mid_project

Mykola_Signayevskyy

2023-02-28

install.packages("webshot") webshot::install_phantomjs()

```
library(MASS)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(leaflet)
```

```
source("/Users/mykola/Desktop/STAT515/third_lesson/hw.R")
```

```
Country <- c("Latvia", "Australia", "Scotland", "Peru", "South Africa", "India")
Height <- c(5.5, 5.4, 5.4, 5.4, 5.2, 5.0)
```

```
average_f_h <- data.frame(Country, Height)
average_f_h
```

Creating data set according to the original graph

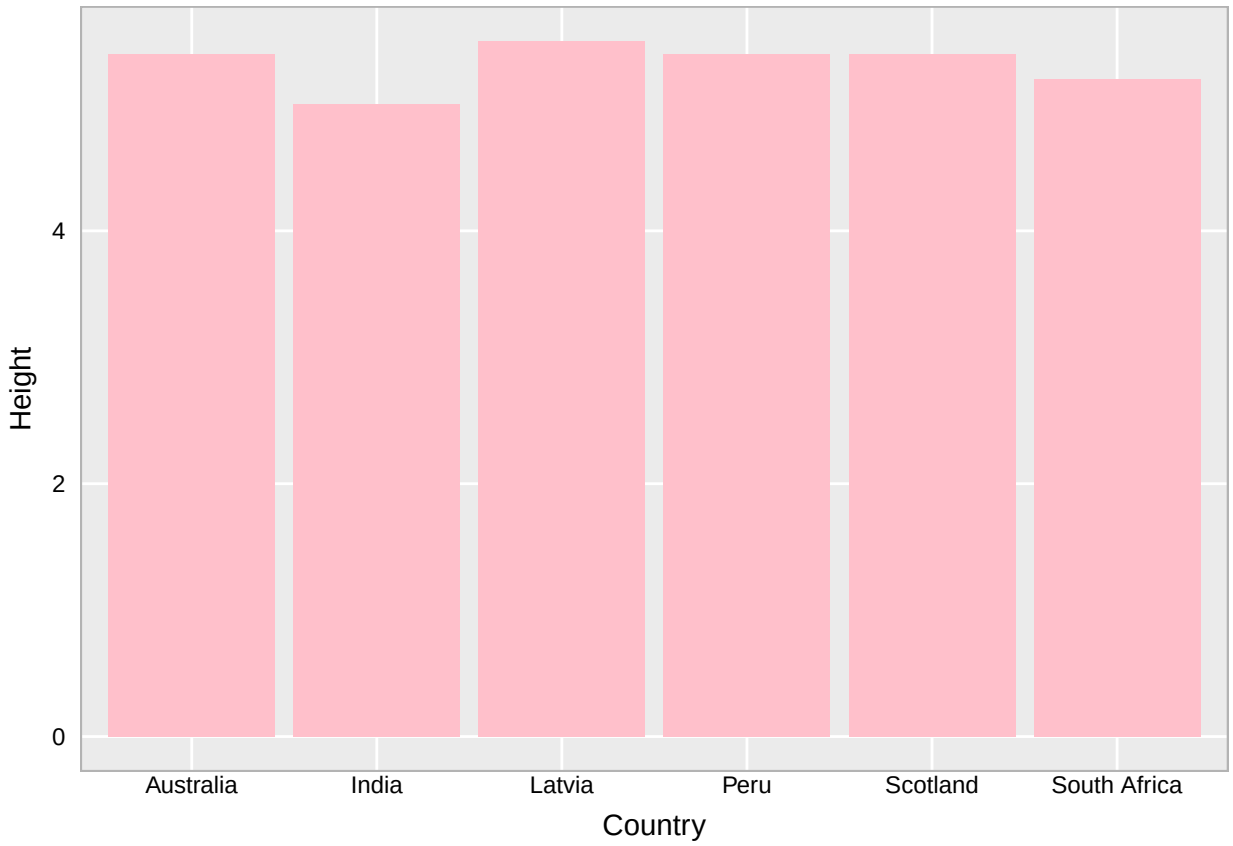
```
##        Country Height
## 1       Latvia    5.5
## 2    Australia    5.4
## 3     Scotland    5.4
## 4         Peru    5.4
## 5 South Africa    5.2
## 6        India    5.0
```

```
newdata <- average_f_h[order(-Height),] #ordering the data by height
newdata
```

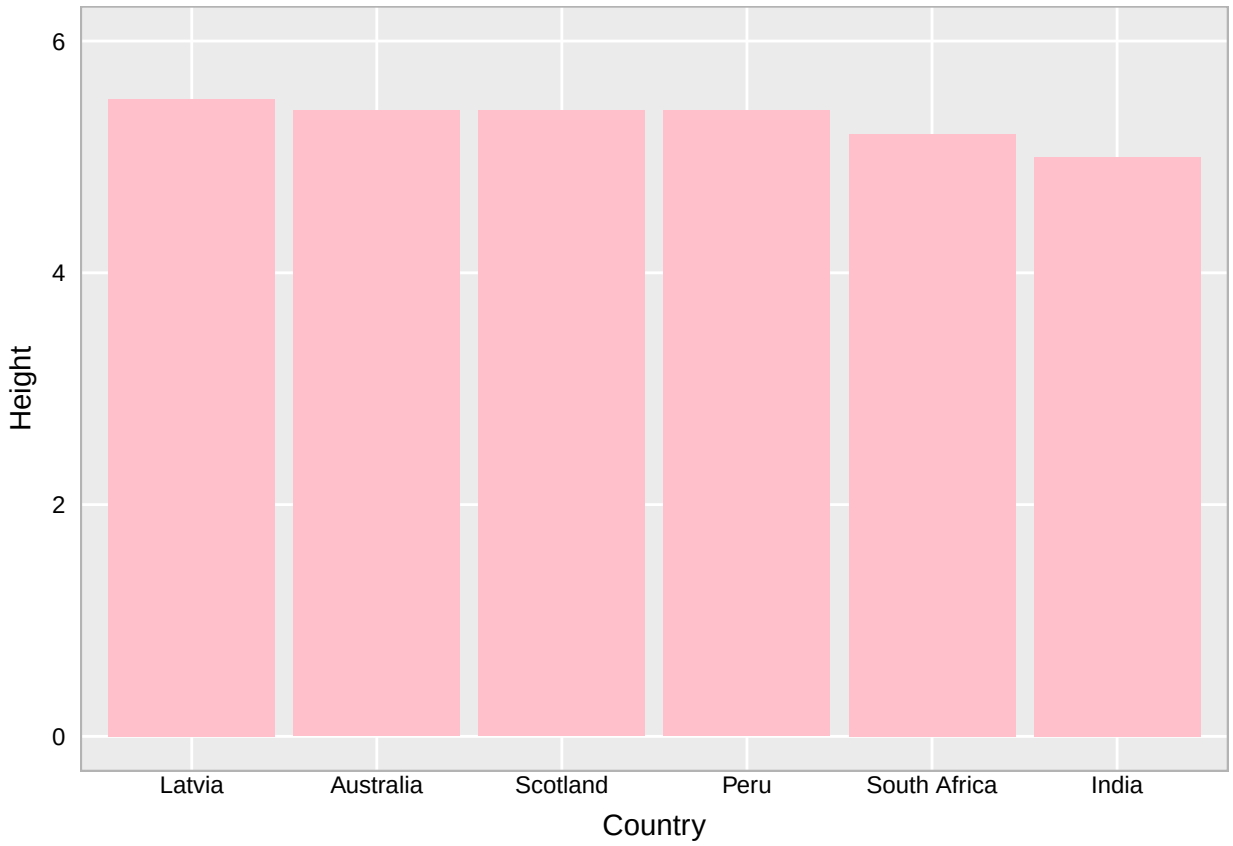
```
##        Country Height
## 1       Latvia    5.5
## 2    Australia    5.4
## 3     Scotland    5.4
## 4         Peru    5.4
## 5 South Africa    5.2
```

```
## 6        India    5.0
```

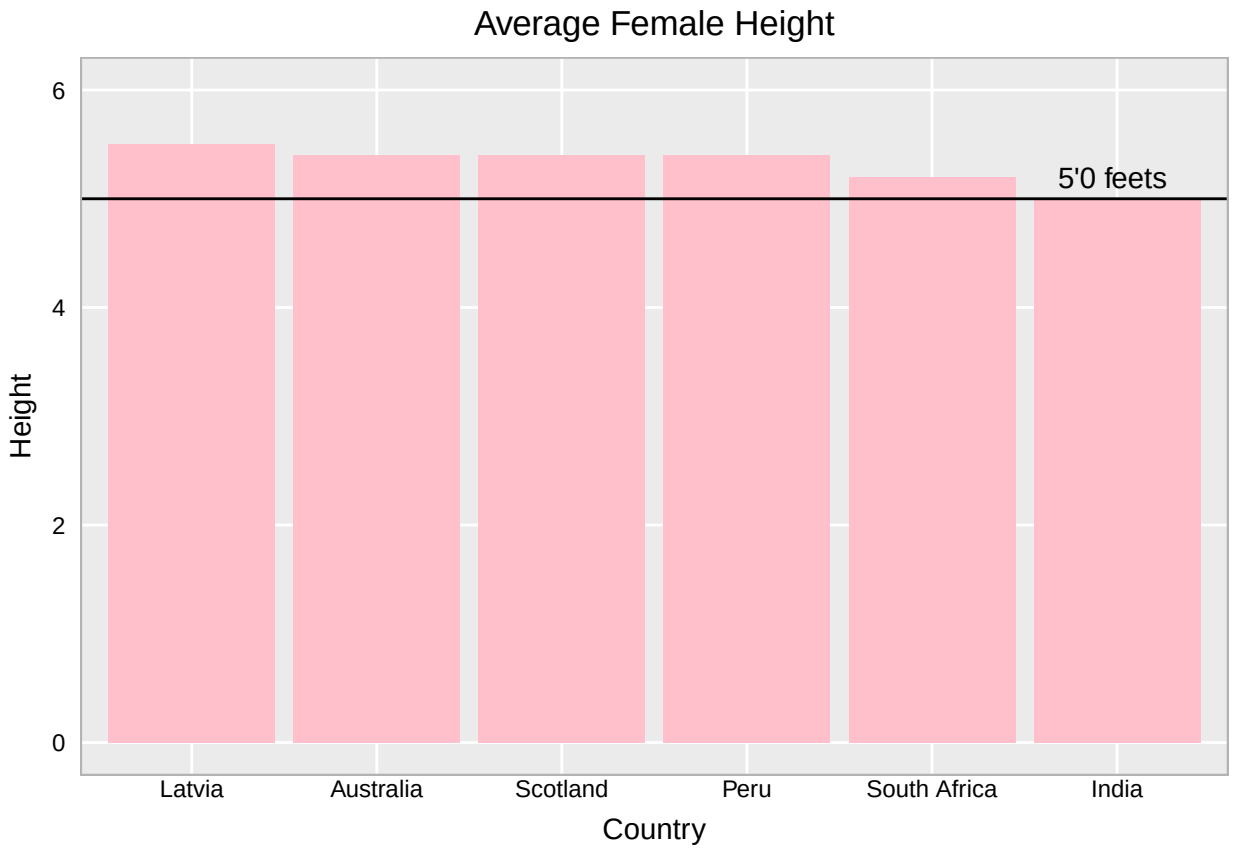
```
p <- ggplot(newdata, aes(x = Country, y = Height)) +
   geom_bar(stat = "identity", fill="pink") + hw
p
```




```
p1 <- p + ylim(0,6) + scale_x_discrete(limits = Country) #ordering bar charts
p1
```




```
p2 <- p1 + geom_hline(yintercept=5) + labs(x="Country",
       y="Height",
       title="Average Female Height")+ annotate("text", x=6, y=5.2, label= "5'0 feets ") + hw #adding l
p2
```



At some point we found the existing dataset with same data

```
height_data <- read_csv('/Users/mykola/Desktop/STAT515/mid_project/Height_data.csv')
```

```
## Rows: 199 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): country, cca3, cca2, region, subregion
## dbl (13): place, pop2023, growthRate, area, ccn3, landAreaKm, density, densi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
table(is.na(height_data)) #checking for a NA cells
```

```
##
## FALSE  TRUE
##  3581     1
```

```
height_data = na.omit(height_data) #removing NA rows
table(is.na(height_data))
```

```
##
## FALSE
##  3564
```

```
filtered_height <- height_data %>%
  select(country, region, meanHeightFemale, meanHeightMale, rank) %>%
```

```
  filter(country %in% c("Latvia", "Australia", "Scotland", "Peru", "South Africa", "India"))
head(filtered_height)
```

```
## # A tibble: 5 x 5
##   country      region        meanHeightFemale meanHeightMale  rank
##   <chr>        <chr>                    <dbl>          <dbl> <dbl>
## 1 Latvia       Europe                    169.           181.     7
## 2 Australia    Oceania                   165.           179.    29
## 3 South Africa Africa                    159.           170.   146
## 4 Peru         South America             154.           167.   177
## 5 India        Asia                      155.           166.   179
```

```
new_row <- c("Scotland", "Europe", 162.5000, 172.7200, 150)
filtered_height <- rbind(filtered_height, new_row)   #adding scotland data as it id not on the dataset
```

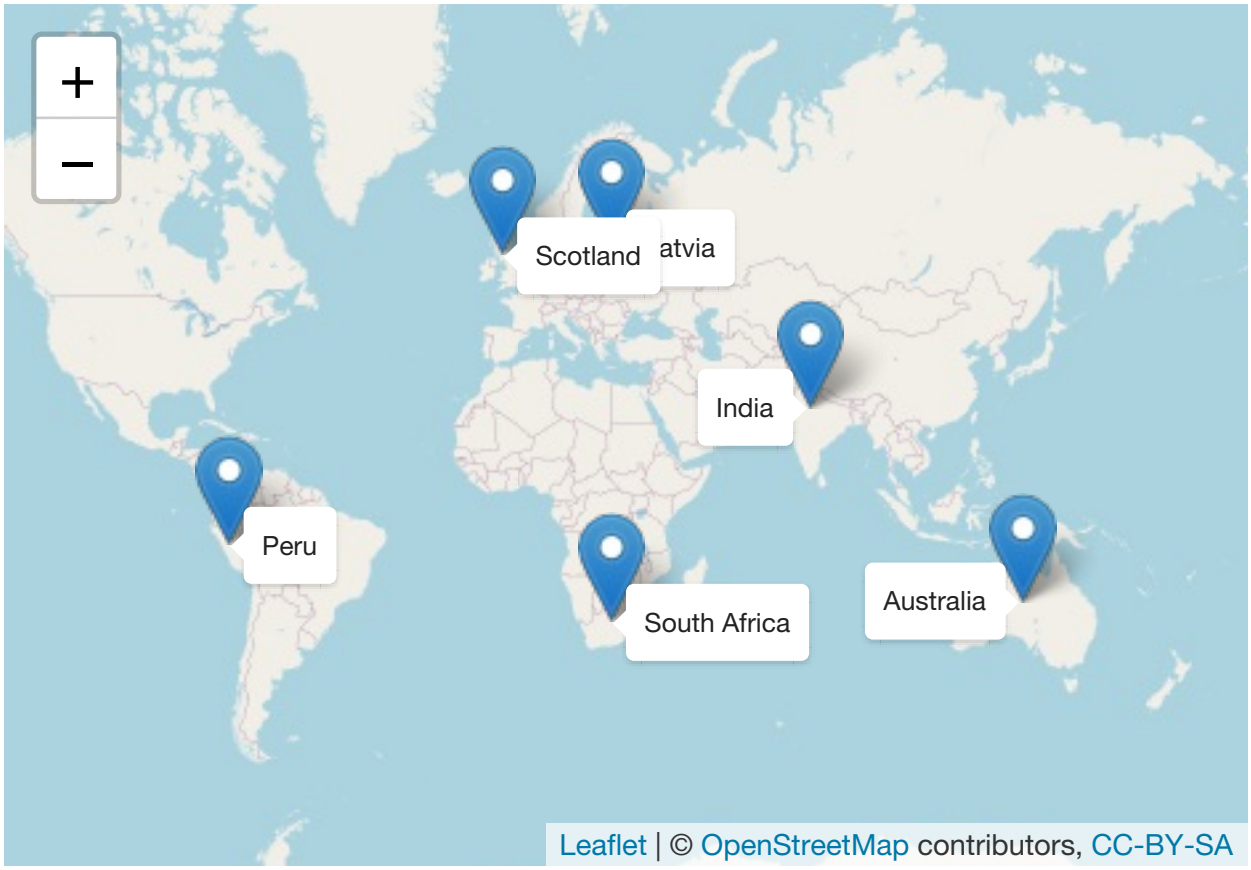
```
lon <- c(24.6032, 133.8826, 24.6727, -76.4000, 77.2167, -4.2514)
lat <- c(56.8796, -23.7005, -28.4792, -9.2800, 25.6448, 55.8609)

filtered_height$Lat <- lat
filtered_height$Lon <- lon
```

```
map1<- leaflet(filtered_height) %>%
  addTiles() %>%   # adding markers
   addMarkers(
    label =  ~filtered_height$country,
    labelOptions = labelOptions(noHide = T))
```

```
## Assuming "Lon" and "Lat" are longitude and latitude, respectively
```

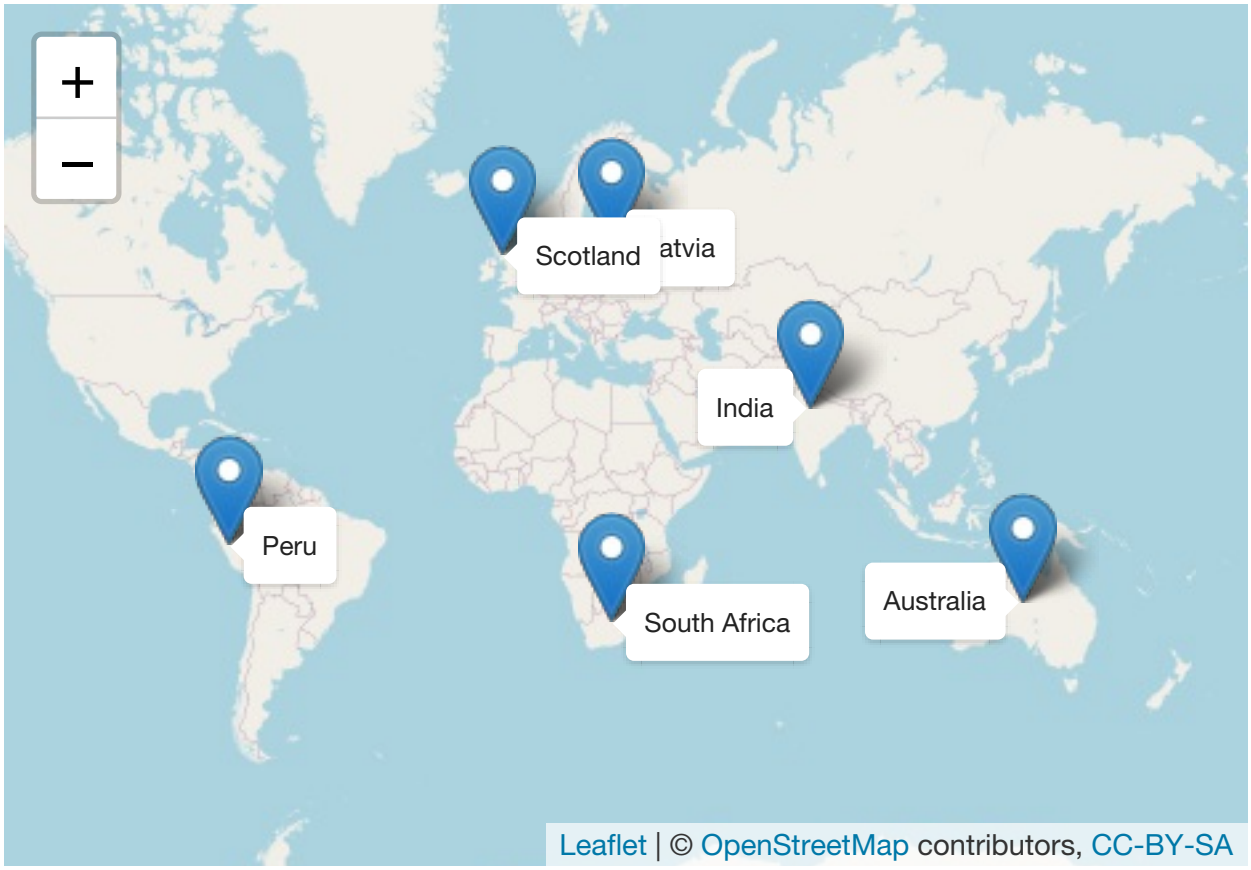
```
map1
```



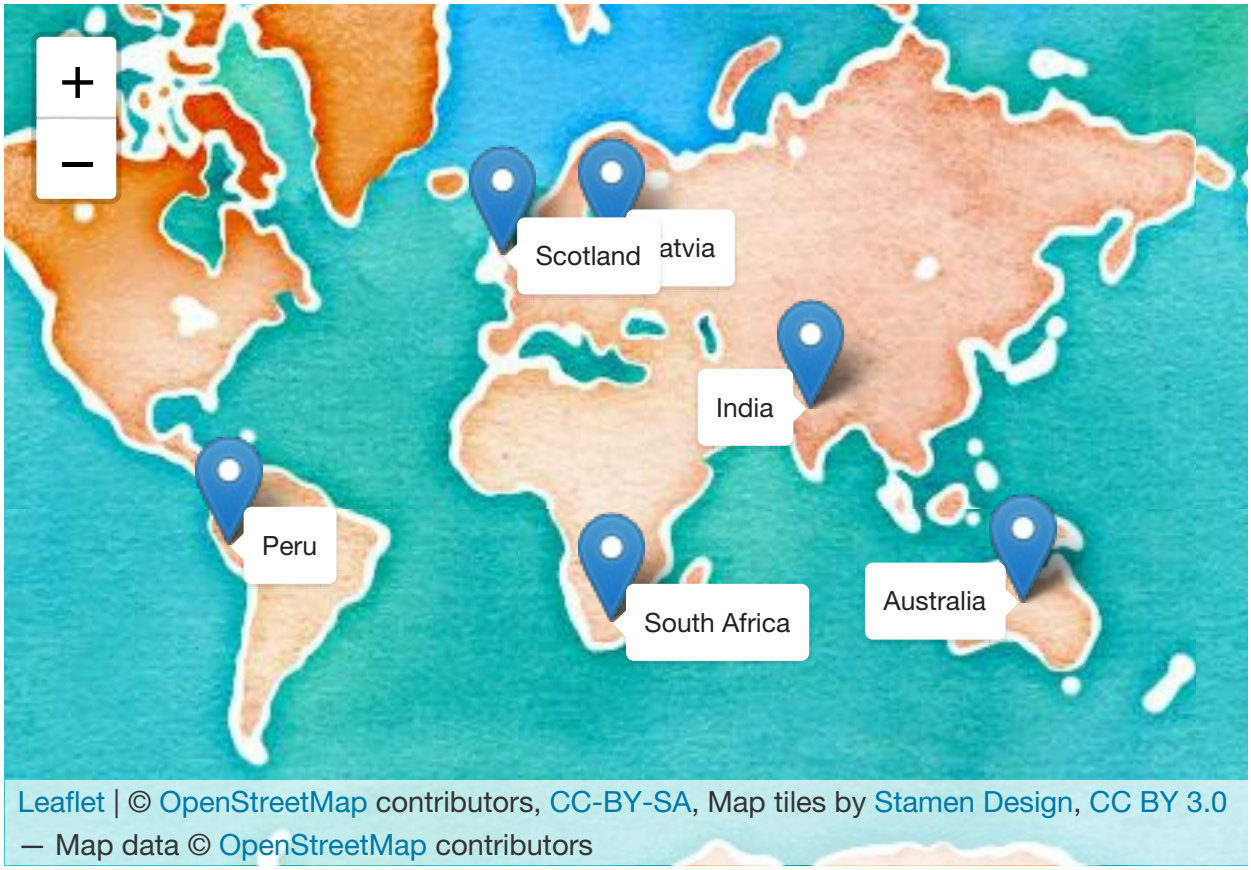

```
map2 <- map1 %>%
  addMarkers(popup = ~filtered_height$meanHeightFemale)
```

```
## Assuming "Lon" and "Lat" are longitude and latitude, respectively
```

```
map2
```




```
map3 <- map2 %>%
  addProviderTiles(providers$Stamen.Watercolor)
map3
```




```
library(plotly)
```

```
##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##     last_plot
## The following object is masked from 'package:MASS':
##
##     select
## The following object is masked from 'package:stats':
##
##     filter
## The following object is masked from 'package:graphics':
##
##     layout
```

```
height2 <- height_data %>% dplyr::select(country, region, meanHeightFemale)

head(height2)
```

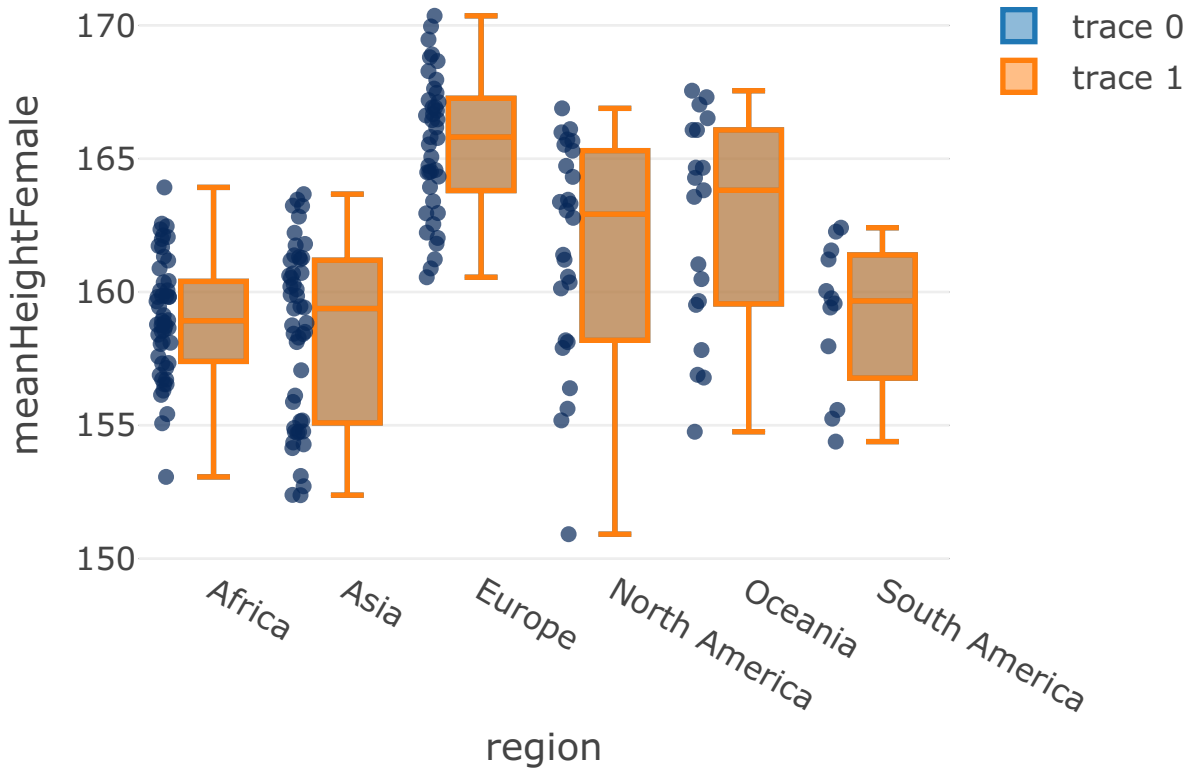
```
## # A tibble: 6 x 3
##   country               region meanHeightFemale
##   <chr>                 <chr>             <dbl>
## 1 Netherlands           Europe             170.
```

```
## 2 Montenegro              Europe              170.
## 3 Bosnia and Herzegovina Europe              167.
## 4 Iceland                 Europe              169.
## 5 Denmark                 Europe              169.
## 6 Czech Republic          Europe              168.
```

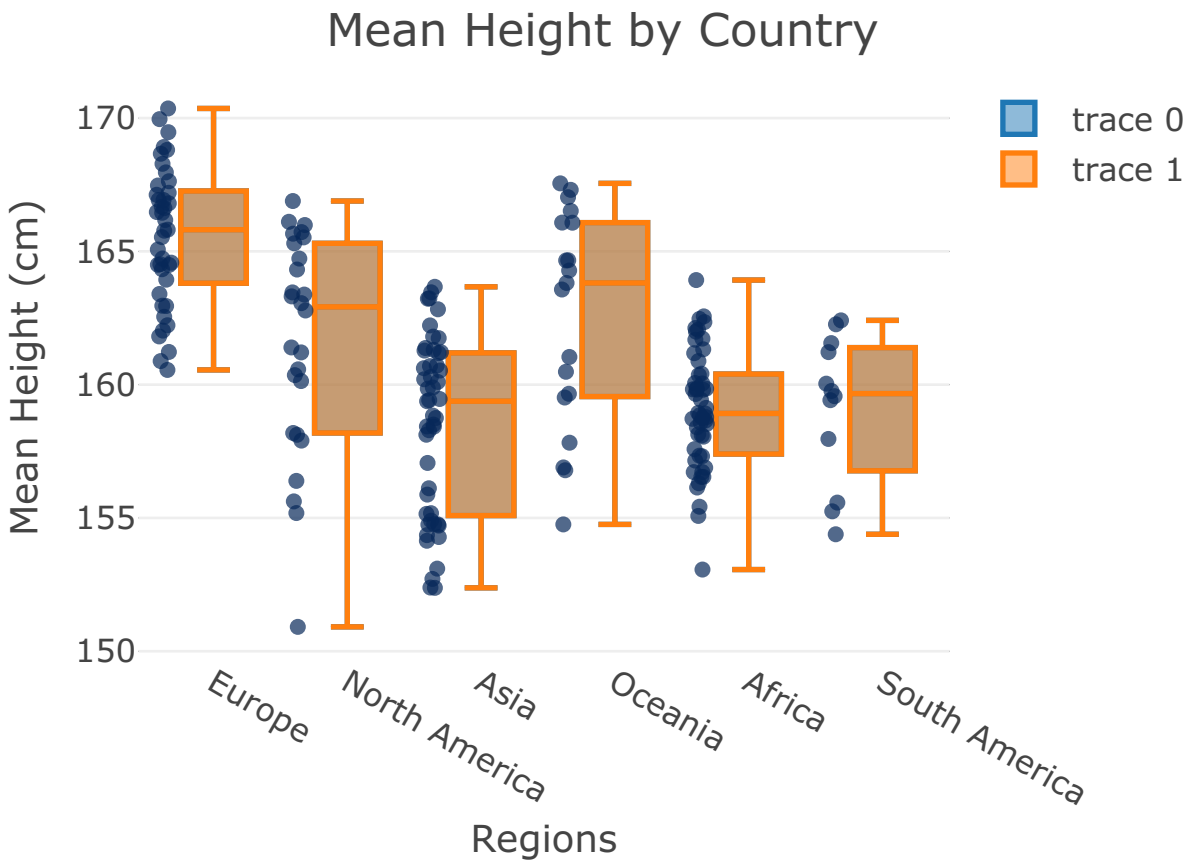
```
# create a box plot with meanHeightFemale as y and country as x
fig <- plot_ly(data = height2, y = ~meanHeightFemale, x = ~region, type = "box")


fig <- fig %>% add_trace(y = ~meanHeightFemale, x = ~region, boxpoints = "all", jitter = 0.3, marker = l

fig
```




```
fig2 <- fig %>%
  layout(title = "Mean Height by Country",
                      xaxis = list(title = "Regions", categoryorder = "array", categoryarray = height_da
                      yaxis = list(title = "Mean Height (cm)"),
                      legend = list(title = "Region"))
fig2
```


Mean Height by Country
trace 0
trace 1
170
165
160
155
150
Mean Height (cm)
Europe
North America
Asia
Oceania
Africa
South America
Regions