" = "United Kingdom"),  
 country = recode(country, "England" = "United Kingdom"),  
 country = recode(country, "Scotland" = "United Kingdom"),  
 country = recode(country, "España" = "Spain"),  
 country = recode(country, "U.k." = "United Kingdom"),  
 country = recode(country, "Uk" = "United Kingdom"),  
 country = recode(country, "United Kindom" = "United Kingdom"))  
  
table(candy\_combined["country"]) # much better!

## country  
## Australia Austria   
## 1121 100   
## Belgium Brasil   
## 100 100   
## Canada Cascadia   
## 35078 103   
## China Costa Rica   
## 306 103   
## Croatia Denmark   
## 100 206   
## Finland France   
## 203 609   
## Germany Greece   
## 1530 103   
## Hong Kong Hungary   
## 206 100   
## Iceland Indonesia   
## 103 103   
## Ireland Japan   
## 412 915   
## Kenya Korea   
## 100 203   
## Mexico Netherlands   
## 712 1021   
## New Zealand Panama   
## 400 100   
## Philippines Portugal   
## 100 100   
## Singapore South Africa   
## 103 103   
## South Korea Spain   
## 203 203   
## Subscribe To Dm4uz3 On Youtube Sweden   
## 103 306   
## Switzerland Taiwan   
## 409 103   
## Uae United Kingdom   
## 103 6220   
## Usa   
## 315050

## Cleaning age column -----------------------------------------------------  
  
  
# oldest person ever was 122. lets take out everything above that as NA  
# lets keep in the 0 values - technically this could be babies in a pram?  
  
# change the column from a character to numeric  
candy\_combined <- candy\_combined %>%  
 mutate(age = as.numeric(age))

## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion

# clean out values bigger than 122 - oldest ever person  
candy\_combined <- candy\_combined %>%  
 mutate(age = ifelse(age>122, NA, age))  
  
  
## Cleaning year column ----------------------------------------------------  
  
  
# change the id value brought over from the bind rows to an actual year  
# alter the column to be numeric rather than character  
candy\_combined <- candy\_combined %>%  
 mutate(year = recode(year, "1" = "2015"),  
 year = recode(year, "2" = "2016"),  
 year = recode(year, "3" = "2017"),  
 year = as.numeric(year))  
  
  
# Write our data to candy\_clean.csv ---------------------------------------  
  
  
candy\_combined %>%  
write\_csv(here("clean\_data/candy\_clean.csv"))

# Answers to Task Brief Questions

# Any other analysis or conclusions

Dont allow people free reign with a survey - limit possible responses