Wrangling Tools

K Arnold

`case_when`

`if-elif` version (Python):

```
if age < 0:
    return "invalid"
elif age < 18:
    return "child"
else:
    return "adult"
```

`case_when` version:

```
age <- 18
case_when(
  age < 0   ~ "invalid",
  age < 18  ~ "child",
  TRUE      ~ "adult"
)
```

```
## [1] "adult"
```

- first to `True` wins in both versions
- `TRUE` corresponds to `else` (the default)

`case_when` vectorizes

Like many R functions, it actually applies to all elements of a vector.

```
age <- c(-1, 0, 17, 18) # a vector
case_when(
  age < 0   ~ "invalid",
  age < 18  ~ "child",
  TRUE      ~ "adult"
)
```

```
## [1] "invalid" "child"   "child"   "adult"
```

`case_when` vs `if_else`

You can write the same thing using both. Which do you prefer?

`if_else`:

```
if_else(age < 0, "invald", "other")
```

```
## [1] "invald" "other"  "other"  "other"
```

```
if_else(
  age < 0, "invald",
  if_else(age < 18, "child", "other"))
```

```
## [1] "invald" "child"  "child"  "other"
```

`case_when`:

```
case_when(
  age < 0   ~ "invalid",
  age < 18  ~ "child",
  TRUE      ~ "adult"
)
```

```
## [1] "invalid" "child"   "child"   "adult"
```

`case_when` in a data frame

```
people <- tribble(
  ~name, ~age,
  "Allen Linford", -1,
  "Seb Dodds", 0,
  "Charleen Lockwood", 17,
  "Ridley Burgin", 18,
)
people %>% mutate(
  adult = case_when(
    age < 0   ~ "invalid",
    age < 18  ~ "child",
    TRUE      ~ "adult"
  )
)
```

```
## # A tibble: 4 x 3
##   name                age adult
##   <chr>             <dbl> <chr>
## 1 Allen Linford        -1 invalid
## 2 Seb Dodds             0 child
## 3 Charleen Lockwood    17 child
## 4 Ridley Burgin        18 adult
```

The recoding pattern

```
population <- read_csv("../../data/worldbank_sp_pop_totl.csv")
population %>% mutate(
  country = case_when(
    country == "United States" ~ "USA",
    iso3c == "GBR"             ~ "UK",   # LHS conditions may use different variables
    TRUE                       ~ country # so can RHS
  )
) %>% filter(str_starts(country, "U")) # Just to see the results
```

```
## # A tibble: 7 x 8
##   iso2c iso3c country              date SP.POP.TOTL obs_status footnote
##   <chr> <chr> <chr>               <dbl>       <dbl> <lgl>      <chr>
## 1 AE    ARE   United Arab Emirates  2019     9770529 NA         <NA>
## 2 GB    GBR   UK                    2019    66834405 NA         Extrapolated assuming the same growth r
## 3 UG    UGA   Uganda                2019    44269594 NA         <NA>
## 4 UA    UKR   Ukraine               2019    44385155 NA         Estimated by World Bank staff.
## 5 UY    URY   Uruguay               2019     3461734 NA         <NA>
## 6 US    USA   USA                   2019   328239523 NA         <NA>
## 7 UZ    UZB   Uzbekistan            2019    33580650 NA         Preliminary.
```

More `case_when` tricks

See `?case_when` for how to:

- Deal with inconsistent data types
- Efficiently encode complicated conditionals
- Reuse `case_when` expressions by making a function

and more!