Assignment 2 Data Wrangling

Yaniv Bronshtein

02/07/2021

Import the libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gapminder)
```

1. Download the dataset on restaurant inspection in csv format from https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/3nnpn8j (Links to an external site.) and convert it to a data frame. read data from csv remove leading and trailing White space

```
res_inspec_results <- read_csv(
  "DOHMH_New_York_City_Restaurant_Inspection_Results.csv",
  trim_ws = TRUE)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   CAMIS = col_double(),
##   ZIPCODE = col_double(),
##   SCORE = col_double(),
##   Latitude = col_double(),
##   Longitude = col_double(),
##   `Community Board` = col_double(),
##   BIN = col_double(),
##   BBL = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```
## Warning: 64 parsing failures.
##   row     col expected actual                                                  file
##  7215 ZIPCODE a double    N/A 'DOHMH_New_York_City_Restaurant_Inspection_Results.csv'
## 12416 ZIPCODE a double    N/A 'DOHMH_New_York_City_Restaurant_Inspection_Results.csv'
## 19460 ZIPCODE a double    N/A 'DOHMH_New_York_City_Restaurant_Inspection_Results.csv'
## 19907 ZIPCODE a double    N/A 'DOHMH_New_York_City_Restaurant_Inspection_Results.csv'
## 20937 ZIPCODE a double    N/A 'DOHMH_New_York_City_Restaurant_Inspection_Results.csv'
## ..... ....... ........ ...... ..................................................
## See problems(...) for more details.
```

Convert data read from file to data frame

```
df <- as.data.frame(res_inspec_results)
```

(1a) Form a new data frame restricted to restaurants in Queens with cuisine to "Pizza". [Hint: Use str_detect() in the library stringr. Examples of usage of str_detect() can be found here: **https://stringr.tidyverse.org/reference/str_detect.html]**

Filter df by specifying the BORO field as Queens and using str_detect() applied in conjunction with fixed() to ascertain that only the rows in the original data frame relating to Pizza in the borough Queens are left are left

```
queens_pizza <- filter(df, BORO == "Queens",
                         str_detect(`CUISINE DESCRIPTION`,
                                    fixed("Pizza", ignore_case = TRUE)))
```

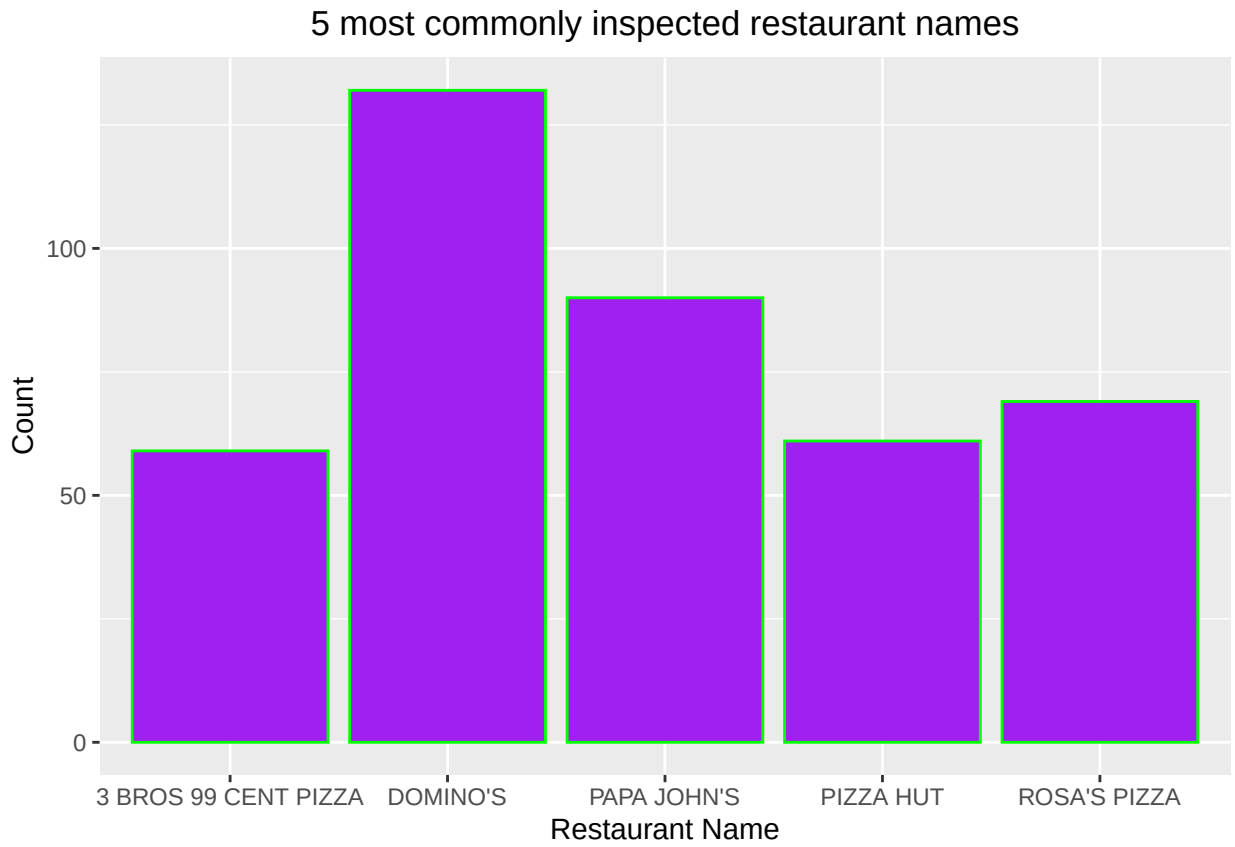
1b). What are the 5 most commonly names inspected restaurants (use the variable "DBA") in the data frame? (queens_pizza)

Perform the following steps 1. Group the queens_pizza data frame by the DBA field 2. Pipe the result of step 1 to summarize which will return a new data frame with just the frequency and the DBA field 3. Pipe the result of step 2 to the ungroup() function 4. Pipe the result of step 3 to the arrange() function to sort the frequency in descending order 5. Pipe the result of step 4 to the head() function which is passed the constant 5 to display only the first 5 rows of the data frame 6. Store the result of steps 1 through 5 in the variable common_names

```
common_names <- queens_pizza %>% group_by(DBA) %>% summarise(n=n()) %>%
  ungroup() %>% arrange(-n)%>%head(5)
```

Plot the results in a bar graph

```
ggplot(data = common_names,
       mapping = aes(x = DBA, y = n)) +
  geom_bar(stat = 'identity', color = "green", fill="purple") +
  labs(
    title = "5 most commonly inspected restaurant names",
    x = "Restaurant Name",
    y = "Count"
  ) +
  theme(plot.title = element_text(hjust = 0.5))
```



Based on the data in the graph, the 5 most inspected restaurants are DOMINOS' PAPA JOHN'S, Rosa's Pizza, PIZZA HUT, AND 3 BROS 99 CENT PIZZA

1c). On what dates has queens pizza parlor "SUSANO'S PIZZERIA & RESTAURANT" been inspected? Filter queens_pizza to match the exact restaurant name. NOTE: we cannot use str_detect() because that would mean including restaurants such as SUSANO'S PIZZERIA & RESTAURANT I

```
queens_pizza_susanos <- filter(queens_pizza,DBA=="SUSANO'S PIZZERIA & RESTAURANT")
```

Next, we make sure to call distinct() to get rid of duplicate dates. We only need the INSPECTION DATE column

```
dates_inspected <- distinct(queens_pizza_susanos["INSPECTION DATE"])
dates_inspected
```

```
##    INSPECTION DATE
## 1       09/25/2018
## 2       07/31/2019
## 3       03/15/2018
## 4       01/08/2020
## 5       03/14/2019
## 6       04/13/2018
## 7       12/09/2019
## 8       09/11/2018
## 9       03/01/2017
```

```
## 10      03/25/2019
## 11      08/14/2019
```

2. The file "gapminder_2007_gini.tsv" is in the "Assignments" folder under the files menu of the course website; it is a subset of the 2007 Gapminder data merged with recent coefficient data (https://en.wikipedia.org/wiki/Gini_coefficient

2a). Create a plot to compare the distributions (e.g., central tendency, dispersion) of the Gini coefficient in different continents. Hint: Use using geom_boxplot(). Details and examples are available here: https://ggplot2.tidyverse.org/reference/geom_boxplot.html]

Read in the data from the tab separated vector file, trimming white space

```
gapminder_data <- read_tsv("gapminder_2007_gini.tsv", trim_ws = TRUE)
```

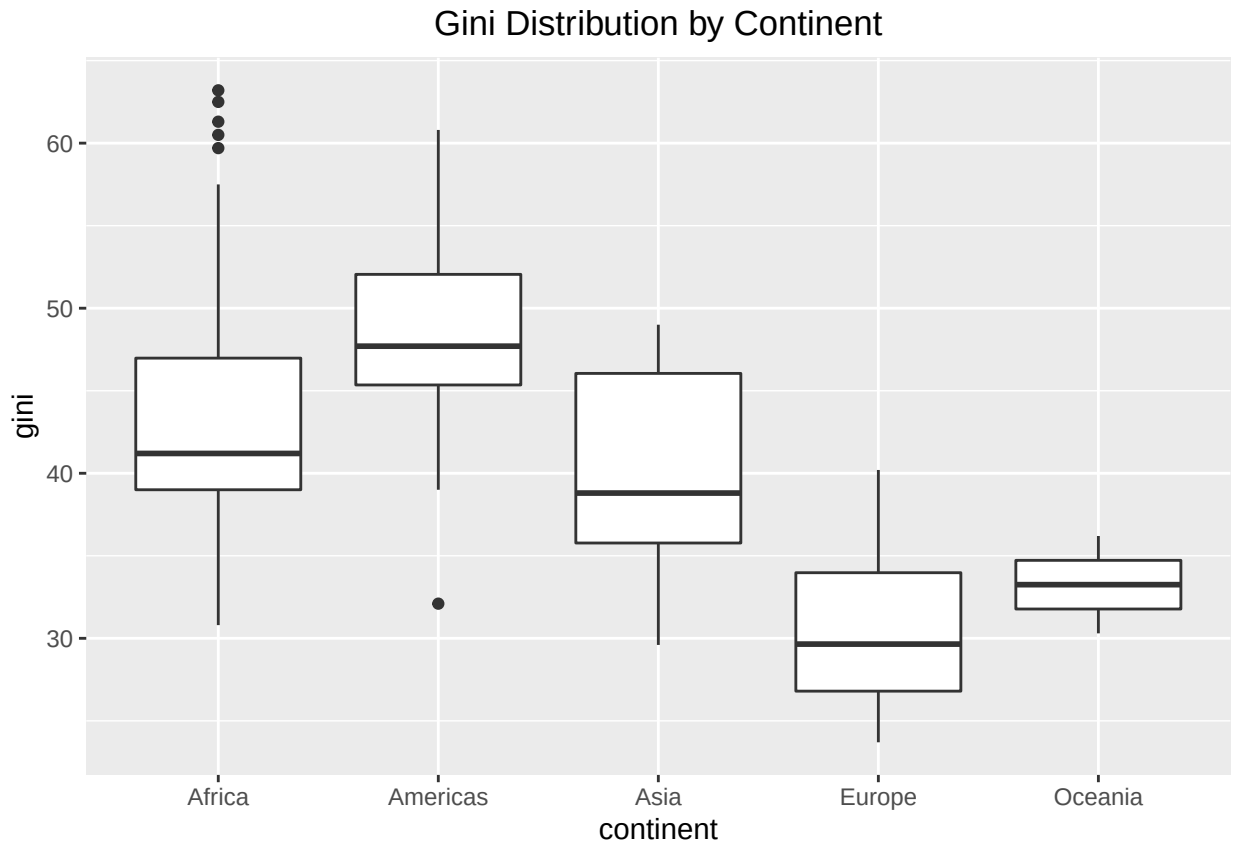
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   country = col_character(),
##   continent = col_character(),
##   year = col_double(),
##   lifeExp = col_double(),
##   pop = col_double(),
##   gdpPercap = col_double(),
##   gini = col_double()
## )
```

Convert the data read from the file to a data frame

```
gap_df <- as.data.frame(gapminder_data)
```

Create the box plot of distribution by continent

```
ggplot(data = gap_df, mapping = aes(x = continent, y = gini)) +
  geom_boxplot() +
  labs(
    title = "Gini Distribution by Continent"
  ) +
  theme(plot.title = element_text(hjust = 0.5))
```



2b) Does the Gini coefficient appear to have any impact on the life expectancy in 2007? Explain your answer using a plot, classified by continents

1. Filter the gap_df data frame by the year 2007 2. Pipe the result of step 1 to group_by() to aggregate the data by continent 3. Pipe the result of step 2 to the summarise() function which returns a data frame with 3 columns: the continents, the median of the gini, and the median of the life expectancy 4. Store the result of steps 1 through 3 in the variable gini_vs_life_exp

```
gini_vs_life_exp <- gap_df %>% filter(year == 2007)  %>% group_by(continent)  %>%
  summarise(continent, gini, lifeExp)
```

```
## `summarise()` has grouped output by 'continent'. You can override using the `.groups` argument.
```

Create a bar plot using the gini_vs_life_exp data frame

```
ggplot(data = gini_vs_life_exp, mapping = aes(x = gini, y = lifeExp, color = continent)) +
  geom_point() +
  geom_smooth(method = "lm", mapping = aes(group = 1)) +

  labs(
    title = "Effect of gini on life expectancy by Continent"
  ) +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

The graph above demonstrates a trend that countries with low gini such as Europe and Oceania have much higher life expectancy than Africa which falls below the linear regression line

(3) Using the original gapminder data frame, please generate a data frame with a new variable called gdp by multiplying the population size by the gdp per capita. To make those large numbers more understandable, please form an new variable called gdp_ratio equal to the gdp divided by the gdp of the United States in 2007. Find the mean gdp_ratio by continent and year, and then plot the mean gdp_ratio over time, distinguishing the continents. Please use both points and lines for the plot

Create a data frame to store the imported data from the R package gapminder

```
df2 <- gapminder
```

Call the dplyr function mutate() to create a calculated column in df2 to store the gdp

```
df2 <- dplyr::mutate(df2, gdp = pop * gdpPercap)
```

To avoid writing extra code, extract the 2007 gap data on the United States

```
us_2007_df <- gap_df %>% filter(country=="United States")
```

Calculate the gdp manually from the single row by multiply the gdp per capita by the population at the time

```
us_gdp_2007 <- us_2007_df["gdpPercap"] * us_2007_df["pop"]
```

Convert the result to a double value and store in the variable denom

```
denom <- as.numeric(us_gdp_2007)
```

Create another calculated column in df2 by calling mutate() on the calculation gdp / denom

```
df2 <- dplyr::mutate(df2, gdp_ratio = gdp / denom )
```

1. Pipe df2 to the function group_by() to group by continent and year 2. Pipe the result of step 1 to the summarise() function which returns a data frame with 3 columns: the continents, the year the mean of the gdp ratio **3. Save the results in the variable q3_info

```
q3_info <- df2 %>% group_by(continent, year)  %>%
  summarise(continent, year, mean_gdp_ratios = mean(gdp_ratio))
```

```
## `summarise()` has grouped output by 'continent', 'year'. You can override using the `.groups` argume
```

Create the plot of all the data

```
ggplot(data = q3_info, mapping = aes(x = year, y = mean_gdp_ratios, color = continent)) +
  geom_point() +
  geom_line()
```