Statistical Computing

Assignment 2

Tinotenda Mutsemi

2024-03-05

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 4.3.3
```

```
library(reshape2)
```

```
#read data
lex_raw <- read.csv("lex.csv")
gdp_pcap_raw <- read.csv("gdp_pcap.csv")
region_raw <- read.xlsx("Data Geographies - v2 - by Gapminder.xlsx", sheetIndex = 2)
population_raw <- read.xlsx("GM-Population - Dataset - v6.xlsx", sheetIndex = 4)
```

```
#Qtn 1a
#data cleaning and selcting required data
lex_2019 <- select(lex_raw, country, X2019)

gdp_pcap_2019 <- select(gdp_pcap_raw, country, X2019)

#region data
#rename region col names
lookup <- c(country = "name",
            region = "four_regions")
region <- select(region_raw, name, four_regions) |> rename(all_of(lookup))
```

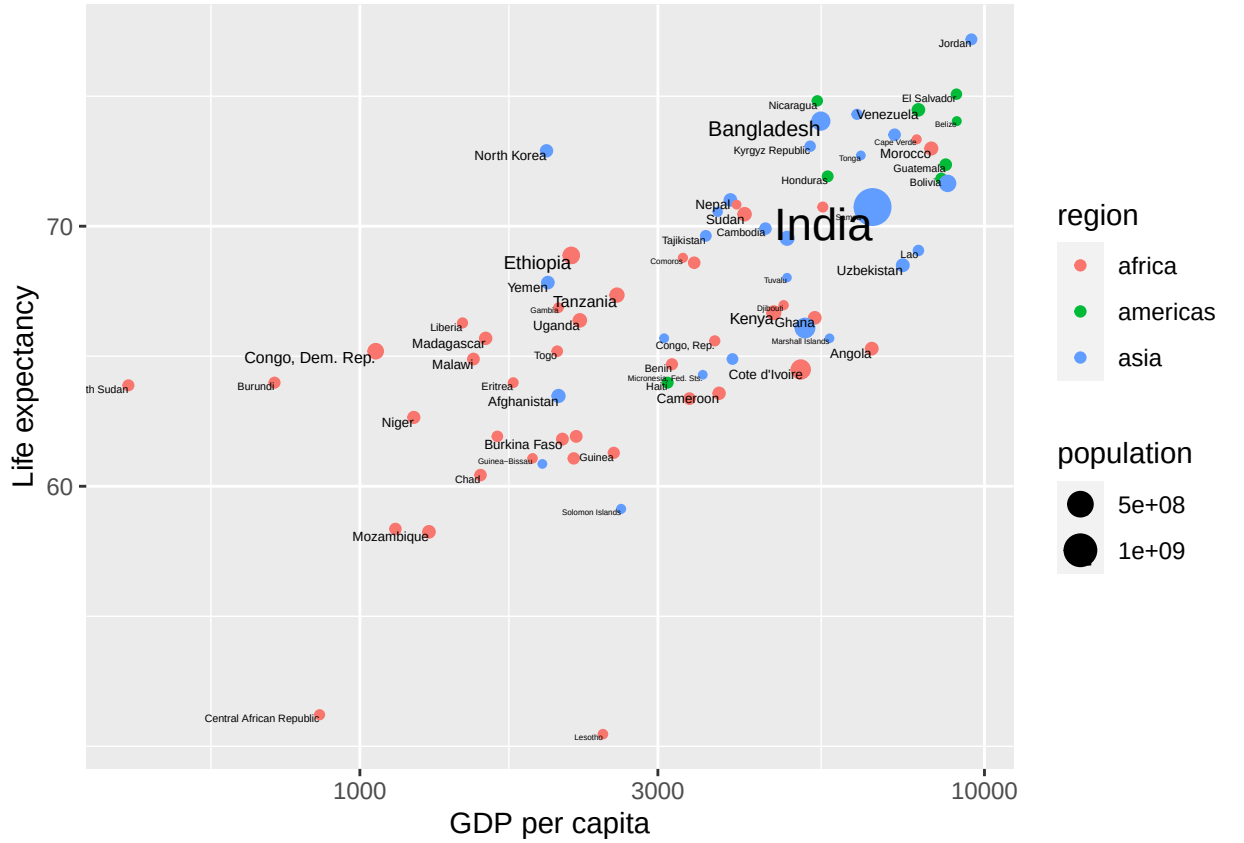
```r
#population data
#deselect col
population_raw2 <- select(population_raw, -geo)
#rename cols
lookup <- c(country = "name",
            year = "time",
            population = "Population")

population_raw2 <- rename(population_raw2, all_of(lookup))

#select 2019 population
population_2019 <- filter(population_raw2, year == 2019)
```

```r
#merge and clean data frames
lex_gdp_2019 <- full_join(lex_2019, gdp_pcap_2019, by = join_by(country))
lex_gdp_region_2019 <- full_join(lex_gdp_2019, region, by = join_by(country))

#rename cols of new df
lookup <- c(lex = "X2019.x",
            gdp_pcap = "X2019.y")
lex_gdp_region_2019 <- rename(lex_gdp_region_2019, all_of(lookup))

#make gdp_pcap numeric
lex_gdp_region_2019$gdp_pcap <- as.numeric(lex_gdp_region_2019$gdp_pcap)
```

```
## Warning: NAs introduced by coercion
```

```r
lex_gdp_region_popu_2019 <- full_join(lex_gdp_region_2019, population_2019, by = join_by(country))
```

```r
plot_lex_gdp_region_popu_2019 <- na.omit(lex_gdp_region_popu_2019)
#scatter plot with ggplot
ggplot(data = plot_lex_gdp_region_popu_2019, mapping = aes(x = gdp_pcap, y = lex,
             colour = region,
             size = population,
             )) +
  geom_point() +
  geom_text(aes(label = country), check_overlap = TRUE, vjust = 1, hjust = 1, col = "black") +
  scale_x_log10() +
  scale_y_log10() +
  labs(
       x = "GDP per capita",
       y = "Life expectancy")
```




```
  # title = "GDP per capita vs Life expectancy 2019",
#save plot
ggsave("lex_gdp_region_popu_2019.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
#Qtn b
region_avg_lex <- lex_gdp_region_popu_2019 |> group_by(region) |> summarise(avg_region_lex = mean(lex, r

#remove na region
region_avg_lex <- na.omit(region_avg_lex)

#sort by avg_region_lex descending
region_avg_lex <- region_avg_lex |> arrange(desc(avg_region_lex))
region_avg_lex
```

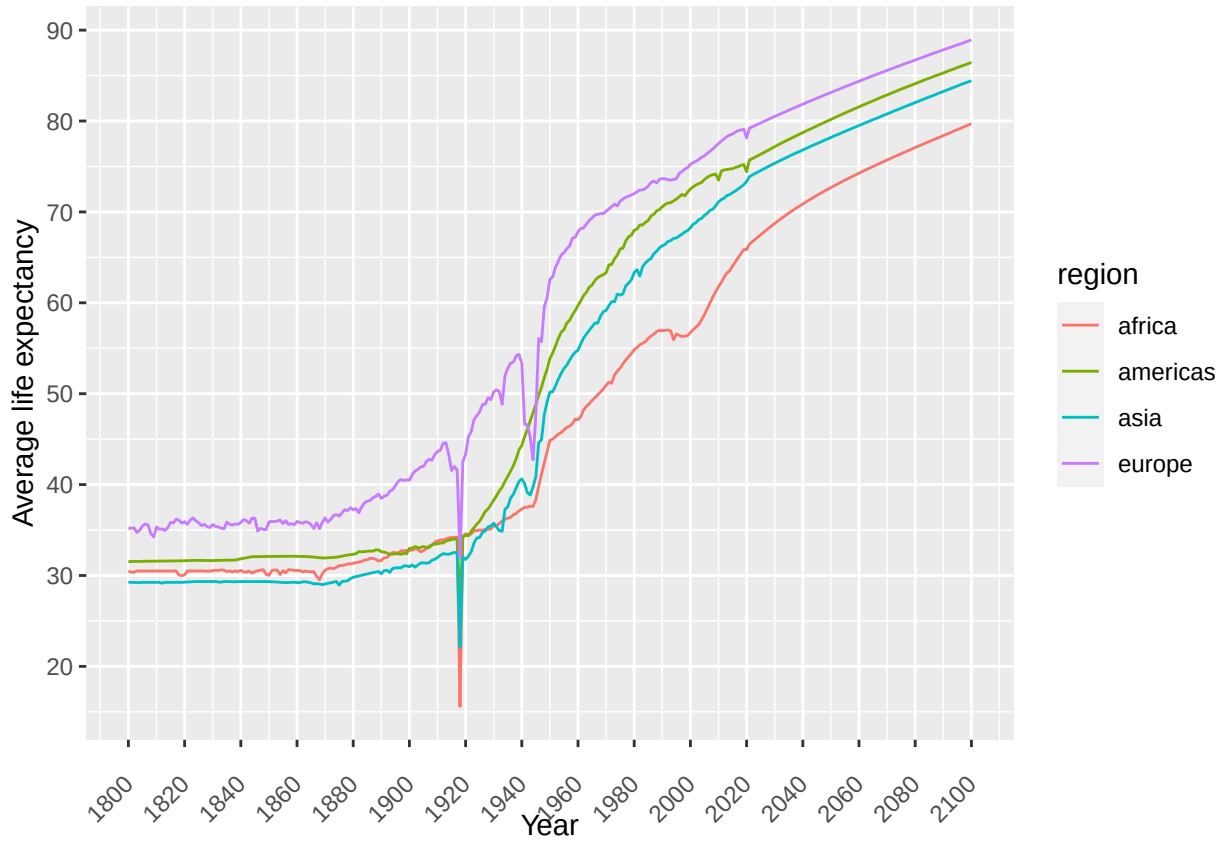
```
## # A tibble: 4 x 3
##   region   avg_region_lex countries_in_region
##   <chr>             <dbl>               <int>
## 1 europe             79.1                  49
## 2 americas           75.2                  35
## 3 asia               73.0                  59
## 4 africa             65.9                  54
```

```r
#Qtn c

#melt lex_raw
lex_melt <- melt(lex_raw, id.vars = "country", value.name = "lex", variable.name = "year")



#merge lex_melt with regions
lex_region <- full_join(lex_melt, region, by = "country")

region_avg_lex <- lex_region |> group_by(region, year) |> summarise(avg_lex = mean(lex, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'region'. You can override using the
## `.groups` argument.
```

```r
#remove leading X from year
region_avg_lex$year <- gsub("X", "", region_avg_lex$year)
#make year numeric
region_avg_lex$year <- as.numeric(region_avg_lex$year)
```

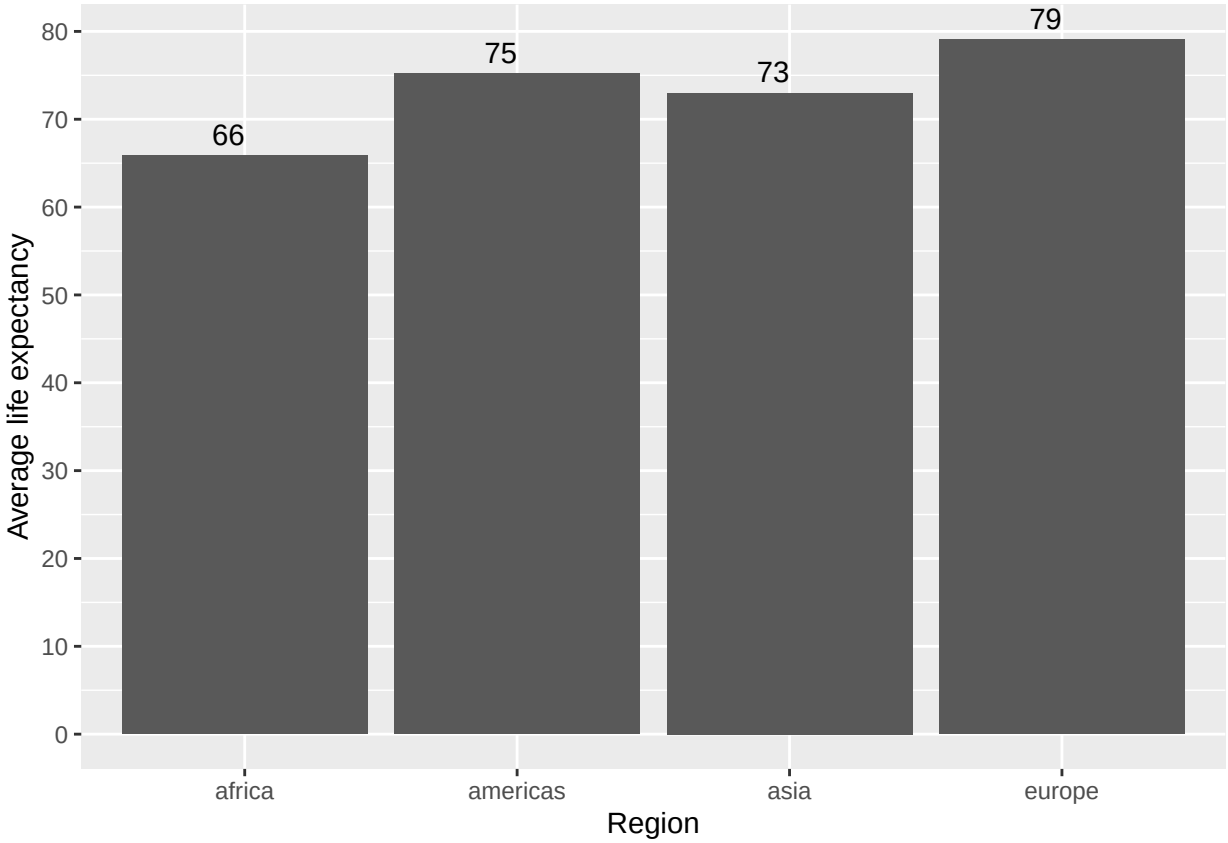
```r
plot_region_avg_lex = na.omit(region_avg_lex)
ggplot(data = plot_region_avg_lex, mapping = aes(x = year, y = avg_lex, group = region, col = region)) +
  geom_line() +
  labs(
       x = "Year",
       y = "Average life expectancy") +
  scale_x_continuous(breaks = seq(1800, 2160,20)) +
  scale_y_continuous(breaks = seq(0, 90, 10)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 1))
```




```
# title = "Region average life expectancy over time",

#save plot
ggsave("region_avg_lex.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
#Qtn d
#select 2019 data
region_avg_lex_2019 <- filter(region_avg_lex, year == 2019)
#drop na region
region_avg_lex_2019 <- na.omit(region_avg_lex_2019)

#plot bar chart
ggplot(data = region_avg_lex_2019, mapping = aes(x = region, y = avg_lex)) +
  geom_bar(stat = "identity") +
  labs(
       x = "Region",
       y = "Average life expectancy") +
  # theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 1)) +
  scale_y_continuous(breaks = seq(0, 90, 10)) +
  geom_text(aes(label = round(avg_lex, 0)), vjust = -0.5, hjust = 1, col = "black")
```




```
# title = "Region average life expectancy 2019",

#save plot
ggsave("region_avg_lex_2019.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
#Qtn e
country_split <- strsplit(lex_raw$country, " ")

#get count two word countries
two_word_countries <- sum(sapply(country_split, length) == 2)
two_word_countries
```

```
## [1] 24
```