Diary Entry

Loy Yee Keen

2023-11-10

Week 9

(1) What is the topic that you have finalised? (Answer in 1 or 2 sentences).

The topic that I have finalised is "Names", specifically "Is there a rise in the gender-neutral names given to babies born in America?"

(2) What are the data sources that you have curated so far? (Answer 1 or 2 sentences).

I have curated a dataset consisting of a list of names given to babies born in the US each year, the gender of the babies, and the count of each name per gender. The data sources span from the years 1880 to 2022. These data sources are extracted from the website of the United States Social Security Administration, an independent agency of the U.S. federal government.

Week 10

(1) What is the question that you are going to answer? (Answer: One sentence that ends with a question mark that could act like the title of your data story),

Is there a rise in the gender-neutral names given to babies born in America?

(2) Why is this an important question? (Answer: 3 sentences, each of which has some evidence, e.g., "According to the United Nations. . . " to justify why the question you have chosen is important)

The Council of Europe underscores the role of gender in shaping power dynamics and opportunities in society. The popularity of gender-neutral names reflects a broader shift toward inclusivity and the challenge of traditional gender roles.

Source: https://www.coe.int/en/web/gender-matters/exploring-gender-and-gender-identity#:~:text=Gender%20is%20of%20key%20importance,equality%20and%20freedom%20from%20discrimination

Additonally, gender-neutral names empower girls and women by challenging gender stereotypes. According to a New York Times article, some parents opt for these names to counter biases and promote strength for their daughters.

Source: https://nypost.com/2018/03/21/why-gender-neutral-baby-names-are-on-the-rise/

This trend aligns with the United Nations' Sustainable Development Goal 5, which aims to achieve gender equality and empower women and girls.

Source: https:// sdgs.un.org/ goals/ goal5

(3) Which rows and columns of the dataset will be used to answer this question? (Answer: Actual names of the variables in the dataset that you plan to use).

I will use multiple datasets to answer the chosen question. All the datasets have the same format; each dataset represents a specific year spanning from 1882 to 2022.

In each dataset, there are 3 columns, corresponding to the name of the baby, the sex of the baby and the count of babies with that name (The original dataset did not define the names of the variables, so I will redefine the variables as Name, Sex and Count).

The number of rows, each corresponding to the observation for each name, differ for every year.

Every row and column of the datasets will be used to answer my chosen question as all of them are relevant in comparing the shifts in naming trends over the years.

I will use rbind to combine the datasets into a single dataset.

(4) Challenges and errors that you faced and how you overcame them.

I encountered difficulties when I read the files using read_csv because the datasets did not have column names. Consequently, the output assigned the first value in each cell of the respective columns as the column names. This approach was erroneous, as the data in the first row represented observations, not variables. To resolve this issue, I tried to look up the answer in the textbook reading (https://r4ds.hadley.nz/data-import) provided in the Lecture 9 slides.

Week 11

(1) List the visualisations that you are going to use in your project (Answer: What are the variables that you are going to plot? How will it answer your larger question?)

- I. Barplot of proportion of US babies with gender-neutral names by year:
  - Variables: y axis = Proportion of gender-neutral names; x-axis = Year.
  - Purpose: Investigate the trend over time, determining whether there is an increase in the use of gender-neutral names for babies in the US.

II. Table of gender-neutral names in 2022:

- Columns: Gender-neutral names in 2022, count of male babies, count of female babies, and the proportion overlap between both sexes.
- Purpose: Provides a more detailed visualisation of the specific names of which the trends would be plotted in III.

III. 5 line plots of the trend of gender-neutral name over time, grouped by gender:

- Variables: x-axis: Proportion of babies with names that have the least male-female proportion overlap, y-axis: Year
- Purpose: Analyse the trend of the top 5 names with the least male-female proportion overlap, understanding whether these names have been consistently gender-neutral or have transitioned over time. This could possibly answer the question of why there is a rise/fall in gender-neutral names over time.

(2) How do you plan to make it interactive? (Answer: features of ggplot2/shiny/markdown do you plan to use to make the story interactive)

- I. Features:
  - A slider for adjusting the number of bars in the barplot.
  - Radio buttons displaying the proportion of male and female babies with gender-neutral names.

II. Features:

- A button to highlight names in the table that are more popular among male babies.
- SelectInput function to choose the number of rows to display in the table.

III. Features:

- SelectInput function to choose a name and display the corresponding plot.
- Forward (and backward) navigation buttons to show the plot for the next 50 years.
- A card providing an explanation of the plot, updating itself when the navigation buttons are pressed. ### (3) What concepts incorporated in your project were taught in the course and which ones were self-learnt? (Answer: Create a table with topics in one column and Weeks in the other to indicate which concept taught in which week is being used. Leave the entry of the Week column empty for self-learnt concepts)

```
## Warning: package 'readxl' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## # A tibble: 52 x 2
##    Topic                          Week
##    <chr>                          <chr>
##  1 library                        2
##  2 pull                           2
##  3 nrow                           2
##  4 logical operators (==, &, <, >) 2 & 4
##  5 ggplot                         2 & 7
##  6 as.integer                     3
##  7 as.character                   3
##  8 vector('list', length=)        3
##  9 list[['']]                     3
## 10 c()                            3
## 11 read_csv                       3
```

```
## 12 $                              3
## 13 filter                          4
## 14 select                          4
## 15 mutate                          4
## 16 arrange(desc())                 4
## 17 seq(from=, to=, by=)            4
## 18 slice                           4
## 19 : eg 1:5                        4
## 20 distinct                        4
## 21 ifelse                          4
## 22 functions                       5
## 23 paste0                          5
## 24 for loop                        6
## 25 rep                             6
## 26 aes(group=)                     7
## 27 aes(colour=)                    7
## 28 geom_col(fill=)                 7
## 29 geom_col(alpha=)                7
## 30 guides()                        7
## 31 facet_wrap(~)                   8
## 32 shiny                           8
## 33 unlist                          <NA>
## 34 lapply()                        <NA>
## 35 abs                             <NA>
## 36 scales = 'free_y'               <NA>
## 37 geom_line                       <NA>
## 38 read_csv(col_names=)            <NA>
## 39 merge                           <NA>
## 40 data.frame                      <NA>
## 41 rbind                           <NA>
## 42 do.call                         <NA>
## 43 css                             <NA>
## 44 is.null                         <NA>
## 45 scale_x_continuous              <NA>
## 46 reactiveVal                     <NA>
## 47 observeEvent                    <NA>
## 48 DT package                      <NA>
## 49 scale_colour_manual             <NA>
## 50 backticks                       <NA>
## 51 guide_legend(title = NULL)      <NA>
## 52 geom_col                        <NA>
```

Explanations for some of the functions I used are as follows:

as.integer(): convert count from type double to type integer

as.character(): convert year from type integer to type character

filter(): filter sex == "M" from the orignal dataset and store in a new variable, filter sex = "F" from the orignal dataset and store in a new variable, filter names that occur more than 500 times for each sex, filter names occurring less than 50 times in total

mutate(): create a new column in the dataset called "proportion overlap"

arrange(desc()): sort names in descending order of "proportion overlap"

seq(from=1882, to=2022, by=10): set the breaks of the bar plot to be in intervals of 10 from 1882 to 2022

distinct(): to reference unique names in the dataframe so that each name does not occur more than once

ifelse: highlight the names in the shiny output table if there are more male babies than female babies with that name plot the lines for the next 50 years and update the text if the forward button is pressed print the text corresponding to a particular name if that name is selected in the selectInput function on shiny

functions function to filter male and female names and store in separate variables function to merge the two variables function to read each dataset (see paste0 function below)

paste0(): concatenate the strings: "yob", year & ".txt" as they have no separators read_year_data <- function(year) { data <- read_csv(paste0("yob", year, ".txt"), col_names = c("Name", "Sex", "Count")) }

for loop: execute functions across multiple names and multiple years from 1882 to 2022

ggplot(data)+aes(group=sex, color=sex)+geom_line(): plot a line plot of proportion of babies with a particular gender-neutral name, with 2 separate lines of different colours for male and female babies.

facet_wrap(~ Name, scales = "free_y"): plot each name in a separate plot and show all the plots at the same time and allow the scales of the y-axis to vary between facets based on the data for each 'Name' otherwise some of the y-axes will be so large that the plot cannot be seen clearly

merge(): merge the 2 datasets, male and female names, into one dataset called gender-neutral names

scale_x_continuous(): sets the breaks for the x-axis. Since each bar corresponds to a year from 1882 to 2022, there are too many bars hence the x-axis looks very cramped

reactiveVal(): store the selected name in selectInput

observeEvent(): observe changes in this stored value (the selected name), and updates the plot/text whenever the user selects a new name or presses a button

backticks: to reference column names that include spaces eg `Proportion Overlap`

geom_col: I have two bar plots. One of the proportion of gender-neutral names for each year and another of the proportion of names occurring less than 50 times for each year. To get the heights of the bars to represent values in the data (proportion), I used geom_col() since the default behaviour of geom_bar() counts the number of cases in each bar instead.

geom_col(fill=): fill the bars with different colors based on the male and female proportions

geom_col(alpha=): to make the plot translucent so the overlapping colours for male and female proportions can be compared more clearly

(4) Include the challenges and errors that you faced and how you overcame them.

The first error I faced was when I used a "for loop" that loops through each dataframe for the years 1882 to 2022, and merges the names that are given to both male and female babies. However, when I tried to access the output outside the loop, I was returned "Error: object 'gender_neutral_names' not found". To overcome this error, I went back to lecture 6 about "for loops" and recalled that we had to pre-allocate space to store the output. I then corrected the code by adding "gender_neutral_names <- vector("list", length =1882:2022)" before the loop.

The next error I faced was when I tried to access a year from the gender_neutral_names list using gender_neutral_names("2022"), but was returned with Error in gender_neutral_names("2022") : could not find function "gender_neutral_names". I then referred to lecture 3 and realised that we need to access columns in a list using gender_neutral_names[["2022"]] instead.

Another error I faced was when I tried to arrange the data in a column named "Proportion Overlap". However, when I printed the output, the data was not arranged. To overcome this challenge, I went online to search for how to reference to column names that include spaces and found out that we need to use backticks around the column name.

I also faced the challenge of not being able to plot a barplot of proportion against year. When I tried to plot a barplot using ggplot(data) + aes(x=year,y=proportion) + geom_bar(), I was returned "Error: stat_count() must not be used with a y aesthetic". I tried to solve this error and typed ?geom_bar into my R console. But I was returned "No documentation for 'geom_bar' in specified packages and libraries: you could try '??geom_bar' ". So I typed ??geom_bar instead and learnt that "If you want the heights of the bars to represent values in the data, use geom_col() instead".

Week 12

Challenges

For my first visualisation, I tried to plot a barplot of the proportion of gender-neutral names given to babies per year using

ggplot(gender_neutral_names) + aes(x = year, y = proportion) + geom_bar()

However, I was returned with "Error occurred in the 1st layer. Caused by error in `setup_params()`:! `stat_count()` must only have an x or y aesthetic".

To overcome this error and plot the barplot successfully, I copied this error into google. The first result came from stackoverflow which said that I need to either use geom_col() or geom_bar(stat="identity"). This is because the default for geom_bar() is (stat=count) which counts the aggregate number of rows for each x value (year), but I am instead providing the y-values (proportion) for the barplot.

I also included radio buttons (none/male/female/both) that fill this gender-neutral barplot according to the proportion of male/female/both that are given gender-neutral names.

I initially tried to do this by binding these 3 dataframes together using rbind, and then using

if(input$proportion_per_sex = female)plot <- -ggplot(total_proportion) + aes(x = year, y = proportion, fill = female_proportio
{ plot<-ggplot(total_proportion)+ aes(x=year, y=proportion, fill=male_proportion) } else {plot<-
total_proportion) +aes(x=year, y=proportion) }

but then this doesnt work since female_proportion and male_proportion are separate columns in the combined dataframe and fill works by colouring binary factor variables within the same column
Instead, I googled if it was possible to layer multiple ggplots in the same plot using eg

ggplot() + geom_col(data = total_proportion) + aes(x = year, y = proportion) + geom_col(data = male_proportion) + aes(x = year, y = proportion) + geom_col(data = female_proportion) + aes(x = year, y = proportion)

Apparently it was possible, so I used this method instead.

For my second visualisation, I had 2 navigation buttons in the side panel that display the line plot when pressed. For example, the original plot (without pressing any buttons) displays an empty plot with x-axis from 1882 to 2022. When the forward button is pressed once, it displays the line from 1882 to 1932 (50 years), then when it is pressed again, it displays from 1882 to 1982 (next 50 years). To accumulate the years, I tried to use «- which I learnt during Week 5's tutorial. My original code was interval_end «- interval_end + 50. However, the plot is empty when I rendered it.

For this visualisation to work effectively, I created two functions, one to store the interval end, and one to store the cumulative end (interval_end <- reactiveVal(1882) and cumulative_end <- reactiveVal(1882)), and used yet another function new_cumulative_end <- cumulative_end() + 50 to accumulate the years instead.

I also faced another challenge for this plot. Even if a name only has records starting from a specific year that is not 1882 (for instance, 1960), the plot does not start the line from 1960 but instead extends the line back 1882, which would otherwise erroneously imply the name's existence before it was actually established. Also, the colour aesthetic for the plot gets mixed up. For instance, if a name only has records for male babies in the 1960s, the plot displays a pink line to represent this trend. However, when there are records for both

sexes in the later years, say 1980, the plot displays 2 lines, but now the male line is blue while the female line is pink. Hence, I included a code chunk to set the proportion to 0 when the year is 1882.