Factors, dates, and strings

EVR 628- Intro to Environmental Data Science

Juan Carlos Villaseñor-Derbez (JC)

Preamble:

Exercise 0 - Set up

Post-it up

1. Download this CSV file to your `EVR 628` project folder. Save it into `data/raw/`.
2. Start a new script, call it `factors_dates_and_strings.R` and save it into your `scripts/01_processing/` folder
3. Add a code outline and load the `tidyverse` and `janitor` packages at the top of your script

Post-it down

```
## SET UP #######################################################################
# Load packages -----------------------------------------------------------------
library(janitor)
library(tidyverse)
```

Exercise 1 - Dates

Post-it up

1. Load your data into an object called `tour_data <-` and clean the column names
2. Inspect your data. How many columns / rows? What are these columns?
3. Create a new object called `tour_data_clean <-` and add a new column called `date` that contains the date. We will build a large pipeline starting here.
4. Remove the `year`, `month`, and `day` columns that we no longer need
5. Place the `date` column all the way to the left

```
# Load data  --------------------------------------------------------------------
tour_data <- read_csv("data/raw/tour_data.csv") |>
```

```r
  clean_names()

dim(tour_data)
colnames(tour_data)
head(tour_data)

## PROCESSING ##################################################################
# Add a date -------------------------------------------------------------------
tour_data_clean <- tour_data |>                  #3) New object
  mutate(date = make_date(year, month, day)) |> # 4) Build my date
  select(date, everything(), -c(year, month, day)) # 5) Remove and organize columns
```

```
[1] 60  6
```

```
[1] "year"       "month"      "day"        "vessel"     "passengers"
[6] "notes"
```

```
# A tibble: 6 x 6
   year month   day vessel passengers notes
  <dbl> <dbl> <dbl> <chr>       <dbl> <chr>
1  2025     1     1 Condor         21 8 Sea lions; 5 Whales; 4 Whale shark
2  2025     1     7 Falcon          6 5 whale; 7 Sea lions; 5 Sea turtle; 4 sea~
3  2025     1    13 Falcon          8 5 Dolphins
4  2025     1    19 Condor          6 4 Sea turtle; 4 sea turtle
5  2025     1    25 Condor         29 2 sea turtle; 8 whale
6  2025     2     1 Falcon         19 3 Whale shark; 7 whale
```

Post-it down

Exercise 2 - Strings

Part 1 - Data contained in a column with strings of text

Post-it up

1. As demonstrated in class, use `separate_longer_delim()` to create one row per species mentioned in the data
2. Standardize the data so all strings appear in lowercase
3. Make sure they are all singular

```r
## PROCESSING ##################################################################
# Add a date -------------------------------------------------------------------
tour_data_clean <- tour_data |>
  mutate(date = make_date(year, month, day)) |>
  select(date, everything(), -c(year, month, day)) |>
  separate_longer_delim(cols = notes, delim = ";") |>  # 1) Separate column into new rows
  mutate(notes = str_to_lower(notes),                  # 2) All strings as lower case
```

```
         notes = str_remove(notes, "s$"))
```

Post-it down

Part 2 - Data STILL contained in a column with strings of text!

Post-it up

1. Look at the documentation for `str_extract()`
2. Run the first two examples at the bottom directly in your console. What seems to be going on?

pause to discuss

3. Using the `str_extract()` function, build a new column called `fauna` that contains the species observed[1]
4. Print your data to the console. Does `fauna` look right?
5. Using the `str_extract()` function, build a new column called `n` that contains the number of organisms observed (make sure it's numeric)
6. Remove the `notes` column

```
## PROCESSING ###############################################################
# Add a date ----------------------------------------------------------------
tour_data_clean <- tour_data |>
  mutate(date = make_date(year, month, day)) |>
  select(date, everything(), -c(year, month, day)) |>
  separate_longer_delim(cols = notes, delim = ";") |>
  mutate(notes = str_to_lower(notes),
         notes = str_remove(notes, "s$"),
         fauna = str_extract(string = notes, pattern = "[a-z ]+$"), # 3) Extract fauna
         fauna = str_squish(fauna),                                 # 4) Remove white space
         n = str_extract(string = notes, pattern = "[:digit:]+"),   # 5) Extract n
         n = as.numeric(n)) |>                                            # and make sure it
  select(-notes)                                                      # 6) Remove notes
```

Post-it down

Exercise 3 - Visualization (a bit of everything)

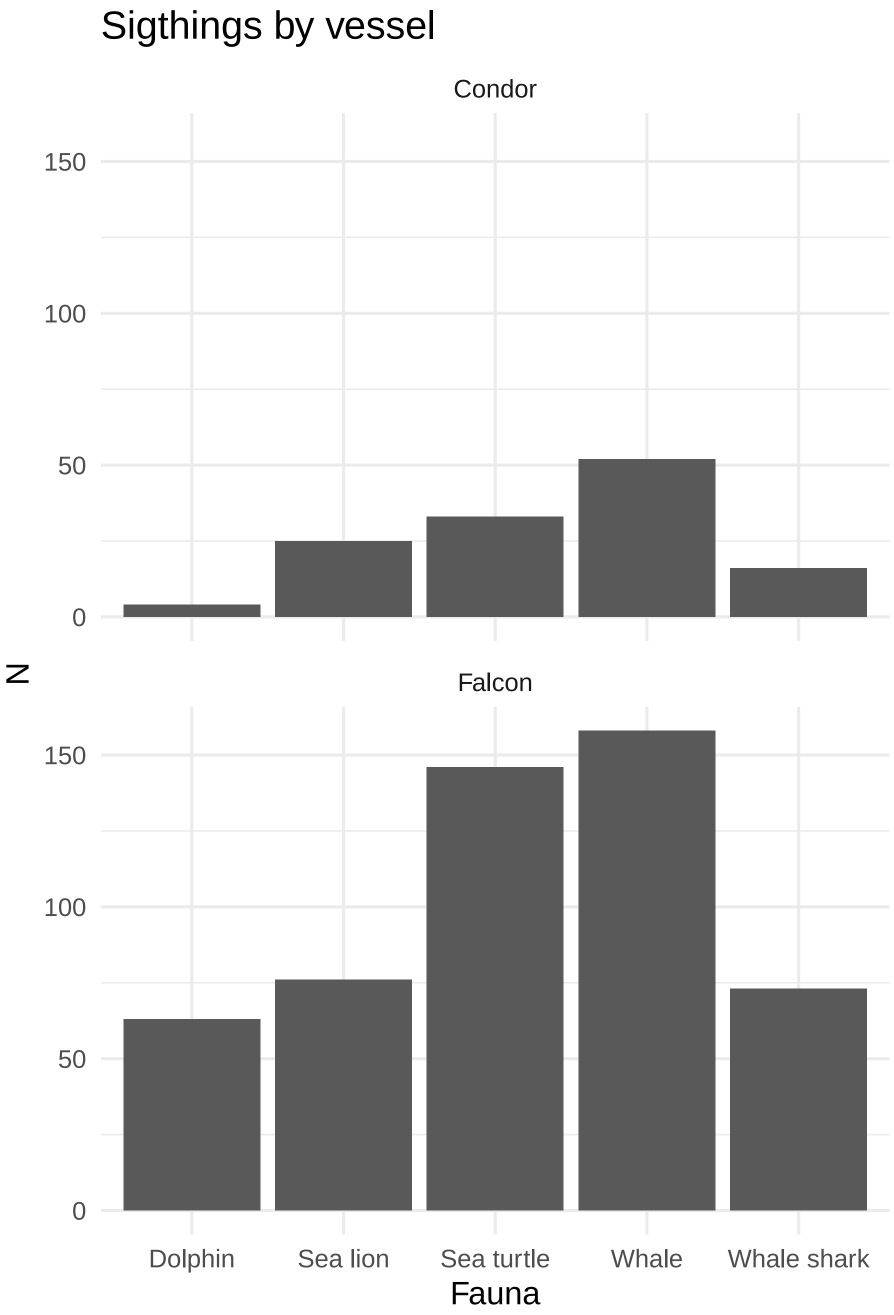
Post-it up

1. Recreate the plot below

Post-it down

[1] Hint: Look at the documentation for `?regex`

```
## VISUALIZE ###################################################################
ggplot(data = tour_data_clean,
       mapping = aes(x = str_to_sentence(fauna), # This is the only str-relevant change here
                     y = n)) +
  stat_summary(geom = "col", fun = "sum") +
  facet_wrap(~vessel, ncol = 1) +
  labs(title = "Sigthings by vessel",
       x = "Fauna",
       y = "N") +
  theme_minimal(base_size = 14)
```


Sigthings by vessel
Condor
Falcon
N
150
100
50
0
Dolphin
Sea lion
Sea turtle
Whale
Whale shark
Fauna