Entering / cleaning data 1

Data cleaning

Cleaning data

Common data-cleaning tasks include:

Task	`dplyr` functions
Renaming columns	`rename`
Extracting certain columns	`select`
Extracting or arranging rows	`slice`, `sample_n`, `filter`, `arrange`
Adding or changing columns	`mutate`

The “tidyverse”

Today, we’ll talk about using functions from the `dplyr` pakcage, as well as a package for working with a specific type of data (`stringr` for character strings, which is part of the “tidyverse”, like the `readr` package.

To use these functions, you’ll need to load those packages:

```
library("dplyr")
library("stringr")
```

Cleaning data

As an example of cleaning data, we'll work with the Daily Show data:

```
daily_show <- read_csv("data/daily_show_guests.csv",
                       skip = 4)
```

```
head(daily_show, 3)
```

```
## # A tibble: 3 x 5
##    YEAR GoogleKnowlege_~ Show  Group
##   <dbl> <chr>            <chr> <chr>
## 1  1999 actor            1/11~ Acti~
## 2  1999 Comedian         1/12~ Come~
## 3  1999 television actr~ 1/13~ Acti~
## # ... with 1 more variable: Raw_Guest_List <chr>
```

Re-naming columns

A first step is often re-naming columns. It can be hard to work with a column name that is:

- long
- doesn't following the naming rules for R objects
- includes upper case

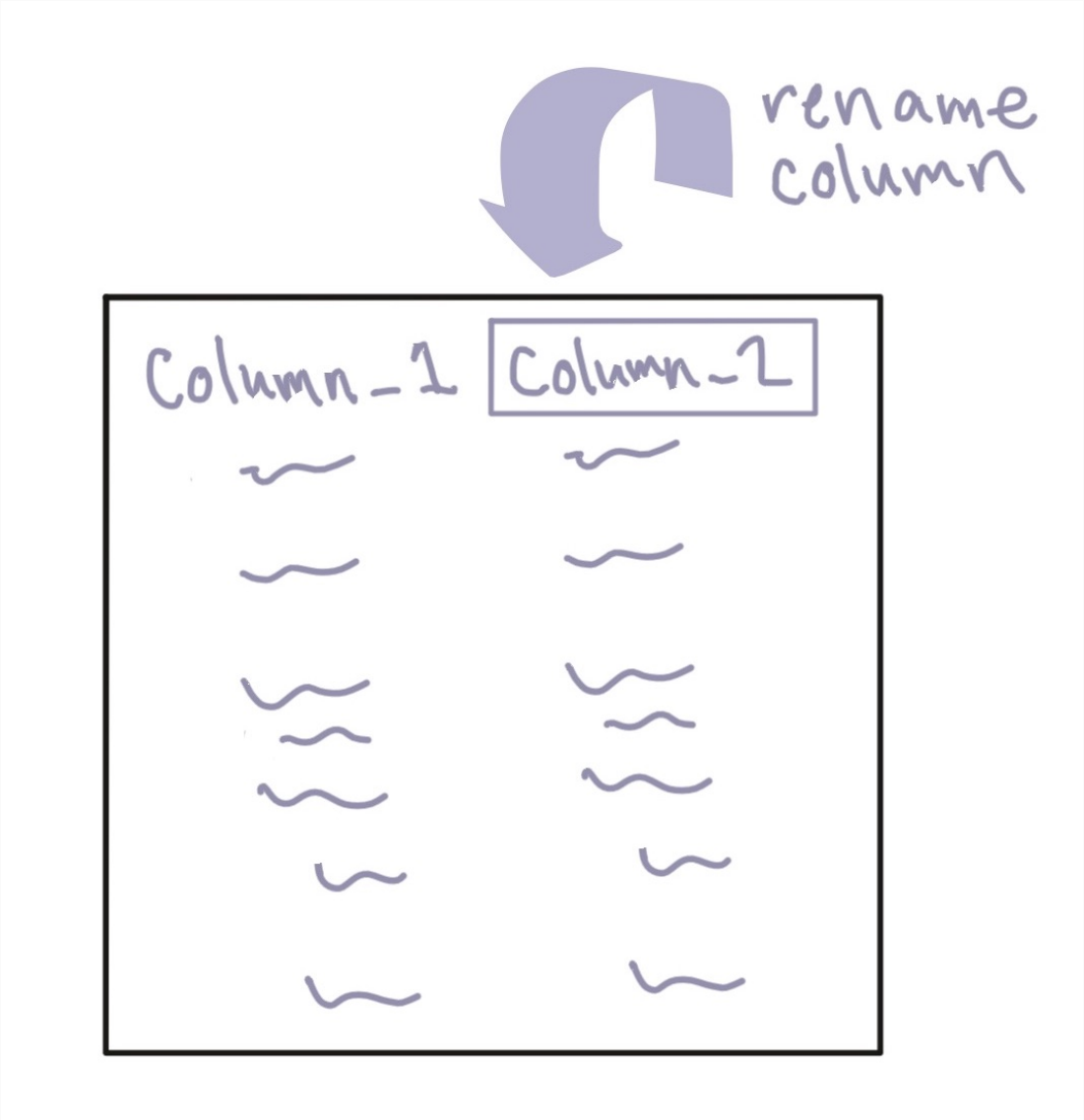
Several of the column names in `daily_show` have some of these issues:

```
colnames(daily_show)
```

```
## [1] "YEAR"
## [2] "GoogleKnowlege_Occupation"
## [3] "Show"
## [4] "Group"
## [5] "Raw_Guest_List"
```

Renaming columns

To rename these columns, use the `rename` function from `dplyr`.



Renaming columns

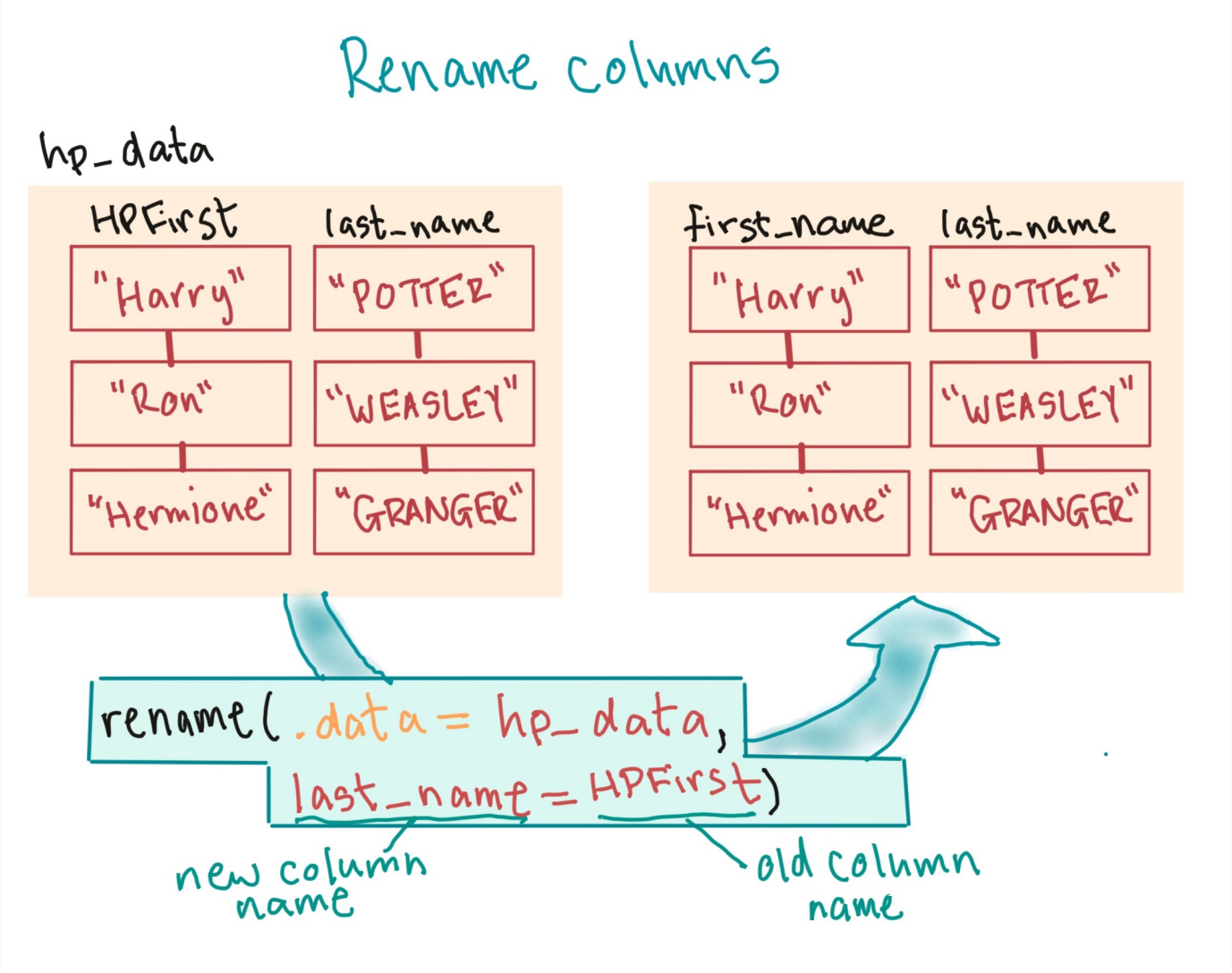
The basic syntax of `rename` is:

```
## Generic code
rename(dataframe,
       new_column_name_1 = old_column_name_1,
       new_column_name_2 = old_column_name_2)
```

If you want to change column names in the saved object, be sure you reassign the object to be the output of `rename`.

Renaming columns

Here's a basic example of using `rename`:



Renaming columns

To rename columns in the daily_show data, then, use:

```
daily_show <- rename(daily_show,
                     year = YEAR,
                     job = GoogleKnowlege_Occupation,
                     date = Show,
                     category = Group,
                     guest_name = Raw_Guest_List)
head(daily_show, 3)
```

```
## # A tibble: 3 x 5
##    year job           date   category guest_name
##   <dbl> <chr>         <chr>  <chr>    <chr>
## 1  1999 actor         1/11/~ Acting   Michael J. ~
## 2  1999 Comedian      1/12/~ Comedy   Sandra Bern~
## 3  1999 television a~ 1/13/~ Acting   Tracey Ullm~
```

Renaming columns

As a quick check, what is the difference between these two calls?

```
# 1.
rename(daily_show,
       year = YEAR,
       job = GoogleKnowlege_Occupation,
       date = Show,
       category = Group,
       guest_name = Raw_Guest_List)
```

```
# 2.
daily_show <- rename(daily_show,
                     year = YEAR,
                     job = GoogleKnowlege_Occupation,
                     date = Show,
                     category = Group,
                     guest_name = Raw_Guest_List)
```