Data tidying and Merging

Juan Carlos Villaseñor-Derbez (JC)

Last week you were asked:

> How much money have tuna purse seiners made since 2000 when fishing for bigeye tuna (*Thunnus obesus*) in the Eastern Pacific Ocean?

We made some simplifying assumptions and got some values (a total of 3,070 M USD since 2000, or about 127 per year). You are now tasked with coming up with more refined estimates. For example, we will account for the fact that the price of fish varies every year.

How we will approach this:

- Find data that shows prices per year and species
- Read them, clean them, tidy them up
- Combine our catch data from last week with this new price data
- Re-calculate our total revenues since 2000

Exersice 1: Tidying price data

Part A: Downloading the data

Post it up

1. In a web browser, go to ffa.int. This is the website for the Pacific Islands Forum Fisheries Agency
2. Hover over "Publication and Statistics" on the top menu
3. Select "Statistics"
4. You will be taken to a site with five items. Download the zip folder called `Economic and Development Indicators and Statistics: Tuna Fishery of the Western and Central Pacific Ocean 2024`
5. As before, place the downloaded zip file in your `EVR628/data/raw` folder and proceed to extract it
6. Open the excel file called `Compendium of Economic and Development Statistics 2024` and study the `Contents` tab
7. Can you identify the price data that we need?

- Which sheet
- What range?

Post it down

Part B: Reading excel data

Post it up

1. Open your RStudio project for EVR628
2. In your console, install the `readxl` package: `install.packages("readxl")`
3. Start a **new** script called `tuna_analysis_prices.R`[1]
4. Add the usual code commenting outline
5. We will need two packages: `readxl`, `janitor`, and `tidyverse`, load them at the top of your script using `library()`
6. Look at the documentation for `read_excel()`
7. Use `read_excel()` to create a new object called `tuna_prices` and read the price data we need. Immediately pipe it into `clean_names`

Post it down

```r
library(readxl)
library(janitor)
library(tidyverse)

tuna_prices <- read_excel(path = "data/raw/Economic-and-Development-Indicators-and-Statistic
                          sheet = "B. Prices",
                          range = "A35:E63",
                          na = "na") |>
  clean_names()
```

Part C: Inspecting price data

Post it up

- What are our column names?
- How many columns do we have?
- Any missing values?
- Do we need to make the data wider or longer?
- Using comments, write out what the target data should be (expand my code chunk see what I wrote)

Post it down

```r
# The final data set should have two columns: year and price. Since we have four
# prices (two markets, two presentations), I will use the average price per year.
# The tidy data set should therefore have four columns: year, market,
# presentation, and price.
```

[1] I would typically suggest to overwrite whatever we had last week in `tuna_analysis.R` because GitHub would keep a version, but I understand you might want to keep the script as is

Part D: Tidy your price data

Post it up

1. Look at the documentation for your `pivot_*` function. What does it say about cases where `names_to` is of length > 1?
2. What about the `names_sep` argument?
3. Use the appropriate `pivot_*` function to reshape your data and save them to a new object called `tidy_tuna_prices`[2]
4. Your resulting data.frame should have 104 rows and 4 columns and look like this:[3]

```
# A tibble: 104 x 4
    year market presentation price
   <dbl> <chr>  <chr>        <dbl>
 1  1997 japan  fresha       8204.
 2  1997 japan  frozenb      8169.
 3  1998 japan  fresha       7703.
 4  1998 japan  frozenb      6320.
 5  1999 japan  fresha       8809.
 6  1999 japan  frozenb      9093.
 7  2000 japan  fresha       9198.
 8  2000 japan  frozenb      8557.
 9  2001 japan  fresha       8260.
10  2001 japan  frozenb      5983.
# i 94 more rows
```

Post it down

! Values in `presentation`

Note that the values in the `presentation` column are not ideal. They end in `a`, `b`, `c`, and `d` due to footnotes included in Excel. For now this doesn't matter because we will quickly remove them. We'll cover some text wrangling in Week 9.

Part E: Calculate mean annual price

Post it up

1. Modify the pipeline that creates `tidy_tuna_prices` to get the mean price per year^[Hint: You will use `group_by()` and `summarize()`, as well as `|>`

```
# A tibble: 28 x 2
    year price
```

[2] Hint: Your `names_to` argument should be a character vector of with two items. `names__sep` should be inspired by our clever use of `snake_case`.

[3] Hint: If you have 112 rows, remember you can use `values_drop_na = T`

```
   <dbl> <dbl>
 1  1997 8186.
 2  1998 7011.
 3  1999 8951.
 4  2000 8877.
 5  2001 5633.
 6  2002 5342.
 7  2003 5285.
 8  2004 5739.
 9  2005 5554.
10  2006 5177.
# i 18 more rows
```

Post it down

Part F: Read the tuna catch data

Post it up

1. Read in the tuna catch data from last week
2. Filter it to retain bigeye tuna caught by the purse seine fleet since 2000
3. Calculate total catch by year

Post it down

> Note: You can copy and paste your code from last week of from below, but make sure your code is organized. Your final data should have 24 rows and 2 columns, as below.

```
# Load the data
tuna_data <- read_csv("data/raw/CatchByFlagGear/CatchByFlagGear1918-2023.csv") |>
  # Clean column names
  clean_names() |>
  # Rename some columns
  rename(year = ano_year,
         flag = bandera_flag,
         gear = arte_gear,
         species = especies_species,
         catch = t)

ps_tuna_data <- tuna_data |>
  filter(species == "BET", # Retain BET values only
         gear == "PS",     # Retain PS values only
         year >= 2000) |>  # Retain data from 2000
  group_by(year) |>        # Specify that I am grouping by year
  # Tell summarize that I want to collapse the catch column by summing all its values
  summarize(catch = sum(catch))
```

```r
ps_tuna_data
```

```
# A tibble: 24 x 2
    year catch
   <dbl> <dbl>
 1  2000 95283
 2  2001 60518
 3  2002 57422
 4  2003 53051
 5  2004 65471
 6  2005 67895
 7  2006 83837
 8  2007 63451
 9  2008 75028
10  2009 76800
# i 14 more rows
```

Part G: Combine your catch and price data

1. Think about what type of join you want
2. What will be on the left and what will be on the right?
3. What is the key?

Post it up

4. Perform the join and save the output to an object called `tuna_revenues`
5. Create a new column that contains the revenue in M USD. Be careful with the units.

Post it down

```
# A tibble: 24 x 4
    year catch price revenue
   <dbl> <dbl> <dbl>   <dbl>
 1  2000 95283 8877.    846.
 2  2001 60518 5633.    341.
 3  2002 57422 5342.    307.
 4  2003 53051 5285.    280.
 5  2004 65471 5739.    376.
 6  2005 67895 5554.    377.
 7  2006 83837 5177.    434.
 8  2007 63451 5054.    321.
 9  2008 75028 5636.    423.
10  2009 76800 6175.    474.
# i 14 more rows
```

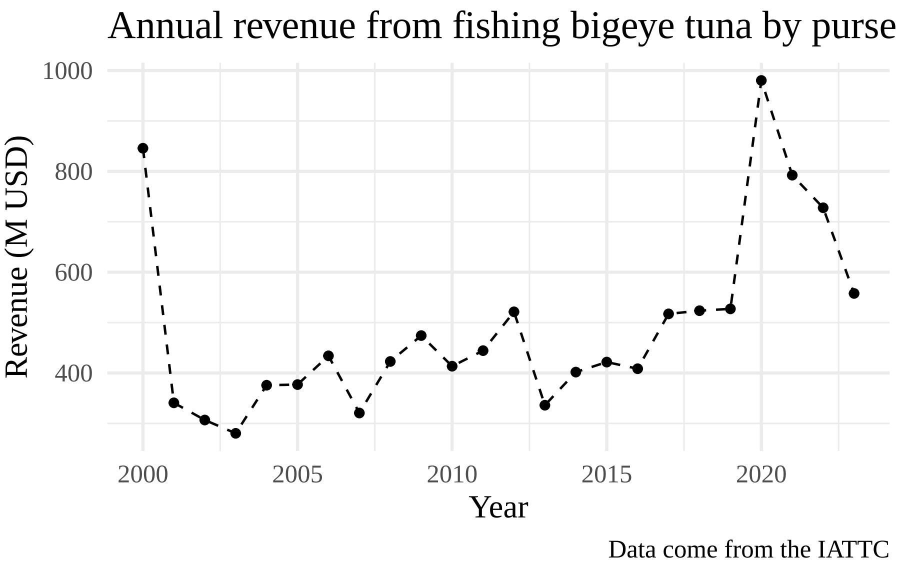
```r
sum(tuna_revenues$revenue)
```

```
[1] 11752.32
```

```
mean(tuna_revenues$revenue)
```

```
[1] 489.68
```

```
# Build  plot
ggplot(data = tuna_revenues,                   # Specify my data
       mapping = aes(x = year, y = revenue)) + # And my aesthetics
  geom_line(linetype = "dashed") +             # Add a dashed line
  geom_point() +                               # With points on top
  labs(x = "Year",                             # Add some labels
       y = "Revenue (M USD)",
       title = "Annual revenue from fishing bigeye tuna by purse seine vessels",
       caption = "Data come from the IATTC") +
  # Modify the theme
  theme_minimal(base_size = 14,               # Font size 14
                base_family = "Times")        # Font family Times
```



How do this plot and numbers compare to what we found last week?