Yield analysis maize part 2

Kathi

Contents

Setup

Installing and loading required packages

```r
# Store package names, required for the analysis in a vector
packages <- c("tidyverse", "DBI", "RSQLite", "broom", "treemapify", "ggrepel", "scales")

# Install packages that are not yet installed
installed_packages <- packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(packages[!installed_packages])
}

# Load packages
invisible(lapply(packages, library, character.only = TRUE))
```

Import data from the SQL database.

In the first part of this project the global timeseries data on maize yield was retrieved from ncf4 files and stored in a database. In this section the data that is required for the planned data analysis is retrieved from the database.

```r
# Create a connection to the DB
con = dbConnect(
```

```r
  drv = RSQLite::SQLite(),
  dbname = "yield.db"
)
```

```r
# Import the yield data from the database
query = "SELECT
            yield,
            year,
            country,
            continent,
            area_ha
          FROM maize_yield
          WHERE TRUE
            AND country IS NOT NULL;"

# Store the retrieved data in a dataframe
yield_data <- dbFetch(dbSendQuery(con, query))

# Change the class of the year and area_ha to numeric
yield_data = yield_data %>%
  mutate(year = as.numeric(year),
         area_ha = as.numeric(area_ha))

# Check the first entries of the dataframe
head(yield_data)
```

```
##   yield year    country  continent  area_ha
## 1    NA 1981 Antarctica Antarctica 1342.707
## 2    NA 1982 Antarctica Antarctica 1342.707
## 3    NA 1983 Antarctica Antarctica 1342.707
## 4    NA 1984 Antarctica Antarctica 1342.707
## 5    NA 1985 Antarctica Antarctica 1342.707
## 6    NA 1986 Antarctica Antarctica 1342.707
```

```r
# Import the demographic data from the database
query = "SELECT
            country,
            population,
            gdp,
            income,
            export,
            import,
            year
        FROM demographic_data;"

# Store the retrieved data in a dataframe
demographic_data <- dbFetch(dbSendQuery(con, query))
```

```
## Warning: Closing open result set, pending rows

## Warning in result_fetch(res@ptr, n = n): Column `population`: mixed type, first
## seen values of type integer64, coercing other values of type string

## Warning in result_fetch(res@ptr, n = n): Column `gdp`: mixed type, first seen
## values of type real, coercing other values of type integer64, string
```

```
## Warning in result_fetch(res@ptr, n = n): Column `income`: mixed type, first seen
## values of type integer64, coercing other values of type real, string

## Warning in result_fetch(res@ptr, n = n): Column `export`: mixed type, first seen
## values of type real, coercing other values of type integer64, string

## Warning in result_fetch(res@ptr, n = n): Column `import`: mixed type, first seen
## values of type real, coercing other values of type integer64, string
```

```
# Change the class of the population, income, import and year to numeric
demographic_data = demographic_data %>%
  mutate(population = as.numeric(population),
         income = as.numeric(income),
         year = as.numeric(year))

# Check the first entries of the dataframe
head(demographic_data)
```

```
##       country population        gdp    income    export     import year
## 1 Afghanistan   13171679 3478787909 3.241e+09 766300000 1080500000 1981
## 2 Afghanistan   12882518         NA        NA 798000000  989600000 1982
## 3 Afghanistan   12537732         NA        NA 806200000 1062900000 1983
## 4 Afghanistan   12204306         NA        NA 841500000 1419400000 1984
## 5 Afghanistan   11938204         NA        NA 697400000 1084300000 1985
## 6 Afghanistan   11736177         NA        NA 550100000 1354700000 1986
```

```
# Close the connection to the database
dbDisconnect(con)
```

```
## Warning in connection_release(conn@ptr): There are 1 result in use. The
## connection will be released when they are closed
```

Data analysis

In this section the global timeseries data on maize yield is analysed by calculating summary statistics and creating visualizations of the results.

Data summary example

This sections creates a dataframe containing summary statistics that are used for several of the following data visualizations. These are:

- the amount of maize produced per ha for each country per year (weighted mean of the yield, weighed by the area at that point)
- the total amount of maize produced per country per year (sum of the yield multiplied by the area at that point)
- the total country area (sum of the area of each point assigned to a country)

```
# Calculate summary statistics and store the results in a dataframe
yield_summary = yield_data %>%
  # Replace NA values in the yield column with 0
    # Assumption: when a yield value is NA it means that in that area
    # no maize was produced!
  mutate(yield = replace_na(yield, 0)) %>%
  group_by(year, country) %>%
  summarise(yield_per_area = weighted.mean(yield, area_ha),
            sum_yield = sum(yield * area_ha, na.rm = TRUE),
```

```
            country_area = sum(area_ha),
            continent = first(continent)) # first() returns the first observation of the
```

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

```
                                             # group in combination with group_by

# Show the first entries of the summary dataframe
head(yield_summary)
```

```
## # A tibble: 6 x 6
## # Groups:   year [1]
##    year country     yield_per_area sum_yield country_area continent
##   <dbl> <chr>                <dbl>     <dbl>        <dbl> <chr>
## 1  1981 Afghanistan         0              0     64707697. Asia
## 2  1981 Aland               0              0       152954. Europe
## 3  1981 Albania             3.18     8889816.     2793843. Europe
## 4  1981 Algeria             0              0    233348440. Africa
## 5  1981 Angola              0.0237   2987265.   125956983. Africa
## 6  1981 Antarctica          0              0   1218741074. Antarctica
```

Data visualizations

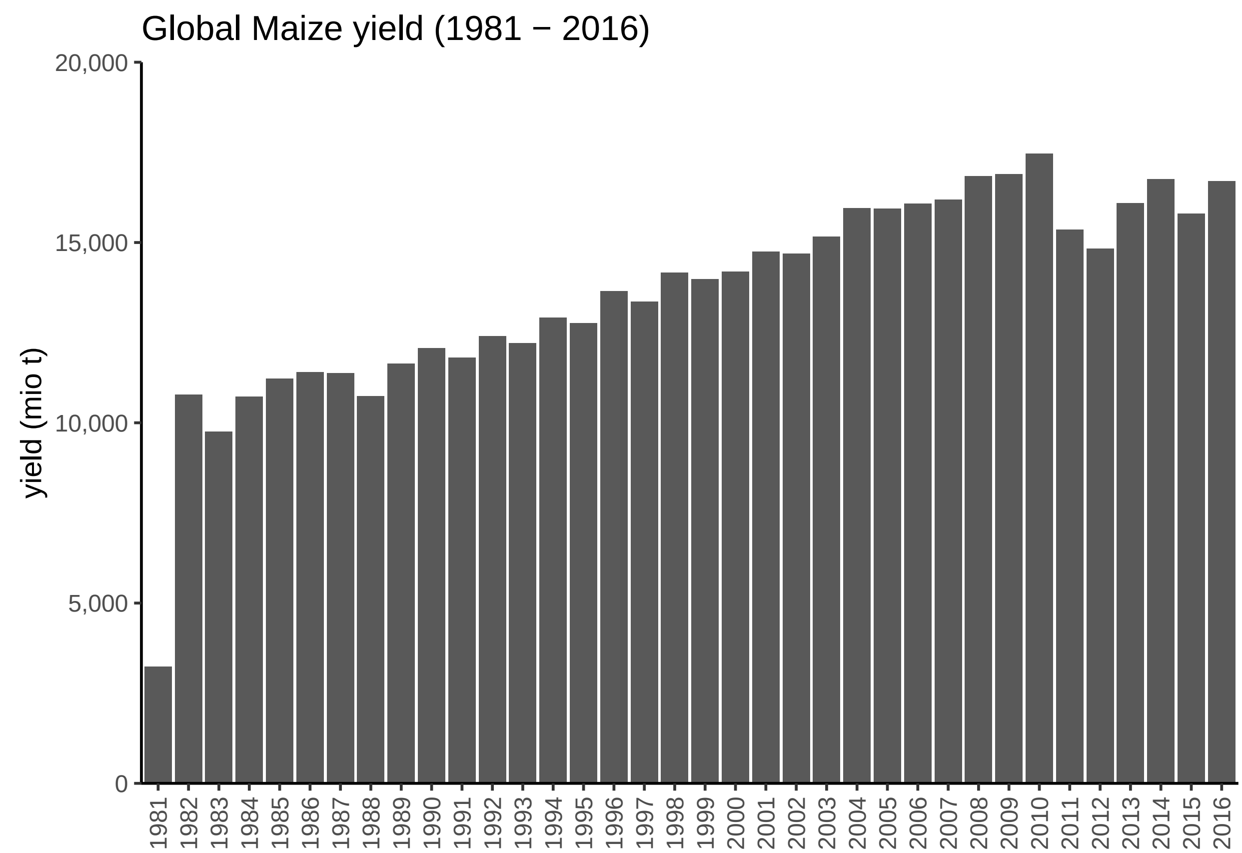
In this section data visualizations are created to illustrate the findings of the data analysis.

How did the global maize yield change over 35 years?

- Calculate the sum of the maize yield produced by all countries per year

```
# Global growth (bar chart)
global_yield_growth_bar <- yield_summary %>%
  # Calculate the total amount of maize produced per year worldwide
  group_by(year) %>%
  summarise(global_yield = sum(sum_yield)) %>%
  # Create a plot of the global yield (in mio t) vs year
  ggplot(aes(x = year, y = global_yield/1000000)) +
    # Create a bar chart and customize its appearance
    geom_bar(stat = "identity") +
    theme_classic() +
    scale_y_continuous(name = "yield (mio t)", # change label of y-axis
                       limits = c(0, 20000), # adjust the min and max of the y-axis
                       breaks = seq(0, 20000, 5000), # adjust the y-axis ticks and labels
                       labels = comma, # adjust the appearance of the y-axis tick labels
                       expand = c(0, 0)) +
    scale_x_continuous(name = NULL, # delete x-axis label
                         breaks = seq(1981, 2016, 1), # adjust the x-axis ticks and labels
                         expand = c(0, 0.1)) +
    guides(x = guide_axis(angle = 90)) + # turn x-axis tick labels by 90 degree
    ggtitle("Global Maize yield (1981 - 2016)") # add a title

# Show the plot
global_yield_growth_bar
```




```
# Save the plot as a jpeg file
ggsave(device = "jpeg", "global_yield_growth_bar", plot = global_yield_growth_bar)
```

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

- Define the timespan without 1981, as that data contains incomplete information when the growing season of maize spans two calendar years (see https://doi.pangaea.de/10.1594/PANGAEA.909132)
- Calculate the sum of the maize yield produced by all countries per year
- Calculate the difference ("delta") in global maize yield to the preceding year in percent

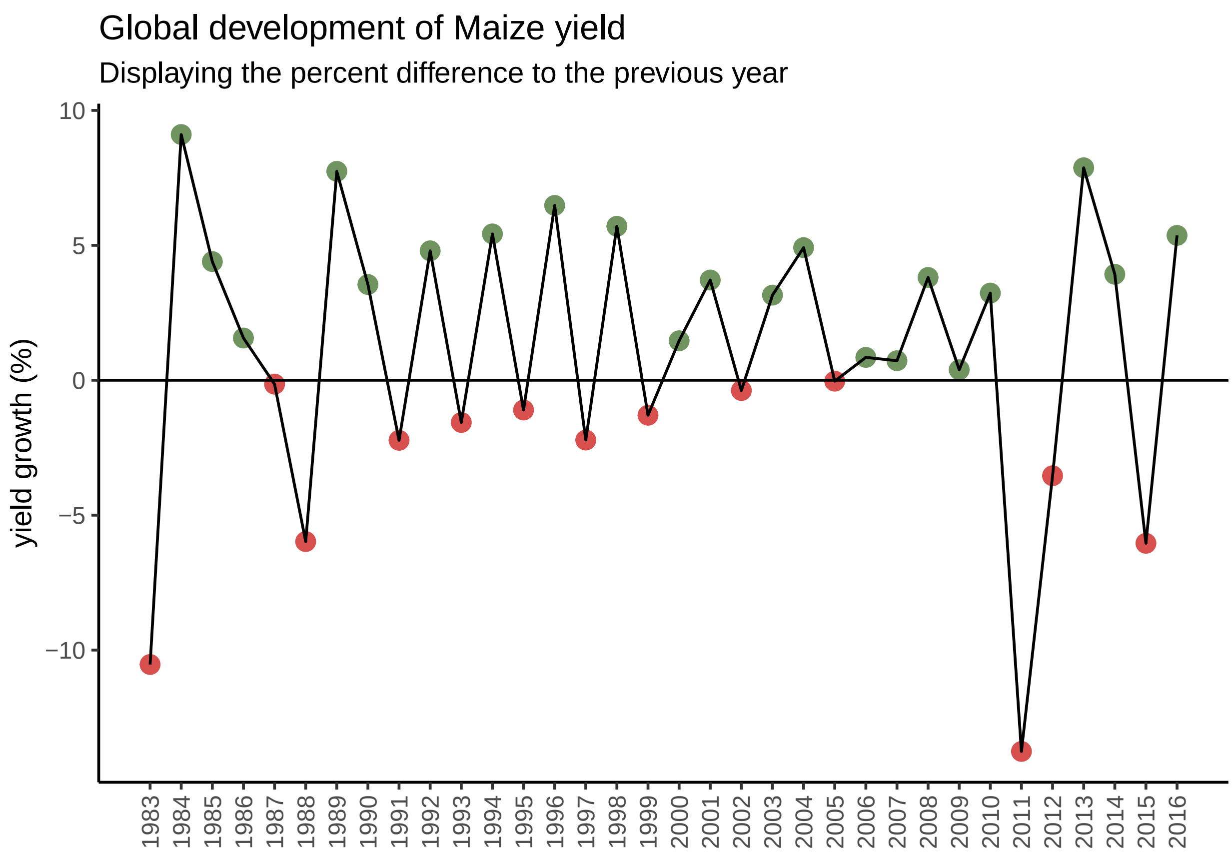
```
# Global growth (line chart)
global_yield_growth_line <- yield_summary %>%
  # Calculate the total amount of maize produced per year
  subset(year >= 1982 & year <= 2016) %>% # exclude 1981 from the analysis due to
                                          # incomplete data
  group_by(year) %>%
  summarise(global_yield = sum(sum_yield)) %>%
  # Calculate the difference in global yield to the preceding year in percent
  mutate(global_yield_delta = 100/global_yield * (global_yield - lag(global_yield)),
         # add a conditional coloring label
         color = case_when(global_yield_delta > 0 ~ "green",
                           global_yield_delta <= 0 ~ "red")) %>%
  # Create a plot of change in global yield (percent) vs year
  ggplot(aes(x = year, y = global_yield_delta)) +
    # Create a connected scatter plot and customize its appearance
    geom_point(aes(color = color), size = 3) +
    geom_line() +
```

```r
    # mark points >0 in green and <0 in red
    scale_color_manual(values = c("green" = "#6f9460", "red" = "#D7504D")) +
    theme_classic() +
    scale_y_continuous(name = "yield growth (%)") + # change label of y-axis
    scale_x_continuous(name = NULL, # delete x-axis label
                       limits = c(1983, 2016), # 1982 is excluded as it serves as basis
                                               # for the growth in 1983 but has no growth
                                               # rate, as no data is available from the
                                               # preceding year.
                       breaks = seq(1983, 2016, 1)) + # adjust the y-axis ticks and labels
    guides(x = guide_axis(angle = 90)) + # turn x-axis tick labels by 90 degree
    theme(legend.position = "none") + # delete the legend
    geom_hline(yintercept = 0) + # add a horizontal line at y=0
    # add a title and a subtitle
    ggtitle("Global development of Maize yield",
            subtitle = "Displaying the percent difference to the previous year")

# Show the plot
global_yield_growth_line
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```r
# Save the plot as a jpeg file
ggsave(device = "jpeg", "global_yield_growth_line", plot = global_yield_growth_line)
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 row(s) containing missing values (geom_path).
```

How ist the maize production divided among continents?

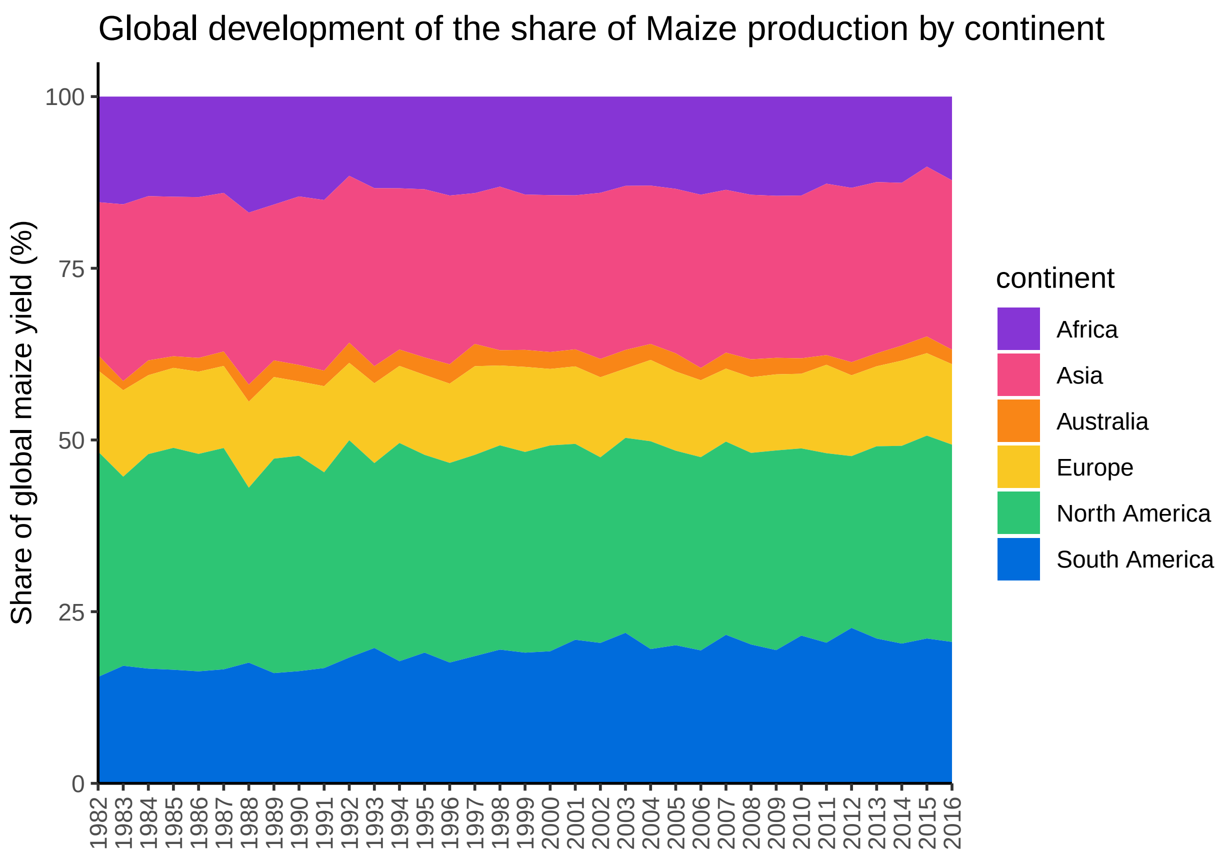
- Calculate the sum of the maize yield produced by each continent per year
- Calculate the proportion on global maize yield of each continent

```
# Create a custom color palette
Colors_continent <- c(Africa = "#8635D5",
                      Asia = "#F24982",
                      Australia = "#F98617",
                      Europe = "#F9C823",
                      "North America" = "#2DC574",
                      "South America" = "#006CDC")

# percentage on yield production by continent over time (stacked area chart)
global_yield_percentage_by_continent <- yield_summary %>%
  subset(year >= 1982 & year <= 2016) %>% # exclude 1981 from the analysis due to
                                          # incomplete data
  group_by(year, continent) %>%
  summarise(sum_yield_continent = sum(sum_yield)) %>%
  # Calculate the proportion on global maize yield of each continent in percent
  mutate(percent_global_yield = (100 / sum(sum_yield_continent)) * sum_yield_continent) %>%
  # Create a stacked area chart change showing the share of maize production by continent
  ggplot(aes(x = year, y = percent_global_yield, fill = continent)) +
    # Create a stacked area chart and customize its appearance
    geom_area() +
    theme_classic() +
    scale_y_continuous(name = "Share of global maize yield (%)", # change label of y-axis
                      limits = c(0, 105), # adjust the min and max of the y-axis
                      breaks = seq(0, 105, 25), # adjust the y-axis ticks and labels
                      labels = comma,
                      expand = c(0, 0)) +
    scale_x_continuous(name = NULL, # delete x-axis label
                       breaks = seq(1982, 2016, 1), # adjust the x-axis ticks and labels
                       expand = c(0, 0)) +
    guides(x = guide_axis(angle = 90)) + # turn x-axis tick labels
    # change the area colors to the custom color palette
    scale_fill_manual(values = Colors_continent) +
    # add a title
    ggtitle("Global development of the share of Maize production by continent")
```

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

```
# Show the plot
global_yield_percentage_by_continent
```




```
# Save the plot as a jpeg file
ggsave(device = "jpeg", "global_yield_percentage_by_continent",
       plot = global_yield_percentage_by_continent)
```

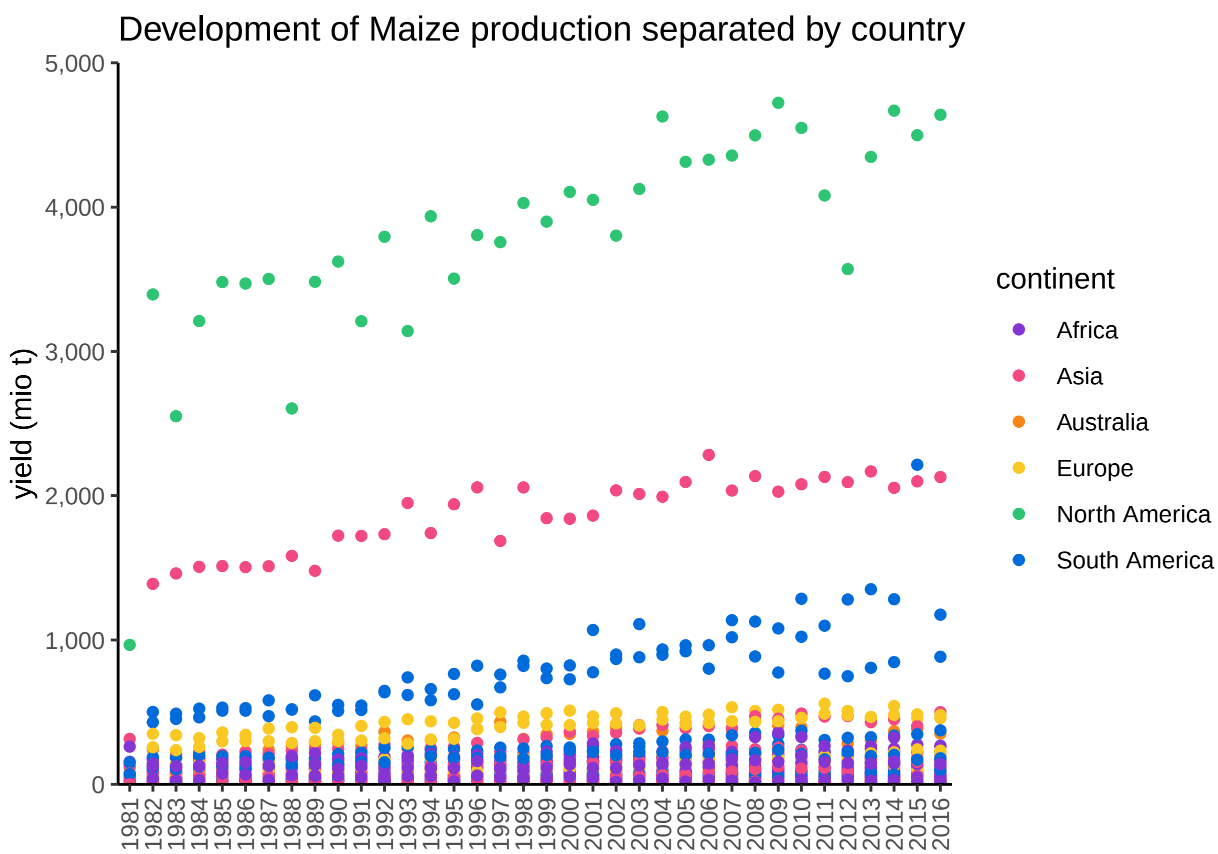
```
## Saving 6.5 x 4.5 in image
```

How is the maize production divided among countries?

```
# Global maize production divided by country (scatter plot)
overview_absolut_yield_by_country <- yield_summary %>%
  subset(sum_yield != 0) %>%
  # Create a scatter plot showing the maize production by country vs year
  ggplot(mapping = aes(x = year, y = sum_yield/1000000, color = continent)) +
    # Create a scatter plot and customize its appearance
    geom_point() +
    theme_classic() +
    scale_y_continuous(name = "yield (mio t)", # change label of y-axis
                       limits = c(0, 5000), # adjust the min and max of the y-axis
                       breaks = seq(0, 5000, 1000), # adjust the y-axis ticks and labels
                       labels = comma,
                       expand = c(0, 0)) +
    scale_x_continuous(name = NULL, # delete x-axis label
                        breaks = seq(1981, 2016, 1), # adjust the x-axis ticks and labels
                        expand = c(0, 0.5)) +
    guides(x = guide_axis(angle = 90)) + # turn x-axis tick labels
    # change point colors to the custom palette
```

```r
    scale_color_manual(values = Colors_continent) +
    ggtitle("Development of Maize production separated by country") # add a title

# Show the plot
overview_absolut_yield_by_country
```




```r
# Save the plot as a jpeg file
ggsave(device = "jpeg", "overview_absolut_yield_by_country",
       plot = overview_absolut_yield_by_country)
```

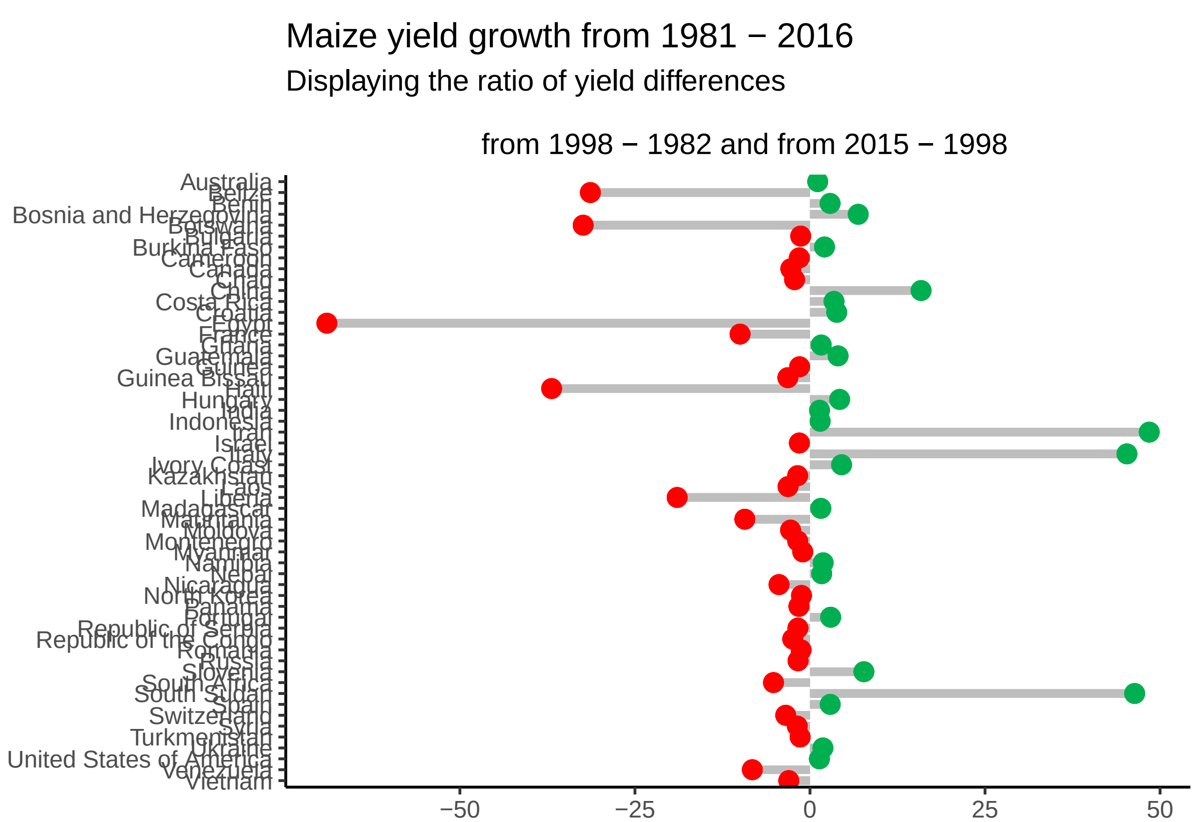
```
## Saving 6.5 x 4.5 in image
```

- Define three points from the timeseries (1982, 1998, 2015).
- Calculate the difference in yield between 1982 and 1998, as well as 1998 and 2015.
- Calculate the quotient of (1998 - 1982)/(2015 - 1998).

```r
# Difference in maize yield growth among countries (lollipop chart)
yield_summary_growth_lollipop <- yield_summary %>%
  dplyr::select(country, continent, sum_yield, year) %>%
  subset(year %in% c(1982, 1998, 2015)) %>% # Select three specific timepoints
  group_by(country) %>%
  arrange(year) %>%
  mutate(yield_1982 = sum_yield[year == 1982],
         yield_1998 = sum_yield[year == 1998],
         yield_2015 = sum_yield[year == 2015],
         yield_growth = (yield_1998 - yield_1982)/(yield_2015 - yield_1998)) %>%
  mutate(label = case_when(yield_growth > 0 ~ "green",
```

```r
                                TRUE ~ "red")) %>% # add a conditional coloring label
  subset(yield_growth > 1 | yield_growth < -1) %>%
  # Create a lollipop chart showing the yield growth by country
  ggplot(aes(x = country, y = yield_growth)) +
    # Create a lollipop chart and customize its appearance
    geom_segment(aes(x = country, xend = country, y = 0, yend = yield_growth),
                 color = "gray", # customize the lines of the lollipop chart
                 lwd = 1.5) +
    geom_point(aes(color = label), size = 3) + # add and customize points
    # change the colors of the points
    scale_color_manual(values = c("green" = "#00B050", "red" = "#FF0000")) +
    theme_classic() +
    theme(legend.position = "none") + # delete legend
    ylab("") + # delete y-axis label
    xlab ("") + # delete x-axis label
    scale_x_discrete(limits = rev) + # reverse the order of x-axis categories
    coord_flip() + # change x- and y-axis
    # add a title and subtitle
    ggtitle(label = "Maize yield growth from 1981 - 2016",
            subtitle = ("Displaying the ratio of yield differences \n
                        from 1998 - 1982 and from 2015 - 1998"))

# Show the plot
yield_summary_growth_lollipop
```



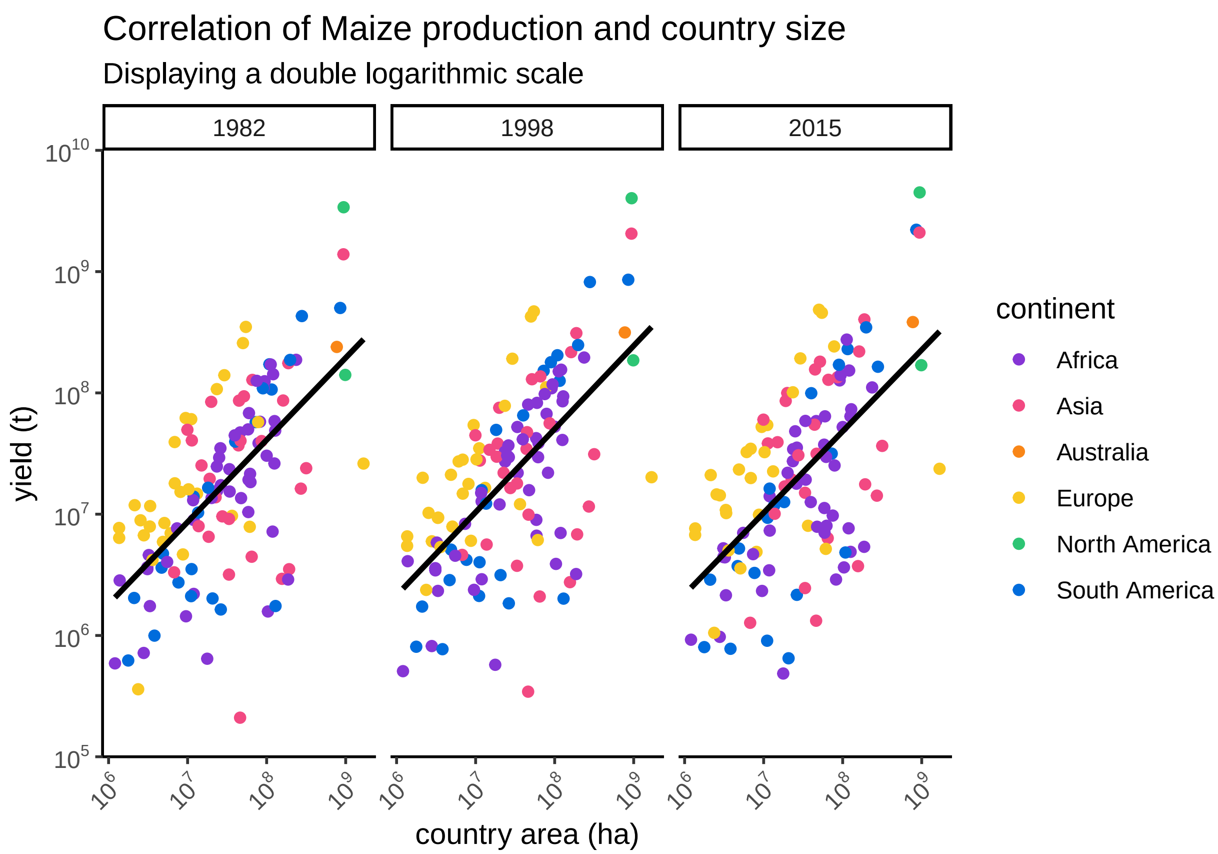
```
# Save the plot as a jpeg file
ggsave(device = "jpeg", "yield_summary_growth_lollipop",
       plot = yield_summary_growth_lollipop, width = 5, height = 7)
```

How does country size influence maize yield?

- Define three points from the timeseries (1982, 1998, 2015)
- Create a scatterplot of yield vs. country size for each year
- Add a regression line
- Fit a regression model

```
# Yield vs country area in three distinct years, double logarithmic scale (scatter plot)
overview_absolut_log_yield_size <- yield_summary %>%
  subset(sum_yield != 0) %>%
  subset(year %in% c(1982, 1998, 2015)) %>% # Select three specific timepoints
   # Create a plot of yield vs country area for each selected year
  ggplot(mapping = aes(x = log10(country_area),
                       y = log10(sum_yield), # double logarithmic scale
                       color = continent)) +
    # Create scatter plot and customize its appearance
    geom_point() +
    facet_wrap("year") + # create three plots (one for each year selected before)
    theme_classic() +
    scale_y_continuous(name = "yield (t)", # change the label of the y-axis
                     limits = c(5, 10), # change min and max of the y-axis
                     # change the appearance of th y-axis tick labels
                     labels = c(expression(10^5), expression(10^6), expression(10^7),
                                expression(10^8), expression(10^9), expression(10^10)),
                     expand = c(0, 0)) +
    scale_x_continuous(name = "country area (ha)", # change the label of the x-axis
                       # change the appearance of th x-axis tick labels
                       labels = c(expression(10^6), expression(10^7),
                                  expression(10^8), expression(10^9))) +
    guides(x = guide_axis(angle = 45)) + # turn x-axis tick labels by 45 degree
    # add a regression line
    geom_smooth(method = "lm", se=FALSE, color="black", formula = y ~ x) +
    # change the color to the custom color palette
    scale_color_manual(values = Colors_continent) +
    # add a title and a subtitle
    ggtitle("Correlation of Maize production and country size",
            subtitle = "Displaying a double logarithmic scale")

# Show the plot
overview_absolut_log_yield_size
```




```r
# Save the plot as a jpeg file
ggsave(device = "jpeg", "overview_absolut_log_yield_size",
       plot = overview_absolut_log_yield_size)
```

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

```r
# Fit a regression model
regression_1982 <- yield_summary %>%
  subset(sum_yield != 0) %>%
  subset(year == "1982") %>%
  lm(log10(sum_yield) ~ log10(country_area), .)

# Show summary statistics
print(summary(regression_1982))
```

```
##
## Call:
## lm(formula = log10(sum_yield) ~ log10(country_area), data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.06283 -0.28963  0.06887  0.41487  1.25926
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.19701    0.56927   3.859 0.000182 ***
## log10(country_area)  0.67692    0.07622   8.881 6.28e-15 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5926 on 124 degrees of freedom
## Multiple R-squared:  0.3888, Adjusted R-squared:  0.3839
## F-statistic: 78.88 on 1 and 124 DF,  p-value: 6.281e-15
```

```
# Add columns (predictions, residuals and cluster assignments)
# to the original dataset based on the statistical model
res_1982 <- augment(regression_1982)
```

```
# Fit a regression model
regression_1998 <- yield_summary %>%
  subset(sum_yield != 0) %>%
  subset(year == "1998") %>%
  lm(log10(sum_yield) ~ log10(country_area), .)

# Show summary statistics
print(summary(regression_1998))
```

```
##
## Call:
## lm(formula = log10(sum_yield) ~ log10(country_area), data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9354 -0.3140  0.1239  0.3942  1.2345
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.21459    0.57918   3.824 0.000207 ***
## log10(country_area)  0.68601    0.07755   8.846 7.61e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6029 on 124 degrees of freedom
## Multiple R-squared:  0.3869, Adjusted R-squared:  0.382
## F-statistic: 78.26 on 1 and 124 DF,  p-value: 7.609e-15
```

```
# Add columns (predictions, residuals and cluster assignments)
# to the original dataset based on the statistical model
res_1998 <- augment(regression_1998)
```

```
# Fit a regression model
regression_2015 <- yield_summary %>%
  subset(sum_yield != 0) %>%
  subset(year == "2015") %>%
  lm(log10(sum_yield) ~ log10(country_area), .)

# Show summary statistics
print(summary(regression_2015))
```

```
##
## Call:
## lm(formula = log10(sum_yield) ~ log10(country_area), data = .)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4916 -0.4616  0.0380  0.4899  1.3154 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          2.30704    0.61905   3.727 0.000295 ***
## log10(country_area)  0.67203    0.08284   8.113 4.46e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.6325 on 122 degrees of freedom
## Multiple R-squared:  0.3504, Adjusted R-squared:  0.3451 
## F-statistic: 65.81 on 1 and 122 DF,  p-value: 4.464e-13
```

```r
# Add columns (predictions, residuals and cluster assignments)
# to the original dataset based on the statistical model
res_2015 <- augment(regression_2015)
```

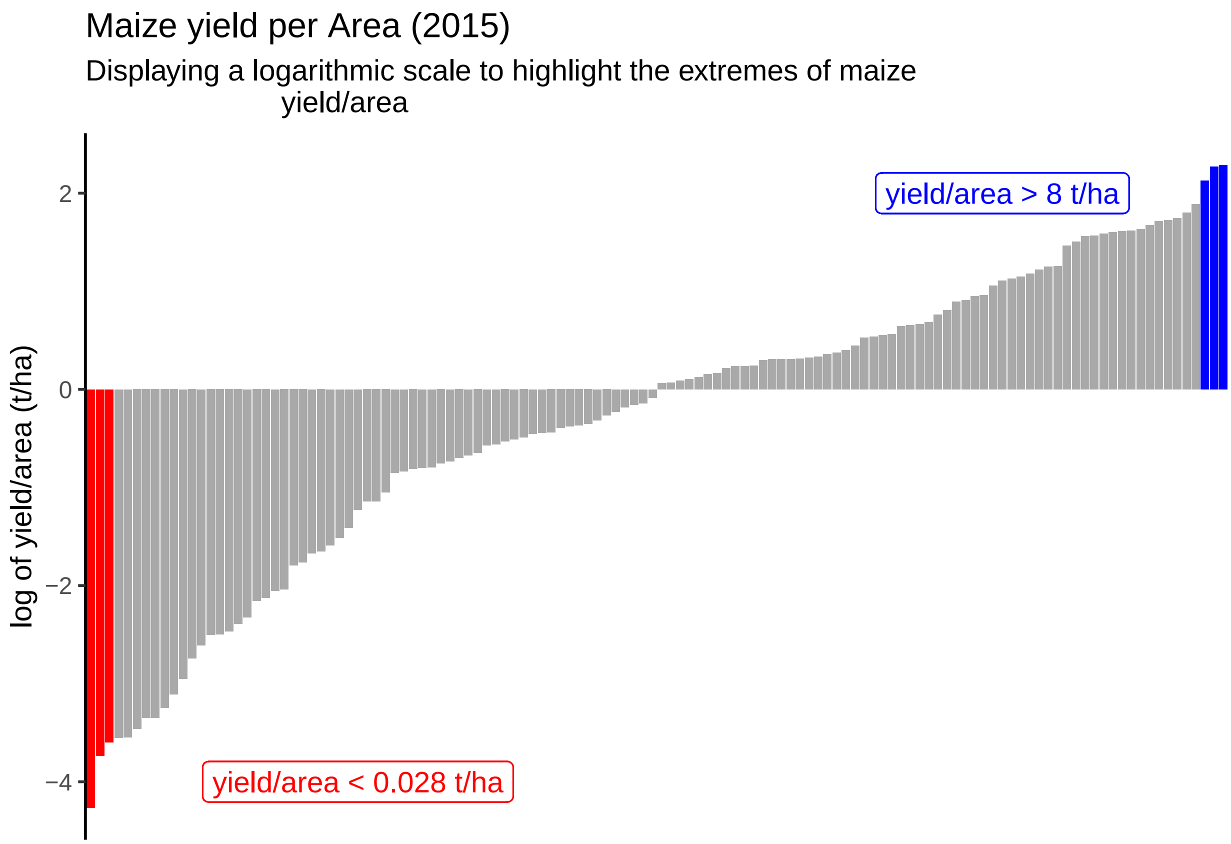
=> A 1 % increase in country area is associated with an approximately 68 % increase in maize yield in each year under investigation.

- Select one time point (2015)
- Create a barplot of yield/area vs. country

```r
# yield/area vs country in 2015 (Bar plot)
yield_area_vs_country_plot <- yield_summary %>%
  subset(sum_yield != 0) %>%
  subset(year == 2015) %>%
  arrange(desc(yield_per_area)) %>%
  # add a column based on rank of the yield_per_area
  mutate(color = case_when(rank(-yield_per_area) <= 3  ~ "blue",
                           rank(yield_per_area) <= 3 ~ "red",
                           TRUE ~ "grey")) %>%
  # Create a plot of yield/area vs country for the selected year
  ggplot(aes(x = reorder(country, yield_per_area),
             y = log(yield_per_area),
             fill = color)) +
    # Create a bar plot and customize its appearance
    geom_bar(stat = "identity") +
    # Change the colors of the bars
    scale_fill_manual(values = c("blue", "darkgrey", "red")) +
    theme_classic() +
    ylab("log of yield/area (t/ha)") + # Change the label of the y-axis
    xlab(NULL) + # Delete the label of the x-axis
    theme(legend.position = "none") + # Delete the legend
    theme(axis.line.x = element_blank(), # Delete the x-axis
          axis.text.x = element_blank(), # Delete the x-axis tick labels
          axis.ticks.x = element_blank()) + # Delete the x-axis ticks
    geom_label(aes(x = 30, y = -4, label = "yield/area < 0.028 t/ha"),
               color = "red", fill = "white") + # Add a label explaining the coloring
    geom_label(aes(x = 100 ,y = 2, label = "yield/area > 8 t/ha"),
               color = "blue", fill = "white") + # Add a label explaining the coloring
    # add a title and a subtitle
    ggtitle("Maize yield per Area (2015)",
            subtitle = "Displaying a logarithmic scale to highlight the extremes of maize
```

```
                         yield/area")

# Show the plot
yield_area_vs_country_plot
```




```
# Save the plot as a jpeg file
ggsave(device = "png", "yield_area_vs_country_plot",
       plot = yield_area_vs_country_plot, width = 9, height = 4)
```

Which are the 6 countries with the highest/lowest yield/area value?

- Create a dataframe
- Filter the countries with the highest/lowest yield/area values based on their rank

```
# write a dataframe that contains the 6 countries with the highest/lowest yield/area value
yield_top_low_performer_yield_per_area_subset <- yield_summary %>%
  subset(sum_yield != 0)%>%
  subset(year == 2015) %>%
  arrange(desc(yield_per_area)) %>%
  dplyr::select(country, sum_yield, country_area , yield_per_area) %>%
  # filter based on the rank of the yield/area value
  filter(rank(-yield_per_area) <= 3 | rank(yield_per_area) <= 3)
```

```
## Adding missing grouping variables: `year`
```

```
# Show the resulting dataframe
yield_top_low_performer_yield_per_area_subset
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year country     sum_yield country_area yield_per_area
##   <dbl> <chr>           <dbl>        <dbl>          <dbl>
## 1  2015 Slovenia    21018759.     2140689.         9.82
## 2  2015 Spain      485008748.    50150367.         9.67
## 3  2015 France     457795832.    54571626.         8.39
## 4  2015 Somaliland    485614.    17702955.         0.0274
## 5  2015 Mongolia     3731034.   156347560.         0.0239
## 6  2015 Russia      23674644.  1682390531.         0.0141
```

```
# Save the dataframe as a .csv file
write.csv2(yield_top_low_performer_yield_per_area_subset,
           file ="yield_top_low_performer_yield_per_area_subset.csv")
```

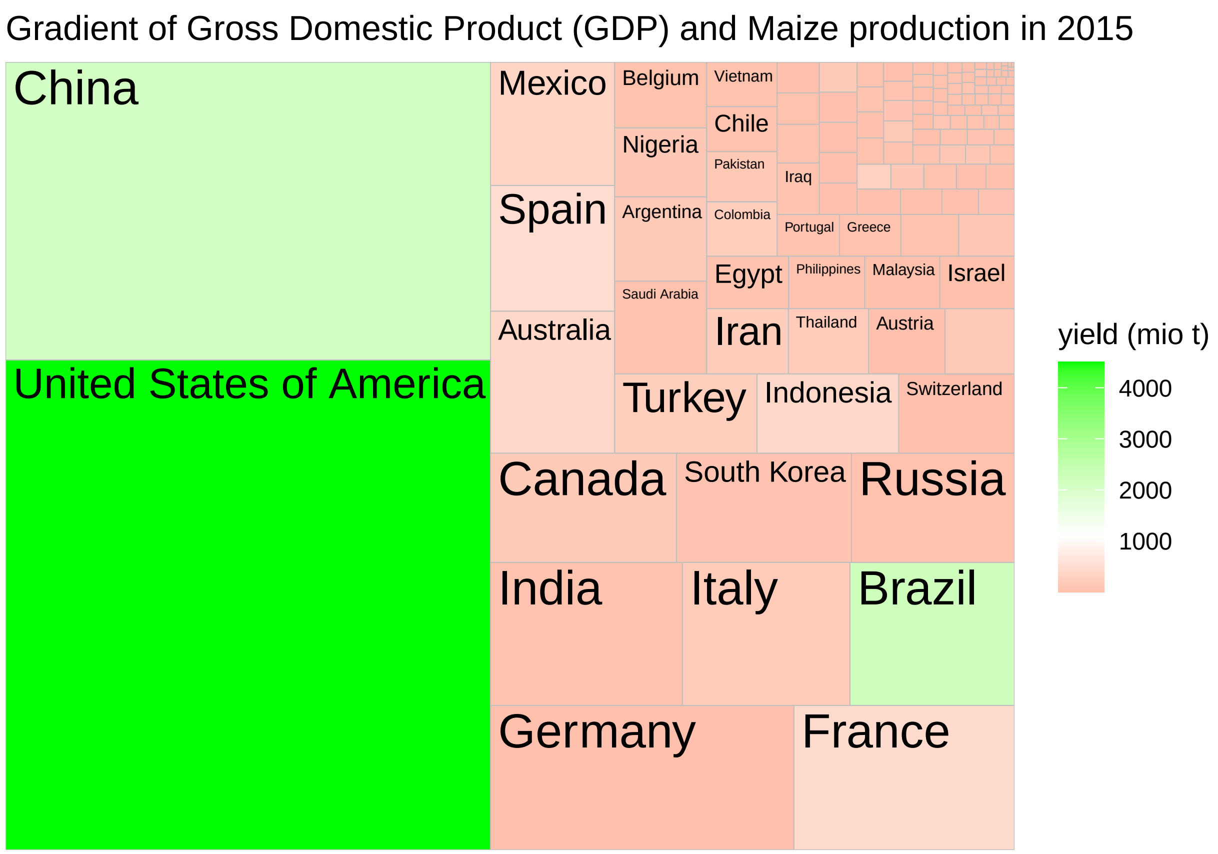
How is prosperity distributed among maize producing countries?

- Join the yield_summary and demographic_data dataframes into one
- Create a treemap in which box size represents the GDP and color gradient represents yield

```
# Create a treemap with GDP as boxsize and yield as heatmap (red to green = low to high)
yield_GDP_yield_treemap <- demographic_data %>%
  right_join(., yield_summary) %>%
  subset(sum_yield != 0 & gdp != "NA" & year == 2015) %>%
  # Create a plot with GDP as boxsize and yield as heatmap
  ggplot(aes(area = gdp, fill = sum_yield/1000000, label = country)) +
    # Create a treemap and customize its appearance
    geom_treemap() +
    geom_treemap_text() + # Add a text label to each tile
    scale_fill_gradient2(low = "red", # Customize the color gradient
                        mid = "white",
                        high = "green",
                        guide = "colorbar",
                        midpoint = 1100) +
    labs(fill = "yield (mio t)") + # Change the legend title
     # Add a title
    ggtitle("Gradient of Gross Domestic Product (GDP) and Maize production in 2015")
```

```
## Joining, by = c("country", "year")
```

```
# Show the plot
yield_GDP_yield_treemap
```




```r
# Save the plot as a jpeg file
ggsave(device = "jpeg", "yield_GDP_yield_treemap",
       plot = yield_GDP_yield_treemap)
```

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

- Join the yield_summary and demographic_data dataframes into one
- Assign ranks for the amount of maize produced and gdp per country

```r
# Create a Slope Chart showing the top10 countries in yield and GDP respectively and the
# respective other rank
yield_GDP_slopechart <- demographic_data %>%
  right_join(., yield_summary) %>%
  subset(sum_yield != 0 & gdp != "NA") %>%
  subset(year == 2015) %>%
  # Assign a rank based on the amount of maize produced
  mutate(rank_yield = rank(-sum_yield),
         # Assign a rank based on the gdp
         rank_gdp = rank(-gdp)) %>%
  subset(rank_yield <= 10 | rank_gdp <= 10) %>%
  # lengthen the data, turning columns of the ranks for yield and gdp into one column
  pivot_longer(cols = c(rank_yield, rank_gdp),
               names_to = "rank_groups",
               values_to = "ranks") %>%
  # add a column based on rank of the yield_per_area
  mutate(cy_label = case_when(rank_groups == "rank_yield" ~ "",
                              TRUE ~ country)) %>%
```

```r
  # Create a plot ranks vs. rank_groups
  ggplot(aes(x = rank_groups, y = -ranks, color = country, group = country)) +
    # Create a slope chart and customize its appearance
    geom_point() +
    geom_text(aes(label = cy_label), # Add country labels to the points
              nudge_x = 0.05,
              size = 3,
              hjust = 0) +
    geom_line() + # connect the rank in yield and gdp for each country
    theme_classic() +
    theme(legend.position = "none") + # remove the legend
    scale_y_continuous(name = "Global Rank", # change the label of the y-axis
                       limits = c(-55, 0), # change min and max of y-axis
                       expand = c(0, 0),
                       # change tick labels of the y-axis
                       labels = c("50", "40", "30", "20", "10", "0")) +
    scale_x_discrete(limits = c("rank_yield",
                                "rank_gdp"), # change the order of x-axis groups
                     label = c("Yield", "GDP"), # change tick labels of the x-axis
                     name = NULL) + # delete the x-axis label
    # add a title
    ggtitle("Gradient of Maize production and Gross Domestic Product (GDP) in 2015")
```

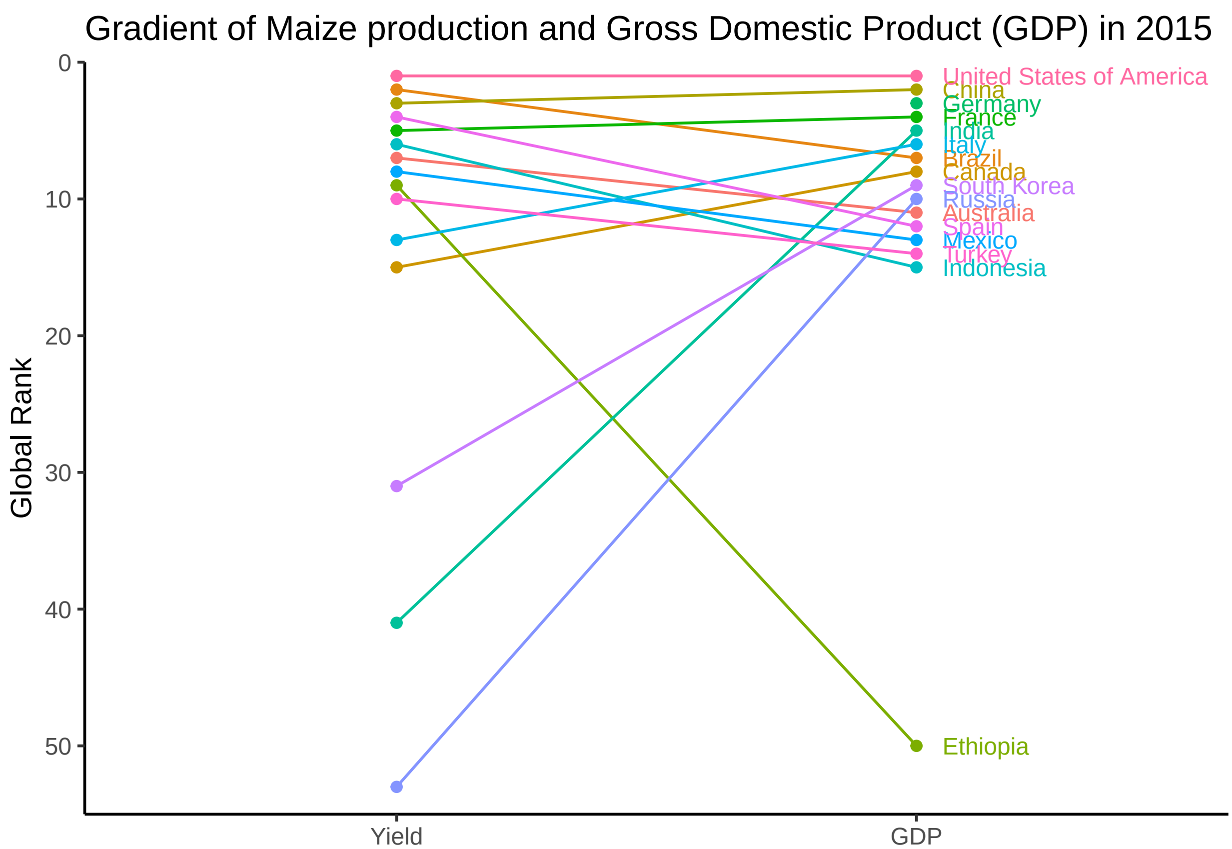
```
## Joining, by = c("country", "year")
```

```r
# Show the plot
yield_GDP_slopechart
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_text).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
# Save the plot as a jpeg file
ggsave(device = "png", "yield_GDP_slopechart",
       plot = yield_GDP_slopechart, width = 6, height = 7)
```

```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_text).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```

How did maize production and population change among countries?

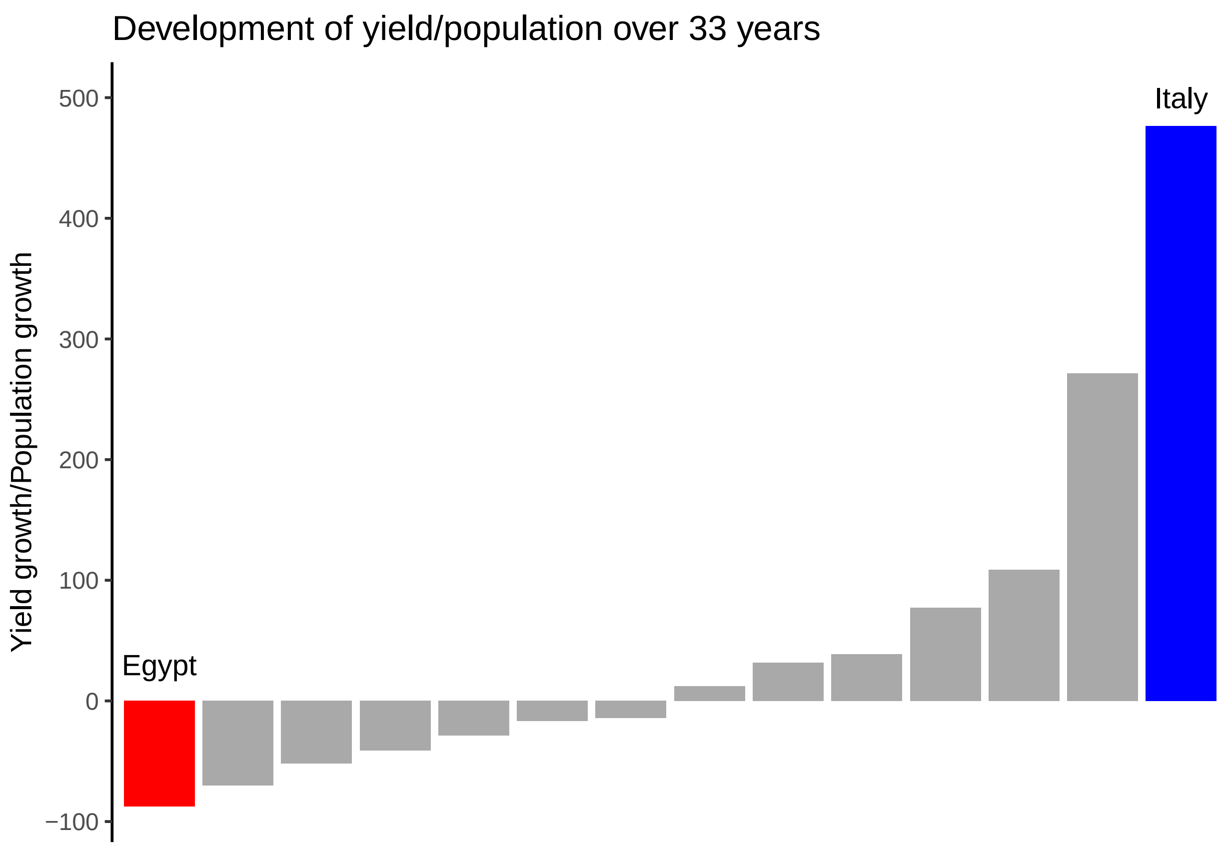
- Define three points from the timeseries (1982, 1998, 2015).
- Calculate the difference in yield and population between 1982 and 1998, as well as 1998 and 2015.
- Calculate the quotient of (1998 - 1982)/(2015 - 1998).
- Calculate yield_growth/population_growth

```
# yield growth/population growth vs country (Bar plot)
yield_population_growth <- demographic_data %>%
  right_join(., yield_summary) %>%
  dplyr::select(country, continent, sum_yield, year, population) %>%
  subset(year == 1982 | year == 1998 | year == 2015) %>%
  group_by(country) %>%
  summarise(yield_1982 =
              sum_yield[year == "1982"],
            yield_1998 =
              sum_yield[year == "1998"],
            yield_2015 =
```

```r
            sum_yield[year == "2015"],
          population_1982 =
            population[year == "1982"],
          population_1998 =
            population[year == "1998"],
          population_2015 =
            population[year == "2015"],
          yield_growth =
            (yield_1998 - yield_1982)/(yield_2015 - yield_1998),
          population_growth =
            (population_1998 - population_1982)/(population_2015 - population_1998),
          yield_population_growth = yield_growth/population_growth) %>%
  subset(yield_population_growth > 10 | yield_population_growth < -10) %>%
  # add a column based on rank of yield_population growth
  mutate(color = case_when(yield_population_growth == max(yield_population_growth) ~ "blue",
                           yield_population_growth == min(yield_population_growth) ~ "red",
                           TRUE ~ "grey")) %>%
  # Create plot yield growth/ population growth vs country
  ggplot(aes(x = reorder(country, yield_population_growth), # reorder the countries on the
                                                            # x-axis based on their y-value
             y = yield_population_growth,
             fill = color)) +
    # Create a bar plot and customize its appearance
    geom_bar(stat = "identity") +
    # change the color of the bars
    scale_fill_manual(values = c("blue", "darkgrey", "red")) +
    theme_classic() +
    guides(x = guide_axis(angle = 90)) + # turn the x-axis tick labels by 90 degree
    ylab("Yield growth/Population growth") + # change the y-axis label
    annotate(geom="text", x = "Egypt", y=30, label="Egypt",
             color="black") + # add a text label
    annotate(geom="text", x = "Italy", y=500, label="Italy", # add a text label
             color="black") +
    theme(legend.position = "none") + # delete the legend
    theme(axis.title.x = element_blank(), # delete the x-axis title
          axis.text.x = element_blank(), # delete the x-axis tick labels
          axis.ticks.x = element_blank(), # delete the x-axis ticks
          axis.line.x = element_blank()) + # delete the x-axis
    # Add a title
    ggtitle("Development of yield/population over 33 years")
```

```
## Joining, by = c("country", "year")
```

```r
# Show the plot
yield_population_growth
```




```r
# Save the plot as a jpeg file
ggsave(device = "png", "yield_population_growth",
       plot = yield_population_growth)
```

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

- Join the yield_summary and demographic_data dataframes into one
- Select data for egypt

```r
# Case study Egypt (Scatter plot)
yield_population_development_egypt <- demographic_data %>%
  right_join(., yield_summary) %>%
  filter(country == "Egypt") %>%
  # Create a plot population vs year and yield vs year
  ggplot(aes(x = year)) +
    # Create a scatter plot with two y-axis
    geom_point(aes(y = population), colour = "black") +
    geom_point(aes(y = sum_yield/1.8), colour = "red") + # Adjust values of second y-axis
    theme_classic() +
    scale_y_continuous(name = "Population", # change the y-axis label
                       limits = c(0, 100000000), # change min and max of the y-axis
                       labels = comma,
                       expand = c(0, 0),
                       sec.axis = sec_axis(~. * 1.8, # adjust values of second y-axis
                                           name = "Yield (t)", # change second y-axis label
                                           labels = comma)) +
    scale_x_continuous(name = NULL,
```

```
                         breaks = seq(1981, 2016, 1)) +
    guides(x = guide_axis(angle = 90)) + # turn the x-axis tick labels by 90 degree
    ggtitle("Egypts development of population and maize production ") + # add a title
    theme(axis.title.y = element_text(color = "black"), # change color of y-axis label
         axis.title.y.right = element_text(color = "red")) # change color of second y-axis
```

```
## Joining, by = c("country", "year")
```

```
                                                                    # label

# Show the plot
yield_population_development_egypt
```


```
# Save the plot as a jpeg file
ggsave(device = "jpeg", "yield_population_development_egypt",
       plot = yield_population_development_egypt)
```

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

- Join the yield_summary and demographic_data dataframes into one
- Select data for italy

```
# Case study Italy
yield_population_development_italy <- demographic_data %>%
  right_join(., yield_summary) %>%
  filter(country == "Italy") %>%
  # Create a plot population vs year and yield vs year
  ggplot(aes(x = as.numeric(year))) +
```

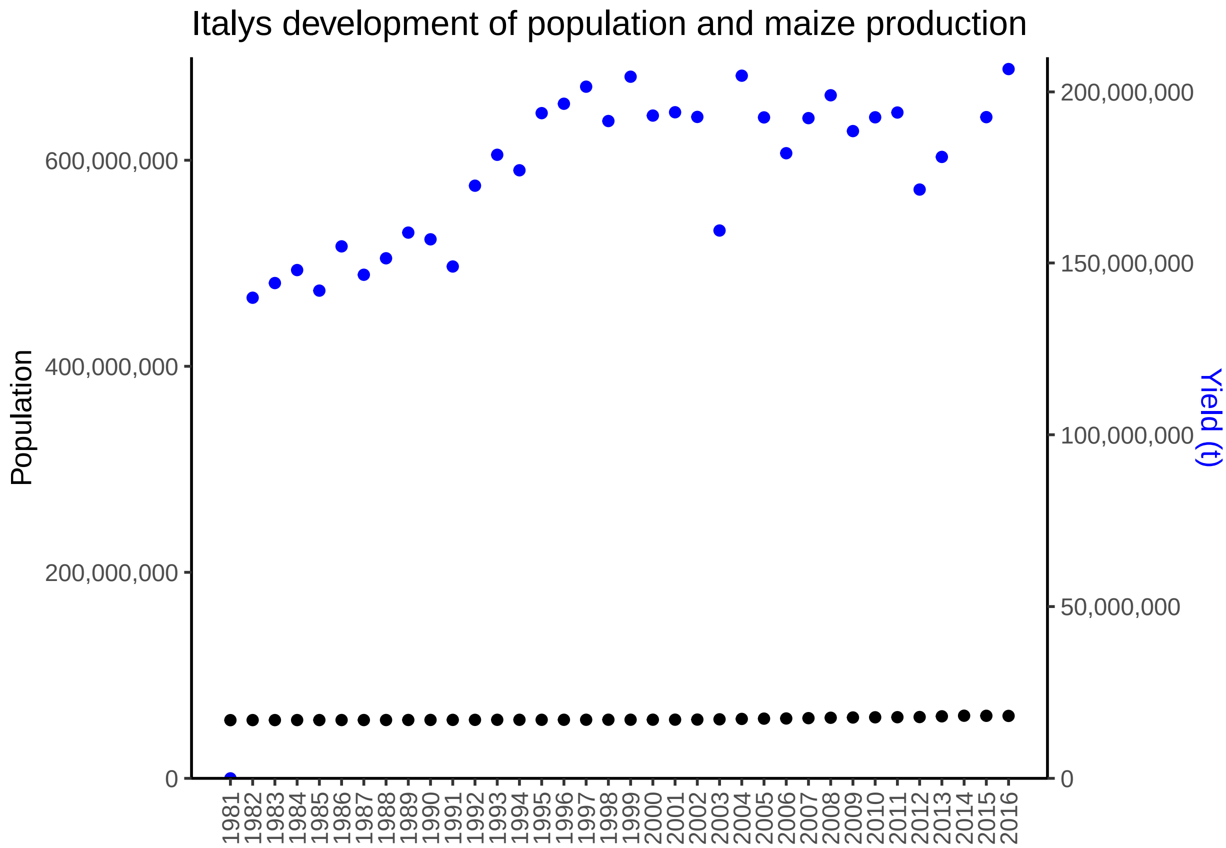
```r
    # Create a scatter plot with two y-axis
    geom_point(aes(y = as.numeric(population)), colour = "black") +
    # Adjust the values of the second y-axis
    geom_point(aes(y = as.numeric(sum_yield)/0.3), colour = "blue") +
    theme_classic() +
    scale_y_continuous(name = "Population", # change the y-axis label
                       limits = c(0, 700000000), # change min and max of the y-axis
                       labels = comma,
                       expand = c(0, 0),
                       sec.axis = sec_axis(~. * 0.3,# adjust values of the second y-axis
                                           name = "Yield (t)", # change second y-axis label
                                           labels = comma)) +
    scale_x_continuous(name = NULL,
                       breaks = seq(1981, 2016, 1)) +
    guides(x = guide_axis(angle = 90)) + # turn the x-axis tick labels by 90 degree
    # add a title
    ggtitle("Italys development of population and maize production ") +
    theme(axis.title.y = element_text(color = "black"), # change color of y-axis label
          axis.title.y.right = element_text(color = "blue")) # change color of second
```

```
## Joining, by = c("country", "year")
```

```r
                                                                      # y-axis label

# Show the plot
yield_population_development_italy
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
# Save the plot as a jpeg file
ggsave(device = "jpeg", "yield_population_development_italy",
       plot = yield_population_development_italy)
```

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

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

Case study germany

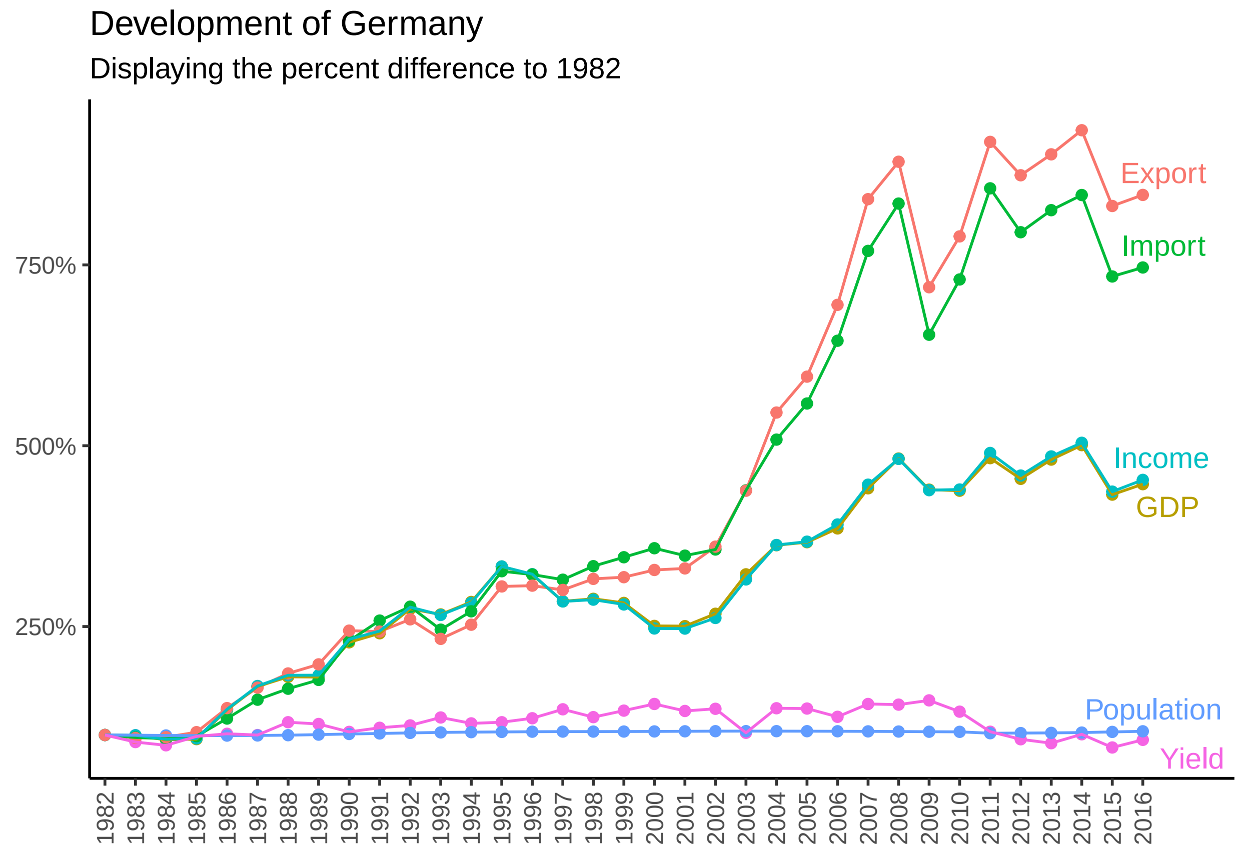
- Join the yield_summary and demographic_data dataframes into one
- Select data for germany
- normalize the data for yield, population, gdp, income, import and export to the value of 1982

```
# Case study Germany
development_germany <- demographic_data %>%
  right_join(., yield_summary) %>%
  filter(country == "Germany" & year > 1981) %>%
  # normalize the data for yield, population, gdp, income, import and export
  # to the value of 1982
  mutate(percent_yield = 100 / sum_yield[year == 1982] * sum_yield,
         percent_pop = 100/population[year == 1982]*population,
         percent_gdp = 100/gdp[year == 1982]*gdp,
         percent_income = 100/income[year == 1982]*income,
         percent_import = 100/import[year == 1982]*import,
         percent_export = 100/export[year == 1982]*export) %>%
  # lengthen the data, turning columns of the percents for yield, population, gdp,
  # income, import and export into one column
```

```r
  pivot_longer(cols = c(percent_yield,
                        percent_pop,
                        percent_gdp,
                        percent_income,
                        percent_import,
                        percent_export),
               names_to = "percent_groups",
               values_to = "percents") %>%
  mutate(label = case_when(year == 2016 & percent_groups == "percent_yield" ~ "Yield",
                           year == 2016 & percent_groups == "percent_pop" ~ "Population",
                           year == 2016 & percent_groups == "percent_gdp" ~ "GDP",
                           year == 2016 & percent_groups == "percent_income" ~ "Income",
                           year == 2016 & percent_groups == "percent_import" ~ "Import",
                           year == 2016 & percent_groups == "percent_export" ~ "Export",
                           TRUE ~ "")) %>% # Create a label for each category
  # Create a plot percents vs year
  ggplot(aes(x = year, y = percents, color = percent_groups)) +
    # Create a connected scatter plot and customize its appearance
    geom_point() +
    geom_line() +
    theme_classic() +
    scale_y_continuous(name = NULL, # delete the y-axis label
                       # change appearance of y-axis tick labels
                       labels = percent_format(scale = 1)) +
    scale_x_continuous(name = NULL, # delete the x-axis label
                       breaks = seq(1982, 2016, 1),
                       expand = expansion(add = c(0.5, 3))) +
    guides(x = guide_axis(angle = 90)) + # turn the x-axis tick labels by 90 degree
    # Add a title and subtitle
    ggtitle("Development of Germany",
            subtitle = "Displaying the percent difference to 1982") +
    geom_text_repel(aes(label = label, hjust = 0.7)) + # Display the category labels
    theme(legend.position = "none") # delete the legend
```

```
## Joining, by = c("country", "year")
```

```r
# Show the plot
development_germany
```




```
# Save the plot as a jpeg file
ggsave(device = "jpeg", "development_germany",
       plot = development_germany, width = 7, height = 5)
```