

Homework 4

Nadine Thomas

Part 1

Playfair's graph uses line and color well. In this graph he uses a red line to represent the population of a country, while a yellow line represents taxation. The use of line for these two variables works well because it is easy to interpret that the longer the line, the bigger the population or taxation is. The use of color is good here because it simply highlights that there are two different variables being compared. Although, since color is also used to differentiate between maritime and land power, I don't think the use of color is great. Using color for different variables and different categories of variables is not ideal.

The main downfall of this graph is shape and size. Circles are used to represent the area of different countries, which can be misleading because area of a circle does not scale linearly. This also creates a large gap in space between the lines for population and taxation. The dashed line in between the two aids the comparison, however it is not easy to compare between countries. Overall, this graph does show that some countries are taxed more than others, but can be confusing.

Part 2

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    4.0.0      v tibble     3.3.0
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.1.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()      masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(ggplot2)
library(ggforce)
europe <- read.csv("playfair_european_nations.csv")
```

```
europe_remake <- europe %>%
  # Reorder countries by area
  mutate(Country = fct_reorder(Country, Area, .desc = TRUE),
    # Split the country names up
    CountrySplit = Country %>%
      str_replace("Emperor's Dominions", "Emperor's\nDominions") %>%
      str_replace("German Empire", "German\nEmpire") %>%
      str_replace("Britain & Ireland", "Britain\n&\nIreland") %>%
      str_replace("Countries under the Dominion of France",
        "Countries under the\nDominion of France"),
    # Make the first two countries names horizontal and the rest vertical
    label_angle = ifelse(row_number() <= 2, 0, 90),
    # Calculate the radius (210 made the circles fit the best for me)
    r = sqrt(Area / pi) / 210)

# Find x0 positions
# Initialize a vector whose length is the total amount of entries
x_positions <- numeric(nrow(europe_remake))
# Initialize the first position as 0
x_positions[1] <- 0
# Padding for space in between circles
padding <- 2

# Since the first element already has a value start with the second element
# and continue until the end of the dataset
for (i in 2:nrow(europe_remake)) {
  # The position of the current element is the position of the previous element
  # plus its radius, plus the radius of the current element, plus padding
  x_positions[i] <- x_positions[i-1] + europe_remake$r[i-1] +
    europe_remake$r[i] + padding
}

# Create a new column x0 and fill it with the positions found earlier
europe_remake$x0 <- x_positions
```

```

# Plot recreation
ggplot(europe_remake) +
  # Circles
  geom_circle(aes(x0 = x0, y0 = 0, r = r, fill = Power), color = "black") +

  # Vertical line for population, subtracting by the radius so that it is on
  # the left side
  geom_segment(aes(x = x0 - r, xend = x0 - r, y = 0, yend = Population),
               size = 1, color = "#d79883") +

  # Vertical line for taxation, adding by the radius so that it is on the
  # right side
  geom_segment(aes(x = x0 + r, xend = x0 + r, y = 0, yend = Taxation),
               size = 1, color = "#debf91") +

  # Line connecting the population and taxation
  geom_segment(aes(x = x0 - r, xend = x0 + r, y = Population, yend = Taxation),
               linetype = "longdash") +

  # Country names under circles
  geom_text(aes(x = x0, y = -r - 5, label = CountrySplit, angle = label_angle),
            size = 4) +

  # Y-axis from 0-30 on both sides
  scale_y_continuous(breaks = seq(0, 30, by = 1), sec.axis = dup_axis())+

  # Setting the color of power
  scale_fill_manual(values = c("Maritime" = "#a1896d", "Land" = "#dba188"))+

  labs(title = "CHART Representing the EXTENT, POPULATION & REVENUES, of the PRINCIPAL NATION
after the DIVISION of POLAND & TREATY of LUNEVILLE.") +

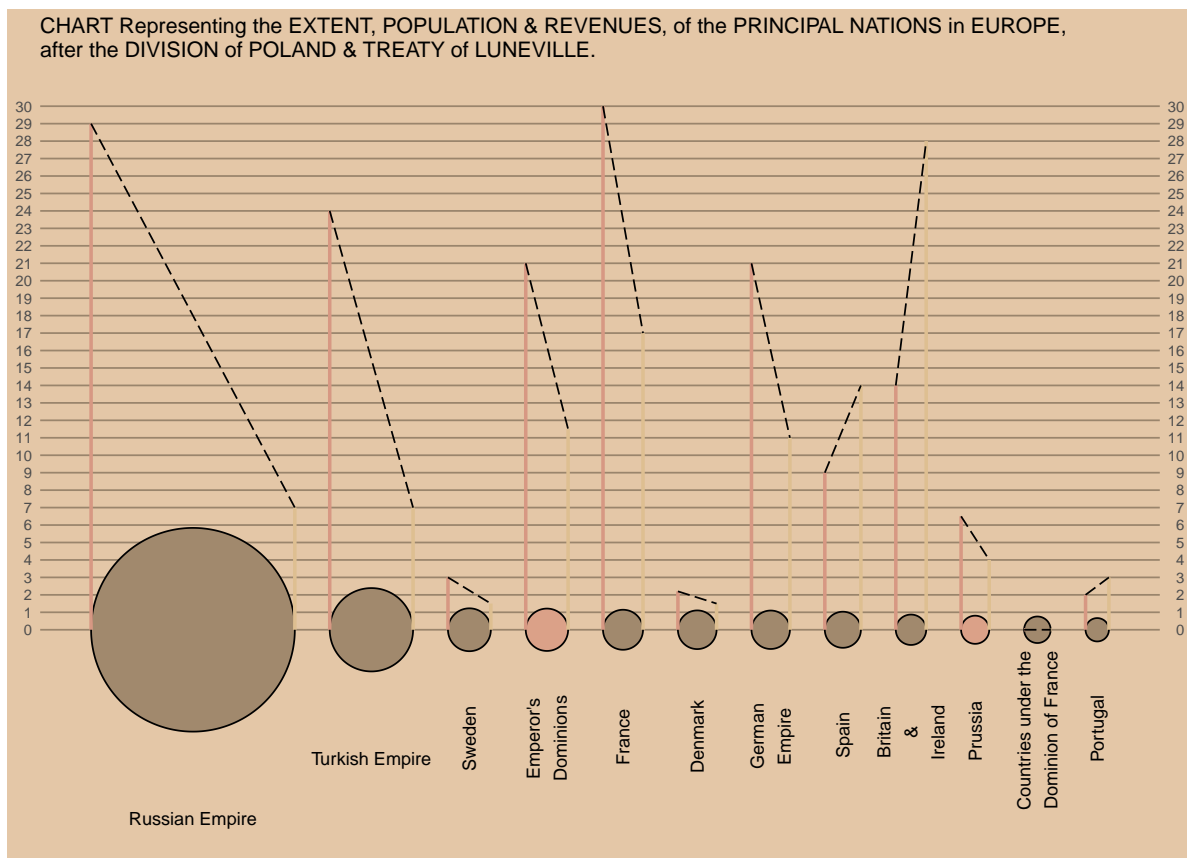
  # Remove x-axis numbers, y-axis label, x-axis grid lines, y-axis minor grid
  # lines and legend for power. Color y-axis major grid lines and background
  theme_minimal() +
  theme(
    axis.title = element_blank(),
    axis.text.x = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_line(color = "#9a856c"),
    panel.grid.minor.y = element_blank(),

```

```
legend.position = "none",
plot.background = element_rect(fill = "#e4c7a7")
) +
```

```
# Recommended from geom_circle
coord_fixed(clip = "off")
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.



Part 2 Sources

?geom_circle

?geom_segment

?geom_text

?scale_y_continuous – looked at breaks and sec.axis

?sec_axis – from scale_y_continuous

?theme – found axis.tile, axis.text.x, legend.position, panel.grid.major.x, panel.grid.minor.x, panel.grid.major.y, panel.grid.minor.y, and plot.background

?element_text – from axis.title and axis.text.x found element_blank()

?element_line – from panel.grid.major.x, panel.grid.minor.x, panel.grid.major.y, and panel.grid.minor.y, also uses element_blank() as well as color

?coord_fixed – geom_circle recommends using this

I got the colors by putting an image of the graph in this website, <https://imagecolorpicker.com/>, and clicking on different features of the graph

Part 3

```
europa_new <- europa %>%
  # Sorts the rows of the dataset from lowest to highest taxation
  mutate(Area = Area / 100000) %>%
  arrange(Taxation) %>%
  mutate(
    # Split long country names for plotting
    CountrySplit = Country %>%
      str_replace("Emperor's Dominions", "Emperor's\nDominions") %>%
      str_replace("German Empire", "German\nEmpire") %>%
      str_replace("Britain & Ireland", "Britain &\nIreland") %>%
      str_replace("Countries under the Dominion of France",
                  "Countries under\nthe Dominion\nof France") %>%
      str_replace("Russian Empire", "Russian\nEmpire") %>%
      str_replace("Turkish Empire", "Turkish\nEmpire"),

    # Factor with levels set in Taxation order
    CountrySplit = factor(CountrySplit, levels = CountrySplit)
  ) %>%

  # Combine Population and Taxation into one column with their values in
  # another column so position_dodge can be used
```

```

pivot_longer(cols = c(Taxation, Population, Area),
              names_to = "Variable",
              values_to = "Value")

ggplot(europe_new, aes(x = CountrySplit, y = Value, fill = Variable))+

  geom_bar(stat = "identity", position = position_dodge())+

  # Separate by power. free_x for scales makes it so where only the countries
  # of that power are labeled in that facet, for space it makes the widths of
  # the bars proportional
  facet_wrap(~Power, scales = "free_x", space = "free_x") +

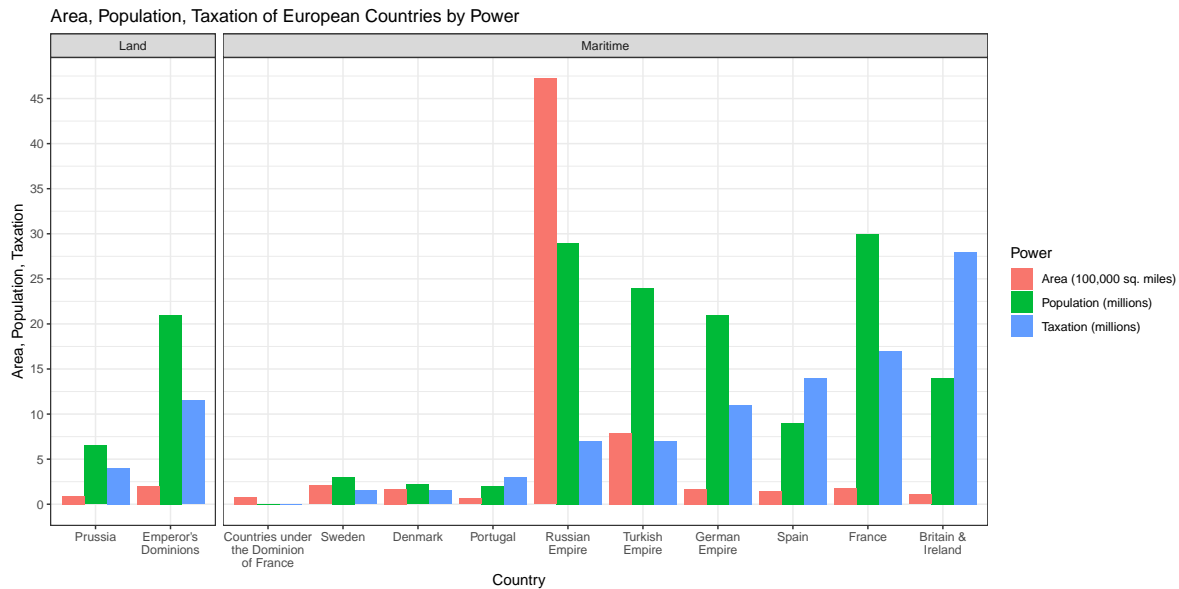
  labs(title = "Area, Population, Taxation of European Countries by Power",
        x = "Country",
        y = "Area, Population, Taxation",
        fill = "Power"
        ) +

  scale_y_continuous(breaks = seq(0,50,5))+

  scale_fill_discrete(
    labels = c("Area (100,000 sq. miles)",
               "Population (millions)",
               "Taxation (millions)")
  ) +

  theme_bw()

```



Part 4

To start I looked up best graph for three numerical and one categorical variables and the AI Overview said a bubble plot or grouped box plot. A grouped box plot wouldn't work for this data, and I made an initial bubble plot and didn't like it. So then I started making a clustered bar chart because I like them.

To make the graph clustered I used `pivot_longer` to join Population and Taxation. Then I mapped the countries to the x-axis, the values from `pivot_longer` to the y-axis, the variables from `pivot_longer` to fill, and Power to `facet_wrap`.

At first this didn't look great because both facets had all the country names even though the country was one or another. To fix this I found `scales = "free_x"`. Then another problem occurred because the facets were still the same size even though land has two countries and maritime has 10. To make the bars of both facets the same width I found `space = "free_x"`.

The next thing I fixed was the position of the country names. Originally, they were overlapping so I added newline characters so that the country names weren't one long line. Then I messed with `fig-width` and `fig-height` until they weren't overlapping.

Another thing I didn't like was that the graph had breaks in increments of 10. I thought this made it difficult to tell what the values of some of the bars were, so I made the breaks in increments of 5.

To really highlight that Britain & Ireland are taxed more than other countries I sorted the countries by taxation. My final steps were adding a title, x and y axis labels, a label for the key, and going through different themes choosing the one I liked best.

One thing I still don't like about this graph is the Countries under the Dominion of France. I don't like it because it looks like there is a gap, but really the values of that country are zero. I'm not sure how to fix this because how else do you represent nothing but a blank space.

Part 4 Addition

I realized after I turned in the homework I forgot to include area. To fix this I included area in the pivot_longer, increased the y-axis to 50