# explanation

Rose Determan

10/6/2021

## Wrangling Code Blocks

### 1. Import necessary packages

```
library(tidyr)
library(dplyr)
library(stringr)
library(reshape2)
```

### 2. Import the literacy rate data and begin processing

I created a list of unique countries in the lit_rate dataframe and classified them into regions. I've included only a few of the country region mapping values since the vectors are rather long. from = country names; to = region names

```
lit_rate <- read.csv("literacy_rate_adult_total_percent_of_people_ages_15_and_above.csv",
                      fileEncoding = "UTF-8-BOM")
from <- c("Aruba","Afghanistan","Angola",..."Zambia","Zimbabwe")
to <- c("Caribbean & Central Amer.","Middle East",..."Africa","Africa")

#create a new column to put the regions
lit_rate$Region <- lit_rate$country

#use mapvalues to map the to values onto the from values
lit_rate$Region <- plyr::mapvalues(lit_rate$Region, from = from, to = to)
```

### 3. Repeat the import process for poverty data

### 4. Begin "tidying" the literacy rates

```
lit_rate[sample(nrow(lit_rate), 5), 1:10]
```

```
##              country X1974 X1975 X1976 X1977 X1978 X1979 X1980 X1981 X1982
## 29     Cote d'Ivoire    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 63              Iran    NA  36.5    NA    NA    NA    NA    NA    NA    NA
## 94        Mozambique    NA    NA    NA    NA    NA  27.1    NA    NA    NA
## 53 Equatorial Guinea    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 58           Croatia    NA    NA    NA    NA    NA    NA    NA    NA    NA
```

Use the `pivot_longer` function. As shown in the sample above, the data set is not formatted in a "tidy" manner. The first step I took was to take the columns names (1974, 1975, etc. ) and make those years into their own column.

```
#create a new var called lit_gather. Pass the lit_rate dataframe to the
```

```
pivot_longer command. Use the year columns as the columns to target, and we want
the new column to be called lit_rate.
lit_gather <- (lit_rate %>% pivot_longer(cols=c('X1974':'X2010'),
                                         values_to = "lit_rate",
                                         values_drop_na = TRUE))
#When the data were imported into R the year column names became strings with an
X in front. We want this new "names" column that was created from the pivot_longer
command to have the years rather than string.
lit_gather$year <- as.numeric(str_sub(lit_gather$name, start = 2, end = 5))
lit_gather <- select(lit_gather, -name)
```

Create a decade variable that is stored as a factor and create a new dataframe with countries summarized by decade. Since not every country has data for every year, I grouped observations by decade and country.

```
#subtract the measured year from the remainder after dividing the year by 10.
For example, 2012/10 remainder = 2. 2012-2 = 2010
lit_gather$decade <- as.factor(lit_gather$year - lit_gather$year %% 10)

#use the group_by and summarise commands to find the mean of each country for
each decade.
lit_dec <- lit_gather %>%
  group_by(decade, country) %>%
  summarise(mean_lit=(mean(lit_rate)))
```

### 5. Repeat the process for the poverty dataset

### 6. Join the literacy and poverty datasets using the `inner_join` command.

This allows you to see countries where there is a literacy rate and a poverty rate in the same year.

```
jn <- inner_join(lit_gather,pov_gather)
```

### 7. Create a long version of the data

To create a boxplot with literacy and poverty side-by-side, the data need to be in one dataframe. The `melt` command takes one column, and makes another "variable" column with the original column name. The new dataframes include the rate and the year. Then, create a new dataframe that combines the two "long" dataframes into one, but excludes the rows with information about the year. Create the region column as a factor. The levels are assigned based on the median literacy rate for the region.

```
lit_long <- melt(lit_gather)
pov_long <- melt(pov_gather)
new <- rbind(lit_long, pov_long)
new <- new[new$variable != "year",]
new$Region <- factor(new$Region, levels=c("Europe","North America", "Asia",
                                         "S. Amer.","Caribbean & Central Amer."
                                          ,"Oceania","Middle East", "Africa"))
```

## Visualization Code Blocks

### 1. Import necessary packages

Additionally, the esquisser command is commented out, but it can be used to interactively visualize the data.

```
library(esquisse)
library(plotly)
library(tidyverse)
library(gridExtra)

#esquisser(data = pov_gather, viewer = "browser")
```

## 2. Figure 1: violin plots

Pass the lit_gather data since it has information for each data point and is not already summarized.

```
rates_violin <- function(data){
    #create a ggplot with the data passed to the function
    ggplot(data, aes(x = decade, y = lit_rate) +
    #add a violon plot and specify fill color
    geom_violin(fill = "#112446") +
    #add appropriate axis titles, plot titles, and source i nfo
    labs(
      x = "Decade",
      y = "Literacy Rate",
      title = "Figure 1: Density of Literacy Rates by Decade",
      caption = "Source: UNESCO Institute of Statistics through
      www.gapminder.org",
    ) +
    #add theme
    theme_minimal()
}
```

## 3. Figure 2: scatter plot

Pass the joined dataframe to the function, since there is both a literacy and a poverty rate for the same year and country.

```
rates_scatter <- function(data){
  #create a ggplot object from the data passed and call the object plt
  plt <-  ggplot(data,
  #set aes and color the points by region
  #add text for the "hover text" in the final plot
    aes(x = lit_rate, y = pov_rate, colour = Region,
        text = paste(
          "Country: ", country, "\n",
          "Year: ", year, "\n",
          "Literacy Rate: ", lit_rate, "\n",
          "Poverty Rate: ", pov_rate, "\n",
          sep = ""
        )) +
    #plot the points
    geom_point(shape = "circle", size = 2, alpha = 0.85) +
    #set the color scale for the points
    scale_color_brewer(palette = "Dark2", direction = 1) +
    #add appropriate x and y axis titles, plot title, and data source
    labs(
      x = "Literacy Rate",
      y = "Poverty Rate",
      title = "Figure 2: Literacy Rate vs Poverty Rate",
      caption = "Source: World Bank, Development Research Group and UNESCO
      Institute of Statistics (UIS) through www.gapminder.org",)+
    theme_minimal()
```

```
   #use ggplotly to plot the data in an interactive manner
    ggplotly(plt,tooltip = "text")
}
```

## 4. Figure 3: region boxplots

Pass the data called "new" which contains the literacy and poverty data in one dataframe. This function outputs boxplots of each region's literacy and poverty rates side by side.

```
boxes <- function(new){
    #create a ggplot object. value =the proportion of either literacy or poverty
    #variable = indicator if data point is a literacy or a poverty point
    ggplot(new, aes(x=Region,y=value, fill=variable))+
    #add boxplot geom
    geom_boxplot() +
                  #add appropriate labels, as in the previous plots
                  labs(x = "Region",
                       y = "Rate",
                       title="Figure 3: Literacy Rate and Poverty by Region",
                       caption = "Source: World Bank, Development Research Group
                       and UNESCO Institute of Statistics (UIS) through
                       www.gapminder.org") +
      #edit the color scale so the legend shows "Literacy" rather than
      #"lit_rate"  and "Poverty" rather than "pov_rate"
      scale_fill_discrete(name = "Rate", labels = c("Literacy", "Poverty"))
}
```