Data Cleaning Part 1

Data Wrangling in R

Data Cleaning

In general, data cleaning is a process of investigating your data for inaccuracies, or recoding it in a way that makes it more manageable.

MOST IMPORTANT RULE - LOOK AT YOUR DATA!

Read in the UFO dataset

Read in data or download from: http://sisbid.github.io/Data-Wrangling/data/ufo/ufo_data_complete.csv.gz

```
ufo <- read_delim("https://sisbid.github.io/Data-Wrangling/
```

```
Warning: One or more parsing issues, call `problems()` on y
e.g.:
  dat <- vroom(...)
  problems(dat)
```

```
Rows: 88875 Columns: 11
-- Column specification --------------------------------
Delimiter: ","
chr (10): datetime, city, state, country, shape, duration (
dbl  (1): duration (seconds)

i Use `spec()` to retrieve the full column specification fo
i Specify the column types or set `show_col_types = FALSE`
```

The "problems"

You saw warning messages when reading in this dataset. We can see these with the `problems()` function from `readr`.

If we scroll through we can see some interesting notes.

```
p <-problems(ufo)
p %>% glimpse()
```

```
Rows: 200
Columns: 5
$ row      <int> 878, 1713, 1815, 2858, 3734, 4756, 5389, 5
$ col      <int> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12
$ expected <chr> "11 columns", "11 columns", "11 columns",
$ actual   <chr> "12 columns", "12 columns", "12 columns",
$ file     <chr> "", "", "", "", "", "", "", "", "", "", ""
```

Any unique problems?

```
count(p, expected, actual)
```

```
# A tibble: 5 x 3
  expected   actual           n
  <chr>      <chr>        <int>
1 11 columns "12 columns"   196
2 a double   "0.5`"           1
3 a double   "2`"             1
4 a double   "2631600  "      1
5 a double   "8`"             1
```

The "problems"

Go look at the raw data around these rows.

```
4/10/2005 22:52 santa cru
38453    el  zumbador (el
```

```
colnames(ufo)
```

```
 [1] "datetime"                "city"                    "state"
 [4] "country"                 "shape"                   "duratio
 [7] "duration (hours/min)"    "comments"                "date po
[10] "latitude"                "longitude"
```

Reading in again

Now we have a chance to keep but clean these values!

```
ufo <-read_csv("https://sisbid.github.io/Data-Wrangling/dat
```

```
Warning: One or more parsing issues, call `problems()` on y
e.g.:
  dat <- vroom(...)
  problems(dat)
```

```
dim(ufo)
```

```
[1] 88875    11
```

```
p <- problems(ufo)

count(p, expected, actual)
```

```
# A tibble: 1 x 3
  expected   actual          n
  <chr>      <chr>       <int>
1 11 columns 12 columns    196
```

Drop the remaining shifted problematic rows for now

Multiply by negative one to drop the rows. Use the Slice function to "select" those rows based on the index.

```
ufo_clean <- ufo %>% slice((pull(p, row))*-1)
```

Checking

```
nrow(ufo)-nrow(ufo_clean)
```

```
[1] 196
```

Clean names with the `clean_names()` function from the `janitor` package

```
colnames(ufo_clean)
```

```
 [1] "datetime"                "city"                    "state"
 [4] "country"                 "shape"                   "duratio
 [7] "duration (hours/min)"    "comments"                "date po
[10] "latitude"                "longitude"
```

```
ufo_clean <- clean_names(ufo_clean)
colnames(ufo_clean)
```

```
 [1] "datetime"              "city"                  "state"
 [4] "country"               "shape"                 "duration_se
 [7] "duration_hours_min"    "comments"              "date_posted
[10] "latitude"              "longitude"
```

Recoding Variables

Exact Swaps - recode function

```
ufo_clean %>% mutate(country = recode(country, gb = "Great
```

```
Rows: 88,679
Columns: 11
$ datetime              <chr> "10/10/1949 20:30", "10/10/1949
$ city                  <chr> "san marcos", "lackland afb", "c
$ state                 <chr> "tx", "tx", NA, "tx", "hi", "tn"
$ country               <chr> "us", NA, "Great Britain", "us",
$ shape                 <chr> "cylinder", "light", "circle", "
$ duration_seconds      <chr> "2700", "7200", "20", "20", "900
$ duration_hours_min    <chr> "45 minutes", "1-2 hrs", "20 sec
$ comments              <chr> "This event took place in early
$ date_posted           <chr> "4/27/2004", "12/16/2005", "1/21
$ latitude              <chr> "29.8830556", "29.38421", "53.2"
$ longitude             <chr> "-97.9411111", "-98.581082", "-2
```

Exact Swaps - recode function

```
ufo_clean %>% mutate(country = recode(country, gb = "Great
                                              "us" = "Unit
```

```
Rows: 88,679
Columns: 11
$ datetime              <chr> "10/10/1949 20:30", "10/10/1949
$ city                  <chr> "san marcos", "lackland afb", "c
$ state                 <chr> "tx", "tx", NA, "tx", "hi", "tn"
$ country               <chr> "United States", NA, "Great Brit
$ shape                 <chr> "cylinder", "light", "circle", "
$ duration_seconds      <chr> "2700", "7200", "20", "20", "900
$ duration_hours_min    <chr> "45 minutes", "1-2 hrs", "20 sec
$ comments              <chr> "This event took place in early
$ date_posted           <chr> "4/27/2004", "12/16/2005", "1/21
$ latitude              <chr> "29.8830556", "29.38421", "53.2"
$ longitude             <chr> "-97.9411111", "-98.581082", "-2
```

Strange country values

Sometimes `country` is `NA` even though `state` is known. A conditional more flexible recoding would be helpful...

```
head(ufo_clean)
```

```
# A tibble: 6 x 11
  datetime      city  state country shape durat~1 durat~2 co
  <chr>         <chr> <chr> <chr>   <chr> <chr>   <chr>   <c
1 10/10/1949 ~ san ~ tx    us      cyli~ 2700    45 min~ Th
2 10/10/1949 ~ lack~ tx    <NA>    light 7200    1-2 hrs 19
3 10/10/1955 ~ ches~ <NA>  gb      circ~ 20      20 sec~ Gr
4 10/10/1956 ~ edna  tx    us      circ~ 20      1/2 ho~ My
5 10/10/1960 ~ kane~ hi    us      light 900     15 min~ AS
6 10/10/1961 ~ bris~ tn    us      sphe~ 300     5 minu~ My
# ... with 1 more variable: longitude <chr>, and abbreviate
#   1: duration_seconds, 2: duration_hours_min, 3: comments
#   5: latitude
```

Deeper look

```
ufo_clean %>% filter(state == "tx") %>% count(country, stat
```

```
# A tibble: 2 x 3
  country state     n
  <chr>   <chr> <int>
1 us      tx     3734
2 <NA>    tx      307
```

```
ufo_clean %>% filter(state == "tx" & is.na(country))
```

```
# A tibble: 307 x 11
   datetime    city  state country shape durat~1 durat~2 co
   <chr>       <chr> <chr> <chr>   <chr> <chr>   <chr>   <c
 1 10/10/1949~ lack~ tx    <NA>    light 7200    1-2 hrs 19
 2 10/10/1994~ merc~ tx    <NA>    cigar 3600    1 hour  ui
 3 10/1/1981 ~ texa~ tx    <NA>    unkn~ 0       <NA>    UH
 4 10/12/2009~ hous~ tx    <NA>    light 60      about ~ A
 5 10/14/1964~ bett~ tx    <NA>    other 0       don&#3~ I
 6 10/15/2004~ dall~ tx    <NA>    fire~ 1800    20-30 ~ NO
```

Case_when

```
casewhen(test ~ value if test is true,
         test2 ~ vlue if test2 is true,
         TRUE ~ value if all above tests are not true) # de
```

```
ufo_clean %>% mutate(country = case_when(
  country == "gb" ~ "Great Britain",
  country == "us" ~"United States",
  TRUE ~ country))%>%
  glimpse()
```

```
Rows: 88,679
Columns: 11
$ datetime            <chr> "10/10/1949 20:30", "10/10/1949
$ city                <chr> "san marcos", "lackland afb", "c
$ state               <chr> "tx", "tx", NA, "tx", "hi", "tn"
$ country             <chr> "United States", NA, "Great Brit
$ shape               <chr> "cylinder", "light", "circle", "
$ duration_seconds    <chr> "2700", "7200", "20", "20", "900
$ duration_hours_min <chr> "45 minutes", "1-2 hrs", "20 sec
```

How many countries?

```
ufo_clean %>% count(country)
```

```
# A tibble: 6 x 2
  country     n
  <chr>   <int>
1 au        592
2 ca       3259
3 de        111
4 gb       2043
5 us      70146
6 <NA>    12528
```

case_when() regions

```
ufo <- ufo %>% mutate(
              region = case_when(
                country %in% c("us", "ca") ~ "North America",
                country %in% c("de") ~ "Europe",
                country %in% "gb" ~ "Great Britain",
                TRUE ~ "Other"
              ))
ufo %>% select(country, region) %>% head()
```

```
# A tibble: 6 x 2
  country region
  <chr>   <chr>
1 us      North America
2 <NA>    Other
3 gb      Great Britain
4 us      North America
5 us      North America
6 us      North America
```

Get US States

```
ufo_clean %>% filter(country == "us") %>% count(state) %>%
```

```
 [1] "ak" "al" "ar" "az" "ca" "co" "ct" "dc" "de" "fl" "ga"
[16] "in" "ks" "ky" "la" "ma" "md" "me" "mi" "mn" "mo" "ms"
[31] "nh" "nj" "nm" "nv" "ny" "oh" "ok" "or" "pa" "pr" "ri"
[46] "ut" "va" "vt" "wa" "wi" "wv" "wy"
```

```
US_states <- ufo_clean %>% filter(country == "us") %>% cou
```

Get Canada States

```
ufo_clean %>% filter(country == "ca") %>% count(state) %>%
```

```
 [1] "ab" "bc" "mb" "nb" "nf" "ns" "nt" "on" "pe" "pq" "qc"
[16] NA
```

```
CA_states <- ufo_clean %>% filter(country == "ca") %>% coun
```

Get Great Britan states

```
ufo_clean %>% filter(country == "gb") %>% count(state) %>%
```

```
 [1] "bc" "la" "ms" "nc" "ns" "nt" "ri" "sk" "tn" "wv" "yt"
```

```
GB_states <- ufo_clean %>% filter(country == "gb") %>% cou
```

A small overlap with us states.

Get DE states

```
ufo_clean %>% filter(country == "de") %>% count(state) %>%
```

```
[1] NA
```

```
DE_states <- ufo_clean %>% filter(country == "de") %>% cou
```

Get AU states

```
ufo_clean %>% filter(country == "au") %>% count(state) %>%
```

```
[1] "al" "dc" "nt" "oh" "sa" "wa" "yt" NA
```

```
AU_states <- ufo_clean %>% filter(country == "au") %>% cou
```

Some overlap with US states.

Get just unique

```
US_states <- US_states[!(US_states %in% AU_states)]
US_states <- US_states[!(US_states %in% GB_states)]
US_states <- US_states[!(US_states %in% CA_states)]

AU_states <- AU_states[!(AU_states %in% US_states)]
AU_states <- AU_states[!(AU_states %in% GB_states)]
AU_states <- AU_states[!(AU_states %in% CA_states)]

CA_states <- CA_states[!(CA_states %in% US_states)]
CA_states <- CA_states[!(CA_states %in% GB_states)]
CA_states <- CA_states[!(CA_states %in% CA_states)]

GB_states <- GB_states[!(GB_states %in% US_states)]
GB_states <- GB_states[!(GB_states%in% AU_states)]
GB_states <- GB_states[!(GB_states %in% CA_states)]
```

How often do rows have a value for country but not the US?

```
ufo_clean %>% filter(country != "us" & !is.na(country)) %>%
```

```
# A tibble: 4 x 2
  country     n
  <chr>   <int>
1 au        592
2 ca       3259
3 de        111
4 gb       2043
```

what is de?

```
ufo_clean %>% filter(country == "de") %>% pull(city)
```

```
  [1] "berlin (germany)"
  [2] "berlin (germany)"
  [3] "obernheim (germany)"
  [4] "ottersberg (germany)"
  [5] "urbach (germany)"
  [6] "bremen (30 km south of) (germany)"
  [7] "sembach (germany)"
  [8] "magdeburg (germany)"
  [9] "neuruppin (germany)"
 [10] "lampertheim (germany)"
 [11] "ramstein (germany)"
 [12] "bremen (germany)"
 [13] "nurenburg (germany)"
 [14] "senftenberg (germany)"
 [15] "schwalmtal (germany)"
 [16] "neuss (germany)"
```

more complicated case_when

Let's make some assumptions for the sake of illustration.

```
ufo_clean <- ufo_clean %>% mutate(prob_country =
      case_when((is.na(country) & state %in% c(US_states))
                (is.na(country) & state %in% c(CA_states))
                (is.na(country) & state %in% c(AU_states))
                (is.na(country) & state %in% c(GB_states))
                   TRUE ~ country))
```

```
count(ufo_clean, prob_country)
```

```
# A tibble: 9 x 2
  prob_country      n
  <chr>         <int>
1 au              592
2 Australia       699
3 ca             3259
4 de              111
5 gb             2043
```

Even more specific

```
ufo_clean <- ufo_clean %>% mutate(prob_country =
       case_when(
       (is.na(country) & state %in% c(US_states))  |
  country == "us" ~ "United States",
       (is.na(country) & state %in% c(CA_states))  |
  country == "ca" ~ "Canada",
       (is.na(country) & state %in% c(AU_states))  |
  country == "au" ~ "Australia",
       (is.na(country) & state %in% c(GB_states))  |
  country == "gb" ~ "Great Britain",
        country == "de" ~ "Germany",
                    TRUE ~ country))
```

We would want to confirm what we recoded with the cities and latitude and longitude, especially to deal with the overlaps in the state lists.

Check counts

```
ufo_clean %>% count(country, prob_country)
```

```
# A tibble: 9 x 3
  country prob_country      n
  <chr>   <chr>          <int>
1 au      Australia        592
2 ca      Canada          3259
3 de      Germany          111
4 gb      Great Britain   2043
5 us      United States  70146
6 <NA>    Australia        699
7 <NA>    Great Britain   5395
8 <NA>    United States   5897
9 <NA>    <NA>             537
```

```
ufo_clean %>% count(prob_country)
```

```
# A tibble: 6 x 2
  prob_country      n
```

Summary

- `recode` makes exact swaps
- `case_when` can use conditionals, need to specify what value for if no conditions are met (can be the original value of a variable)

Lab

https://sisbid.github.io/Data-Wrangling/labs/data-cleaning-lab.Rmd