R bootcamp - suicide

Eva-Maria Bonin

Last edited Sunday 12/03/2023

Contents

Introduction **2**

Data sources **2**

Methods **2**

Pre-processing 2

Final dataset 6

Exploratory data analysis **8**

Summarising the dataset 8

Exlploring data by continent 9

Model **14**

Special chapter **14**

Conclusions **14**

Introduction

Data sources

World Bank

Global Burden of Disease study

Methods

Pre-processing

Obtaining and merging data files.

- Create a column to capture the gender associated with an indicator value and removed the string indicating gender from the indicator descriptions:

```
# Creating gender column in WB data
tmp_wb_long <- tmp_wb_long %>%
  mutate(sex = case_when(
    str_detect(indicator, "female") ~ "Female",
    str_detect(indicator, "male") ~ "Male"),
    sex = ifelse(is.na(sex), "Both", sex))

# Remove "male" and "female" from indicator descriptions
tmp_wb_long$indicator <- gsub("_male","",as.character(tmp_wb_long$indicator))
tmp_wb_long$indicator <- gsub("_female","",as.character(tmp_wb_long$indicator))

# dropping ISO2C column; pretty sure there's an easier way.
tmp_wb_long <- subset(tmp_wb_long, select = -c(iso2c))
```

Merging datasets and adding in variable for continent:

```
# Merging
tmp_wb_gbd_long <- rbind(tmp_wb_long, gbd)
tmp_wb_gbd_long[tmp_wb_gbd_long == 'NULL'] <- NA


# Generating wide version of dataset
tmp_wb_gbd_wide <- tmp_wb_gbd_long %>%
  pivot_wider(names_from = "indicator",
              values_from = "value"
  )

# Adding in continent
continent <- read.csv("../data/continents.csv")
tmp_wb_gbd_wide <- left_join(tmp_wb_gbd_wide, continent, by = "iso3c")
```

During the initial cleaning, I generated a list of all unique country names in the combined dataset to check for duplicates caused by variance in spelling. These issues were resolved by manually modifying country names in the GBD datafile, to reflect the names used in the WB data.

I renamed the variables in the wide dataset to ensure consistency between data coming from the WB and GBD datasets.

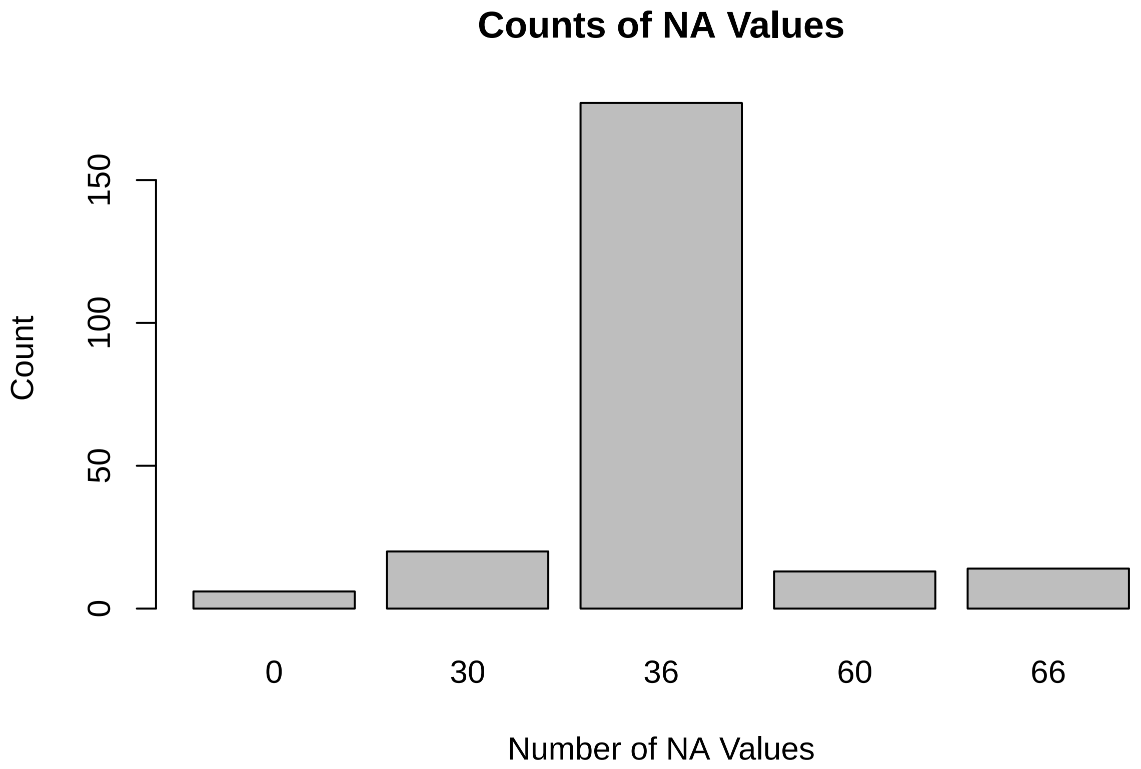
Now the data were ready to address missing data, either by removing unusable rows or by performing imputation. This consisted of three different approaches:

1. GDP per capita, years of compulsory education and total population were available overall, but not by sex. This overall value was used to fill missing values in "male" and "female".
2. Countries with no data at all on suicide rates were removed from the dataset.
3. For all over variables, missing data were replaced by manifestation of the *sex* variable. In a first step, the mean for the country and year was used to replace missing values. Where missing values remained, these were replaced by the mean for the country overall, then for the region, and where missing values still remained, they were replaced by the mean for the continent.

```
# First step: Replace missing values by male and female with overall value

## Filter out values for sex == "Both"
tmp_wb_gbd_wide.both <- tmp_wb_gbd_wide %>%
  filter(sex == "Both") %>%
  select(!sex)

## Join the data, making sure that country and year are matched correctly.
tmp_wb_gbd_wide.tmp2 <-
  left_join(x = tmp_wb_gbd_wide, y = tmp_wb_gbd_wide.both,
            by = c("country", "year"),
            suffix = c("", ".y"))


# Replacing missing data in original variable with overall value.
tmp_wb_gbd_wide <- tmp_wb_gbd_wide.tmp2 %>%
  plyr::mutate(sui = coalesce(sui, sui.y),
               gdp_pc = coalesce(gdp_pc, gdp_pc.y),
               edu = coalesce(edu, edu.y),
               pop_t = coalesce(pop_t, pop_t.y))


# Remove columns with .y suffix
columns_to_remove <- grep("\\.y", names(tmp_wb_gbd_wide))
tmp_wb_gbd_wide <- tmp_wb_gbd_wide[,-columns_to_remove]
```

After this operation, we have data for 230 countries.

```
# Second step: Drop countries with no data on suicide rates
# Counting missing values on suicide rates by country

tmp_na_count <- tmp_wb_gbd_wide %>%
  group_by(country) %>%
  dplyr::summarize(count_na = sum(is.na(sui)))

# Count the occurrences of each value in count_na
tmp_counts <- table(tmp_na_count$count_na)

# create a bar plot of the counts
barplot(tmp_counts, xlab = "Number of NA Values", ylab = "Count", main = "Counts of NA Values")
```

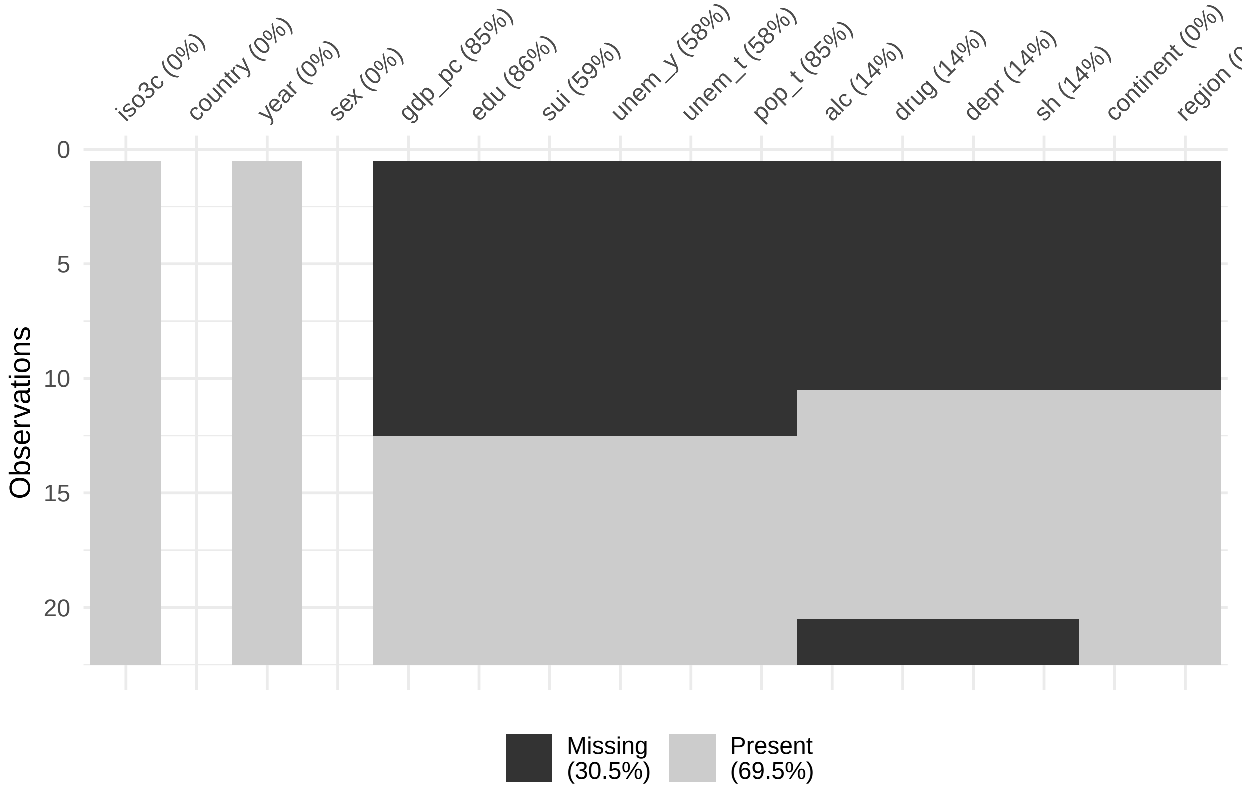


After some investigation to confirm that these countries really had no data on the suicide rate, I generated a list of countries that were missing 60 or 66 instances of the suicide rate and dropped them from the dataset.

We now have 216 countries in the dataset.

Prior to performing the imputation in the third and final step, I visualised the distribution of missing values using the *vis_miss* library.

```
# Group the data by country
tmp_df_by_country <- tmp_wb_gbd_wide %>%
  group_by(country, sex) %>%
  arrange(year)

# Create a heatmap of missing data using vis_miss()
vis_miss(tmp_df_by_country)
```



Most data are missing in the economic indicators and mental health data obtained from the GBD dataset. Data were then imputed to address this.

```r
# Impute missing values using median by group (country, year).

na1 <- tmp_wb_gbd_wide %>% is.na() %>% colSums()

tmp_df_imputed <- tmp_wb_gbd_wide %>%
  group_by(country, year) %>%
  mutate_if(is.numeric, na.aggregate, FUN = median) %>%
  ungroup()

na2 <- tmp_df_imputed %>% is.na() %>% colSums()
```

Before the first imputation, there were 6.4518×10^4 missing values in the dataset. After the imputation, there were 5.2854×10^4 missing values in the dataset.

```r
# By country only

tmp_df_imputed2 <- tmp_df_imputed %>%
  group_by(country) %>%
  mutate_if(is.numeric, na.aggregate, FUN = median) %>%
  ungroup()

na3 <- tmp_df_imputed2 %>% is.na() %>% colSums()
```

After the imputation, there were 1.1082×10^4 missing values in the dataset.

```r
# Repeating imputation by region

tmp_df_imputed3 <- tmp_df_imputed2 %>%
  group_by(region) %>%
  mutate_if(is.numeric, na.aggregate, FUN = median) %>%
  ungroup()

na4 <- tmp_df_imputed3 %>% is.na() %>% colSums()
```

After the imputation, there were 744 missing values in the dataset.

```r
# And finally by continent

tmp_df_imputed4 <- tmp_df_imputed3 %>%
  group_by(continent) %>%
  mutate_if(is.numeric, na.aggregate, FUN = median) %>%
  ungroup()

na5 <- tmp_df_imputed4 %>% is.na() %>% colSums()
```

After the final imputation, there were 480 missing values in the dataset.

Final dataset

The final dataset was saved in wide format (*suicide_final*) and in long format (*suicide_final_long*). Final preparations included

- Adding a variable of *date* type, based on the variable *year*, to facilitate the use of time series functions;
- Adding a calculated column showing total estimated deaths from suicide (*deaths*);
- Labelling variables;

```r
# Final preparations
suicide_final <- tmp_df_imputed4

# Create year1 variable in date type to allow for time series analysis

suicide_final$year1 <- ymd(paste0(suicide_final$year, "-01-01"))
str(suicide_final$year1)
```

```
##  Date[1:13242], format: "2012-01-01" "2012-01-01" "2012-01-01" "2013-01-01" "2013-01-01" ...
```

```r
# Adding calculated column: total suicide deaths.
# Assuming population is 50/50 male / female.
suicide_final <- suicide_final %>%
  mutate(deaths = case_when(
    sex %in% c("Male", "Female") ~ sui * pop_t/100000 * 0.5,
    TRUE ~ sui * pop_t/100000
  ))
```

```r
# Generating long dataset
indicators <- c("gdp_pc", "edu", "sui_female", "sui_male", "sui", "deaths", "unem_y_female", "unem_y_mal
suicide_final_long <- suicide_final %>%
  pivot_longer(cols = c(5:14),
               names_to = "indicator",
               values_to = "value")


# Labelling wide dataset
suicide_final = apply_labels(suicide_final,
                             iso3c = "ISO3C",
                             country = "Country",
                             year = "Year",
                             gdp_pc = "GDP per capita (constant 2015 US$)",
                             edu = "Compulsory education, duration (years)",
                             sui = "Suicide mortality rate (per 100,000 population)",
                             unem_y = "Unemployment, youth total (% of total labor force ages 15-24) (mo
                             unem_t = "Unemployment, total (% of total labor force) (modeled ILO estimat
                             pop_t = "Total population",
                             alc = "Alcohol abuse",
                             drug = "Drug abuse",
                             sh = "Self harm",
                             sex = "Sex",
                             depr = "Depression",
                             continent = "Continent",
                             region = "Sub-region",
                             deaths = "Deaths by suicide",
                             year1 = "Year in date format")

# Labelling long dataset
suicide_final_long = apply_labels(suicide_final_long,
                                  iso3c = "ISO3C",
                                  country = "Country",
                                  year = "Year",
                                  sex = "Sex",
                                  depr = "Depression",
                                  continent = "Continent",
                                  region = "Sub-region",
                                  year1 = "Year in date format",
                                  indicator = "Indicator",
                                  values = "Value"
                                  )
```

Finally, I write the data to disk to preserve it.

```r
write_xlsx(suicide_final,"../data/suicide_final.xlsx")
write_xlsx(suicide_final_long,"../data/suicide_final_long.xlsx")

# Removing temporary objects
rm(list = ls()[grep("^tmp_", ls())])
```

Exploratory data analysis

Summarising the dataset

An overview of the variables in the dataset is provided using the *table1* function. Note that this is filtered for sex == "Both", but shows the average of data for all countries and years.

```
# tmp_both <- suicide_final %>% filter(sex == "Both")
# table1 <- table1(~ sui + deaths + pop_t + edu + gdp_pc + unem_y + unem_t + alc + drug + depr + sh | f
#
# table1
```

To show the top 20 countries with the highest average suicide rates at any point, I calculated the average by country and sex and sort by the mean:

```
# Filtering
sui_country_time <- suicide_final %>%
  group_by(country, sex) %>%
  summarise(mean_sui=mean(sui),
            .groups = 'drop')

# Sorting
sui_country_time_sorted <- sui_country_time[order(sui_country_time$mean_sui, decreasing = TRUE),]

print(sui_country_time_sorted, n = 20)
```

```
## # A tibble: 648 x 3
##    country                sex        mean_sui
##    <labelled>             <labelled>    <dbl>
##  1 Lesotho                Male         102.
##  2 Lesotho                Both          79.8
##  3 Lesotho                Female        58.3
##  4 Guyana                 Male          48.3
##  5 Eswatini               Male          45.4
##  6 Lithuania              Male          42.2
##  7 Korea, Rep.            Male          40.0
##  8 Russian Federation     Male          39.7
##  9 Kiribati               Male          38.8
## 10 Guyana                 Both          38.4
## 11 Micronesia, Fed. Sts. Male          33.8
## 12 Eswatini               Both          33.0
## 13 Belarus                Male          32.2
## 14 Lithuania              Both          31.3
## 15 Suriname               Male          31.1
## 16 South Africa           Male          30.5
## 17 Ukraine                Male          29.9
## 18 Russian Federation     Both          29.6
## 19 Kiribati               Both          29.4
## 20 Korea, Rep.            Both          28.5
## # ... with 628 more rows
```

We can see that all 20 highest values are for males.

```
# Max only for males
max_m <- (head(sui_country_time_sorted %>% filter(sex == "Male"), n=1))[1]

# Min only for males
min_m <- tail(sui_country_time_sorted %>% filter(sex == "Male" & mean_sui > 0), n=1)

# Value for Switzerland
swiss_m <- sui_country_time_sorted %>% filter(sex == "Male" & country == "Switzerland")
```

The highest value is for the country of Lesotho, the lowest for the country of Barbados, Male, 0.713636363636364. For comparison, the value for males in Switzerland is Switzerland, Male, 17.1454545454545. These three countries are chosen for the case study comparisons.

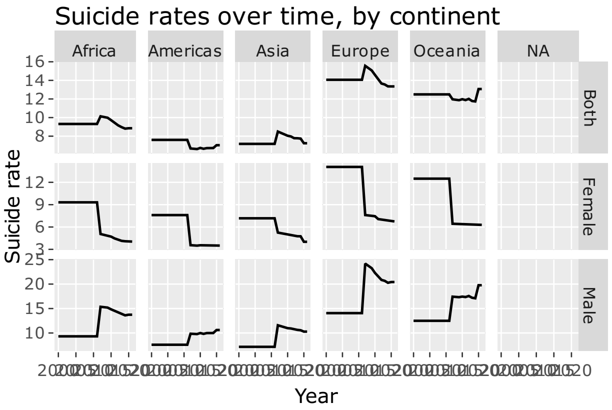
Exlploring data by continent

Due to the many countries in the dataset, visual analysis is challenging. An overview is provided by showing the average data by continent.

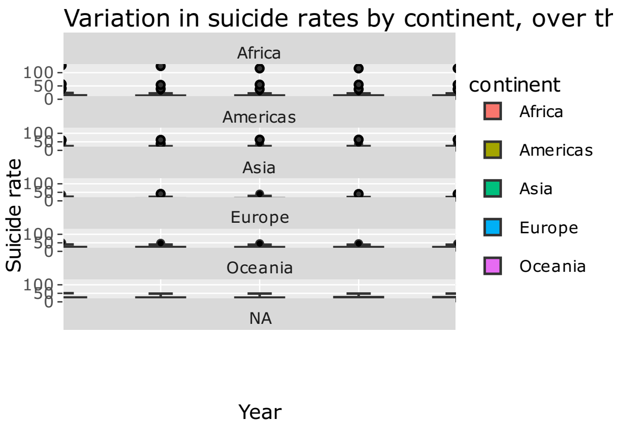
- A line graph shows trends in suicide rates over time.
- Box plots for the last 3 years show the distribution of the variable, to give an idea of the variability within continents.
- A cumulative graph shows total deaths by suicide.

I used the *plotly* package with *ggplot2* to make the graphs more interactive.

```
# Summarise suicide data by continent

sui_continent <- suicide_final %>%
  group_by(continent, sex, year1) %>%
  summarise(mean_sui=mean(sui), mean_deaths = mean(deaths),
            .groups = 'drop')


# Line graph: suicide rates over time
m_c <- sui_continent %>% ggplot(aes(x=year1,y=mean_sui)) +
  geom_line() +
    facet_grid(sex~continent, scales = "free_y") +
  ylab("Suicide rate") +
  xlab("Year") +
  ggtitle("Suicide rates over time, by continent")
m_c <- ggplotly(m_c)
m_c
```



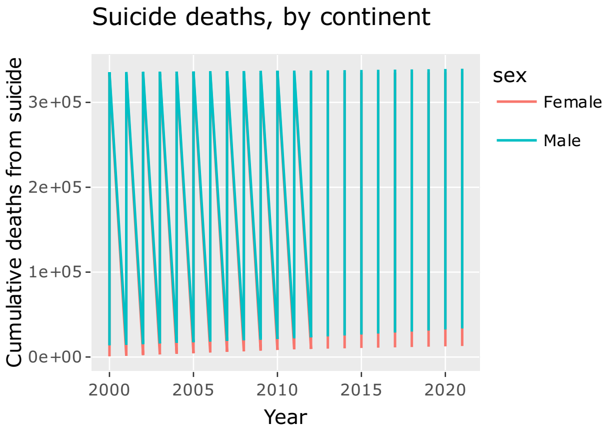
```r
# limiting to last three years
p <- suicide_final %>% filter(sex == "Male", year > 2016) %>%
  ggplot(aes(x=year, y=sui, fill=continent)) +
  geom_boxplot() +
  facet_wrap(~continent, ncol = 1) +
  xlab("Year") +
  ylab("Suicide rate") +
  ggtitle("Variation in suicide rates by continent, over the last 3 years")

p <- ggplotly(p)
p
```



```r
# Suicide deaths by continent
q <- sui_continent %>% filter(sex != "Both") %>%
  ggplot(aes(x = year1, y=cumsum(mean_deaths), colour = sex)) +
  geom_line() +
  ggtitle("Suicide deaths, by continent") +
  xlab("Year")+
  ylab("Cumulative deaths from suicide")
  facet_wrap(~ continent, scales = "free_y")
```

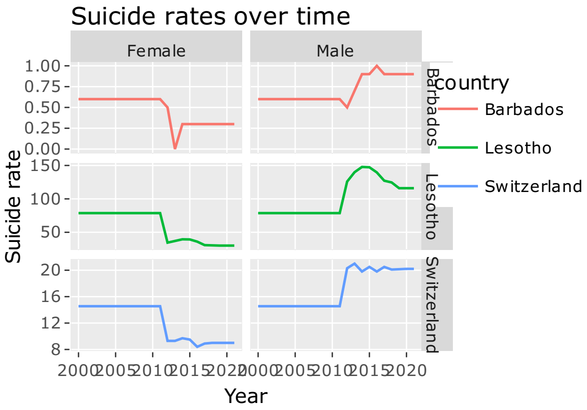
```
## <ggproto object: Class FacetWrap, Facet, gg>
##     compute_layout: function
##     draw_back: function
##     draw_front: function
##     draw_labels: function
##     draw_panels: function
##     finish_data: function
##     init_scales: function
##     map_data: function
##     params: list
##     setup_data: function
##     setup_params: function
##     shrink: TRUE
##     train_scales: function
##     vars: function
##     super:  <ggproto object: Class FacetWrap, Facet, gg>
```

```
q <- ggplotly(q)
q
```

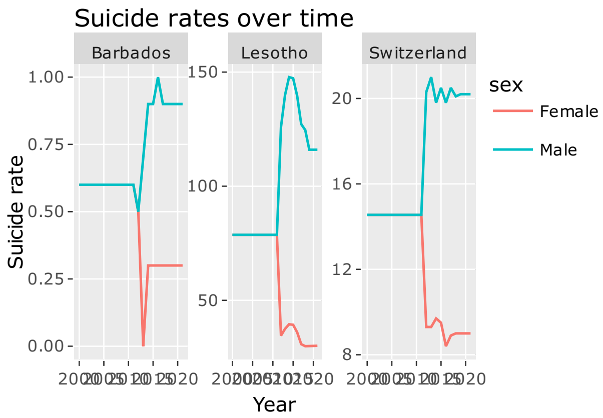


In a case study, I compare Switzerland to the countries with the highest and lowest male suicide rates.

```
# Creating filtered dataset
cs_countries <- c("Switzerland", "Lesotho", "Barbados")
case_study <- suicide_final %>%
  filter(country %in% cs_countries, sex != "Both")

case1 <- ggplot(case_study, aes(x = year, y = sui, color = country)) +
  geom_line() +
  facet_grid(country ~ sex, scales = "free_y") +
  ylab("Suicide rate") +
  xlab("Year") +
  ggtitle("Suicide rates over time")

case1 <- ggplotly(case1)
case1
```



```
#' suicide rate over time plus cumulative deaths

#+ case2

# Multiple lines in the same plot
case2 <- case_study %>%
  ggplot(mapping = aes(y = sui, x=year, colour = sex)) +
  geom_line() +
  facet_wrap(~ country, scales = "free_y")+
  ylab("Suicide rate") +
  xlab("Year") +
  ggtitle("Suicide rates over time")

case2 <- ggplotly(case2)
case2
```



Model

Regression?

Special chapter

Heat map or shiny R

Conclusions