PresentationShiny

Christiana Reichsthaler

3/12/2018

```
library(gapminder)
```

```
## Warning: package 'gapminder' was built under R version 3.4.2
```

```
library(tibble)
library(stringr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(knitr)
library(forcats)
library(readxl)
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'tidyr' was built under R version 3.4.3
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## as.difftime(): lubridate, base
## date():        lubridate, base
## filter():      dplyr, stats
## intersect():   lubridate, base
## lag():         dplyr, stats
## setdiff():     lubridate, base
## union():       lubridate, base
```

```
library(dplyr)
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 3.4.3
```

```
library(pacman)
if(!file.exists("./data")) {dir.create("./data")}

# Import of Data and cleaning up of the Data
TourismData <- "http://api.worldbank.org/v2/en/indicator/ST.INT.ARVL?downloadformat=excel"
download.file(TourismData, destfile = "./data/tmp.xls", mode = "wb")
TourismData <- read_excel("./data/tmp.xls")
```

```
TourismData1 <- select(TourismData, "Data Source", "World Development Indicators", "X__38", "X__39", "X_
names(TourismData1) <- c("country", "country_code", "year1995", "1996", "1997", "1998", "1999", "year20
TourismData1 = TourismData1 [-1:-4,]

TourismData1 <- TourismData1[-c(5, 34,36, 47, 59, 60, 61, 62, 63, 66, 72, 93, 96, 100, 101, 102, 103, 1

TourismData1$year1995 <- as.numeric(TourismData1$year1995)
TourismData1$year2000 <- as.numeric(TourismData1$year2000)
TourismData1$year2005 <- as.numeric(TourismData1$year2005)
TourismData1$year2010 <- as.numeric(TourismData1$year2010)
TourismData1$year2016 <- as.numeric(TourismData1$year2016)

TourismData1
```

```
## # A tibble: 218 x 25
##                 country country_code year1995  `1996`  `1997`  `1998`
##                   <chr>        <chr>    <dbl>   <chr>   <chr>   <chr>
##  1          Afghanistan          AFG       NA    <NA>    <NA>    <NA>
##  2               Angola          AGO     9000   21000   45000   52000
##  3              Albania          ALB   304000  287000  119000  184000
##  4              Andorra          AND       NA    <NA>    <NA>    <NA>
##  5 United Arab Emirates          ARE  2315000 2572000 2476000 2991000
##  6            Argentina          ARG  2289000 2614000 2764000 3012000
##  7              Armenia          ARM    12000   13000   23000   32000
##  8       American Samoa          ASM    34000   35000   26000   36000
##  9  Antigua and Barbuda          ATG   220000  228000  240000  234000
## 10            Australia          AUS  3726000 4165000 4318000 4167000
## # ... with 208 more rows, and 19 more variables: `1999` <chr>,
## #   year2000 <dbl>, `2001` <chr>, `2002` <chr>, `2003` <chr>,
## #   `2004` <chr>, year2005 <dbl>, `2006` <chr>, `2007` <chr>,
## #   `2008` <chr>, `2009` <chr>, year2010 <dbl>, `2011` <chr>,
## #   `2012` <chr>, `2013` <chr>, `2014` <chr>, `2015` <chr>,
## #   year2016 <dbl>, `2017` <chr>
```

```
#Project Plan
#I downloaded data on international arrivals from 1995 to 2017 from a website called "Our World in Data

#Timeline: I already cleaned up the data and I hope to create the shiny app this weekend.
#I will be working alone

#My Plan is to mainly look at the years below and see what the increase/decrease is over a period of 5

TourismData1 %>%
select("country","year1995", "year2000", "year2005", "year2010", "year2016") %>%
filter(country == "United States")
```

```
## # A tibble: 1 x 6
##         country year1995 year2000 year2005 year2010 year2016
##           <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 United States 43318000 51238000 49206000 60010000 75608000
```

```
Selectcountries <- TourismData1 %>%
select(country,"year1995", "year2000", "year2005", "year2010", "year2016") %>%
filter(country %in% c("France", "Ukraine", "Bahamas, The", "Syrian Arab Republic"))
Selectcountries
```

```
## # A tibble: 4 x 6
##                country year1995 year2000 year2005 year2010 year2016
##                  <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1         Bahamas, The  1598000  1544000  1608000  1370000  1482000
## 2               France 60033000 77190000 74988000 76647000 82570000
## 3 Syrian Arab Republic   815000  2100000  3571000  8546000       NA
## 4              Ukraine  3716000  6431000 17631000 21203000 13333000
```

```
MostVisited <- TourismData1 %>%
select(country,year2016) %>%
  arrange(desc(year2016))
print(MostVisited, n=218)
```

```
## # A tibble: 218 x 2
##                            country   year2016
##                              <chr>      <dbl>
##   1                          World 1244960000
##   2                 European Union  482410000
##   3                         France   82570000
##   4                  United States   75608000
##   5                          Spain   75315000
##   6                          China   59270000
##   7                          Italy   52372000
##   8                 United Kingdom   35814000
##   9                        Germany   35555000
##  10                         Mexico   35079000
##  11                       Thailand   32530000
##  12                         Turkey   30289000
##  13                        Austria   28121000
##  14                       Malaysia   26757000
##  15           Hong Kong SAR, China   26553000
##  16                         Greece   24799000
##  17             Russian Federation   24571000
##  18                          Japan   24040000
##  19                     South Asia   21293000
##  20                         Canada   19824000
##  21                   Saudi Arabia   18049000
##  22                         Poland   17471000
##  23                    Korea, Rep.   17242000
##  24                    Netherlands   15828000
##  25               Macao SAR, China   15703600
##  26                          India   14569000
##  27                        Croatia   13809000
##  28                        Ukraine   13333000
##  29                      Singapore   12914000
##  30                      Indonesia   11519000
##  31                       Portugal   11223000
##  32                        Denmark   10781000
##  33                        Morocco   10332000
##  34                        Romania   10223000
##  35                        Bahrain   10158000
##  36                        Ireland   10100000
##  37                   South Africa   10044000
##  38                        Vietnam   10013000
##  39                        Belarus    9423500
##  40                 Czech Republic    9321000
##  41                    Switzerland    9205000
##  42                      Australia    8263000
##  43                       Bulgaria    8252000
##  44                        Belgium    7481000
##  45                         Sweden    6782000
##  46                         Brazil    6578000
##  47                     Kazakhstan    6509000
##  48                        Georgia    6361000
##  49                    Philippines    5967000
##  50                         Norway    5960000
##  51             Dominican Republic    5959300
##  52                        Tunisia    5724000
##  53                          Chile    5641000
##  54                      Argentina    5559000
##  55                        Hungary    5302000
##  56               Egypt, Arab Rep.    5258000
##  57                       Cambodia    5012000
##  58             Iran, Islamic Rep.    4942000
##  59                        Albania    4070000
##  60                           Cuba    3968000
##  61                         Jordan    3858000
##  62                           Peru    3744000
##  63                    Puerto Rico    3736000
##  64                    New Zealand    3370000
##  65                       Colombia    3317000
##  66                        Lao PDR    3315000
##  67                         Cyprus    3187000
##  68                        Estonia    3147000
##  69                        Uruguay    3037000
##  70                       Slovenia    3032000
##  71                          Qatar    2938200
##  72                Kyrgyz Republic    2930000
##  73                     Costa Rica    2925000
##  74                        Myanmar    2907000
##  75                         Israel    2900000
##  76                        Andorra    2831000
##  77                        Finland    2789000
##  78                      Lithuania    2296000
##  79                           Oman    2292000
##  80                        Jamaica    2182000
##  81                       Zimbabwe    2168000
##  82                      Sri Lanka    2051000
##  83                     Azerbaijan    2044000
##  84                        Algeria    2039000
##  85                Slovak Republic    2027000
##  86                         Panama    2007000
##  87                          Malta    1966000
##  88                      Guatemala    1906000
##  89                        Nigeria    1889000
##  90                         Latvia    1793000
##  91                        Iceland    1792000
##  92                        Lebanon    1688000
##  93                     Montenegro    1662000
##  94                     Mozambique    1639000
##  95                  Cote d'Ivoire    1583000
##  96                           Guam    1535000
##  97                      Nicaragua    1504000
##  98                   Bahamas, The    1482000
##  99                        Namibia    1469000
## 100                    El Salvador    1434000
## 101                        Ecuador    1418000
## 102                         Uganda    1323000
## 103                       Paraguay    1308000
## 104                       Maldives    1286000
## 105                         Serbia    1281000
## 106                      Mauritius    1275000
## 107                          Kenya    1268000
## 108                        Armenia    1260000
## 109                       Tanzania    1233000
## 110                        Lesotho    1196000
## 111                     Luxembourg    1054000
## 112                        Bolivia     959000
## 113                         Zambia     956000
## 114                      Swaziland     947000
## 115                         Rwanda     932000
## 116                       Ethiopia     871000
## 117                         Malawi     849000
## 118                           Fiji     792000
## 119         Bosnia and Herzegovina     777000
## 120                          Nepal     753000
## 121                       Barbados     632000
## 122                  Venezuela, RB     601000
## 123                     Cabo Verde     598000
## 124       Northern Mariana Islands     531000
## 125      Sint Maarten (Dutch part)     528000
## 126                 Macedonia, FYR     510000
## 127       Turks and Caicos Islands     454000
## 128                        Curacao     441000
## 129            Trinidad and Tobago     410000
## 130         British Virgin Islands     408000
## 131                       Mongolia     404000
## 132              West Bank and Gaza     400000
## 133                         Angola     397000
## 134                         Belize     386000
## 135                 Cayman Islands     385000
## 136                      St. Lucia     348000
## 137                           Togo     338000
## 138                         Monaco     336000
## 139                     Seychelles     303000
## 140                     Madagascar     293000
## 141                          Benin     267000
## 142            Antigua and Barbuda     265000
## 143                       Suriname     256000
## 144                        Bermuda     244000
## 145                         Guyana     235000
## 146                    Congo, Rep.     224000
## 147              Brunei Darussalam     219000
## 148                         Bhutan     210000
## 149               French Polynesia     192000
## 150                        Burundi     187000
## 151                           Mali     173200
## 152                    Gambia, The     161000
## 153                        Grenada     156000
## 154                   Burkina Faso     152000
## 155                          Niger     152000
## 156                        Eritrea     142000
## 157                          Palau     138000
## 158                          Samoa     134000
## 159                        Moldova     121000
## 160                  New Caledonia     116000
## 161                        Vanuatu      95100
## 162 St. Vincent and the Grenadines      79000
## 163                       Dominica      78000
## 164                    Timor-Leste      66000
## 165                  Liechtenstein      60000
## 166                     San Marino      60000
## 167                          Tonga      59100
## 168                   Sierra Leone      55000
## 169          Micronesia, Fed. Sts.      29600
## 170          Sao Tome and Principe      29000
## 171                        Comoros      26800
## 172                Solomon Islands      23200
## 173                 American Samoa      20100
## 174               Marshall Islands       9800
## 175                       Kiribati       5700
## 176                         Tuvalu       2500
## 177                    Afghanistan         NA
## 178           United Arab Emirates         NA
## 179                     Bangladesh         NA
## 180                       Botswana         NA
## 181       Central African Republic         NA
## 182                       Cameroon         NA
## 183               Congo, Dem. Rep.         NA
## 184                       Djibouti         NA
## 185                  Faroe Islands         NA
## 186                          Gabon         NA
## 187                          Ghana         NA
## 188                      Gibraltar         NA
## 189                         Guinea         NA
## 190                  Guinea-Bissau         NA
## 191              Equatorial Guinea         NA
## 192                      Greenland         NA
## 193                       Honduras         NA
## 194                          Haiti         NA
## 195                    Isle of Man         NA
## 196                           Iraq         NA
## 197            St. Kitts and Nevis         NA
## 198                         Kuwait         NA
## 199                        Liberia         NA
## 200                          Libya         NA
## 201       St. Martin (French part)         NA
## 202                     Mauritania         NA
## 203                          Nauru         NA
## 204                       Pakistan         NA
## 205               Papua New Guinea         NA
## 206      Korea, Dem. People’s Rep.         NA
## 207                          Sudan         NA
## 208                        Senegal         NA
## 209                        Somalia         NA
## 210                    South Sudan         NA
## 211           Syrian Arab Republic         NA
## 212                           Chad         NA
## 213                     Tajikistan         NA
## 214                   Turkmenistan         NA
## 215                     Uzbekistan         NA
## 216          Virgin Islands (U.S.)         NA
## 217                         Kosovo         NA
## 218                    Yemen, Rep.         NA
```

```
Graph <- TourismData1 %>%
  select(country,year2016) %>%
  arrange(desc(year2016)) %>%
  group_by(year2016)


LeastVisited <- TourismData1 %>%
select(country, year2016) %>%
arrange(year2016)
  print(LeastVisited, n=10)
```

```
## # A tibble: 218 x 2
##                   country year2016
##                     <chr>    <dbl>
##  1                 Tuvalu     2500
##  2               Kiribati     5700
##  3       Marshall Islands     9800
##  4         American Samoa    20100
##  5        Solomon Islands    23200
##  6                Comoros    26800
##  7 Sao Tome and Principe    29000
##  8 Micronesia, Fed. Sts.    29600
##  9           Sierra Leone    55000
## 10                  Tonga    59100
## # ... with 208 more rows
```

```
HighestOverallIncrease <- TourismData1 %>%
  select(country, year1995, year2016) %>%
  mutate(indec = year2016 - year1995) %>%
  arrange(desc(indec))
print(HighestOverallIncrease, n=10)
```

```
## # A tibble: 218 x 4
##            country  year1995   year2016     indec
##              <chr>     <dbl>      <dbl>     <dbl>
##  1           World 524006000 1244960000 720954000
##  2 European Union 258870000  482410000 223540000
##  3          Spain  32971000   75315000  42344000
##  4          China  20034000   59270000  39236000
##  5 United States  43318000   75608000  32290000
##  6       Thailand   6952000   32530000  25578000
##  7         Turkey   7083000   30289000  23206000
##  8         France  60033000   82570000  22537000
##  9          Italy  31052000   52372000  21320000
## 10        Germany  14847000   35555000  20708000
## # ... with 208 more rows
```

```
HighestOverallDecrease <- TourismData1 %>%
  select(country, year1995, year2016) %>%
  mutate(indec = year2016 - year1995) %>%
  arrange(indec)
print(HighestOverallDecrease, n=10)
```

```
## # A tibble: 218 x 4
##                      country year1995 year2016    indec
##                        <chr>    <dbl>    <dbl>    <dbl>
##  1                   Poland 19215000 17471000 -1744000
##  2                  Eritrea   315000   142000  -173000
##  3 Northern Mariana Islands   676000   531000  -145000
##  4                  Bermuda   387000   244000  -143000
##  5             Bahamas, The  1598000  1482000  -116000
##  6            Venezuela, RB   700000   601000   -99000
##  7           American Samoa    34000    20100   -13900
##  8            Liechtenstein    59000    60000     1000
##  9                   Tuvalu      900     2500     1600
## 10                 Kiribati     3900     5700     1800
## # ... with 208 more rows
```

```
#What could be the reason behind this? eg. Venezuela: political/economic crisis in recent years

mean(TourismData1$year1995, na.rm=TRUE)
```

```
## [1] 7056151
```

```
mean(TourismData1$year2016, na.rm=TRUE)
```

```
## [1] 16726561
```

```
library(ggmap)
library(maptools)
```

```
## Loading required package: sp

## Warning: package 'sp' was built under R version 3.4.3

## Checking rgeos availability: TRUE
```

```
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

```r
visited <- c("USA", "Austria","Bahamas", "Costa Rica", "Jamaica", "Brazil", "Ireland", "UK", "France", 
ll.visited <- geocode(visited)
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=USA&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Austria&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Bahamas&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Costa%20Rica&sensor=

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Jamaica&sensor=false

## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Jamaica"

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Brazil&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Ireland&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=UK&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=France&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Spain&sensor=false

## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Belgium&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Netherlands&sensor=

## Warning: geocode failed with status OVER_QUERY_LIMIT, location =
## "Netherlands"

## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Luxemburg&sensor=fa

## Warning: geocode failed with status OVER_QUERY_LIMIT, location =
## "Luxemburg"

## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Italy&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Switzerland&sensor=

## Warning: geocode failed with status OVER_QUERY_LIMIT, location =
## "Switzerland"

## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lichtenstein&sensor
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Denmark&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Sweden&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Czech%20Republic&sens
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Croatia&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hungary&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Slovenia&sensor=fals

## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Slovenia"

## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Slovakia&sensor=fals
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Boasnia%20and%20Her
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Montenegro&sensor=fa
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Albania&sensor=false

## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Albania"

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Bulgaria&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Greece&sensor=false

## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Greece"
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Cyprus&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Turkey&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=United%20Arab%20Emira
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lebanon&sensor=fals
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Syria&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Isreal&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Palestine&sensor=fa
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Sri%20Lanka&sensor=f
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Malaysia&sensor=fals
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Indonesia&sensor=fal
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Taiwan&sensor=false
```

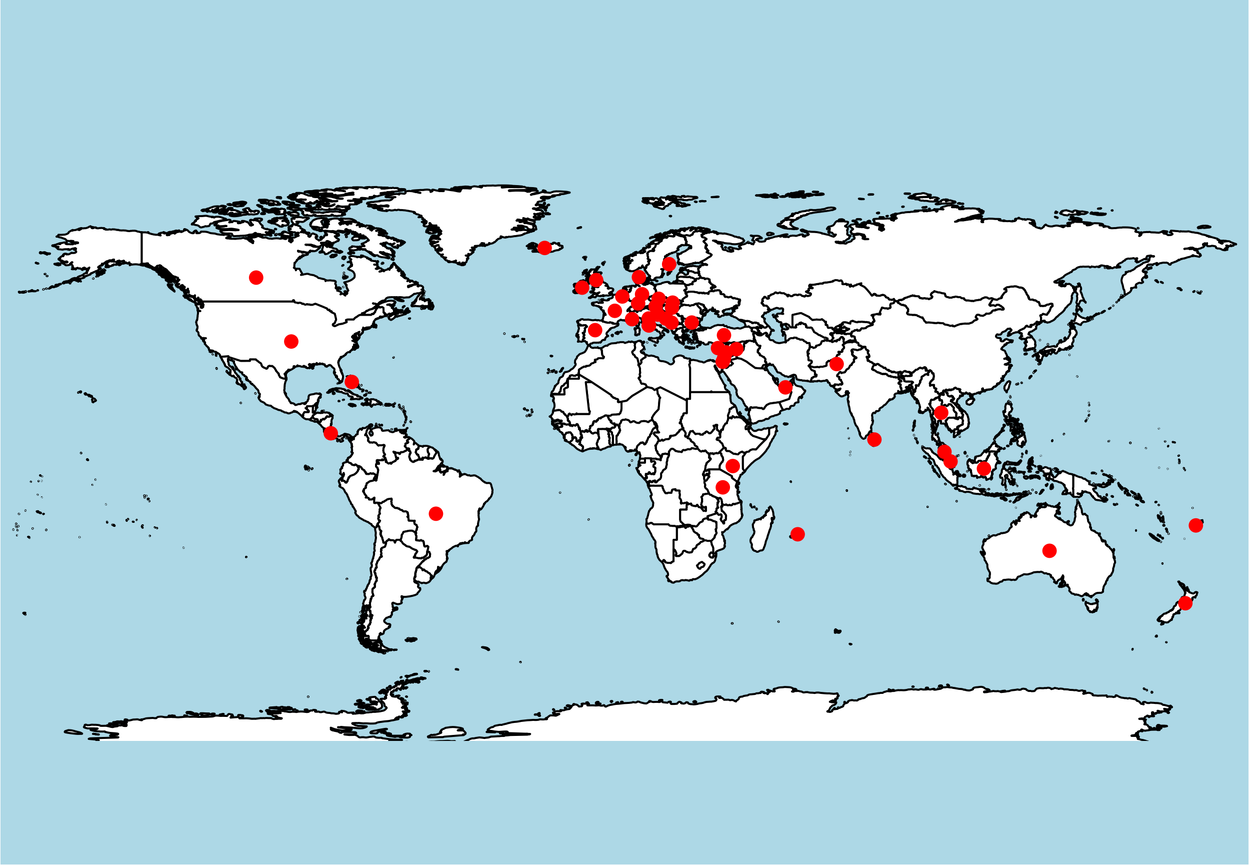
```
## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Taiwan"
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Singapore&sensor=fal
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Thailand&sensor=fals
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Mauritius&sensor=fa
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Kenya&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Tanzania&sensor=fal
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Australia&sensor=fa
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=New%20Zealand&senso
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Fiji&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Malta&sensor=false
```

```
## Warning: geocode failed with status OVER_QUERY_LIMIT, location = "Malta"
```

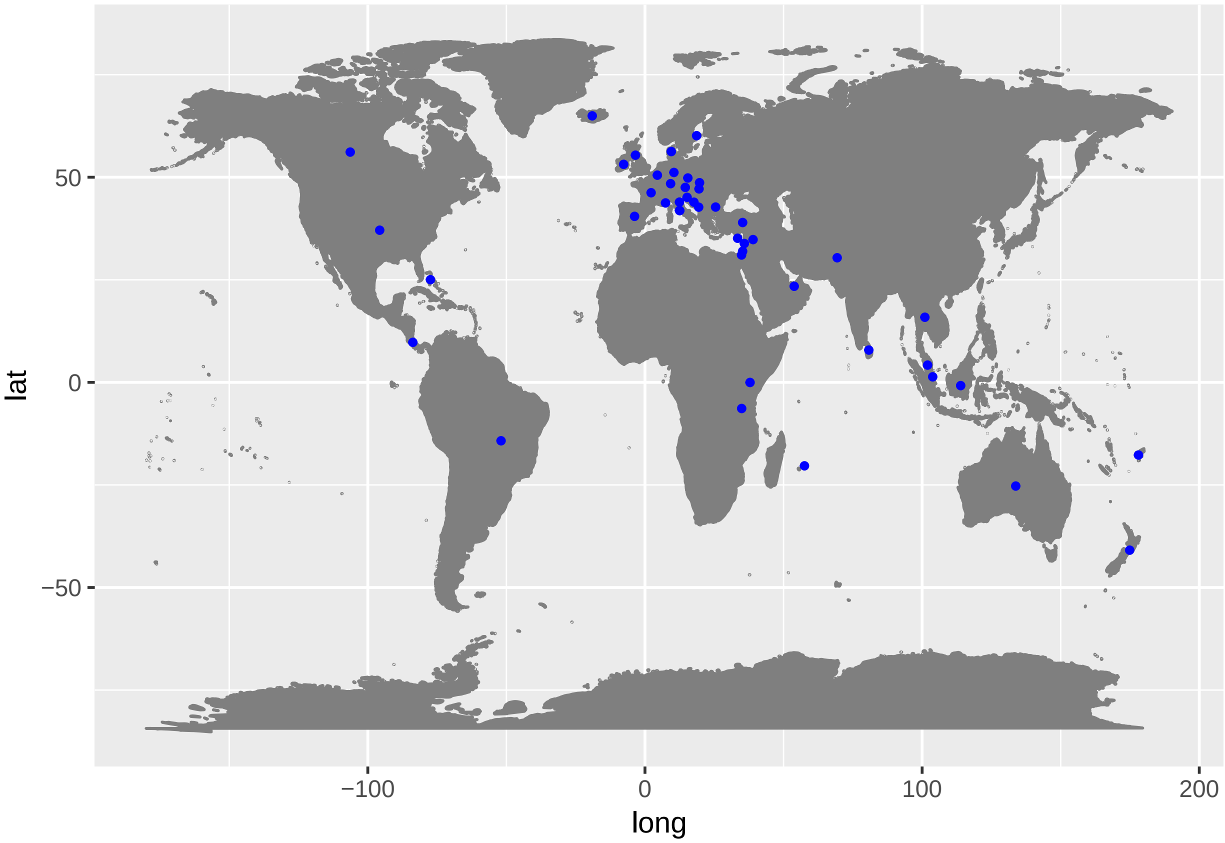
```
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Vatican%20City&sens
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=%20San%20Marino&sens
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Monaco&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Canada&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Iceland&sensor=fals
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Pakistan&sensor=fal
```

```r
# I created a map with all 55 countries that I have visited so far. The code is from https://www.r-blog
# I am also planning to use the code of the maps for something of the data above. Maybe show the top 10

visit.x <- ll.visited$lon
visit.y <- ll.visited$lat

map("world", fill=TRUE, col="white", bg="lightblue", ylim=c(-80, 100), mar=c(0,0,0,0))
points(visit.x,visit.y, col="red", pch=16)
```



```
mp <- NULL
mapWorld <- borders("world", colour="gray50", fill="gray50")
mp <- ggplot() +   mapWorld
mp <- mp+ geom_point(aes(x=visit.x, y=visit.y) ,color="blue", size=1)
mp
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



```
#Just an idea: not sure if I will use it in the presentation because I have not completely figured it o

Data1 <- select(TourismData, "Data Source", "World Development Indicators", "X__38", "X__39", "X__40", 
names(Data1) <- c("country", "country_code", "year1995", "year1996", "year1997", "year1998", "year1999"
Data1 = Data1 [-1:-4,]

Data1 <- Data1[-c(5, 34,36, 47, 59, 60, 61, 62, 63, 66, 72, 93, 96, 100, 101, 102, 103, 105, 108, 126, 

Data1$year1995 <- as.numeric(Data1$year1995)
Data1$year2000 <- as.numeric(Data1$year2000)
Data1$year2005 <- as.numeric(Data1$year2005)
Data1$year2010 <- as.numeric(Data1$year2010)
Data1$year2016 <- as.numeric(Data1$year2016)

gap <- gapminder

das <- gap %>%
  left_join(Data1,
            by = c("country")) %>%
select(country, country_code, year1997, year2002, year2007, continent, year, lifeExp, pop, gdpPercap) %
  rename(visitors = year2007)%>%
  mutate(gdp = pop * gdpPercap) %>%
  filter(year > 1995)
```

```
## Warning: Column `country` joining factor and character vector, coercing
## into character vector
```

```r
VisGdp <- das %>%
  select(country, visitors, continent, year, lifeExp, pop, gdpPercap, gdp) %>%
  filter(year == 2007) %>%
  arrange(visitors, gdp)
  print(VisGdp, n=5)
```

```
## # A tibble: 142 x 8
##        country visitors continent  year lifeExp      pop gdpPercap
##          <chr>    <chr>    <fctr> <int>   <dbl>    <int>     <dbl>
## 1      Lebanon  1017000      Asia  2007  71.993  3921278 10461.059
## 2      Albania  1062000    Europe  2007  76.423  3600523  5937.030
## 3  Netherlands 11008000    Europe  2007  79.762 16570613 36797.933
## 4       Panama  1103000  Americas  2007  75.537  3242173  9809.186
## 5 Saudi Arabia 11531000      Asia  2007  72.777 27601038 21654.832
## # ... with 137 more rows, and 1 more variables: gdp <dbl>
```