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 M USD 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 (The "data tidying" part)
- Combine our catch data from last week with this new price data (The "merging" part)
- Re-calculate our total revenues since 2000

This will require three pipelines: - Tidy price data (Exercise 1) - Wrangle catch data (Exercise 2) - Combine tidy prices and catch data (Exercise 3)

Pipelines 1 and 3 contain tools covered this week. You should already be familiar with pipeline 2.

Exercise 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 three packages: `readxl`, `janitor`, and `tidyverse`, load them at the top of your script using `library()`
6. Use `?read_excel()` to look at the documentation for the function
7. Use `read_excel()` to create a new object called `tuna_prices` and read the price data we need[2]. Immediately pipe it into `clean_names`.

Post-it down

Part C: Inspecting price data

Be prepared to discuss the following points:

Post-it up

1. Inspect the column names of `tuna_prices` using `colnames()` in your console.
2. How many columns and rows do we have?
3. Any missing values?
4. Do we need to make the data wider or longer?
5. Using comments, write out what the target data should be (expand my code chunk see what I wrote)

Post-it down

See an example of my description below

```
[1] "year"          "japan_fresha"  "japan_frozenb" "us_freshc"    
[5] "us_frozend"   
```

```
[1] 28  5
```

[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

[2] Hint: You will need to specify a file path, a sheet, and a range of cells.

```
# A tibble: 4 x 5
   year japan_fresha japan_frozenb us_freshc us_frozend
  <dbl>        <dbl>         <dbl>     <dbl>      <dbl>
1  1997        8204.         8169.        NA         NA
2  1998        7703.         6320.        NA         NA
3  1999        8809.         9093.        NA         NA
4  2000        9198.         8557.        NA         NA
```

```
# 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.
```

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`[3]
4. Your resulting tibble should have 104 rows and 4 columns and look like this:[4]

```
# 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

[3] 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`.

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

❗ 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[5]

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

Exercise 2: Tidying tuna catch data (again)

Part A: Read the tuna catch data

> Note: You can copy-paste and modify your code from last week, but make sure your code is organized.

Post-it up

1. Read in the tuna catch data from last week
2. Filter it to retain bigeye tuna (`BET`) caught by the purse seine fleet (`PS`) since 2000
3. Calculate **total** catch by year. Your final data should have 24 rows and 2 columns, as below

[5] Hint: You will use `group_by()` and `summarize()`, as well as `|>`

Post-it down

```
# 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
```

Exercise 3: Combine your catch and price data

Part A: Plan the join

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?
4. Write down, using human language, what you want to do.

Post-it up

Part B: Perform the join

1. Perform the join and save the output to an object called `tuna_revenues`
2. Create a new column that contains the annual revenue in M USD. Pay attention to 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
```

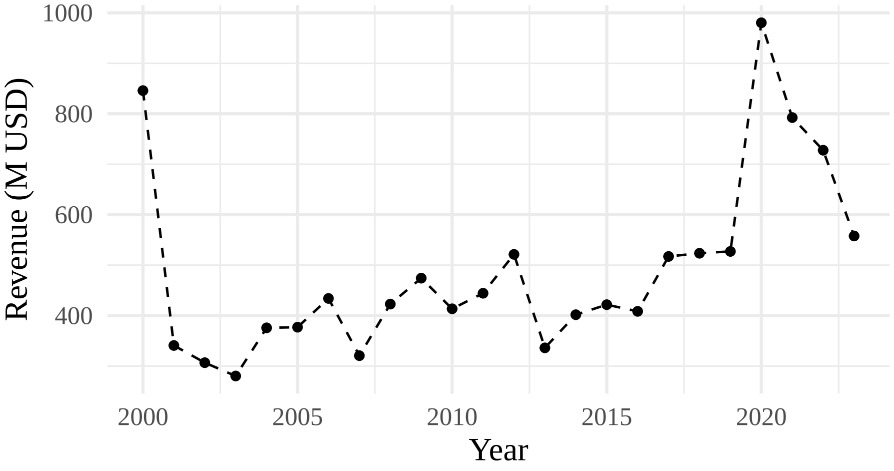
Part C: Answer the questions again

1. How much TOTAL revenue since 2000?
2. How much mean ANNUAL revenue since 2000?
3. Make a figure
4. How do these plot and numbers compare to what we found last week?

```
[1] 11752.32
```

```
[1] 489.68
```

Annual revenue from fishing bigeye tuna by purse



Data come from the IATTC