Assignment 1

Pinyapat Manasboonpermpool

07 October 2021

1 Task 1

Introduction

In Task 1, we begin to analyse the data set *starwars*, which contains characteristics of many of the characters in the Star Wars universe.

First, the package *tidyverse* should be installed and loaded to be able to access to the data set *starwars*.

To view the information of the data set, the head() function is coded to present all the variables and observations. There are 14 variables in total which are classified as *name, height, mass, hair_color, skin_color, eye_color, birth_year, sex, gender, homeworld, species, films, vehicles*, and *starships*. Each variable of column consists of its observation in row, consisting the total of 87 observations.

```
library(tidyverse)
head(starwars)
```

```
## # A tibble: 6 x 14
##   name  height  mass hair_color skin_color eye_color birth_year
##   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl>
## 1 Luke~    172    77 blond      fair       blue            19
## 2 C-3PO    167    75 <NA>       gold       yellow         112
## 3 R2-D2     96    32 <NA>       white, bl~ red             33
## 4 Dart~    202   136 none       white      yellow          41.9
## 5 Leia~    150    49 brown      light      brown           19
## 6 Owen~    178   120 brown, gr~ light      blue            52
## # ... with 7 more variables: sex <chr>, gender <chr>,
## #   homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

The main focus of Task 1 is to summarize the average mean of weighted mass which is grouped by the eye colors specifically to the category of home world - *Tatooine*.

To begin with, we use the filter() function to specify the category of home world - *Tatooine*. Then, as the expected outcome of the average mean we which to compute is related to the mass data, those *missing* values of observations should be included by using the function drop_na(). Followed by the function group(), this is used to take a consideration of the data set of variable *eye_color* to compute the average mean of weighted mass. Lastly, the average computation is named as *avg_mass* and this is added as a variable by using of summarize() function.

```
# Pull some data from the starwars data set
starwars_tatooine_summary <-
  starwars %>%
  filter(homeworld == "Tatooine") %>%
  drop_na(mass) %>%
  group_by(eye_color) %>%
  summarize(avg_mass = mean(mass))
```

The result of the average weighted mass can be demonstrated by the visualization of the bar charts.

According to the data, the bar charts are laid out four distinctive types of eye colors on the X-axis and the average weight on the Y-axis for the case of home world *Tatooine*.

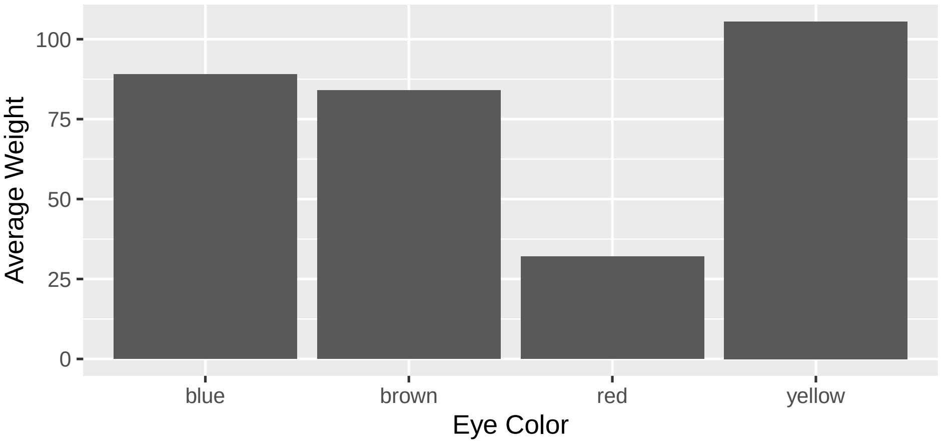

Figure 1: The average mass of eye colors in the homeworld of Tatooine

Conclusion

The charts indicate the result of the yellow eye color with the highest average weight of above 100 (kg) in *Tatooine*.Following with the other types, the blue color group is with the second higested weight of above 90 (kg), the brown color group is with the weight of below 90 (kg), and lastly the red color group is with the lowest average weight of above 25 (kg), respectively. (Figure 1).

2 Task 2

Introduction

In Task 2, the data set *table4a* which contains the information about the tuberculosis (TB) cases in three countries, *Afghanistan, Brazil, China*, during the two period of years in 1999 and 2000.

First, we view *table4a* data set and notice that there are three rows and three columns which the information of both the variables and observations has an untidy look. This data table is inapplicable to be used for an analysis.

To elaborate, apart from the country column, the other two columns on the right consist of values in numbers which can be assumed as cases and years located without the indication of columns names of the variables. The reason to explain this untidy table is that the data table have some *missing* variables names in the columns, as well as, the values of observations are not put in rows as expected. Therefore, we should adjust the data table accordingly before taking any further steps.

Based on the observations of data in the table, the *missing* variables to be set in the table are *year* and *cases*.

```
library(tidyverse)
table4a
```

```
## # A tibble: 3 x 3
##   country     ‘1999‘ ‘2000‘
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

Conclusion

To adjust the data of *table4a*, we need to create the columns for those missing variables by using the function of pivot_longer(). This way, we create the new column for the years and the new column of the number of tuberculosis cases in 1999 and 2000. This tidy data set is called *table4a_tidy*.

```
# tidy up table4a and call it a new variable "table4a_tidy"
table4a_tidy <- table4a %>%
                pivot_longer(c("1999", "2000"),
                names_to = "Year",
                values_to = "Cases") %>%
                rename(Country = country)
```

Table 1: The number of tuberculosis cases outbreak in 1999 and 2000

Country	Year	Cases
Afghanistan	1999	745
Afghanistan	2000	2,666
Brazil	1999	37,737
Brazil	2000	80,488
China	1999	212,258
China	2000	213,766

3 Task 3

Introduction

In Task 3, when viewing the data set *table4b* which contains the information about the population in three countries, *Afghanistan, Brazil, China*, during the two period of years in 1999 and 2000, we notice that it has the same untidy look of data table similar to *table4a*. We can then repeat the same steps as in the *table4a_tidy* in order to create the columns for new variables of year and population.

```
library(tidyverse)
table4b
```

```
## # A tibble: 3 x 3
##   country         ‘1999‘     ‘2000‘
## * <chr>            <int>      <int>
## 1 Afghanistan   19987071   20595360
## 2 Brazil       172006362  174504898
## 3 China       1272915272 1280428583
```

To adjust the data of *table4b*, we need to create the columns for those missing variables by using the function of pivot_longer(). This way, we create the new column for the years and the new column of the number of population in 1999 and 2000. This tidy data set is called *table4b_tidy*.

```
#tidy up table4b and call it a new variable "table4b_tidy"
  library(dplyr)
table4b_tidy <- table4b %>%
                pivot_longer(
                c("1999", "2000"),
                names_to = "Year",
                values_to = "Population") %>%
                rename(Country = country)
```

Table 2: The number of population in Afghannistan, Brazil, China in 1999 and 2000

Country	Year	Population
Afghanistan	1999	19,987,071
Afghanistan	2000	20,595,360
Brazil	1999	172,006,362
Brazil	2000	174,504,898

To analyze the data of two tables *table4a* and *table4b*, we merge the two data sets using the left_join() function which allows the two data sets of tables that have the common variables names such as the *country* and *year* columns in this case to combine. As a result, we create a new combined data set between *table4a_tidy* and *table4b_tidy* which allows us to compute for the proportional rate of tuberculosis in population.

```
table4 <- left_join(table4a_tidy, table4b_tidy) %>%
          mutate(Rate = (Cases/Population))
```

Table 3: The data set of cases and population rate in Afghanistan, Brazil, China in 1999 and 2000

Country	Year	Cases	Population	Rate
Afghanistan	1999	745	19,987,071	0.000037
Afghanistan	2000	2,666	20,595,360	0.000129
Brazil	1999	37,737	172,006,362	0.000219
Brazil	2000	80,488	174,504,898	0.000461
China	1999	212,258	1,272,915,272	0.000167
China	2000	213,766	1,280,428,583	0.000167

Conclusion

According to the new data set of table, there were increased rates of tuberculosis cases in populations in *Afghanistan* and *Brazil* in the two periods of 1999 and 2000. However, China experienced no significant changes in the rates of tuberculosis cases in their population in both years of 1999 and 2000.