

# Problem set 2

Put your name here

## Table of contents

Getting started . . . . .	1
Read the data . . . . .	1
Clean the data . . . . .	2
Manipulate the data using <code>dplyr</code> . . . . .	2
Tidying the data . . . . .	4

## Getting started

```
library(tidyverse)
library(janitor)
```

## Read the data

```
Rows: 195 Columns: 9
-- Column specification -----
Delimiter: ","
chr (1): Breed
dbl (8): 2013 Rank, 2014 Rank, 2015 Rank, 2016 Rank, 2017 Rank, 2018 Rank, 2...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
Rows: 195 Columns: 17
-- Column specification -----
Delimiter: ","
chr (3): Breed, Coat Type, Coat Length
dbl (14): Affectionate With Family, Good With Young Children, Good With Othe...
```

- i Use ``spec()`` to retrieve the full column specification for this data.
- i Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

```
breed_rank <- read_csv("data/breed_rank.csv")
breed_traits <- read_csv("data/breed_traits.csv")
```

## Clean the data

Display variables.

```
names(breed_rank)
```

```
[1] "Breed"      "2013 Rank" "2014 Rank" "2015 Rank" "2016 Rank" "2017 Rank"
[7] "2018 Rank" "2019 Rank" "2020 Rank"
```

```
names(breed_traits)
```

```
[1] "Breed"                                "Affectionate With Family"
[3] "Good With Young Children"            "Good With Other Dogs"
[5] "Shedding Level"                      "Coat Grooming Frequency"
[7] "Drooling Level"                     "Coat Type"
[9] "Coat Length"                        "Openness To Strangers"
[11] "Playfulness Level"                  "Watchdog/Protective Nature"
[13] "Adaptability Level"                 "Trainability Level"
[15] "Energy Level"                      "Barking Level"
[17] "Mental Stimulation Needs"
```

Make better names.

```
breed_traits <- breed_traits |>
  clean_names()
```

## Manipulate the data using dplyr

Make a summary.

```
breed_traits |>
  group_by(shedding_level) |>
  summarise(n = n())
```

```
# A tibble: 6 x 2
  shedding_level    n
      <dbl> <int>
1           0     1
2           1    27
3           2    41
4           3   109
5           4    16
6           5     1
```

Filter the shedding\_level 0.

```
breed_traits <- breed_traits |>
  filter(shedding_level != 0)
```

Check if manipulation was successful.

```
breed_traits |> count(shedding_level)
```

```
# A tibble: 5 x 2
  shedding_level    n
      <dbl> <int>
1           1    27
2           2    41
3           3   109
4           4    16
5           5     1
```

Make an untidy data frame.

```
untidy_scores <- breed_traits |>
  mutate(untidy_score = shedding_level +
         coat_grooming_frequency + drooling_level) |>
  select(breed, untidy_score)
```

Arrange scores in descending order.

```
untidy_scores |>
  arrange(desc(untidy_score))
```

```
# A tibble: 194 x 2
  breed                untidy_score
  <chr>                <dbl>
1 Bernese Mountain Dogs      11
2 Leonbergers                11
3 Newfoundlands              10
4 Bloodhounds                10
5 St. Bernards               10
6 Old English Sheepdogs      10
7 Dogues de Bordeaux         10
8 Neapolitan Mastiffs        10
9 Black Russian Terriers     10
10 Tibetan Mastiffs          10
# i 184 more rows
```

## Tidying the data

How does this data set fail to meet the criteria for tidy data?

There are three interrelated rules which make a dataset tidy:

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

We have a year and a rank variable, but neither of these variables have their own column. Shown above is one observation, by dog breed. But that “one” observation is actually eight separate observations: the rank in 2013, the rank in 2014, etc. Each observation needs to have its own row.

Make pivoted data with a year and a rank variable.

```
ranks_pivoted <- breed_rank |>
  pivot_longer(`2013 Rank`:`2020 Rank`,
              names_to = "year",
              values_to = "rank")
```

Rename breed and make the year variable numeric.

```
ranks_pivoted <- ranks_pivoted |>
  rename(breed = Breed) |>
  mutate(year = parse_number(year))
```

Filter data to only Bernese Mountain Dogs.

```
ranks_pivoted <- ranks_pivoted |>  
  filter(str_detect(breed, "Bernese"))
```

Plot rankings across time.

```
ranks_pivoted |>  
  ggplot(aes(x = year, y = rank, label = rank)) +  
  geom_point(size = 3) +  
  geom_text(vjust = 2)
```

