Data Cleaning of the Food-Choice Preferences of College Students Dataset

This dataset includes information on food choices, nutrition, preferences, childhood favorites, and other information from college students. There are 126 responses from students. Data is raw and uncleaned. The dataset can be accessed here.

Functions used in this project include:

- case match
- case when
- str_subset
- str_replace, and others

Import all necessary libraries

```
library(tidyverse)
library(ggplot2)
library(stringr)
library(rebus)
library(magrittr)
```

Import the dataset and view the first 20 rows

```
food_choices <- read.csv("food_coded.csv", header = T)
```

assess the imported dataframe

```
glimpse(food_choices)
```

```
## Rows: 125
## Columns: 61
## $ GPA                          <chr> "2.4", "3.654", "3.3", "3.2", "3.5", "2.2~
## $ Gender                       <int> 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1,~
## $ breakfast                    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ calories_chicken             <int> 430, 610, 720, 430, 720, 610, 610, 720, 4~
## $ calories_day                 <dbl> NaN, 3, 4, 3, 2, 3, 3, 3, NaN, 3, 3, 4, 3~
## $ calories_scone               <dbl> 315, 420, 420, 420, 420, 980, 420, 420, 4~
## $ coffee                       <int> 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2,~
## $ comfort_food                 <chr> "none", "chocolate, chips, ice cream", "f~
## $ comfort_food_reasons         <chr> "we dont have comfort ", "Stress, bored, ~
## $ comfort_food_reasons_coded   <int> 9, 1, 1, 2, 1, 4, 1, 1, 2, 1, 2, 3, 3, 1,~
## $ cook                         <dbl> 2, 3, 1, 2, 1, 3, 2, 3, 3, 3, 1, 3, 5, 2,~
## $ comfort_food_reasons_coded.1 <int> 9, 1, 1, 2, 1, 4, 1, 1, 2, 1, 2, 3, 3, 1,~
## $ cuisine                      <dbl> NaN, 1, 3, 2, 2, NaN, 1, 1, 1, 1, 1, 1, 1~
## $ diet_current                 <chr> "eat good and exercise", "I eat about thr~
## $ diet_current_coded           <int> 1, 2, 3, 2, 2, 2, 3, 1, 1, 1, 1, 1, 1, 2,~
## $ drink                        <dbl> 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 2, 2,~
## $ eating_changes               <chr> "eat faster ", "I eat out more than usual~
## $ eating_changes_coded         <int> 1, 1, 1, 1, 3, 1, 2, 2, 2, 1, 3, 4, 2, 1,~
## $ eating_changes_coded1        <int> 1, 2, 3, 3, 4, 3, 5, 5, 8, 3, 4, 5, 5, 3,~
## $ eating_out                   <int> 3, 2, 2, 2, 2, 1, 2, 2, 5, 3, 2, 1, 1, 4,~
## $ employment                   <dbl> 3, 2, 3, 3, 2, 3, 3, 2, 2, 3, 1, 2, 3, 2,~
## $ ethnic_food                  <int> 1, 4, 5, 5, 4, 4, 5, 2, 5, 5, 5, 5, 4, 5,~
## $ exercise                     <dbl> 1, 1, 2, 3, 1, 2, 1, 2, NaN, 1, 1, 1, 3, ~
## $ father_education             <dbl> 5, 2, 2, 2, 4, 1, 4, 3, 5, 5, 2, 3, 3, 2,~
## $ father_profession            <chr> "profesor ", "Self employed ", "owns busi~
## $ fav_cuisine                  <chr> "Arabic cuisine", "Italian", "italian", "~
## $ fav_cuisine_coded            <int> 3, 1, 1, 3, 1, 6, 4, 5, 1, 1, 4, 1, 4, 1,~
## $ fav_food                     <dbl> 1, 1, 3, 1, 3, 3, 1, 1, 3, 1, 1, 1, 3, 1,~
## $ food_childhood               <chr> "rice  and chicken ", "chicken and biscui~
## $ fries                        <int> 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ fruit_day                    <int> 5, 4, 5, 4, 4, 2, 4, 5, 4, 5, 5, 5, 4, 5,~
## $ grade_level                  <int> 2, 4, 3, 4, 4, 2, 4, 2, 1, 1, 3, 2, 1, 3,~
## $ greek_food                   <int> 5, 4, 5, 5, 4, 2, 5, 3, 5, 5, 1, 5, 3, 4,~
## $ healthy_feeling              <int> 2, 5, 6, 7, 6, 4, 4, 3, 7, 3, 9, 1, 9, 8,~
## $ healthy_meal                 <chr> "looks not oily ", "Grains, Veggies, (mor~
## $ ideal_diet                   <chr> "being healthy ", "Try to eat 5-6 small m~
## $ ideal_diet_coded             <int> 8, 3, 6, 2, 2, 2, 2, 2, 6, 2, 7, 2, 1, 2,~
## $ income                       <dbl> 5, 4, 6, 6, 6, 1, 4, 5, 5, 4, 3, 5, 5, 5,~
## $ indian_food                  <int> 5, 4, 5, 5, 2, 5, 5, 1, 5, 4, 1, 5, 3, 3,~
## $ italian_food                 <int> 5, 4, 5, 5, 5, 5, 5, 3, 5, 5, 5, 5, 4, 5,~
## $ life_rewarding               <dbl> 1, 1, 7, 2, 1, 4, 8, 3, 8, 3, 8, 1, 9, 10~
## $ marital_status               <dbl> 1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2,~
## $ meals_dinner_friend          <chr> "rice, chicken,  soup", "Pasta, steak, ch~
## $ mother_education             <dbl> 1, 4, 2, 4, 5, 1, 4, 2, 5, 5, 4, 4, 4, 4,~
## $ mother_profession            <chr> "unemployed", "Nurse RN ", "owns business~
## $ nutritional_check            <int> 5, 4, 4, 2, 3, 1, 4, 4, 2, 5, 2, 5, 2, 2,~
## $ on_off_campus                <dbl> 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 2,~
## $ parents_cook                 <int> 1, 1, 1, 1, 1, 2, 2, 1, 2, 3, 1, 1, 2, 2,~
## $ pay_meal_out                 <int> 2, 4, 3, 2, 4, 5, 2, 5, 3, 3, 2, 3, 2, 3,~
## $ persian_food                 <dbl> 5, 4, 5, 5, 2, 5, 5, 1, 5, 4, 2, 5, 3, 3,~
## $ self_perception_weight       <dbl> 3, 3, 6, 5, 4, 5, 4, 3, 4, 3, 1, 2, 5, 3,~
## $ soup                         <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1,~
## $ sports                       <dbl> 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 1,~
## $ thai_food                    <int> 1, 2, 5, 5, 4, 4, 5, 1, 5, 4, 2, 5, 3, 5,~
## $ tortilla_calories            <dbl> 1165, 725, 1165, 725, 940, 940, 940, 725,~
## $ turkey_calories              <int> 345, 690, 500, 690, 500, 345, 690, 500, 3~
## $ type_sports                  <chr> "car racing", "Basketball ", "none", "nan~
## $ veggies_day                  <int> 5, 4, 5, 3, 4, 1, 4, 4, 3, 5, 5, 5, 3, 5,~
## $ vitamins                     <int> 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2,~
## $ waffle_calories              <int> 1315, 900, 900, 1315, 760, 1315, 1315, 13~
## $ weight                       <chr> "187", "155", "I’m not answering this. ",~
```

This dataframe consists of 125 rows and 61 columns.

To check for the number of null values in each column, I’ll use the `ColSums` function, store it in a tibble then rename the columns of the tibble

```r
nulls <- tibble(names(food_choices), colSums(is.na(food_choices)))
names(nulls) <- c('column', 'n')
nulls %>%
  filter(n>0) %>%
  arrange(-n) # arrange in descending order of number of nulls
```

```
## # A tibble: 20 x 2
##    column                         n
##    <chr>                         <dbl>
##  1 calories_day                     19
##  2 comfort_food_reasons_coded       19
##  3 cuisine                          17
##  4 exercise                         13
##  5 employment                        9
##  6 cook                              3
##  7 mother_education                  3
##  8 drink                             2
##  9 fav_food                          2
## 10 sports                            2
## 11 calories_scone                    1
## 12 father_education                  1
## 13 income                            1
## 14 life_rewarding                    1
## 15 marital_status                    1
## 16 on_off_campus                     1
## 17 persian_food                      1
## 18 self_perception_weight            1
## 19 soup                              1
## 20 tortilla_calories                 1
```

From the query above, it is evident that there are 20 columns with at least one null value. To handle null values, there are different approaches, including deleting the rows which contain these null values. This might not be the best choice here considering the relatively small size of the dataset. Instead, I have chosen to replace null values with the modal (mode) value for categorical values and with the mean for other numeric columns.

```r
mode_calories <- names(sort(table(food_choices$calories_day), decreasing = T))[1] #extract the modal va
food_choices$calories_day[is.na(food_choices$calories_day)] <- mode_calories # replace null values with
```

repeat this step for the cuisine column

```r
mode_cuisine <- names(sort(table(food_choices$cuisine), decreasing = T))
food_choices$cuisine[is.na(food_choices$cuisine)] <- mode_cuisine
```

There are other columns in the dataframe containing missing values, however, it won't be efficient to repeat the same code for so many columns. Instead, I've utilized a "for" loop that checks for the modal value in all these columns and replaces the nulls with them.

```r
# food_choices' is your data frame
#

for(col in c('exercise', 'employment', 'cook', 'mother_education')) {
  mode_val <- names(sort(table(food_choices[[col]]), decreasing = TRUE)[1]) # extract the modal value
  food_choices[[col]][is.na(food_choices[[col]])] <- mode_val
}
```

The above step has been repeated here.

```r
for(col in c('drink', 'fav_food', 'sports')) {
  mode_1 <- names(sort(table(food_choices[[col]]), decreasing = T)[1])
  food_choices[[col]][is.na(food_choices[[col]])] <- mode_1
}
```

Now, all null values have been treated and this can once again be confirmed using the `ColSums` function.

```
colSums(is.na(food_choices))
```

```
##                          GPA                       Gender
##                            0                            0
##                    breakfast             calories_chicken
##                            0                            0
##                 calories_day               calories_scone
##                            0                            1
##                       coffee                 comfort_food
##                            0                            0
##         comfort_food_reasons   comfort_food_reasons_coded
##                            0                           19
##                         cook comfort_food_reasons_coded.1
##                            0                            0
##                      cuisine                 diet_current
##                            0                            0
##           diet_current_coded                        drink
##                            0                            0
##               eating_changes         eating_changes_coded
##                            0                            0
##        eating_changes_coded1                   eating_out
##                            0                            0
##                   employment                  ethnic_food
##                            0                            0
##                     exercise             father_education
##                            0                            1
##            father_profession                  fav_cuisine
##                            0                            0
##            fav_cuisine_coded                     fav_food
##                            0                            0
##               food_childhood                        fries
##                            0                            0
##                    fruit_day                  grade_level
##                            0                            0
##                   greek_food              healthy_feeling
##                            0                            0
##                 healthy_meal                   ideal_diet
##                            0                            0
##             ideal_diet_coded                       income
##                            0                            1
##                  indian_food                 italian_food
##                            0                            0
##               life_rewarding               marital_status
##                            1                            1
##          meals_dinner_friend             mother_education
##                            0                            0
##            mother_profession            nutritional_check
##                            0                            0
##                on_off_campus                 parents_cook
##                            1                            0
##                 pay_meal_out                 persian_food
##                            0                            1
##       self_perception_weight                         soup
##                            1                            1
##                       sports                    thai_food
##                            0                            0
##            tortilla_calories              turkey_calories
##                            1                            0
##                  type_sports                  veggies_day
##                            0                            0
##                     vitamins              waffle_calories
##                            0                            0
##                       weight
##                            0
```

This column is a duplicate so it will be dropped.

```r
food_choices <- food_choices %>%
  select(-comfort_food_reasons_coded)
```

Due to the large number of columns, I will clean only a select few that will be useful in answering the questions highlighted below.

- Is there any impact of regular exercise on an individual's weight? (gender, exercise, and weight columns will be necessary for this question)
- are students who exercise regularly more likely to make better food choices?

Before that, the GPA column presents an exciting opportunity to flex our data cleaning muscle. First, I'll use the `table` function to have a general overview of the values contained within this column.

- There's a value "3.79 bitch" that needs to be cleaned, and a regex pattern will be used to isolate this value and then clean it.
- replace unknown values with the average gpa value

```r
table(food_choices$GPA) # gpa distribution
```

```
##
##        2.2       2.25        2.4        2.6       2.71        2.8        2.9
##          1          1          1          2          1          5          2
##          3        3.1        3.2      3.292        3.3       3.35        3.4
##         11          3         10          1          9          1          9
##        3.5        3.6      3.605       3.63       3.65      3.654       3.67
##         13          7          1          1          1          1          1
##       3.68        3.7       3.73       3.75       3.77 3.79 bitch        3.8
##          1         10          1          1          1          1          6
##       3.83       3.87      3.882       3.89        3.9      3.904       3.92
##          2          1          1          1          7          1          1
##          4        nan  Personal     Unknown
##          4          2          1          1
```

```r
pat <- "\\d*\\.\\d*\\s" # regex to check for decimal numbers followed by a space
matched_gpa <- str_subset(food_choices$GPA, pat)
```

```r
cleaned_gpa<- sub(" bitch", "", str_subset(food_choices$GPA, pat)) # use the sub function to replace th
food_choices$GPA[which(food_choices$GPA %in% matched_gpa)] <- cleaned_gpa

dgt <- "\\d*\\.?\\d+" # regex to check for all decimal numbers
char_gpa <- str_subset(food_choices$GPA, dgt)
round(mean(as.numeric(char_gpa)), 2)
```

```
## [1] 3.42
```

```r
wrds <- "^[^0-9]"
unc <- str_subset(food_choices$GPA, wrds)
food_choices$GPA[which(food_choices$GPA %in% unc)] <- round(mean(as.numeric(char_gpa)), 2) # replace gp

food_choices$GPA <- as.numeric(food_choices$GPA) # convert to numeric
class(food_choices$GPA)
```

```
## [1] "numeric"
```

For the gpa column, there were no missing values, however some inconsistencies in formatting were noted and duly corrected. I replaced values that were "unknown" with the mean value of the column.

Change the numerical values in the gender column to align with the data dictionary using the `case_match` function Gender:

1 - Female 2 - Male

```r
food_choices %>% select(Gender) %>% unique() # inspect the values in this column
```

```
##   Gender
## 1      2
## 2      1
```

```r
food_choices$Gender <- food_choices %>% select(Gender) %>% mutate(Gender= case_match(Gender, 1 ~ 'Femal
food_choices$Gender <-as.character(food_choices$Gender$Gender)
```

utilize case when to change the values in the exercise column exercise:

- 1 - Everyday
- 2 - 2 -3 times weekly
- 3 - Once a week

```r
food_choices %>% select(exercise) %>% unique() #inspect for missing values
```

```
##   exercise
## 1        1
## 3        2
## 4        3
```

```r
food_choices$exercise <-  food_choices %>% select(exercise) %>% mutate(exercise = case_when(exercise ==
food_choices$exercise <- as.character(food_choices$exercise$exercise)
```

In the weight column, the following inconsistencies have been observed

- 144lbs
- Not sure, 240
- i’m not answering this.
- NA

I’ll use the `str_replace` function for data cleaning here. Also, unknown values were replaced with the mean of the weight column

```r
table(food_choices$weight)
```

```
## 
##                       100                      105                      110 
##                         1                        1                        1 
##                       112                      113                      115 
##                         1                        2                        1 
##                       116                      118                      120 
##                         1                        1                        3 
##                       123                      125                      127 
##                         1                        5                        1 
##                       128                      129                      130 
##                         2                        2                        4 
##                       135                      137                      138 
##                         8                        1                        1 
##                       140                  144 lbs                      145 
##                         8                        1                        4 
##                       150                      155                      156 
##                         7                        6                        1 
##                       160                      165                      167 
##                         3                        5                        2 
##                       168                      169                      170 
##                         1                        1                        7 
##                       175                      180                      184 
##                         6                        6                        1 
##                       185                      187                      190 
##                         6                        1                        5 
##                       192                      195                      200 
##                         1                        1                        4 
##                       205                      210                      230 
##                         1                        2                        1 
##                       260                      264                      265 
##                         1                        1                        1 
## I’m not answering this.                      nan            Not sure, 240 
##                         1                        2                        1
```

```r
# clean up values containing "lbs" and "not sure"
lbs <- str_subset(food_choices$weight, "lbs$")
ns <- str_subset(food_choices$weight, "Not")
```

```
food_choices$weight[which(food_choices$weight %in% lbs)] <- str_replace(lbs, "144 lbs", "144")
food_choices$weight[which(food_choices$weight %in% ns)] <- str_replace(ns, "Not sure, 240$", "240")

# to avoid data loss, replace unspecified values with the mean of the column
unsp <- str_subset(food_choices$weight, "^[^0-9].*") #regex to check for non-numeric entries
sp <- str_subset(food_choices$weight, "^[0-9].*")
food_choices$weight[which(food_choices$weight %in% unsp)] <- round(mean(as.numeric(sp)), 0) # replace t
food_choices$weight <-  as.numeric(food_choices$weight)
```

To answer the 2nd question, these variables will be of interest; `exercise`, `nutritional_checks`, `veggie_day`, `fruit_day`. The `exercise` column has been cleaned already.

The values to be replaced in the nutritional check column:

- 1 - Never
- 2 - On certain products
- 3 - Very rarely
- 4 - On most products
- 5 - On everything

```
table(food_choices$nutritional_check)
```

```
##
##  1  2  3  4  5
## 10 36 20 43 16
```

```
# the integer responses need to be changed to a more meaningful format
food_choices$nutritional_check <-  food_choices %>% select(nutritional_check) %>% mutate(nutritional_che
food_choices$nutritional_check <- as.character(food_choices$nutritional_check$nutritional_check)
```

Same will be done using the veggies_day column. Importantly, the values were converted to factors, to aid the reintegration into the dataframe.

```
table(food_choices$veggies_day)
```

```
##
##  1  2  3  4  5
##  3 11 21 37 53
```

```
# same has to be done for the veggies_day column
food_choices$veggies_day <-  food_choices %>% select(veggies_day) %>% mutate(veggies_day = case_match(ve
food_choices$veggies_day <- factor(food_choices$veggies_day$veggies_day, levels = c("very unlikely", "un
```

Repeat this step for the fruit_day column also.

```
table(food_choices$fruit_day)
```

```
##
##  1  2  3  4  5
##  1  4 24 33 63
```

```r
food_choices$fruit_day <- food_choices %>% select(fruit_day) %>% mutate(fruit_day = case_match(fruit_da
food_choices$fruit_day <- factor(food_choices$fruit_day$fruit_day, levels = c("very unlikely", "unlikel
```

convert the values in these columns to lowercase to ensure consistent formatting

```r
food_choices$comfort_food <-  str_to_lower(food_choices$comfort_food)
food_choices$comfort_food_reasons<- str_to_lower(food_choices$comfort_food_reasons)
```

Now, to dive into deeper waters, I want to clean the father_profession column. There are 2 steps in this phase; - Trim words that contain extra spaces - There are some misspelled words in this column that need to be corrected.

```r
table(str_to_lower(food_choices$father_profession))
```

```
## 
##                       accountant                          architect
##                                  2                                  1
##                        assembler                             banker
##                                  1                                  1
##                     beacon light             beverage and food sales
##                                  1                                  1
##       biohemical waste elimination                       business guy
##                                  1                                  1
##                       business man                     business owner
##                                  1                                  4
##                       car salesman                     ceo of company
##                                  1                                  1
##                                cfo                clinical researcher
##                                  1                                  1
##             commercial real estate                  commidity trader 
##                                  1                                  1
##       commissioner of erie county                       construction 
##                                  1                                  2
##           construction management               contract negotiations
##                                  1                                  1
##                 corporate manager                        cross-guard 
##                                  1                                  1
##                       dairy farmer                      dairy farmer 
##                                  1                                  1
##                         dead beat                            deceased
##                                  1                                  1
##          delivery man for fritolay                            dentist
##                                  1                                  1
##                           dentist                     design engineer
##                                  2                                  1
##                             doctor                electrical engineer
##                                  1                                  2
##                           engineer                          engineer 
##                                  1                                  1
##       european logistics director                             fireman
##                                  1                                  1
##                ford plant employee                        ge salesman
##                                 1                                 1
##                          handyman             high school principal
##                                 1                                 1
##                  his own business                      hockey coach
##                                 1                                 1
##                      home marker                    house appraiser
##                                 1                                 1
##                 hvac professional                  hvac technician
##                                 1                                 1
##                               idk     information systems architect
##                                 1                                 1
##                        insurance                                 it
##                                 1                                 2
##                        journalist                      landscaping
##                                 1                                 1
##                            lawyer                  manager at pepsi
##                                 2                                 1
##                         mechanic                mechanical engineer
##                                 2                                 1
##              mechanical engineer                                nan
##                                 1                                 3
##                          not sure                      optometrist
##                                 1                                 1
##           owner of new york lunch                     owns business
##                                 1                                 1
##                 owns his business owns his own promotional company
##                                 1                                 1
##                    pharmaceutical                physical therapist
##                                 1                                 1
##                      police force                    police officer
##                                 1                                 1
##                   police officer                          politician
##                                 1                                 1
##  president of automotive company                          profesor
##                                 1                                 1
##                   project manager  radio telecommunications manager
##                                 1                                 1
##                           realtor                            retire
##                                 1                                 1
##                           retired                           retired
##                                 1                                 1
##              retired - bus driver                      risk manager
##                                 1                                 1
##                             sales                    sales manager
##                                 1                                 1
##                          salesman  school library media specialist
##                                 2                                 1
##                     self employed                    self employed
##                                 1                                 1
##       self employed construction                     senior manager
##                                 1                                 1
##     sergeant correctional officer                service technition
##                                 1                                 1
##                   shirt designer              small business owner
##                                 1                                 2
##                solar engineering       store manager at giant eagle
##                    subcontractor                         supervisor
##                                 1                                 1
##                       taxi driver                            teacher
##                                 1                                 2
##                   transportation                       truck driver
##                                 1                                 1
##                     truck driver                       union worker
##                                 1                                 1
##                    united nations                          unknown
##                                 1                                 1
##                       ups driver        vice president of a company
##                                 1                                 1
##                            vp of                          vp of gnc
##                                 1                                 1
##                            welder             works for kirila fire
##                                 1                                 1
```

```
reg <- "\\s$"
# some words end with spaces which have to be cleaned
string_space <- str_subset(food_choices$father_profession, reg) # extract words that end with spaces
food_choices$father_profession <-  ifelse(food_choices$father_profession %in% string_space, sub(reg, ""
food_choices$father_profession)
```

To handle the misspelled words, I'll use the hunspell library. The hunspell library will act as a spellchecker here. The code, albeit bulky, has been well labelled to make the logic here easy to follow.

```
library(hunspell)
```

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

```
exempt_words <- c("Idk", "nan", "HVAC", "GNC", "Kirila") #words the spellchecker should ignore
# write a function that checks for misspelled words that are not part of the exempt words and corrects
correct_spelling <- function(sentence, exempt_words) {
  words <- unlist(strsplit(food_choices$father_profession, "\\s+")) #split all sentences into individua
  wrong_words <- unlist(hunspell(words, dict = dictionary("en_US")))
  wrong_words <- setdiff(wrong_words, exempt_words) # remove words found in exempt words
  suggestions <- hunspell_suggest(wrong_words, dict = dictionary("en_US")) # spelling corrections for p
  corrected_words <- vector("list", length = length(wrong_words)) # store the corrected words in a vect

  for (i in seq_along(wrong_words)) { #  iterate over the wrong words to check if corrections exist and
    if(length(suggestions[[i]])>0) {
      corrected_words[[i]] <- suggestions[[i]][1]
    } else {
      corrected_words[[i]] <- wrong_words[i]
    }
  }
  corrected_sentence <- sentence
  for (i in seq_along(wrong_words)) { # use gsub to replace occurrences of wrong spellings with the rig
    corrected_sentence <- gsub(wrong_words[i], corrected_words[[i]], corrected_sentence)
```

```r
  }
  return(corrected_sentence)
}

# iterate through each observation in the column and then apply the correction function
food_choices$father_profession <-sapply(food_choices$father_profession, function(x) correct_spelling(x,

# replace some unhelpful values with unknown
food_choices %>%
  select(father_profession) %>%
  mutate(father_profession= ifelse(father_profession %in% c("idk", "nan", "not sure"), "unknown", fathe
```

```
##                    father_profession
## 1                          professor
## 2                      Self employed
## 3                      owns business
## 4                          Assembler
## 5                                 IT
## 6                        Taxi Driver
## 7                     Shirt designer
## 8                       Business guy
## 9              High School Principal
## 10        self employed construction
## 11                               Idk
## 12                        accountant
## 13                             VP of
## 14                    business owner
## 15                       landscaping
## 16                      Hockey Coach
## 17                       Optometrist
## 18                      Construction
## 19                          Engineer
## 20                         architect
## 21                               CFO
## 22                     subcontractor
## 23              small business owner
## 24            Commercial Real Estate
## 25                  Manager at Pepsi
## 26                         Insurance
## 27           Beverage and Food Sales
## 28                         Dead beat
## 29               Electrical Engineer
## 30  Radio Telecommunications Manager
## 31                           unknown
## 32                          deceased
## 33                            Lawyer
## 34                      Dairy Farmer
## 35       Vice President of a company
## 36                 Solar Engineering
## 37                          engineer
## 38                       cross-guard
## 39     Biochemical Waste Elimination
## 40                           Retired
## 41  School Library Media Specialist
## 42                            Welder
## 43                   Design Engineer
## 44                        Accountant
## 45               Electrical Engineer
## 46                            Banker
## 47                          Mechanic
## 48                         Assembler
## 49                   House Appraiser
## 50                           unknown
## 51                           Fireman
## 52                  Commodity trader
## 53                      Construction
## 54                 HVAC Professional
## 55    Sergeant correctional officer
## 56                      union worker
## 57                          Salesman
## 58                 Owns his business
## 59                Physical Therapist
## 60 Owns his own promotional company
## 61                       Optometrist
## 62                      Construction
## 63                      police force
## 64                         VP of GNC
## 65           Owner of New York Lunch
## 66                           Dentist
## 67              small business owner
## 68  President of Automotive company
## 69                        UPS driver
## 70                         Insurance
## 71              Retired - Bus Driver
## 72                         Dead beat
## 73                    Police Officer
## 74                      Risk Manager
## 75                            retire
## 76                      car salesman
## 77                      dairy farmer
## 78                      Dairy Farmer
## 79                     self employed
## 80             Contract negotiations
## 81                          engineer
## 82                                IT
## 83             Works for Kirila Fire
## 84                           Realtor
## 85  School Library Media Specialist
## 86                            Lawyer
## 87                Service Technician
## 88                        Accountant
## 89                          handyman
## 90                     Self employed
## 91                   Project manager
## 92                           Teacher
## 93                      Truck Driver
## 94                    Senior Manager
## 95     information systems architect
## 96                        Supervisor
## 97        Delivery Man For Frito lay
## 98                           unknown
## 99                    Business Owner
## 100                   business owner
## 101                            VP of
## 102                         salesman
## 103              Mechanical Engineer
## 104                      GE Salesman
## 105                   Business Owner
## 106              Ford Plant employee
## 107              Clinical Researcher
## 108             Small business owner
## 109                            Sales
## 110                    subcontractor
## 111                          Retired
## 112                          unknown
## 113                       UPS driver
## 114                          Teacher
## 115                       Politician
## 116                   Pharmaceutical
## 117                     Business Man
## 118                 His own business
## 119                          Dentist
## 120                   United Nations
## 121                   Transportation
## 122                           Doctor
## 123                   CEO of company
## 124     Store manager at Giant Eagle
## 125                       Journalist
```

Overall, this project has helped to solidify my understanding of some functions essential for data cleaning in R. I hope anyone reading this has also found it useful!

AJANAKU AYOMIDE