Uniformity

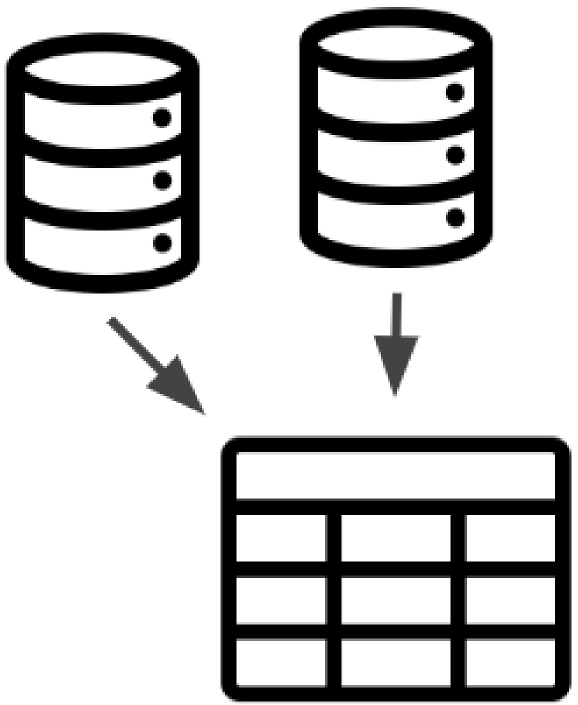
CLEANING DATA IN R

Maggie Matsui
Content Developer @ DataCamp

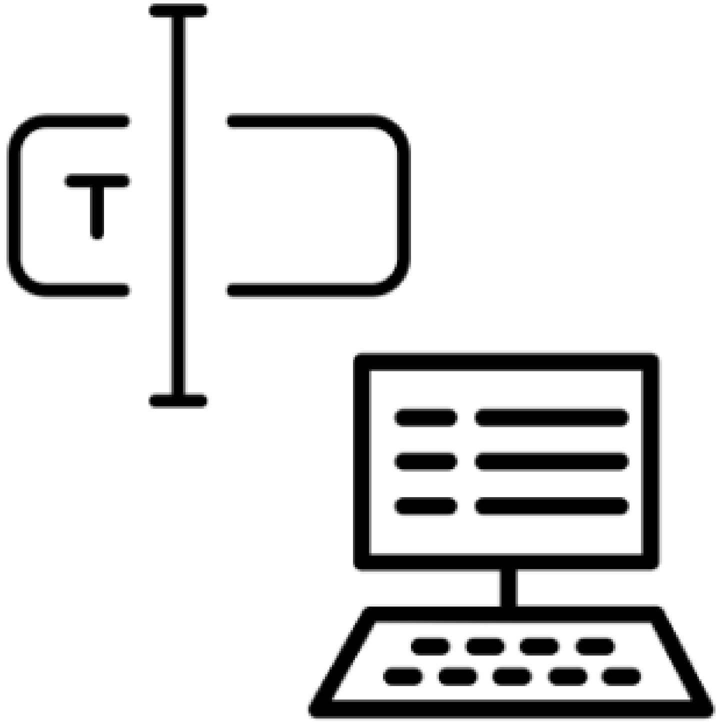
Uniformity

- Different units or formats
  - **Temperature:** `°C` vs. `°F`
  - **Weight:** `kg` vs. `g` vs. `lb`
  - **Money:** USD `$` vs. GBP `£` vs. JPY `¥`
  - **Date:** `DD-MM-YYYY` vs. `MM-DD-YYYY` vs. `YYYY-MM-DD`

Where do uniformity issues come from?



Multiple data sources



Data entry errors

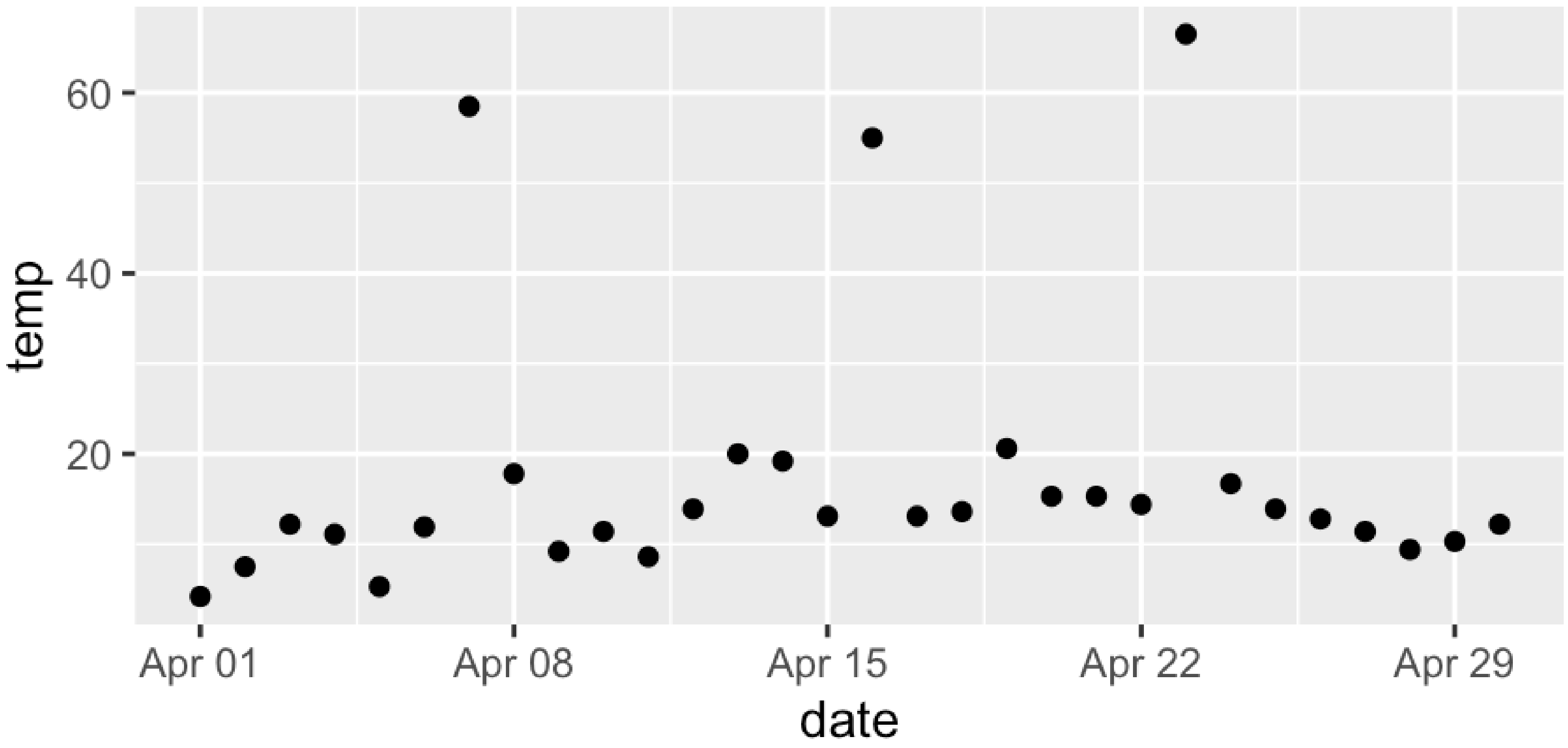
Finding uniformity issues

```
head(nyc_temps)
```

```
        date temp
1 2019-04-01  4.2
2 2019-04-02  7.5
3 2019-04-03 12.2
4 2019-04-04 11.1
5 2019-04-05 41.5
6 2019-04-06 11.9
```

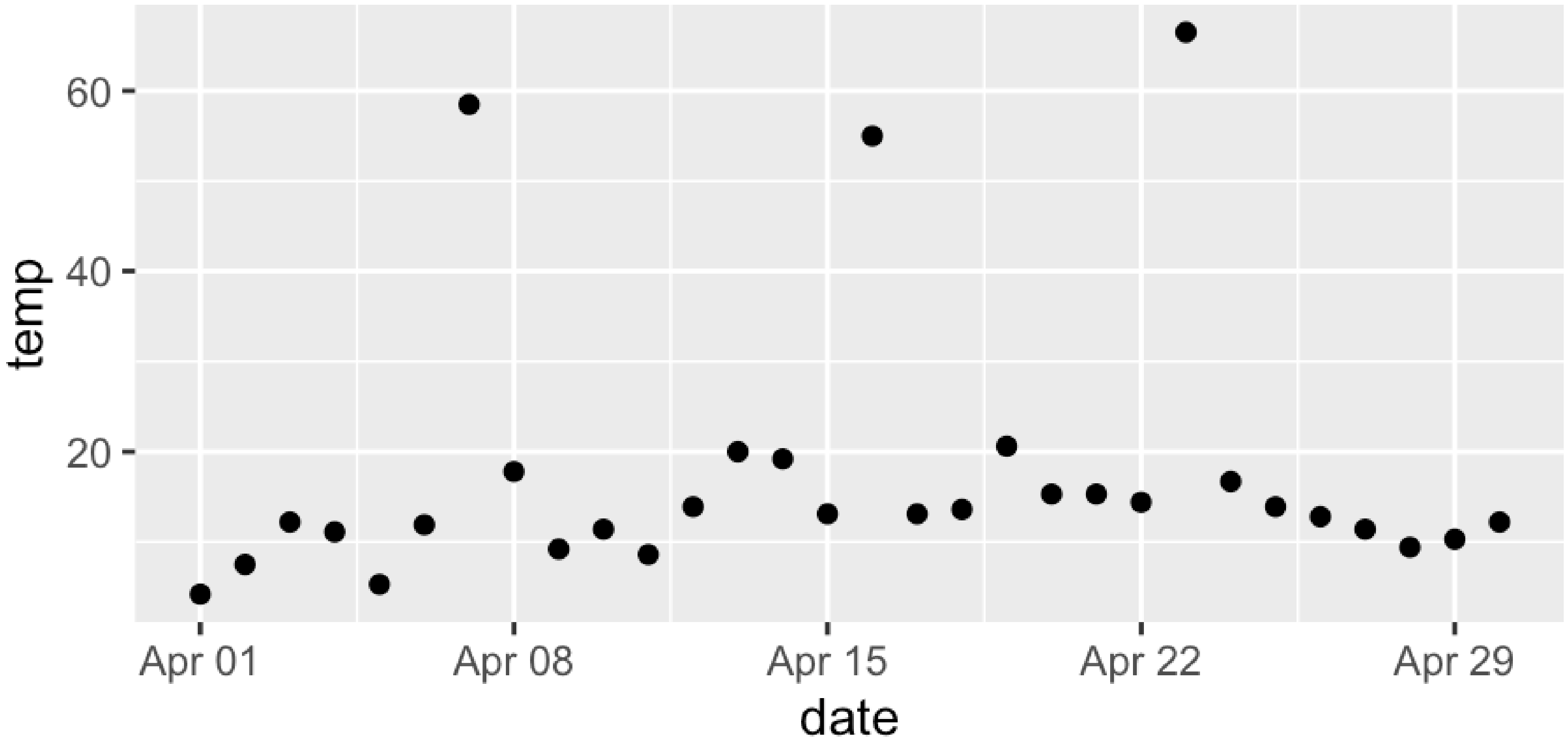
Finding uniformity issues

```
library(ggplot2)
ggplot(nyc_temps, aes(x = date, y = temp)) +
  geom_point()
```



What to do?

- There's no one best option. ***It depends on your dataset!***
- Do your research to understand where your data comes from



- Data from Apr 7, 16, and 23 is from an external source that measured temps in °F

Unit conversion

$$C = (F - 32) \times \frac{5}{9}$$

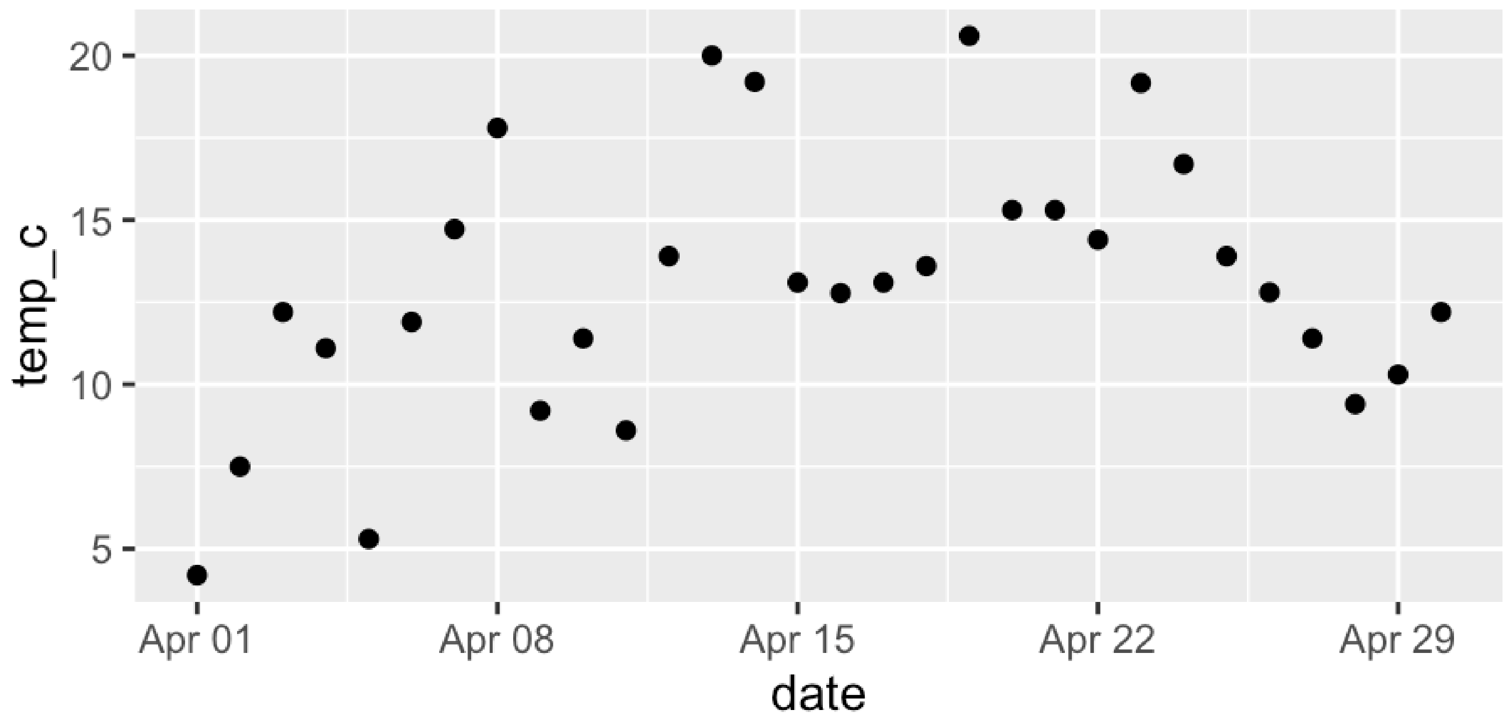
```
ifelse(condition, value_if_true, value_if_false)
```

```
nyc_temps %>%
  mutate(temp_c = ifelse(temp > 50, (temp - 32) * 5 / 9, temp))
```

```
         date temp   temp_c
1  2019-04-01  4.2  4.20000
...
7  2019-04-07 58.5 14.72222
...
```

Unit conversion

```
nyc_temps %>%
  mutate(temp_c = ifelse(temp > 50, (temp - 32) * 5 / 9, temp)) %>%
  ggplot(aes(x = date, y = temp_c)) +
    geom_point()
```



Date uniformity

```
nyc_temps
```

```
              date temp_c
1       2019-11-23   5.12
2         01/15/19  -0.67
3   April 24, 2019  17.46
4         08/30/19  26.46
5 October 3, 2019  14.63
6       2019-03-17   3.47
```

Date string	**Date format**
`"2019-11-23"`	`"%Y-%m-%d"`
`"01/15/19"`	`"%m-%d-%y"`
`"April 24, 2019"`	`"%B %d, %Y"`

`?strptime` in R console

Parsing multiple formats

```
library(lubridate)
parse_date_time(nyc_temps$date,
                orders = c("%Y-%m-%d", "%m/%d/%y", "%B %d, %Y"))
```

```
"2019-11-23 UTC" "2019-01-15 UTC" "2019-04-24 UTC" "2019-08-30 UTC"
"2019-10-03 UTC" "2019-03-17 UTC"
```

```
parse_date_time("Monday, January 3",
                orders = c("%Y-%m-%d", "%m/%d/%y", "%B %d, %Y"))
```

```
NA
```

Ambiguous dates

Is 02/04/2019 in February or April?

- Depends on your data!

Options include:

- Treat as missing
- If your data comes from multiple sources, infer based on source
- Infer based on other data in the dataset

Let's practice!

CLEANING DATA IN R

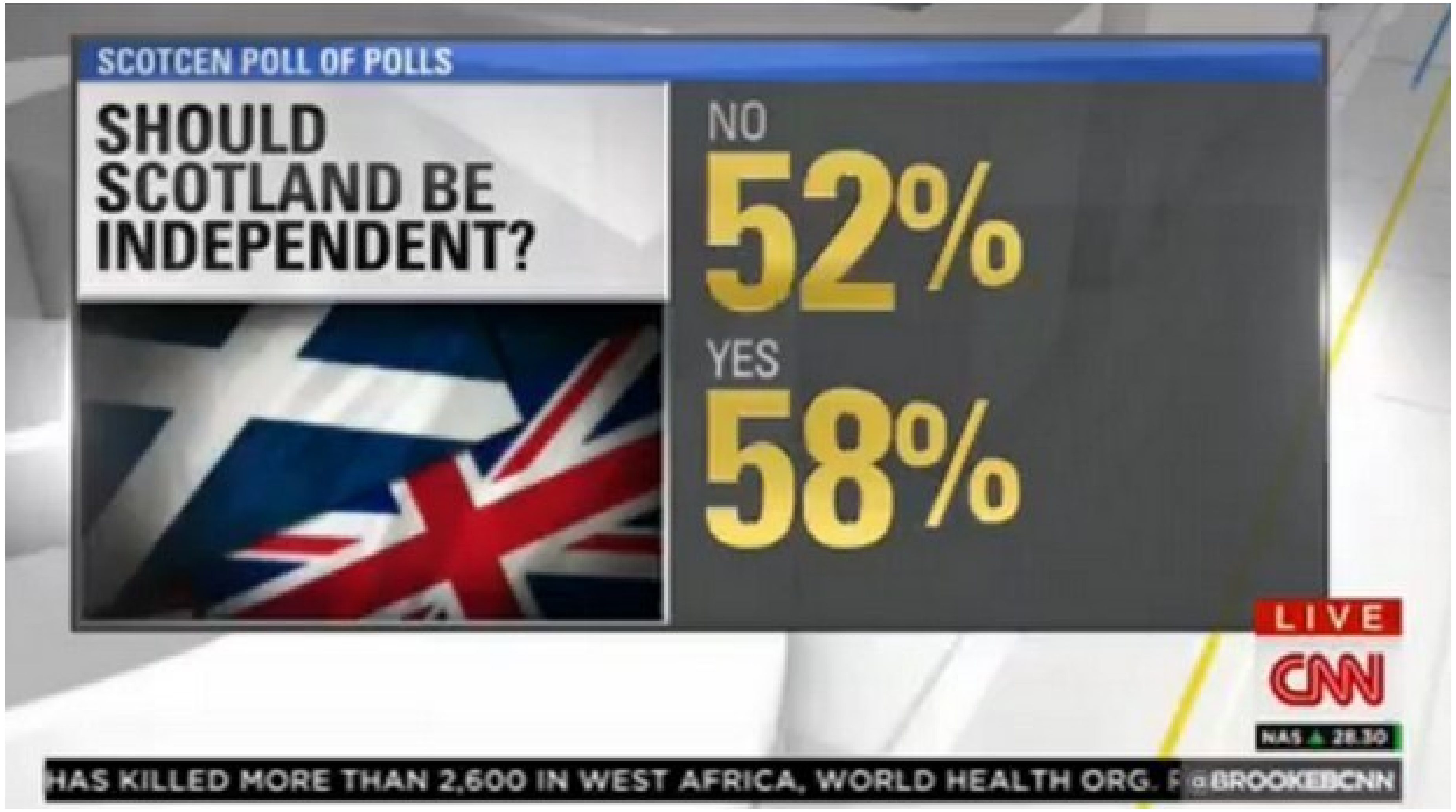
Cross field validation

CLEANING DATA IN R

Maggie Matsui
Content Developer @ DataCamp

What is cross field validation?

- Cross field validation = a sanity check
- *Does this value make sense based on other values?*



[1] https://www.buzzfeednews.com/article/katienotopoulos/graphs-that-lied-to-us

Credit card data

```
head(credit_cards)
```

```
  date_opened dining_cb groceries_cb  gas_cb total_cb acct_age
1  2018-07-05     26.08        83.43   78.90   188.41        1
2  2016-01-23   1309.33         4.46 1072.25  2386.04        4
3  2016-03-25    205.84       119.20  800.62  1125.66        4
4  2018-06-20     14.00        16.37   18.41    48.78        1
5  2017-02-08     98.50       283.68  281.70   788.33        3
6  2014-11-18    889.28      2626.34 2973.62  6489.24        5
```

Validating numbers

```
credit_cards %>%
  select(dining_cb:total_cb)
```

```
   dining_cb groceries_cb  gas_cb total_cb
1      26.08        83.43   78.90   188.41
2    1309.33         4.46 1072.25  2386.04
3     205.84       119.20  800.62  1125.66
4      14.00        16.37   18.41    48.78
5      98.50       283.68  281.70   788.33
6     889.28      2626.34 2973.62  6489.24
```

Validating numbers

```
credit_cards %>%
  mutate(theoretical_total = dining_cb + groceries_cb + gas_cb) %>%
  filter(theoretical_total != total_cb) %>%
  select(dining_cb:theoretical_total)
```

```
  dining_cb groceries_cb  gas_cb total_cb theoretical_total
1     98.50       283.68  281.70   788.33            663.88
2   3387.53       363.85 2706.42  4502.94           6457.80
```

Validating date and age

```
credit_cards %>%
  select(date_opened, acct_age)
```

```
   date_opened acct_age
1   2018-07-05        1
2   2016-01-23        4
3   2016-03-25        4
4   2018-06-20        1
5   2017-02-08        3
6   2014-11-18        5
```

Calculating age

```
library(lubridate)
date_difference <- as.Date("2015-09-04") %--% today()
date_difference
```

```
2015-09-04 UTC--2020-03-09 UTC
```

```
as.numeric(date_difference, "years")
```

```
4.511978
```

```
floor(as.numeric(date_difference, "years"))
```

```
4
```

Validating age

```
credit_cards %>%
  mutate(theor_age = floor(as.numeric(date_opened %--% today(), "years"))) %>%
  filter(theor_age != acct_age)
```

```
  date_opened acct_age dining_cb groceries_cb  gas_cb total_cb theor_age
1  2016-03-25        4    814.34       471.58 3167.41  4453.33         3
2  2018-03-06        3    238.48       186.05  213.84   638.37         2
```

What next?

Dropping Data

Set to missing and impute

Apply rules from domain knowledge

Let's practice!

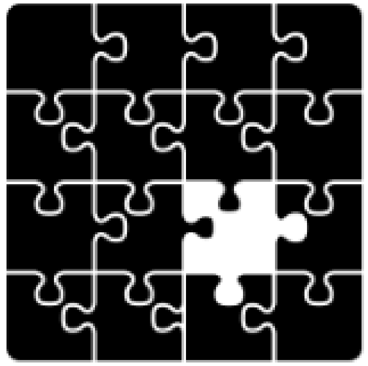
CLEANING DATA IN R

Completeness

CLEANING DATA IN R

Maggie Matsui
Content Developer @ DataCamp

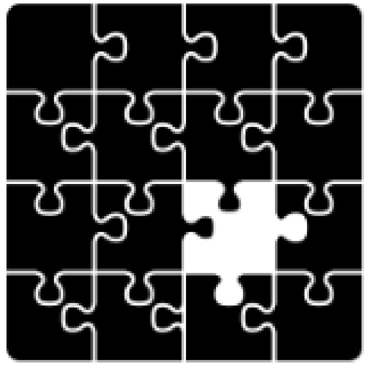
What is missing data?



Occurs when no data value is stored for a variable in an observation

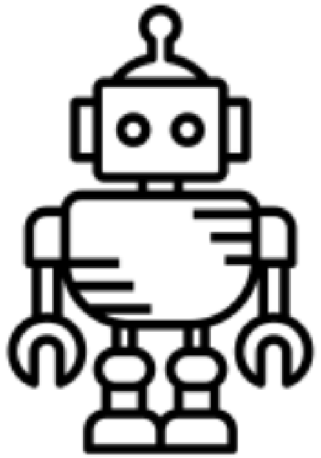
Can be represented as `NA`, `nan`, `0`, `99`, `.` ...

What is missing data?



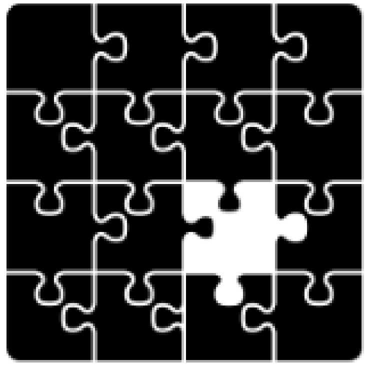
Occurs when no data value is stored for a variable in an observation

Can be represented as `NA`, `nan`, `0`, `99`, `.` ...



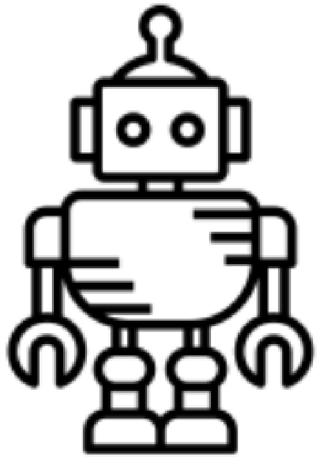
Technical error

What is missing data?



Occurs when no data value is stored for a variable in an observation

Can be represented as `NA` , `nan` , `0` , `99` , `.` ...



Technical error

Human error

Air quality

```
head(airquality)
```

```
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

Air quality

```
head(airquality)
```

```
   Ozone Solar.R Wind Temp Month Day
1      41     190  7.4   67     5   1
2      36     118  8.0   72     5   2
3      12     149 12.6   74     5   3
4      18     313 11.5   62     5   4
5 --> NA  --> NA 14.3   56     5   5
6      28  --> NA 14.9   66     5   6
```

Finding missing values

```
is.na(airquality)
```

```
     Ozone Solar.R  Wind  Temp Month   Day
[1,] FALSE   FALSE FALSE FALSE FALSE FALSE
[2,] FALSE   FALSE FALSE FALSE FALSE FALSE
[3,] FALSE   FALSE FALSE FALSE FALSE FALSE
[4,] FALSE   FALSE FALSE FALSE FALSE FALSE
[5,]  TRUE    TRUE FALSE FALSE FALSE FALSE
[6,] FALSE    TRUE FALSE FALSE FALSE FALSE
```

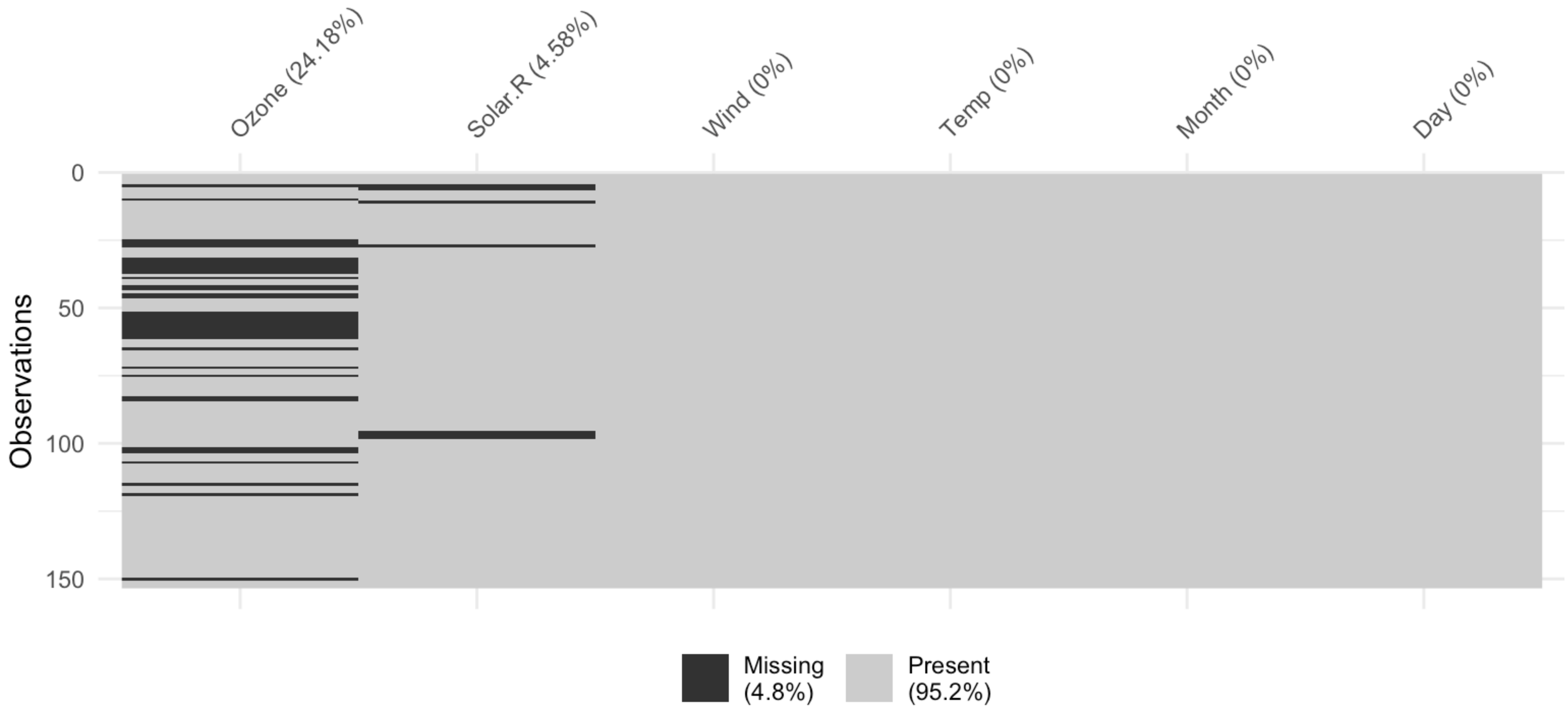
Counting missing values

```
# Count missing vals in entire dataset
sum(is.na(airquality))
```

```
44
```

Visualizing missing values

```
library(visdat)
vis_miss(airquality)
```



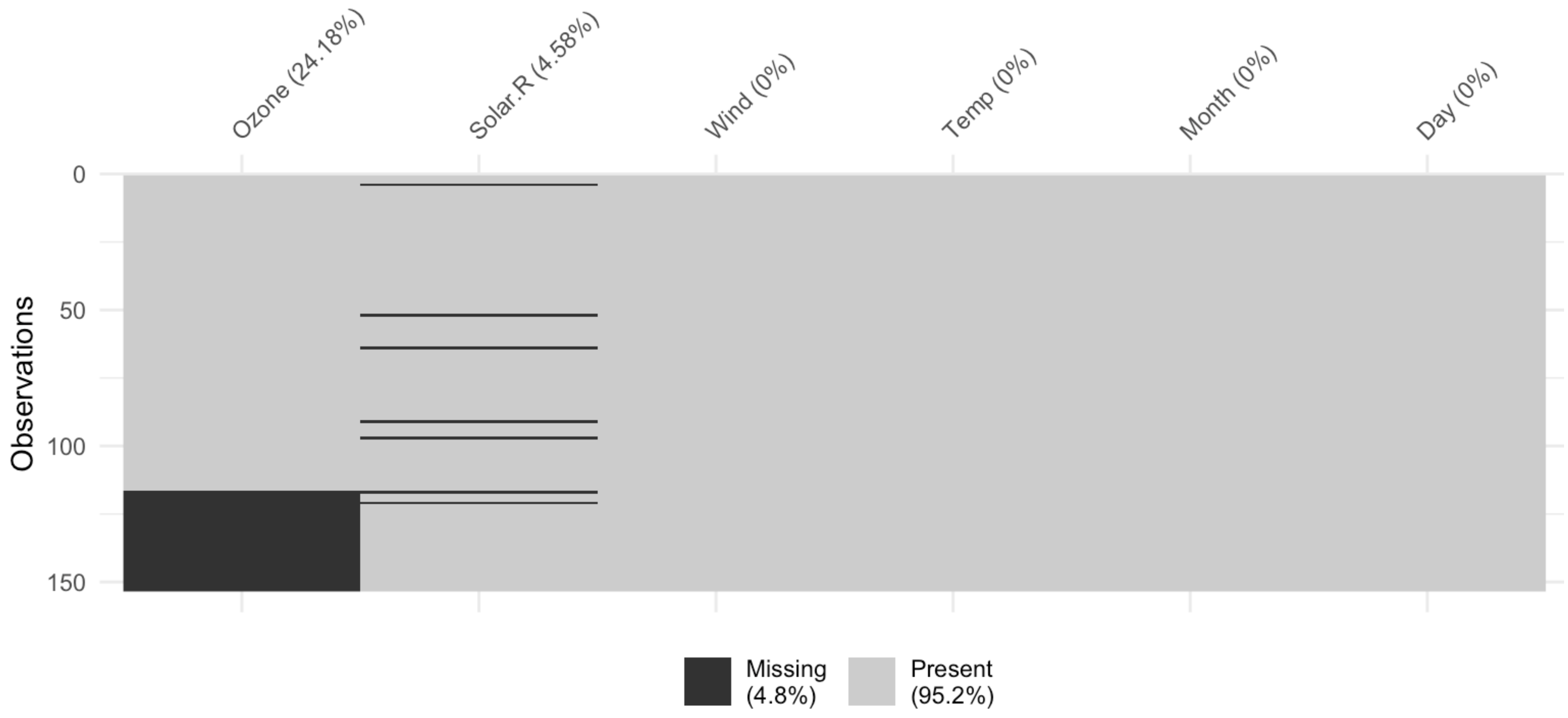
Investigating missingness

```
airquality %>%
  mutate(miss_ozone = is.na(Ozone)) %>%
  group_by(miss_ozone) %>%
  summarize_all(median, na.rm = TRUE)
```

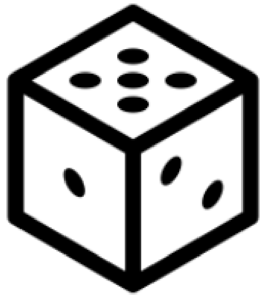
```
  miss_ozone Ozone Solar.R  Wind  Temp Month   Day
  <lgl>      <dbl>   <int> <dbl> <dbl> <dbl> <dbl>
1 FALSE       31.5     207   9.7    65     7    16
2 TRUE        NA       194   9.7    99     6    15
```

Investigating missingness

```
airquality %>%
  arrange(Temp) %>%
  vis_miss()
```

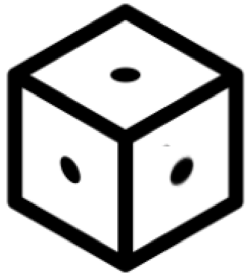


Types of missing data



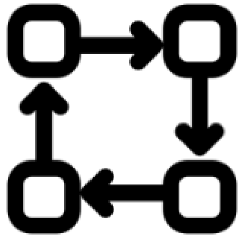
Missing Completely at Random

(MCAR)



Missing at Random

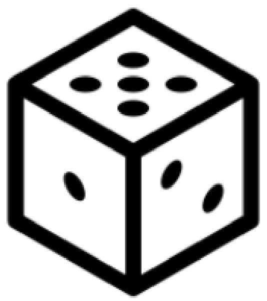
(MAR)



Missing Not at Random

(MNAR)

Types of missing data

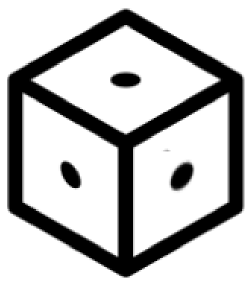


Missing Completely at Random

(MCAR)

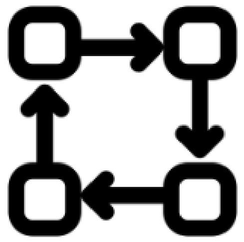
No systematic relationship between missing data and other values

Data entry errors when inputting data



Missing at Random

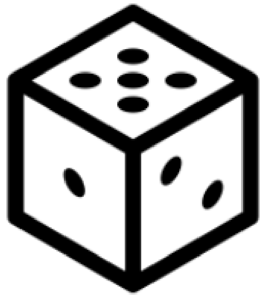
(MAR)



Missing Not at Random

(MNAR)

Types of missing data

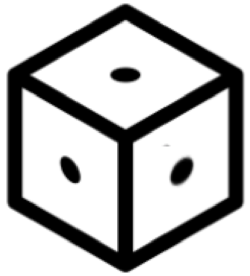


Missing Completely at Random

(MCAR)

No systematic relationship between missing data and other values

Data entry errors when inputting data

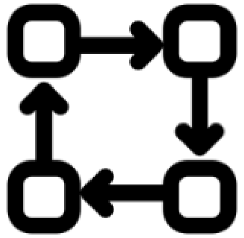


Missing at Random

(MAR)

Systematic relationship between missing data and other <u>observed</u> values

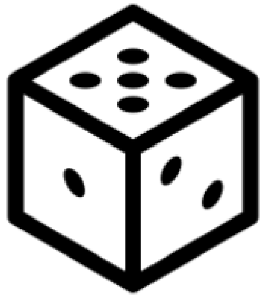
Missing ozone data for high temperatures



Missing Not at Random

(MNAR)

Types of missingness

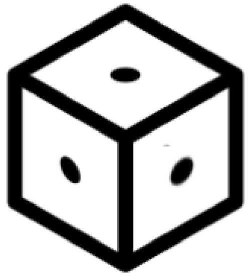


Missing Completely at Random

(MCAR)

No systematic relationship between missing data and other values

Data entry errors when inputting data

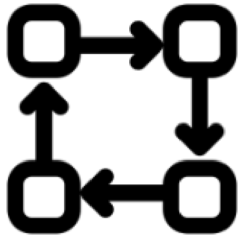


Missing at Random

(MAR)

*Systematic relationship between missing data and other **observed** values*

Missing ozone data for high temperatures



Missing Not at Random

(MNAR)

*Systematic relationship between missing data and **unobserved** values*

Missing temperature values for high temperatures

Dealing with missing data

Simple approaches:

1. Drop missing data
2. Impute (fill in) with statistical measures *(mean, median, mode..)* or domain knowledge

More complex approaches:

1. Impute using an algorithmic approach
2. Impute with machine learning models

Learn more in ***Dealing with Missing Data in R***

Dropping missing values

```
airquality %>%
  filter(!is.na(Ozone), !is.na(Solar.R))
```

```
   Ozone Solar.R  Wind  Temp Month   Day
   <int>   <int> <dbl> <int> <int> <int>
1     41     190   7.4    67     5     1
2     36     118     8    72     5     2
3     12     149  12.6    74     5     3
4     18     313  11.5    62     5     4
5     23     299   8.6    65     5     7
6     19      99  13.8    59     5     8
```

Replacing missing values

```
airquality %>%
  mutate(ozone_filled = ifelse(is.na(Ozone), mean(Ozone, na.rm = TRUE), Ozone))
```

```
   Ozone Solar.R  Wind  Temp Month   Day ozone_filled
   <int>   <int> <dbl> <int> <int> <int>        <dbl>
1     41     190   7.4    67     5     1          41
2     36     118   8      72     5     2          36
3     12     149  12.6    74     5     3          12
4     18     313  11.5    62     5     4          18
5     NA      NA  14.3    56     5     5          42.1
```

Let's practice!

CLEANING DATA IN R