IE6600
Computation and Visualization for Analytics
Final Project

Team Members

Jianjian Liu

Hui Du

Jessica Anna James

Background:

In our project, we use the WHO Motality database to create few user friendly plots to help people understand this topic.

We hope that through our project, it will be easier for people to understand the changes in the proportion of mortality causes and the number of deaths in different regions, so as to understand the health status and problems faced by human beings all over the world.

Or you can pay more attention to your own health by understanding the causes of high death numbers in different ages, genders, regions.

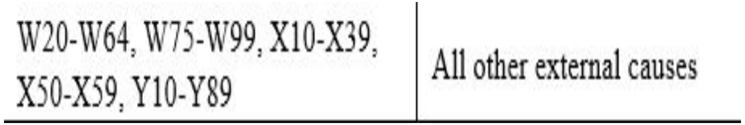
Data: introduction to Data

Data source: https://www.who.int/data/data-collection-tools/who-mortality-database

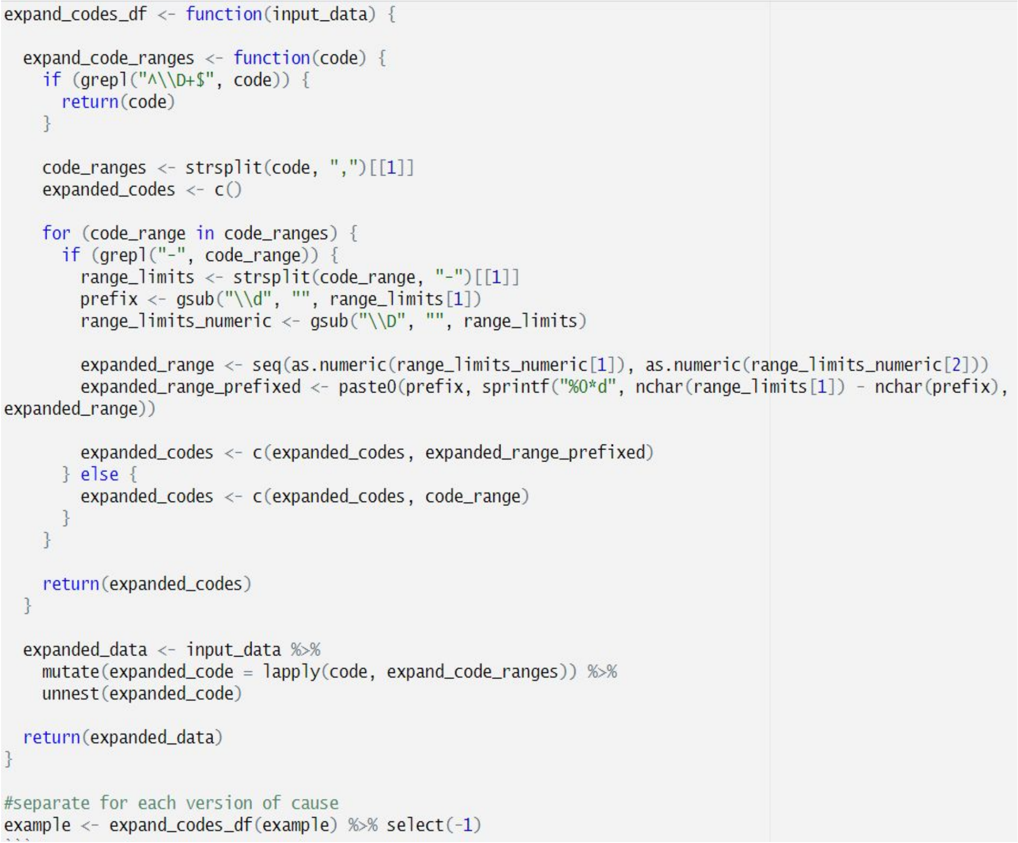
1. **Cause code:**
   a. data for Cause name and Cause code.
   b. https://icd.who.int/browse10/2019/en. and data manual.
2. **Mortality (2008-2021):**
   a. include Country code, Year, Cause code, Sex, deaths number in each age distribution etc.
   b. separate into three files: Mortality, ICD-10: (part 3/5), (part 4/5), (part 5/5)
3. **Country codes**: data for Country name and Country code.
4. **Population and live births**: use for calculated mortality rate in population.
5. **Documentation**: data manual.

Data: 1.Cause code:

In Cause code table, usually several groups of codes are put together to represent the same cause of death, which makes this data cannot be directly merged into the cause of Mortality data. We had to create a new column to display each code one by one.

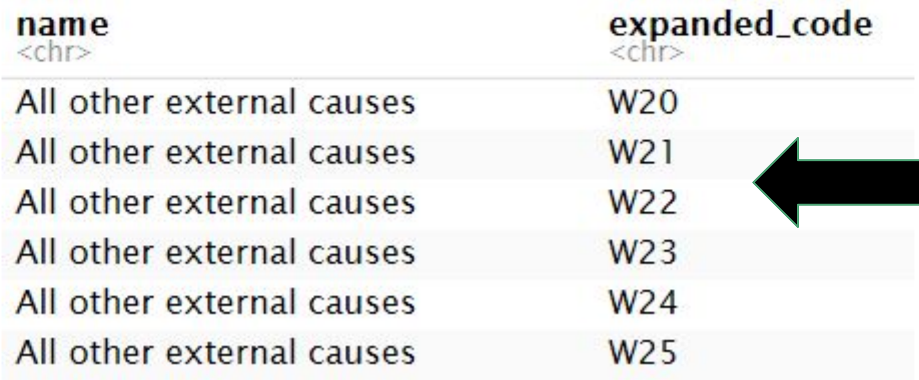


W20-W64, W75-W99, X10-X39, X50-X59, Y10-Y89	All other external causes

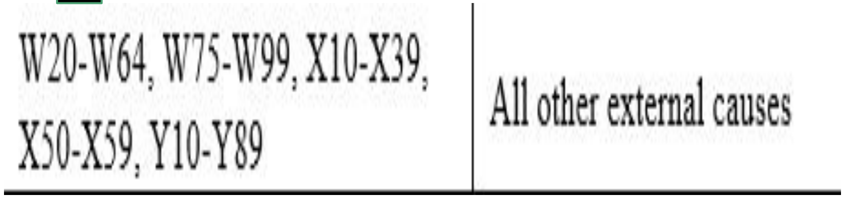


```
expand_codes_df <- function(input_data) {

  expand_code_ranges <- function(code) {
    if (grepl("^\\D+$", code)) {
      return(code)
    }

    code_ranges <- strsplit(code, ",")[[1]]
    expanded_codes <- c()

    for (code_range in code_ranges) {
      if (grepl("-", code_range)) {
        range_limits <- strsplit(code_range, "-")[[1]]
        prefix <- gsub("\\d", "", range_limits[1])
        range_limits_numeric <- gsub("\\D", "", range_limits)

        expanded_range <- seq(as.numeric(range_limits_numeric[1]), as.numeric(range_limits_numeric[2]))
        expanded_range_prefixed <- paste0(prefix, sprintf("%0*d", nchar(range_limits[1]) - nchar(prefix),
expanded_range))

        expanded_codes <- c(expanded_codes, expanded_range_prefixed)
      } else {
        expanded_codes <- c(expanded_codes, code_range)
      }
    }

    return(expanded_codes)
  }

  expanded_data <- input_data %>%
    mutate(expanded_code = lapply(code, expand_code_ranges)) %>%
    unnest(expanded_code)

  return(expanded_data)
}

#separate for each version of cause
example <- expand_codes_df(example) %>% select(-1)
```

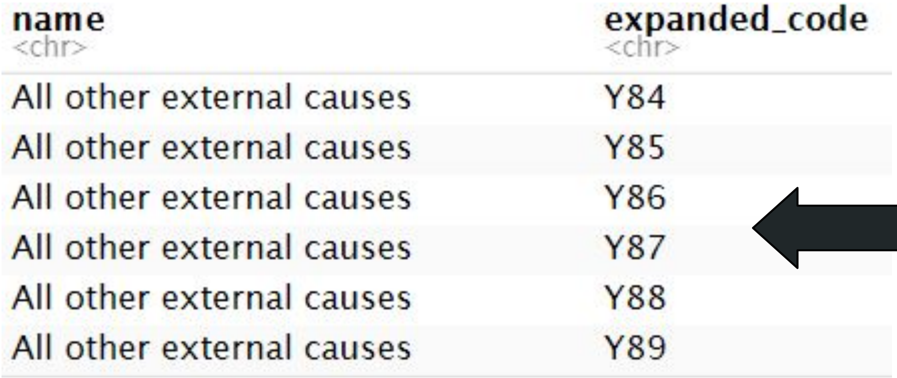
Data: 1.Cause code: output of example



name <chr>	expanded_code <chr>
All other external causes	W20
All other external causes	W21
All other external causes	W22
All other external causes	W23
All other external causes	W24
All other external causes	W25



W20-W64, W75-W99, X10-X39, X50-X59, Y10-Y89	All other external causes



name <chr>	expanded_code <chr>
All other external causes	Y84
All other external causes	Y85
All other external causes	Y86
All other external causes	Y87
All other external causes	Y88
All other external causes	Y89

Data: 2. Mortality (2008-2021)

Data frame: combine 3 separated data frame into one data frame

Columns(selected) :Year, Country, Cause, Sex, Total, Age group(age distribution)

Missing Value: set missing value of Age group as 0.

Age(distribution): make all age distribution consistent of each year.

Merge: Merge with Country name and Cause name.

Reshape Data: gather() Age_group, mortality

Data: Mortality rate data

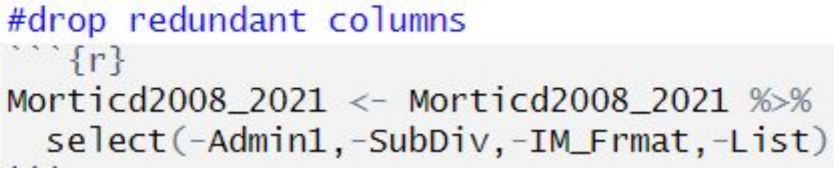
Process population data in the same way as the death data.

after two data reshaped, merge two data together:

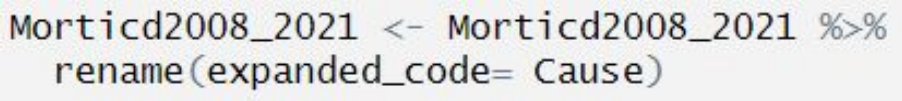
create new column(mortality rate) by using:

$$\textit{mortality rate} = \frac{\textit{mortality number}\left(\textit{of the age and gender}\right)}{\textit{total population}\left(\textit{of the age and gender}\right)}$$

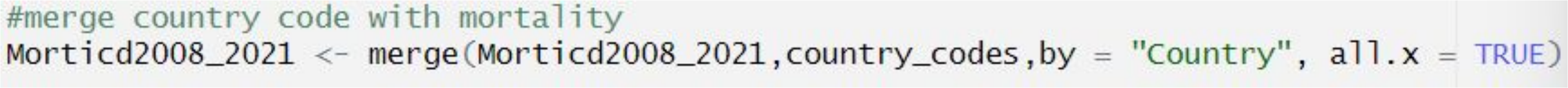
Data: some functions we use tidy data.



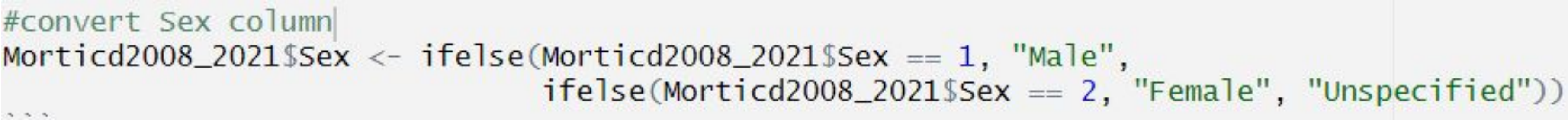
~~~
#drop redundant columns
```{r}
Morticd2008_2021 <- Morticd2008_2021 %>%
  select(-Admin1,-SubDiv,-IM_Frmat,-List)
```
~~~



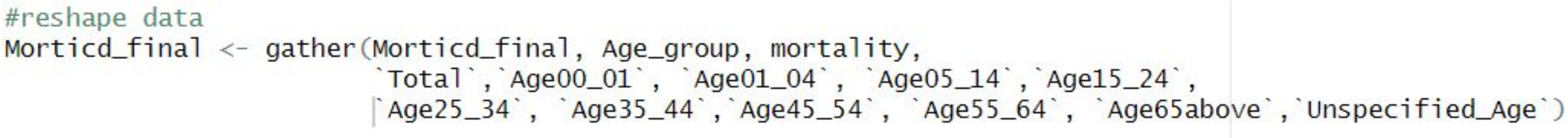
```
Morticd2008_2021 <- Morticd2008_2021 %>%
  rename(expanded_code= Cause)
```



```
#merge country code with mortality
Morticd2008_2021 <- merge(Morticd2008_2021,country_codes,by = "Country", all.x = TRUE)
```



~~~
#convert Sex column
Morticd2008_2021$Sex <- ifelse(Morticd2008_2021$Sex == 1, "Male",
                               ifelse(Morticd2008_2021$Sex == 2, "Female", "Unspecified"))
```
~~~

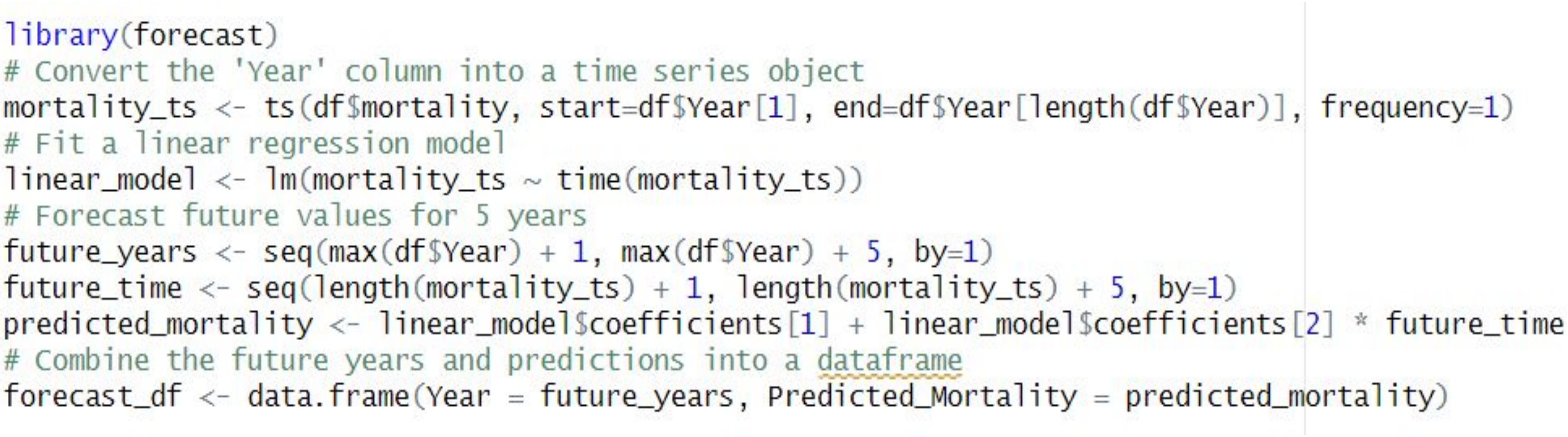


```
#reshape data
Morticd_final <- gather(Morticd_final, Age_group, mortality,
                        `Total`,`Age00_01`, `Age01_04`, `Age05_14`,`Age15_24`,
                        `Age25_34`, `Age35_44`,`Age45_54`, `Age55_64`, `Age65above`,`Unspecified_Age`)
```

Data: Linear regression predicts trends.

Disadvantage: Linearity assumption, Overfitting, etc.

Limitation: not accurate for specific points in the future time



```r
library(forecast)
# Convert the 'Year' column into a time series object
mortality_ts <- ts(df$mortality, start=df$Year[1], end=df$Year[length(df$Year)], frequency=1)
# Fit a linear regression model
linear_model <- lm(mortality_ts ~ time(mortality_ts))
# Forecast future values for 5 years
future_years <- seq(max(df$Year) + 1, max(df$Year) + 5, by=1)
future_time <- seq(length(mortality_ts) + 1, length(mortality_ts) + 5, by=1)
predicted_mortality <- linear_model$coefficients[1] + linear_model$coefficients[2] * future_time
# Combine the future years and predictions into a dataframe
forecast_df <- data.frame(Year = future_years, Predicted_Mortality = predicted_mortality)
```

Functions of our Project

We have created an R Shiny application that is designed to provide a comprehensive and interactive platform allowing users to explore Mortality trends, patterns, and insights with respect to the causes, time frame, gender and age

.

We have implemented the following functions in our application :

1)Histogram

2)Line graphs

3)Choropleth

4)Radar plots

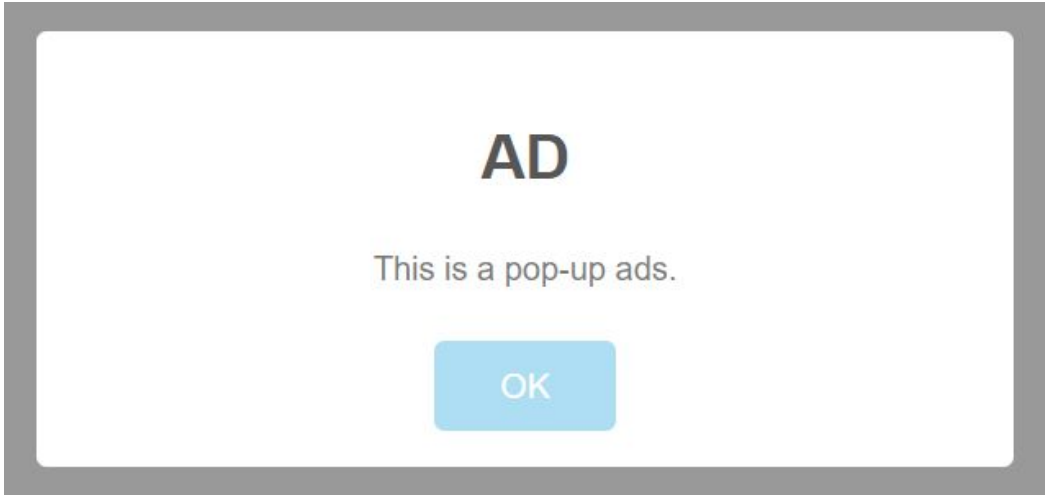
By implementing the above functions, we have realized various factors that contribute to the mortality rate in the world

Two Interesting Shiny App Implementations

2. Pop-up Advertise function

```
install.packages("shinyalert")
```

```
shinyalert() + Sys.sleep()  =
```



AD

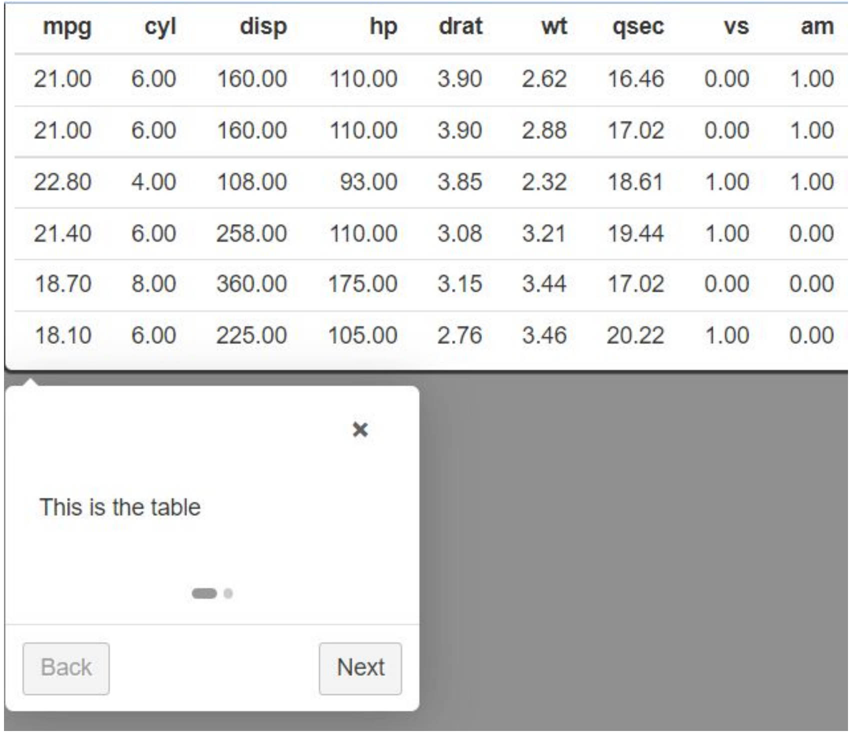
This is a pop-up ads.

Two Interesting Shiny App Implementations

1. Interactive tutorials function

   install.packages("shiny")

   introjsUI() + introBox() =



mpg	cyl	disp	hp	drat	wt	qsec	vs	am
21.00	6.00	160.00	110.00	3.90	2.62	16.46	0.00	1.00
21.00	6.00	160.00	110.00	3.90	2.88	17.02	0.00	1.00
22.80	4.00	108.00	93.00	3.85	2.32	18.61	1.00	1.00
21.40	6.00	258.00	110.00	3.08	3.21	19.44	1.00	0.00
18.70	8.00	360.00	175.00	3.15	3.44	17.02	0.00	0.00
18.10	6.00	225.00	105.00	2.76	3.46	20.22	1.00	0.00

This is the table

THANK YOU

Thank You