

STATS 205: Final Project Write-Up

Brian Liu

6/14/2019

1. Background of the data and why it is interesting or important

The data we are using is the data from WHO suicide statistics from Kaggle. This gives population-based statistics on suicide rate.

The reason this data is interesting and important is that suicide is prevalent in many times and places around the world, but many places and times have different suicide rates. When it comes to suicide, there are many potential factors or attributes that may be correlated with an increased risk of suicide, such as:

- a person's sex
- the age group a person belongs to
- the generation a person was born in

The goal is to find significant correlations between these factors and suicide rates: that is, does x factor positively predict suicide rate?

The simple inspiration is suicide prevention: If we can identify the factors that correlate positively with, or predict high suicide rates, then we can target our suicide prevention efforts towards populations with those high-risk factors or attributes.

2. Explanation of the method studied and its properties

We will use the statistical techniques of **nonparametric bootstrap** and parametric bootstrap methods to aid in prediction, with linear regression as well, and use cross-validation to test if, given new data for a population, this population is at risk of suicide. In other words, predict if the suicide rate would be abnormally or significantly high, and then compare the performance between the two methods (nonparametric and parametric).

Nonparametric bootstrap

Nonparametric bootstrap is

Parametric bootstrap

Linear regression

Cross validation

3. Data analysis or simulation study

We will use the crude rate of suicide per 100,000 people.

This analysis provides information on age-standardized rates...

```
who_suicide_statistics_df <- read.csv("who_suicide_statistics.csv")
head(who_suicide_statistics_df)
```

```
##   country year    sex      age suicides_no population
## 1 Albania 1985 female 15-24 years         NA      277900
## 2 Albania 1985 female 25-34 years         NA      246800
## 3 Albania 1985 female 35-54 years         NA      267500
## 4 Albania 1985 female  5-14 years         NA      298300
## 5 Albania 1985 female 55-74 years         NA      138700
## 6 Albania 1985 female  75+ years         NA       34200
```

```
colnames(who_suicide_statistics_df)
```

```
## [1] "country"      "year"          "sex"           "age"           "suicides_no"
## [6] "population"
```

Filter and save countries with missing suicide rate.

```
library(tidyverse)
```

```
## Registered S3 methods overwritten by 'ggplot2':
```

```
##   method      from
## [.quosures    rlang
## c.quosures     rlang
## print.quosures rlang
```

```
## -- Attaching packages ----- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1    v purrr   0.3.2
## v tibble  2.1.1    v dplyr   0.8.1
## v tidyr   0.8.3    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.4.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
filtered_suicide_df <- drop_na(who_suicide_statistics_df, "suicides_no")
head(filtered_suicide_df)
```

```
##   country year    sex      age suicides_no population
## 25 Albania 1987 female 15-24 years         14      289700
## 26 Albania 1987 female 25-34 years          4      257200
## 27 Albania 1987 female 35-54 years          6      278800
## 28 Albania 1987 female  5-14 years          0      311000
## 29 Albania 1987 female 55-74 years          0      144600
## 30 Albania 1987 female  75+ years          1       35600
```

After filtering countries with missing suicide rate, take a random sample of 100 countries and make sure each continent has approximately equal countries.

Filter countries by continent:

```
library(countrycode)
filtered_suicide_df$continent <- countrycode(sourcevar = filtered_suicide_df[, "country"],
                                             origin = "country.name",
                                             destination = "continent")
```

```
## Warning in countrycode(sourcevar = filtered_suicide_df[, "country"], origin = "country.name", : Some
```

```
## Warning in countrycode(sourcevar = filtered_suicide_df[, "country"], origin = "country.name", : Some
head(filtered_suicide_df)
```

```
##      country year    sex      age suicides_no population continent
## 25 Albania 1987 female 15-24 years          14    289700      Europe
## 26 Albania 1987 female 25-34 years           4    257200      Europe
## 27 Albania 1987 female 35-54 years           6    278800      Europe
## 28 Albania 1987 female  5-14 years           0    311000      Europe
## 29 Albania 1987 female 55-74 years           0    144600      Europe
## 30 Albania 1987 female  75+ years            1     35600      Europe
```

```
write.csv(filtered_suicide_df, 'filtered_suicide.csv')
```

Let us find out which continents are counted:

```
# Get list of continents
list_of_continents <- unique(filtered_suicide_df$continent); list_of_continents
```

```
## [1] "Europe" "Americas" "Asia" "Oceania" "Africa" NA
```

Therefore,

$$\frac{100 \text{ countries}}{6 \text{ continents}} \approx 16 \text{ to } 17 \text{ countries per continent}$$

we should randomly sample 17 countries from each continent.

Notably, there are countries that are not on any of the listed continents. Let us see which ones those are:

```
not_in_a_continent = filtered_suicide_df[is.na(filtered_suicide_df$continent),]
write.csv(not_in_a_continent, 'not_in_a_continent.csv')
head(not_in_a_continent)
```

```
##      country year    sex      age suicides_no population continent
## 32317 Rodrigues 2001 female 15-24 years           0         NA      <NA>
## 32318 Rodrigues 2001 female 25-34 years           0         NA      <NA>
## 32319 Rodrigues 2001 female 35-54 years           0         NA      <NA>
## 32320 Rodrigues 2001 female  5-14 years           0         NA      <NA>
## 32321 Rodrigues 2001 female 55-74 years           0         NA      <NA>
## 32322 Rodrigues 2001 female  75+ years           0         NA      <NA>
```

```
unique(not_in_a_continent$country)
```

```
## [1] Rodrigues      Virgin Islands (USA)
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
```

Let us make the choice not to include these countries in the analysis, since there are only two countries.

```
# Take off `NA` from list of continents
list_of_continents <- list_of_continents[-length(list_of_continents)]
list_of_continents
```

```
## [1] "Europe" "Americas" "Asia" "Oceania" "Africa"
```

We will now create six dataframes, filtered by list of countries for each continent.

```
# library(rlist)
countries_per_continent <- list()

for (i in seq_along(list_of_continents))
```

```
{
  countries_per_continent[[i]] <- filtered_suicide_df[filtered_suicide_df$continent == list_of_continents[i]]
}
```

```
length(countries_per_continent)
```

```
## [1] 5
```

```
length(countries_per_continent)
```

```
## [1] 5
```

```
for (i in seq_along(countries_per_continent))
{
  print(head(countries_per_continent[[i]]))
  print(length(countries_per_continent[[i]]))
  cat("\n")
}
```

```
##      country year    sex      age suicides_no population continent
## 25 Albania 1987 female 15-24 years          14      289700      Europe
## 26 Albania 1987 female 25-34 years           4      257200      Europe
## 27 Albania 1987 female 35-54 years           6      278800      Europe
## 28 Albania 1987 female  5-14 years           0      311000      Europe
## 29 Albania 1987 female 55-74 years           0      144600      Europe
## 30 Albania 1987 female  75+ years            1       35600      Europe
```

```
## [1] 7
```

```
##
```

```
##      country year    sex      age suicides_no population continent
## 373 Anguilla 1983 female 15-24 years           0           NA      Americas
## 374 Anguilla 1983 female 25-34 years           0           NA      Americas
## 375 Anguilla 1983 female 35-54 years           0           NA      Americas
## 376 Anguilla 1983 female  5-14 years           0           NA      Americas
## 377 Anguilla 1983 female 55-74 years           0           NA      Americas
## 378 Anguilla 1983 female  75+ years            0           NA      Americas
```

```
## [1] 7
```

```
##
```

```
##      country year    sex      age suicides_no population continent
## 1501 Armenia 1981 female 15-24 years           5      348000      Asia
## 1502 Armenia 1981 female 25-34 years           6      242200      Asia
## 1503 Armenia 1981 female 35-54 years           6      333500      Asia
## 1504 Armenia 1981 female  5-14 years           0      295200      Asia
## 1505 Armenia 1981 female 55-74 years          10      164300      Asia
## 1506 Armenia 1981 female  75+ years            7       43100      Asia
```

```
## [1] 7
```

```
##
```

```
##      country year    sex      age suicides_no population continent
## 2161 Australia 1979 female 15-24 years          71      1236800      Oceania
## 2162 Australia 1979 female 25-34 years          86      1138500      Oceania
## 2163 Australia 1979 female 35-54 years         171      1572100      Oceania
## 2164 Australia 1979 female  5-14 years           1      1246500      Oceania
## 2165 Australia 1979 female 55-74 years         135      1137800      Oceania
## 2166 Australia 1979 female  75+ years          15       309900      Oceania
```

```
## [1] 7
```

```
##
```

```
##      country year    sex      age suicides_no population continent
## 7669 Cabo Verde 2011 female 15-24 years          1      56039      Africa
## 7670 Cabo Verde 2011 female 25-34 years          0      38528      Africa
## 7671 Cabo Verde 2011 female 35-54 years          2      49078      Africa
## 7672 Cabo Verde 2011 female 5-14 years           0      56558      Africa
## 7673 Cabo Verde 2011 female 55-74 years          2      19887      Africa
## 7674 Cabo Verde 2011 female 75+ years            0       7582      Africa
## [1] 7
```

This text links to very important information about why a `for` loop doesn't print anything.¹

[Link to Pandoc Markdown formatting](#)

Randomly sample 17 countries from each continent:

```
list_of_continents
```

```
## [1] "Europe" "Americas" "Asia" "Oceania" "Africa"
```

```
for (i in seq_along(countries_per_continent))
{
  print(list_of_continents[i])
  countries <- unique(countries_per_continent[[i]]$country)
  print(countries)
  print(length(countries))
  cat("\n")
}
```

```
## [1] "Europe"
## [1] Albania          Austria            Belarus
## [4] Belgium          Bosnia and Herzegovina Bulgaria
## [7] Croatia          Czech Republic    Denmark
## [10] Estonia          Finland           France
## [13] Germany          Greece            Hungary
## [16] Iceland          Ireland           Italy
## [19] Latvia           Lithuania          Luxembourg
## [22] Malta            Monaco            Montenegro
## [25] Netherlands      Norway            Poland
## [28] Portugal         Republic of Moldova <NA>
## [31] Romania          Russian Federation San Marino
## [34] Serbia           Slovakia           Slovenia
## [37] Spain            Sweden             Switzerland
## [40] TFYR Macedonia   Ukraine            United Kingdom
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 42
##
## [1] "Americas"
## [1] Anguilla          Antigua and Barbuda
## [3] Argentina         Aruba
## [5] Bahamas           Barbados
## [7] Belize            Bermuda
## [9] Bolivia           Brazil
## [11] British Virgin Islands Canada
```

¹Basically, `for` loops are functions themselves. R prints out the result of a command automatically, but functions are not inherently a command, and since `for` loops are functions, nothing will be printed. The solution is to have `print(command())` within the `for` loop to get output for your `for` loop. You will never again spend hours trying to find out why a `for` loop doesn't print anything because you're no longer an R newbie.

## [13] Cayman Islands	Chile
## [15] Colombia	Costa Rica
## [17] Cuba	Dominica
## [19] Dominican Republic	Ecuador
## [21] El Salvador	Falkland Islands (Malvinas)
## [23] French Guiana	Grenada
## [25] Guadeloupe	Guatemala
## [27] Guyana	Haiti
## [29] Honduras	Jamaica
## [31] Martinique	Mexico
## [33] Montserrat	Netherlands Antilles
## [35] Nicaragua	Panama
## [37] Paraguay	Peru
## [39] Puerto Rico	<NA>
## [41] Saint Kitts and Nevis	Saint Lucia
## [43] Saint Pierre and Miquelon	Saint Vincent and Grenadines
## [45] Suriname	Trinidad and Tobago
## [47] Turks and Caicos Islands	United States of America
## [49] Uruguay	Venezuela (Bolivarian Republic of)
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe	
## [1] 50	
##	
## [1] "Asia"	
## [1] Armenia	Azerbaijan
## [3] Bahrain	Brunei Darussalam
## [5] Cyprus	Georgia
## [7] Hong Kong SAR	Iran (Islamic Rep of)
## [9] Iraq	Israel
## [11] Japan	Jordan
## [13] Kazakhstan	Kuwait
## [15] Kyrgyzstan	Macau
## [17] Malaysia	Maldives
## [19] Mongolia	Occupied Palestinian Territory
## [21] Oman	Philippines
## [23] Qatar	Republic of Korea
## [25] <NA>	Saudi Arabia
## [27] Singapore	Sri Lanka
## [29] Syrian Arab Republic	Tajikistan
## [31] Thailand	Turkey
## [33] Turkmenistan	United Arab Emirates
## [35] Uzbekistan	
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe	
## [1] 35	
##	
## [1] "Oceania"	
## [1] Australia Fiji Kiribati New Zealand <NA>	
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe	
## [1] 5	
##	
## [1] "Africa"	
## [1] Cabo Verde	Egypt Mauritius
## [4] Mayotte	Morocco Reunion
## [7] <NA>	Sao Tome and Principe Seychelles
## [10] South Africa	Tunisia Zimbabwe

```
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 12
```

Since there are only 5 countries in Oceania and 12 countries in Africa, we will use all 5 countries of Oceania and all 12 countries of Africa.

```
samples_of_countries <- list()
num_samples <- 17
for (i in seq_along(countries_per_continent))
{
  countries <- unique(countries_per_continent[[i]]$country)
  current_sample <- list()
  if (length(countries) >= num_samples)
  {
    current_sample <- sample(countries, 17)
  } else {
    current_sample <- sample(countries, length(countries))
  }
  samples_of_countries[[i]] <- current_sample
}
```

Let's see the countries that we will be sampling:

```
total <- 0
for (i in seq_along(samples_of_countries))
{
  print(list_of_continents[i])
  print(samples_of_countries[[i]])
  print(length(samples_of_countries[[i]]))
  total <- total + length(samples_of_countries[[i]])
  cat("\n")
}
```

```
## [1] "Europe"
## [1] Ukraine Belarus Austria
## [4] Ireland Luxembourg TFYR Macedonia
## [7] Romania Netherlands Serbia
## [10] France Slovakia Bosnia and Herzegovina
## [13] Hungary Monaco Poland
## [16] Greece Russian Federation
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 17
##
## [1] "Americas"
## [1] Costa Rica Suriname Montserrat
## [4] Trinidad and Tobago Haiti Cuba
## [7] Uruguay Honduras Netherlands Antilles
## [10] Bahamas British Virgin Islands Peru
## [13] Saint Kitts and Nevis Bermuda Aruba
## [16] Belize Grenada
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 17
##
## [1] "Asia"
## [1] Turkey United Arab Emirates
## [3] Turkmenistan Tajikistan
```

```
## [5] Iran (Islamic Rep of)      Macau
## [7] Kuwait                     <NA>
## [9] Cyprus                     Japan
## [11] Occupied Palestinian Territory Malaysia
## [13] Republic of Korea          Kyrgyzstan
## [15] Jordan                     Thailand
## [17] Singapore
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 17
##
## [1] "Oceania"
## [1] Kiribati      New Zealand <NA>      Australia   Fiji
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 5
##
## [1] "Africa"
## [1] Mauritius      Reunion      Egypt
## [4] Tunisia      Seychelles   Morocco
## [7] Sao Tome and Principe Cabo Verde    Zimbabwe
## [10] South Africa   Mayotte      <NA>
## 141 Levels: Albania Anguilla Antigua and Barbuda Argentina ... Zimbabwe
## [1] 12
```

```
total
```

```
## [1] 68
```

Let's filter the original dataframe only to include countries that we have sampled:

```
countries_to_test <- list()
a <- 0
for (i in seq_along(samples_of_countries))
{
  # find out a way to access each country name
  # print each country name
  for (j in seq_along(samples_of_countries[[i]]))
  {
    sample <- samples_of_countries[[i]]
    country_string <- toString(sample[[j]])
    countries_to_test[a] <- country_string
    a <- a + 1
  }
}

length(countries_to_test)
```

```
## [1] 67
```

```
countries_to_test
```

```
## [[1]]
## [1] "Belarus"
##
## [[2]]
## [1] "Austria"
##
## [[3]]
```



```

## [1] "Ireland"
##
## [[4]]
## [1] "Luxembourg"
##
## [[5]]
## [1] "TFYR Macedonia"
##
## [[6]]
## [1] "Romania"
##
## [[7]]
## [1] "Netherlands"
##
## [[8]]
## [1] "Serbia"
##
## [[9]]
## [1] "France"
##
## [[10]]
## [1] "Slovakia"
##
## [[11]]
## [1] "Bosnia and Herzegovina"
##
## [[12]]
## [1] "Hungary"
##
## [[13]]
## [1] "Monaco"
##
## [[14]]
## [1] "Poland"
##
## [[15]]
## [1] "Greece"
##
## [[16]]
## [1] "Russian Federation"
##
## [[17]]
## [1] "Costa Rica"
##
## [[18]]
## [1] "Suriname"
##
## [[19]]
## [1] "Montserrat"
##
## [[20]]
## [1] "Trinidad and Tobago"
##
## [[21]]

```

```

## [1] "Haiti"
##
## [[22]]
## [1] "Cuba"
##
## [[23]]
## [1] "Uruguay"
##
## [[24]]
## [1] "Honduras"
##
## [[25]]
## [1] "Netherlands Antilles"
##
## [[26]]
## [1] "Bahamas"
##
## [[27]]
## [1] "British Virgin Islands"
##
## [[28]]
## [1] "Peru"
##
## [[29]]
## [1] "Saint Kitts and Nevis"
##
## [[30]]
## [1] "Bermuda"
##
## [[31]]
## [1] "Aruba"
##
## [[32]]
## [1] "Belize"
##
## [[33]]
## [1] "Grenada"
##
## [[34]]
## [1] "Turkey"
##
## [[35]]
## [1] "United Arab Emirates"
##
## [[36]]
## [1] "Turkmenistan"
##
## [[37]]
## [1] "Tajikistan"
##
## [[38]]
## [1] "Iran (Islamic Rep of)"
##
## [[39]]

```

```

## [1] "Macau"
##
## [[40]]
## [1] "Kuwait"
##
## [[41]]
## [1] "NA"
##
## [[42]]
## [1] "Cyprus"
##
## [[43]]
## [1] "Japan"
##
## [[44]]
## [1] "Occupied Palestinian Territory"
##
## [[45]]
## [1] "Malaysia"
##
## [[46]]
## [1] "Republic of Korea"
##
## [[47]]
## [1] "Kyrgyzstan"
##
## [[48]]
## [1] "Jordan"
##
## [[49]]
## [1] "Thailand"
##
## [[50]]
## [1] "Singapore"
##
## [[51]]
## [1] "Kiribati"
##
## [[52]]
## [1] "New Zealand"
##
## [[53]]
## [1] "NA"
##
## [[54]]
## [1] "Australia"
##
## [[55]]
## [1] "Fiji"
##
## [[56]]
## [1] "Mauritius"
##
## [[57]]

```

```

## [1] "Reunion"
##
## [[58]]
## [1] "Egypt"
##
## [[59]]
## [1] "Tunisia"
##
## [[60]]
## [1] "Seychelles"
##
## [[61]]
## [1] "Morocco"
##
## [[62]]
## [1] "Sao Tome and Principe"
##
## [[63]]
## [1] "Cabo Verde"
##
## [[64]]
## [1] "Zimbabwe"
##
## [[65]]
## [1] "South Africa"
##
## [[66]]
## [1] "Mayotte"
##
## [[67]]
## [1] "NA"

```

4. Interpretation of the results or discussion

5. References