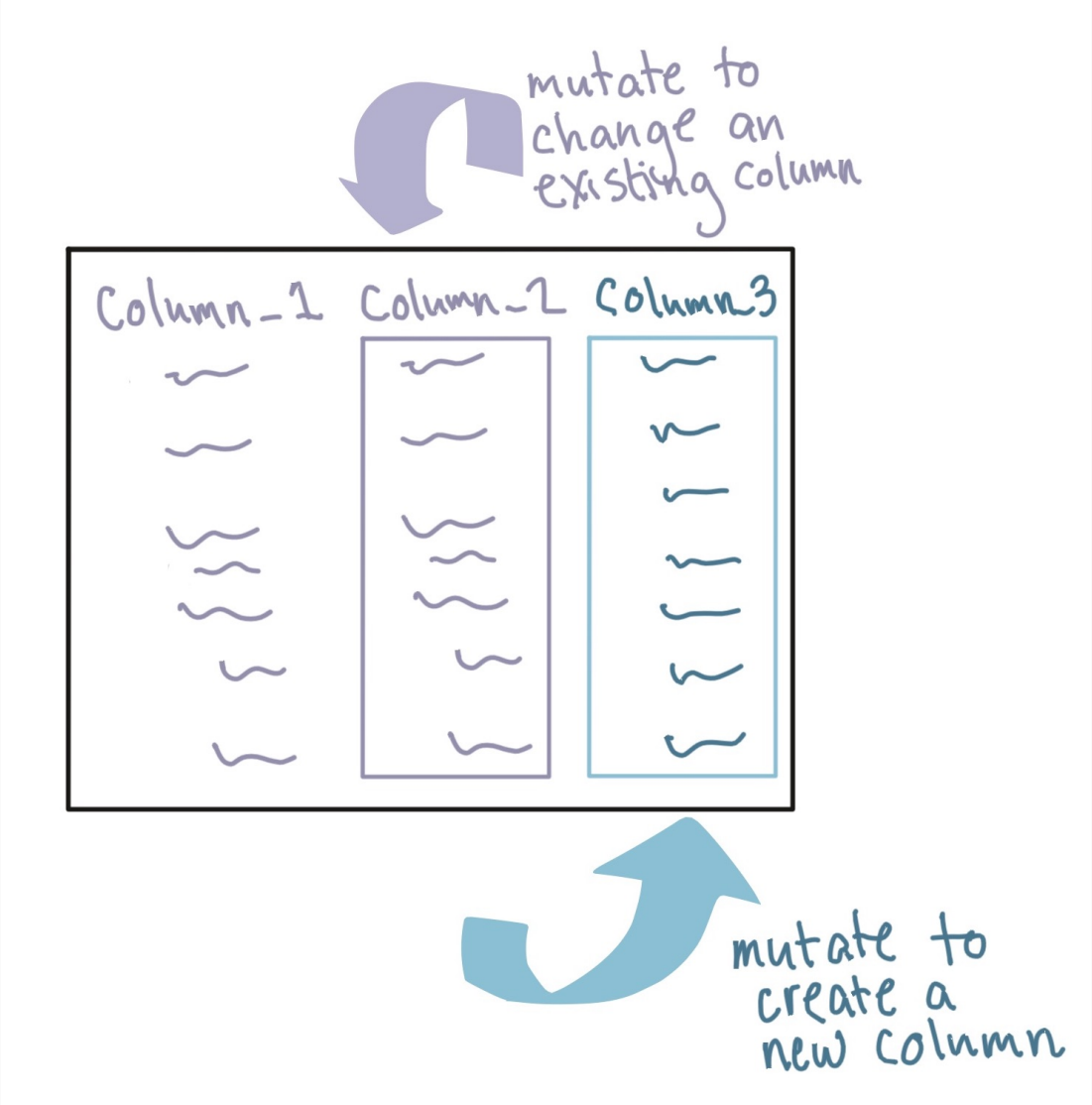
Entering / cleaning data 1

Data cleaning: Adding or changing columns

Add or change columns

You can **change** a column or **add** a new column using the `mutate` function from the `dplyr` package.



Add or change columns

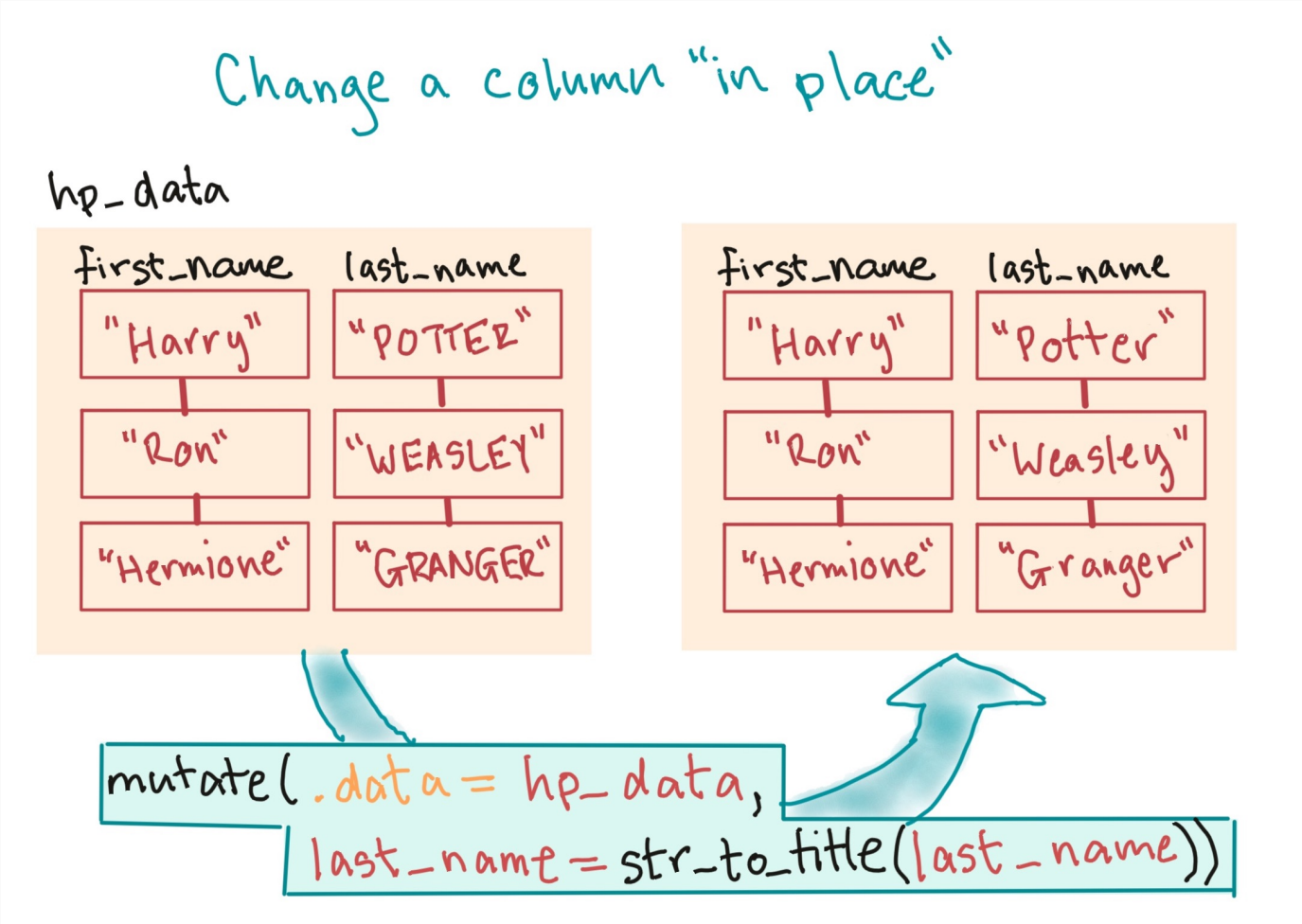
The `mutate` function has the syntax:

```
# Generic code
mutate(.data =dataframe,
       changed_column = function(original column),
       new_column = function(original columns))
```

- If you want to just **change** a column (in place), use its original name on the left of the equation.
- If you want to **add** a new column, use a new column name (you pick what that name will be) on the left of the equation (this will be the name of the new column).

Add or change columns

You can use `mutate` to *change* an existing column.



Add or change columns

For example, the `job` column in `daily_show` sometimes uses upper case and sometimes does not:

```
daily_show$job[1:10]
```

```
##  [1] "neurosurgeon"   "scientist"
##  [3] "physician"      "doctor"
##  [5] "astronaut"      "Astrophysicist"
##  [7] "Surgeon"        "physician"
##  [9] "Astrophysicist" "Neuroscientist"
```

Add or change columns

There is a package called `stringr` for working with character strings.

We could use the `str_to_lower` function from the `stringr` package to make all listings lowercase:

```
library("stringr")
daily_show <- mutate(.data = daily_show,
                     job = str_to_lower(string = job))
```

Add or change columns

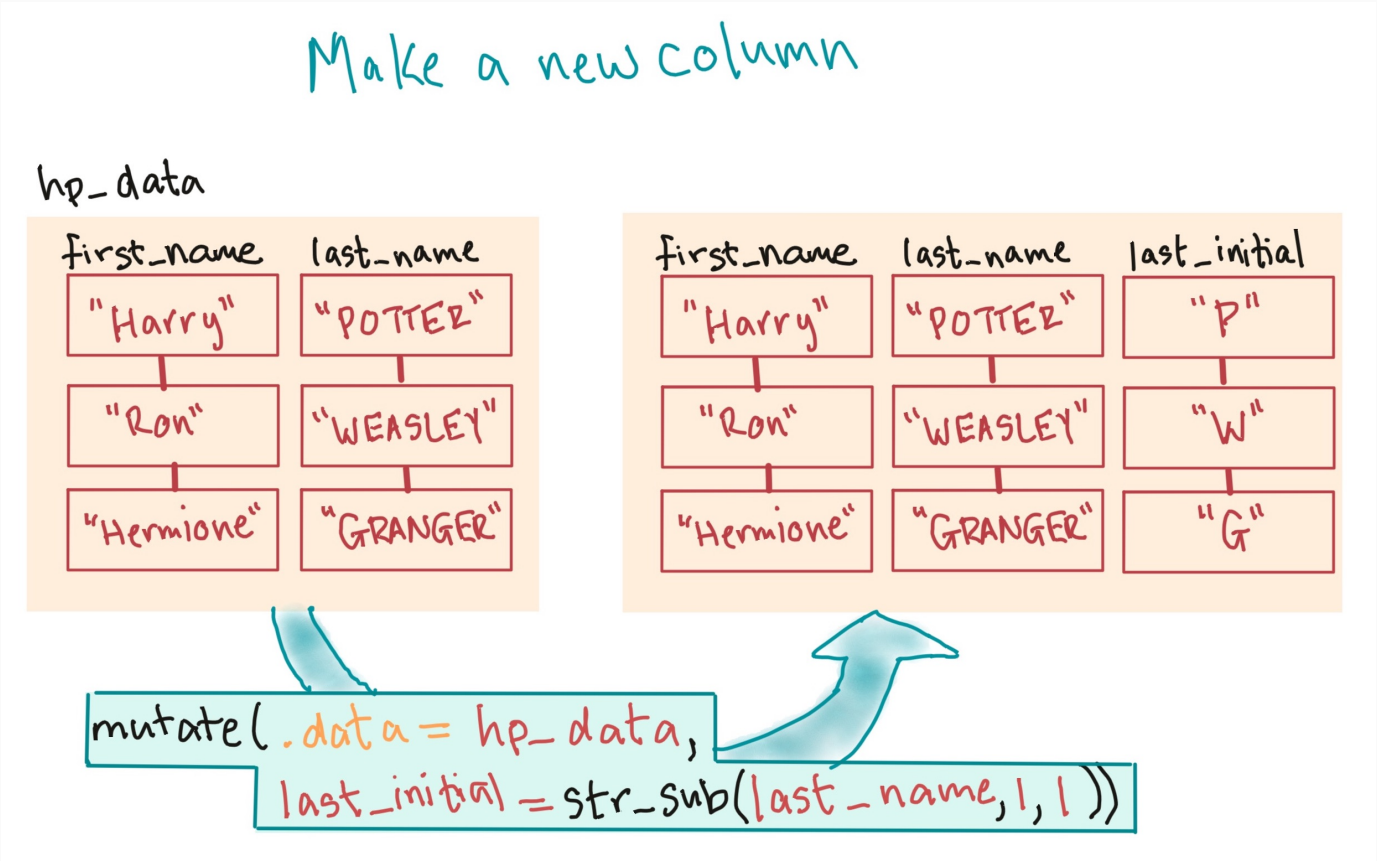
Now all job listings are in lowercase:

```
daily_show$job[1:10]
```

```
##  [1] "neurosurgeon"   "scientist"
##  [3] "physician"      "doctor"
##  [5] "astronaut"      "astrophysicist"
##  [7] "surgeon"        "physician"
##  [9] "astrophysicist" "neuroscientist"
```

Add or change columns

Alternatively, you can also use `mutate` to *add* a new column to the dataframe.



Add or change columns

For example, you could add a column called `uc_job` with the job name in uppercase letters:

```
daily_show <- mutate(.data = daily_show,
                     uc_job = str_to_upper(string = job))
slice(select(.data = daily_show, job, uc_job), 1:3)
```

```
## # A tibble: 3 x 2
##   job          uc_job
##   <chr>        <chr>
## 1 neurosurgeon NEUROSURGEON
## 2 scientist    SCIENTIST
## 3 physician    PHYSICIAN
```