Data Cleaning

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
cities <- read_csv("C:/Users/willi/Downloads/CITIES_26072022091653320.csv")
```

```
## Rows: 39320 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): METRO_ID, Metropolitan areas, VAR, Variables, Unit Code, Unit, Powe...
## dbl (4): TIME, Year, PowerCode Code, Value
## lgl (4): Reference Period Code, Reference Period, Flag Codes, Flags
##
## 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.
```

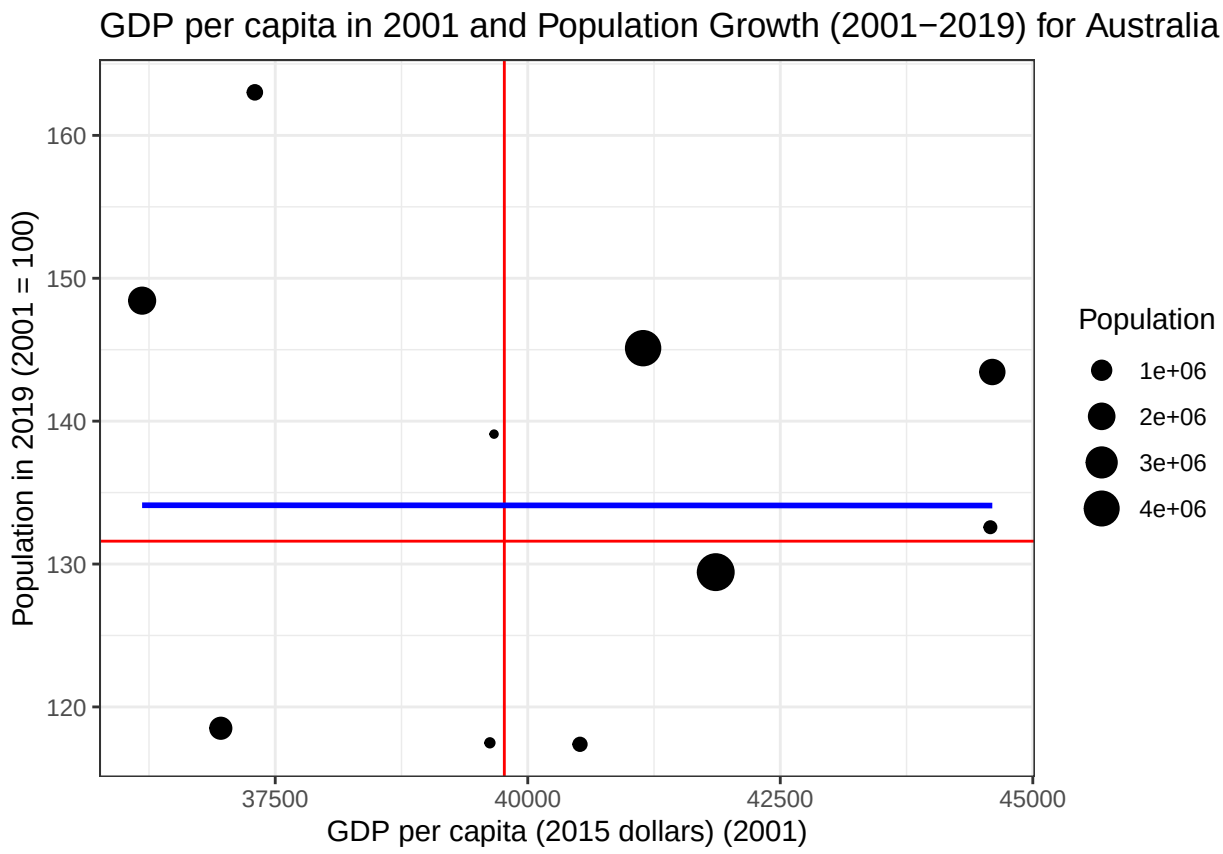
```
cities = select(cities, METRO_ID, `Metropolitan areas`, VAR, Year, Value)

convert_country = function(x){
  recode(x, US="United States", FR="France", IE="Ireland",
         DE="Germany",ME="Mexico",JP="Japan",CL="Chile",EE="Estonia",
         UK="United Kingdom",PL="Poland",PT="Portugal",IT="Italy",KO="Korea",
         ES="Spain",NL="Netherlands",LT="Lithuania",BE="Belgium",CA="Canada",
         CO="Colombia",AU="Australia",FI="Finland",NO="Norway",SE="Sweden",
         AT="Austria",CZ="Czechia",HU="Hungary",DK="Denmark",LU="Luxembourg",
         SI="Slovenia",CH="Switzerland",EL="Greece",LV="Latvia",SK="Slovakia",
         BG="Bulgaria",HR="Croatia",MT="Malta",NZ="New Zealand",RO="Romania",
         TR="Turkey",DN="Denmark",GR="Greece",IR="Ireland",PO="Poland",
         PR="Portugal",SW="Sweden",SV="Slovenia",GB="United Kingdom",TU="Turkey",
         ML="Malta")
}
cities = cities %>%
  mutate(Country = convert_country(substr(METRO_ID, 1, 2))) %>%
  mutate(VAR = recode(VAR, GDP_REAL_PPP = "GDP", T_T = "Population",
                      GDP_PC_REAL_PPP = "GDP_per_capita"))
names(cities) = c("ID","Metro","Var","Year","Value","Country")
```

```r
wider = cities %>%
  pivot_wider(names_from = Year, values_from = Value)



from_to = function(wider, from = "2002", to = "2018"){
  wider[[to]]/wider[[from]]*100
}

wider$`2019_2001` = from_to(wider, from = "2001", to = "2019")
Countries_20192001 = c("Australia", "Canada","Germany","Poland","United Kingdom","United States")
```

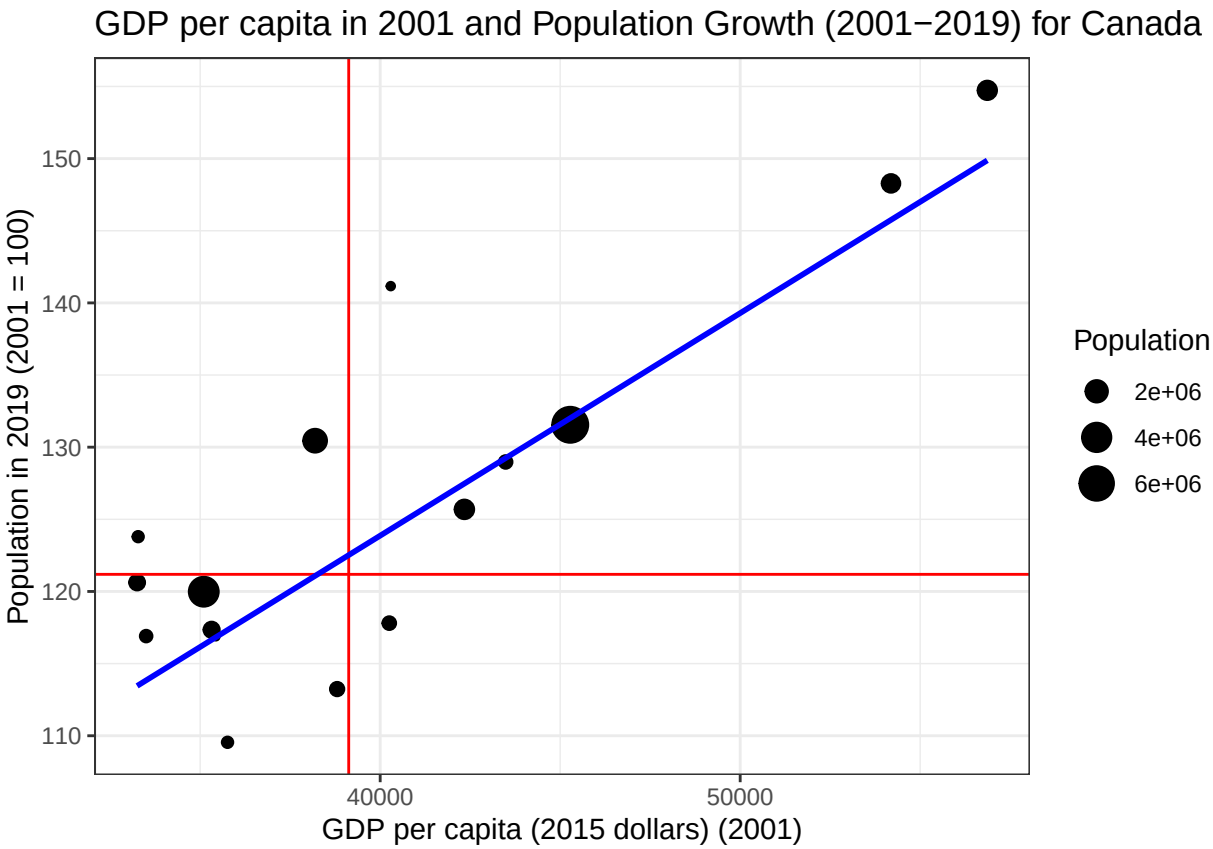
```r
Population_Growth = wider %>%
  filter(Var == "Population" & Country %in% Countries_20192001) %>%
  filter(Metro != "Austria") %>%
  select(Country, Metro, `2019_2001`, `2010`)
GDP_2002 = wider %>%
  filter(Var == "GDP_per_capita" & Country %in% Countries_20192001) %>%
  filter(Metro != "Austria") %>%
  select(Country, Metro, `2001`)
test = left_join(Population_Growth, GDP_2002, by = c("Country", "Metro"))
names(test) = c("Country", "Metro","PopulationGrowth","Population","GDP")

plots = test %>%
  drop_na() %>%
  group_by(Country) %>%
  do(plot = ggplot(data = ., aes(x = GDP, y = PopulationGrowth, size = Population)) +
                    scale_size() + theme_bw() +
                    geom_point(data = . %>% filter(Country != Metro)) +
                    geom_vline(aes(xintercept = GDP),
                        data = . %>% filter(Country == Metro), color = "red") +
                    geom_hline(aes(yintercept = PopulationGrowth),
                        data = . %>% filter(Country == Metro), color = "red") +
                    xlab("GDP per capita (2015 dollars) (2001)") + ylab("Population in 2019 (2001 = 100)
                    ggtitle(paste0("GDP per capita in 2001 and Population Growth (2001-2019) for ",.$Cou
                    geom_smooth(method = "lm", mapping = aes(weight = Population),
                        color = "blue", show.legend = FALSE, se = FALSE))
```

```r
plots[[2]]
```

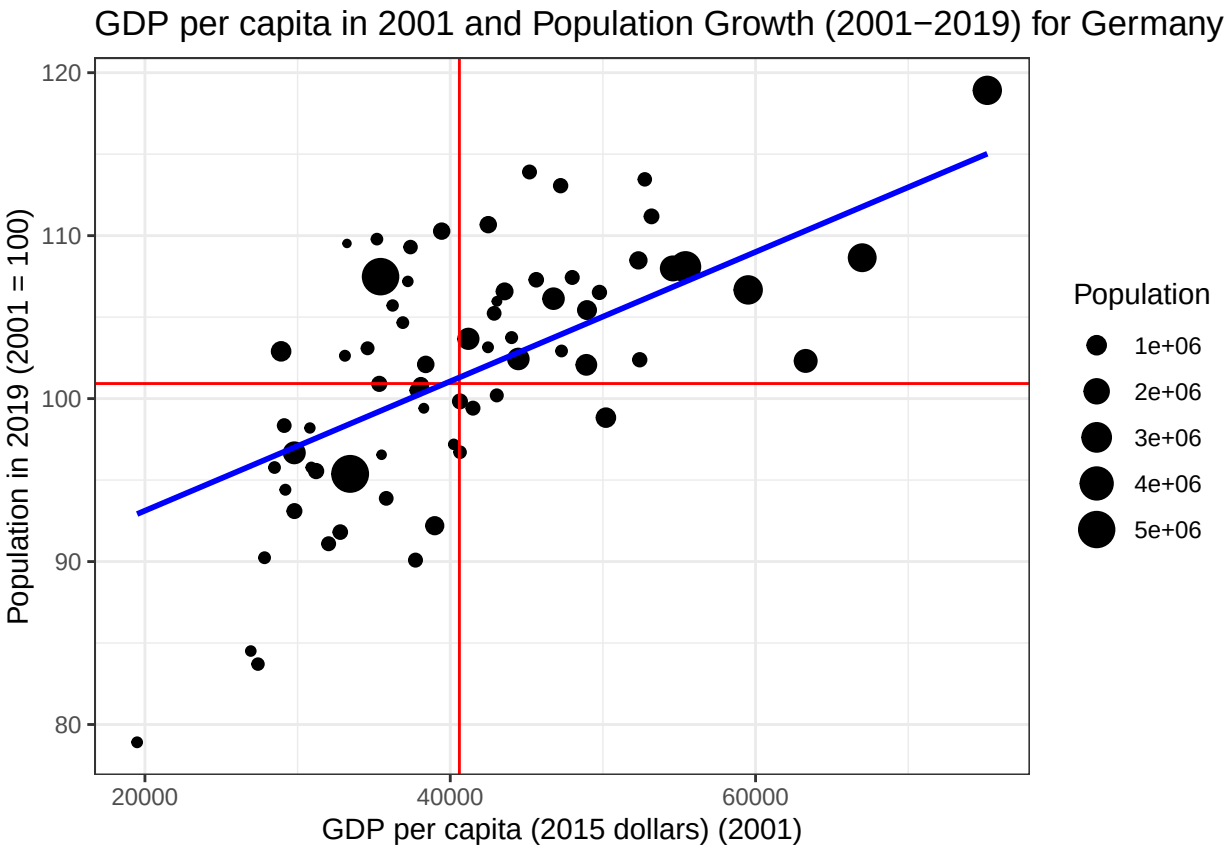
```
## [[1]]

## `geom_smooth()` using formula 'y ~ x'
```



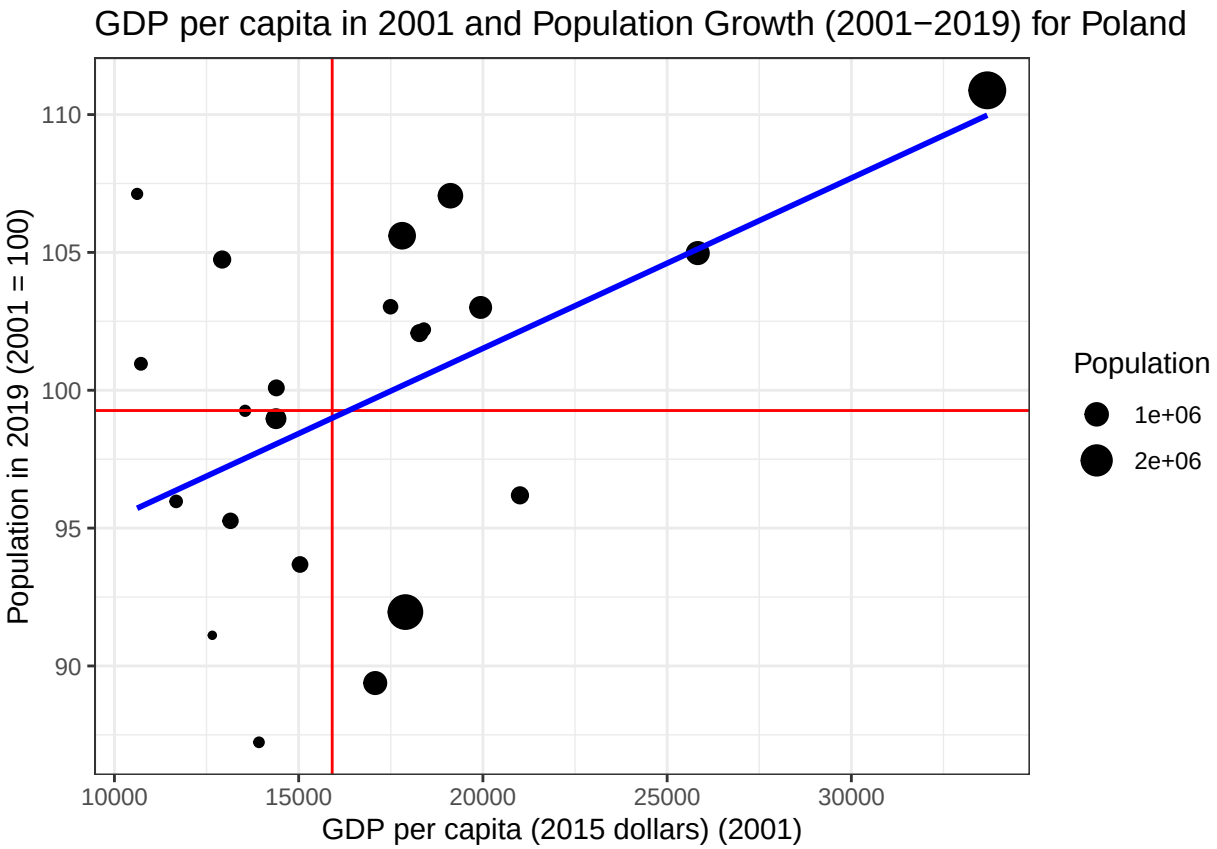

```
##
## [[2]]

## `geom_smooth()` using formula 'y ~ x'
```



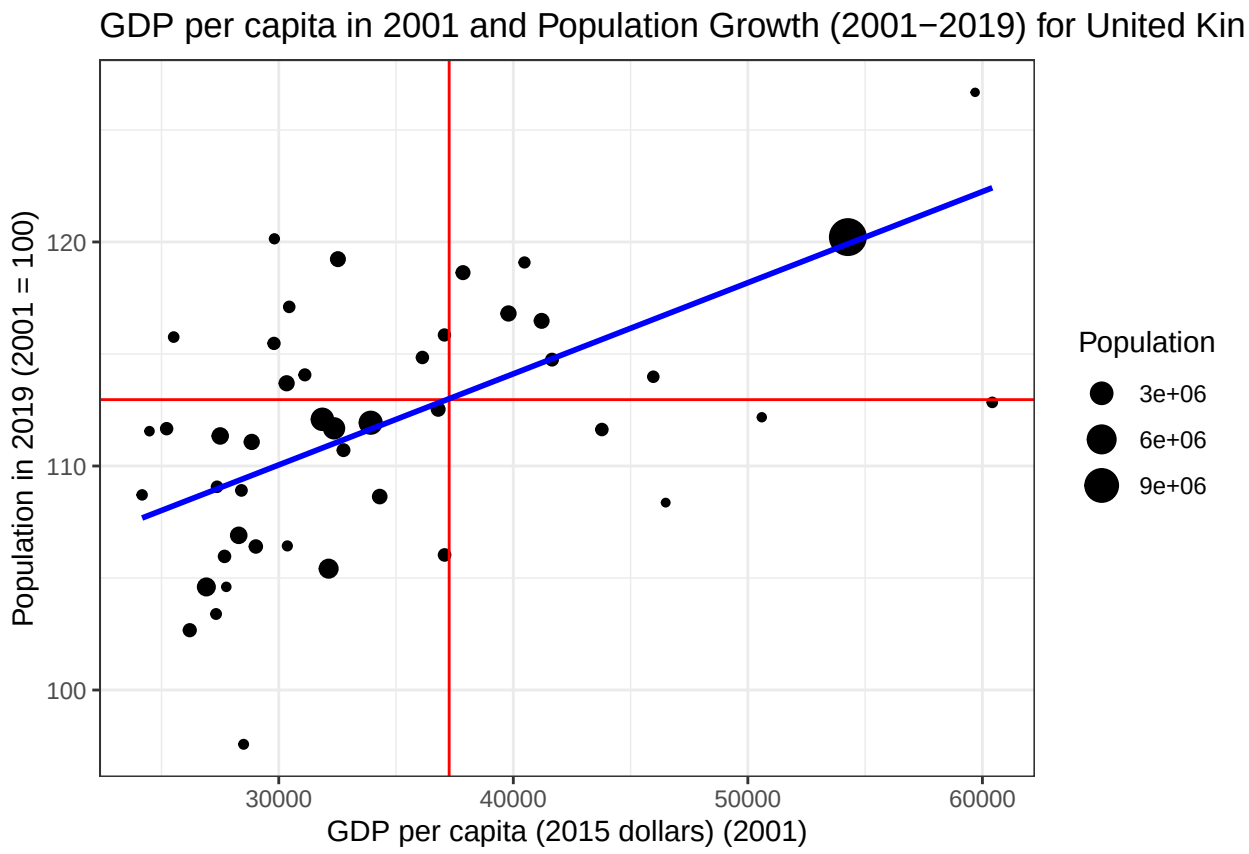

```
##
## [[3]]

## `geom_smooth()` using formula 'y ~ x'
```



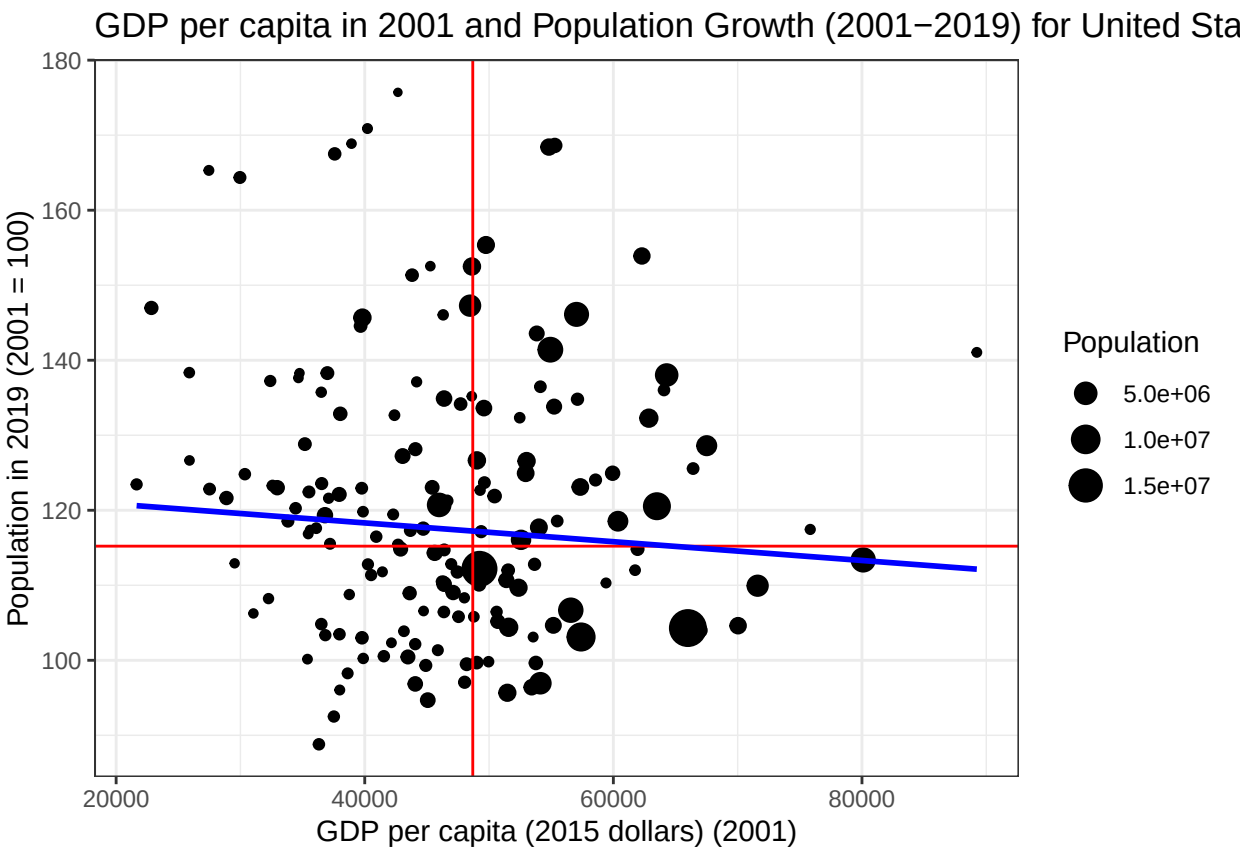

```
##
## [[4]]

## `geom_smooth()` using formula 'y ~ x'
```




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

## `geom_smooth()` using formula 'y ~ x'
```

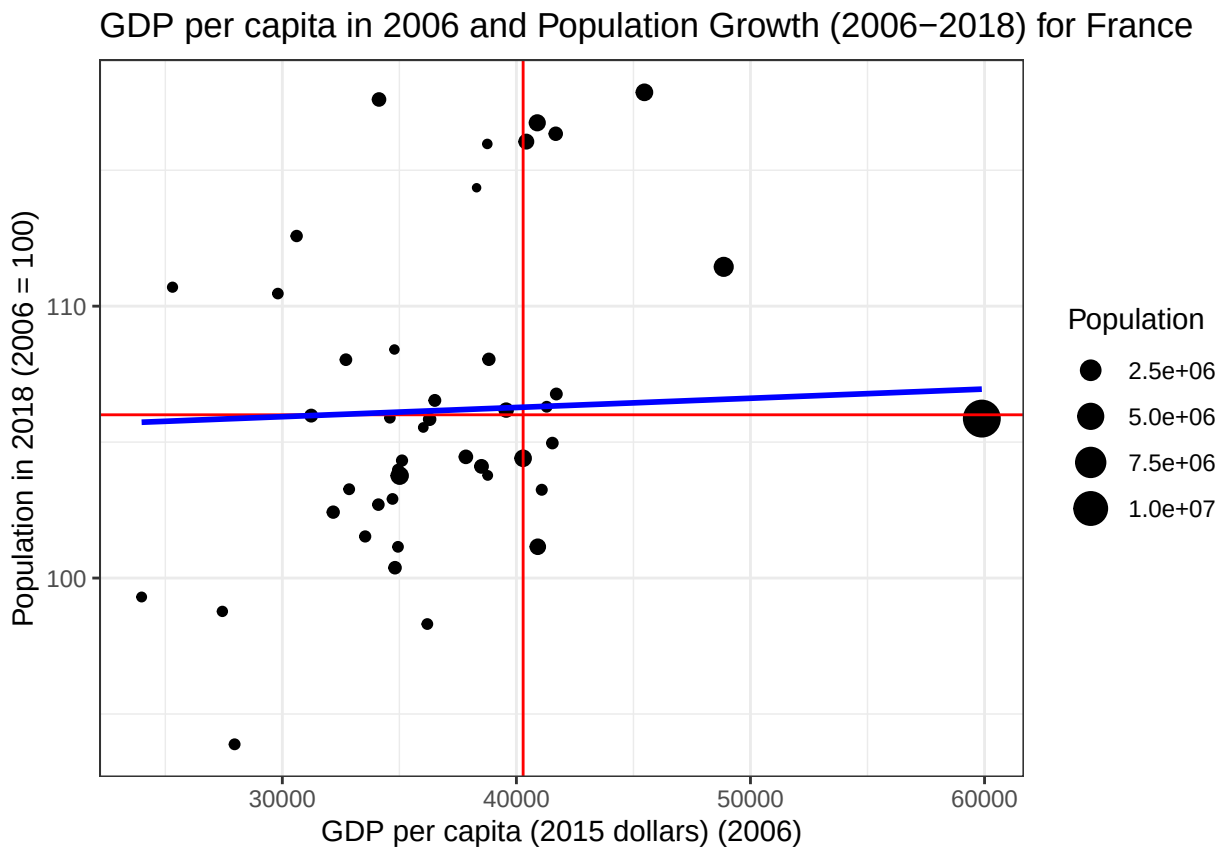



```
##
## [[6]]

## `geom_smooth()` using formula 'y ~ x'
```



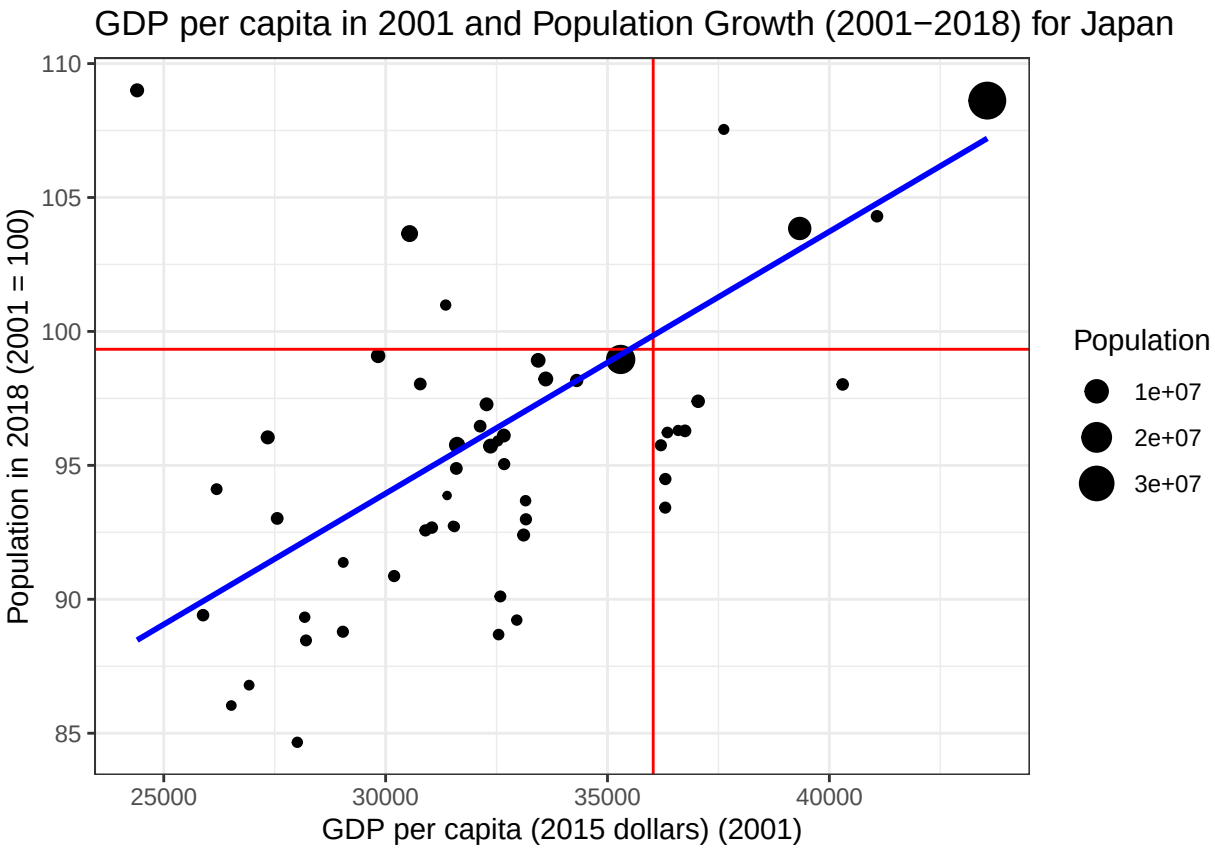

```
wider$`2018_2006` = from_to(wider, from = "2006", to = "2018")
Population_Growth = wider %>%
  filter(Var == "Population" & Country == "France") %>%
  select(Country, Metro, `2018_2006`, `2010`)
GDP_2002 = wider %>%
  filter(Var == "GDP_per_capita" & Country == "France") %>%
  select(Country, Metro, `2006`)
test2 = left_join(Population_Growth, GDP_2002, by = c("Country", "Metro"))
names(test2) = c("Country", "Metro","PopulationGrowth","Population","GDP")

plot2 = ggplot(data = test2, aes(x = GDP, y = PopulationGrowth, size = Population)) +
                    scale_size() + theme_bw() +
                    geom_point(data = . %>% filter(Country != Metro)) +
                    geom_vline(aes(xintercept = GDP),
                        data = . %>% filter(Country == Metro), color = "red") +
                    geom_hline(aes(yintercept = PopulationGrowth),
                        data = . %>% filter(Country == Metro), color = "red") +
                    xlab("GDP per capita (2015 dollars) (2006)") + ylab("Population in 2018 (2006 = 100
                    ggtitle(paste0("GDP per capita in 2006 and Population Growth (2006-2018) for France
                    geom_smooth(method = "lm", mapping = aes(weight = Population),
                        color = "blue", show.legend = FALSE, se = FALSE)
plot2
```

```
## `geom_smooth()` using formula 'y ~ x'
```



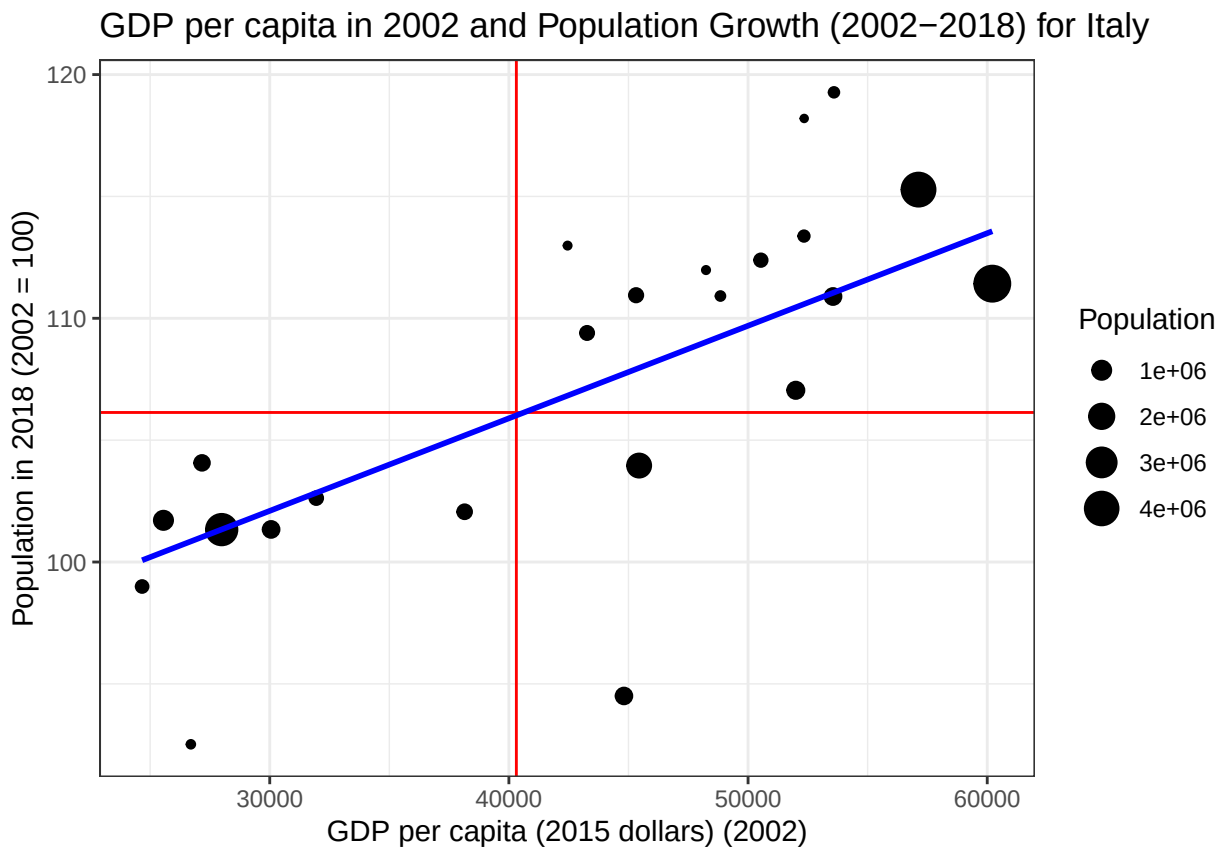

```
wider$`2018_2001` = from_to(wider, from = "2001", to = "2018")
Population_Growth = wider %>%
  filter(Var == "Population" & Country == "Japan") %>%
  select(Country, Metro, `2018_2001`, `2010`)
GDP_2002 = wider %>%
  filter(Var == "GDP_per_capita" & Country == "Japan") %>%
  select(Country, Metro, `2001`)
test3 = left_join(Population_Growth, GDP_2002, by = c("Country", "Metro"))
names(test3) = c("Country", "Metro","PopulationGrowth","Population","GDP")

plot3 = ggplot(data = test3, aes(x = GDP, y = PopulationGrowth, size = Population)) +
                    scale_size() + theme_bw() +
                    geom_point(data = . %>% filter(Country != Metro)) +
                    geom_vline(aes(xintercept = GDP),
                        data = . %>% filter(Country == Metro), color = "red") +
                    geom_hline(aes(yintercept = PopulationGrowth),
                        data = . %>% filter(Country == Metro), color = "red") +
                    xlab("GDP per capita (2015 dollars) (2001)") + ylab("Population in 2018 (2001 = 100)
                    ggtitle(paste0("GDP per capita in 2001 and Population Growth (2001-2018) for Japan")
                    geom_smooth(method = "lm", mapping = aes(weight = Population),
                        color = "blue", show.legend = FALSE, se = FALSE)
plot3
```

```
## `geom_smooth()` using formula 'y ~ x'
```



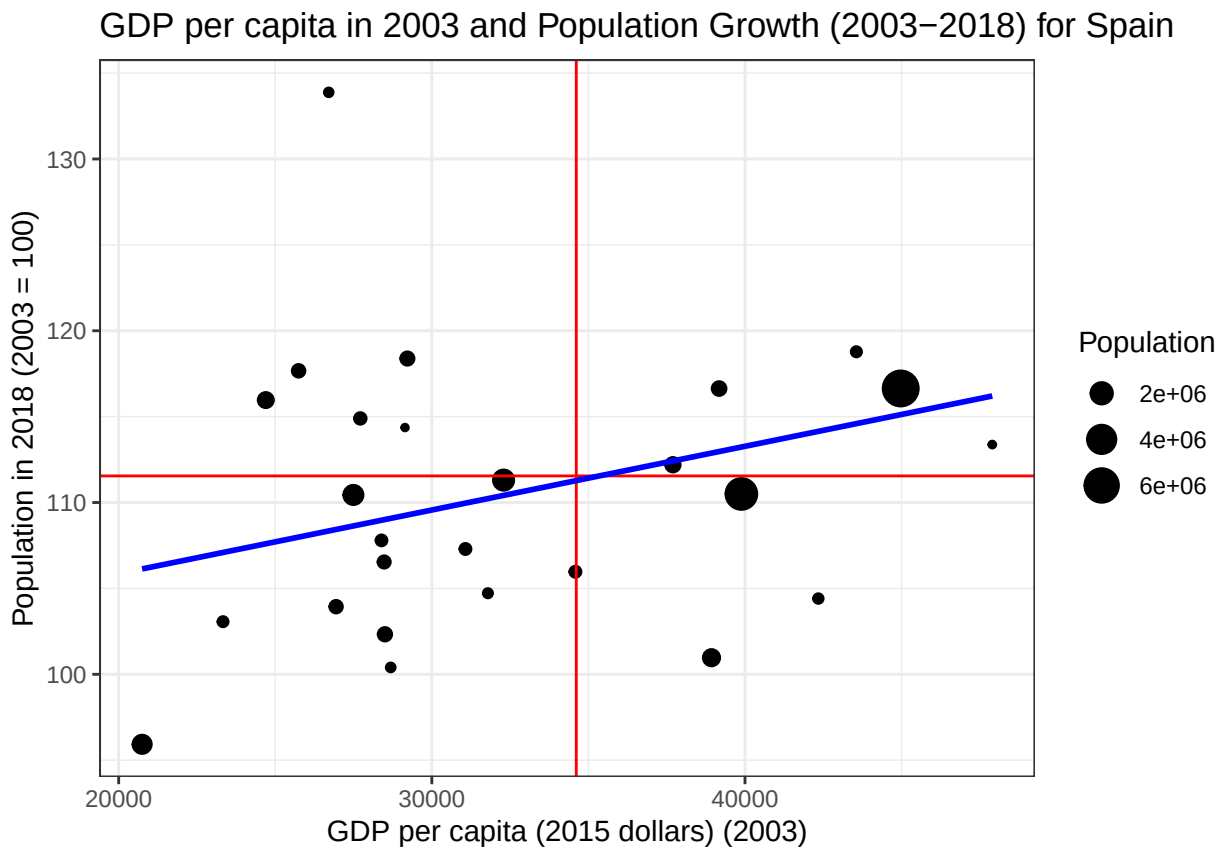

```
wider$`2018_2002` = from_to(wider, from = "2002", to = "2018")
Population_Growth = wider %>%
  filter(Var == "Population" & Country == "Italy") %>%
  select(Country, Metro, `2018_2002`, `2010`)
GDP_2002 = wider %>%
  filter(Var == "GDP_per_capita" & Country == "Italy") %>%
  select(Country, Metro, `2002`)
test4 = left_join(Population_Growth, GDP_2002, by = c("Country", "Metro"))
names(test4) = c("Country", "Metro","PopulationGrowth","Population","GDP")

plot4 = ggplot(data = test4, aes(x = GDP, y = PopulationGrowth, size = Population)) +
                    scale_size() + theme_bw() +
                    geom_point(data = . %>% filter(Country != Metro)) +
                    geom_vline(aes(xintercept = GDP),
                        data = . %>% filter(Country == Metro), color = "red") +
                    geom_hline(aes(yintercept = PopulationGrowth),
                        data = . %>% filter(Country == Metro), color = "red") +
                    xlab("GDP per capita (2015 dollars) (2002)") + ylab("Population in 2018 (2002 = 100
                    ggtitle(paste0("GDP per capita in 2002 and Population Growth (2002-2018) for Italy"
                    geom_smooth(method = "lm", mapping = aes(weight = Population),
                        color = "blue", show.legend = FALSE, se = FALSE)
plot4
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
wider$`2018_2003` = from_to(wider, from = "2003", to = "2018")
Population_Growth = wider %>%
  filter(Var == "Population" & Country == "Spain") %>%
  select(Country, Metro, `2018_2003`, `2010`)
GDP_2002 = wider %>%
  filter(Var == "GDP_per_capita" & Country == "Spain") %>%
  select(Country, Metro, `2003`)
test5 = left_join(Population_Growth, GDP_2002, by = c("Country", "Metro"))
names(test5) = c("Country", "Metro","PopulationGrowth","Population","GDP")

plot5 = ggplot(data = test5, aes(x = GDP, y = PopulationGrowth, size = Population)) +
                    scale_size() + theme_bw() +
                    geom_point(data = . %>% filter(Country != Metro)) +
                    geom_vline(aes(xintercept = GDP),
                        data = . %>% filter(Country == Metro), color = "red") +
                    geom_hline(aes(yintercept = PopulationGrowth),
                        data = . %>% filter(Country == Metro), color = "red") +
                    xlab("GDP per capita (2015 dollars) (2003)") + ylab("Population in 2018 (2003 = 100
                    ggtitle(paste0("GDP per capita in 2003 and Population Growth (2003-2018) for Spain"
                    geom_smooth(method = "lm", mapping = aes(weight = Population),
                        color = "blue", show.legend = FALSE, se = FALSE)
plot5
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
percent = rep(0, 478)
result = rbind(test, test2, test3, test4, test5)
aggregates = result %>% filter(Country == Metro)

result$Percent = result$Population/left_join(result, select(aggregates, Country, Population), by = "Cou

result = result %>% filter(Country != Metro)

write.csv(result, file = "data.csv", row.names=FALSE)
write.csv(aggregates, file = "aggregates.csv", row.names=FALSE)
lines = result %>%
  group_by(Country) %>%
  do(A = lm(PopulationGrowth ~ GDP, data = ., weights = Population))
lines2 = sapply(lines[[2]], function(x){
  R2 = summary(x)$r.squared
  Pval = summary(x)$coefficients["GDP","Pr(>|t|)"]
  Inter = coef(x)[1]
  Slope = coef(x)[2]
  return(c(R2 = R2, Pval = Pval, Inter, Slope))
})

lines3 = t(lines2)
lines4 = data.frame(Country = lines[[1]], lines3)
names(lines4) = c("Country","R2","Pval","Intercept","Slope")
write.csv(lines4, file = "regression_lines.csv", row.names=FALSE)
```