Chapter 4

The Caveman Coder

2025-10-28

Hands-On: My First ggplot2 Thematic Map

Step 1: Load necessary packages

```
library(pacman)
pacman::p_load(tidyverse)
```

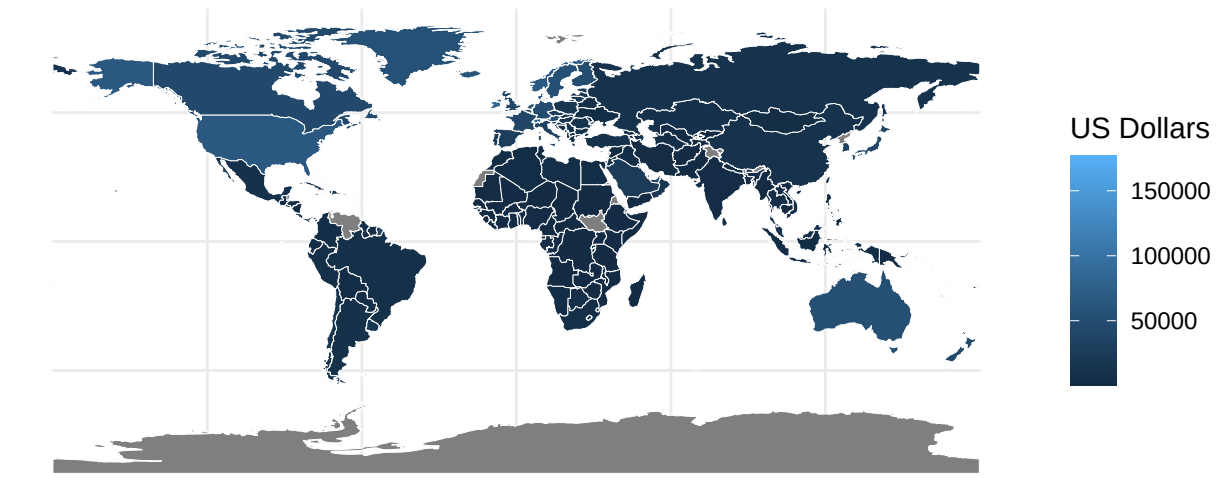
Step 2: Load the cleaned and joined data

```
world_data_loaded <- sf::st_read("world_data_cleaned_joined.gpkg")
```

```
## Reading layer `world_data_cleaned_joined' from data source
##   `/home/norman/Documents/ThirdBrain/x1_Projects/RProjects/Notes-on-Mapping-the-World-with-R/world_da
##   using driver `GPKG'
## Simple feature collection with 257 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.65872
## Geodetic CRS:  WGS 84
```

Step 3: Plot a basic choropleth

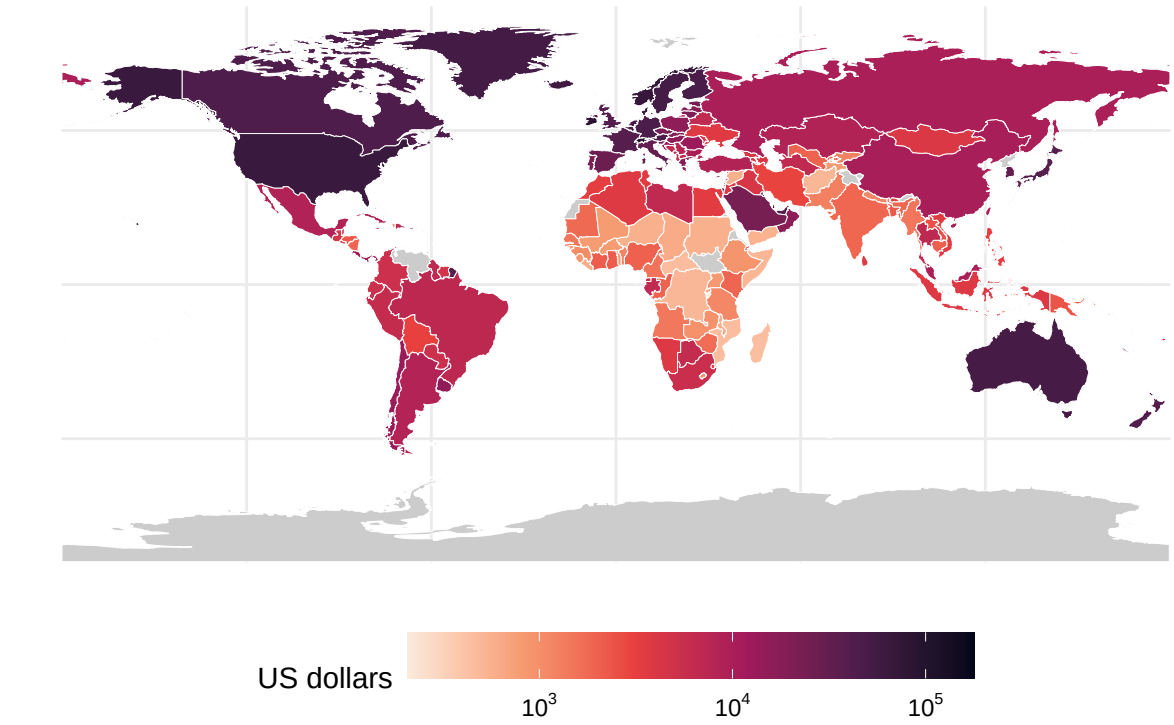
```
map_plot_basic <- world_data_loaded |>
  ggplot() +
  geom_sf(
    mapping = aes(fill = gdp_per_capita),
    color = "white",
    linewidth = 0.1
  ) +
  labs(
    title = "GDP per capita (2020)",
    fill = "US Dollars"
  ) +
  theme_minimal()

map_plot_basic
```



Customizing colors

Create a choropleth map using the Viridis "rocket" palette with log10 scaling:

```
map_plot_viridis <- world_data_loaded |>
  ggplot() +
  geom_sf(
    mapping = aes(fill = gdp_per_capita),
    color = "white",
    linewidth = 0.1
  ) +
  scale_fill_viridis_c(
    option = "rocket",
    direction = -1,
    name = "US dollars",
    na.value = "grey80",
    trans = "log10",
    labels = scales::label_log(digits = 2) # use spacial labels for log scale
  ) +
  labs(title = "GDP per capita (2020)") +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    legend.key.width = grid::unit(1.5, "cm")
  )

map_plot_viridis
```



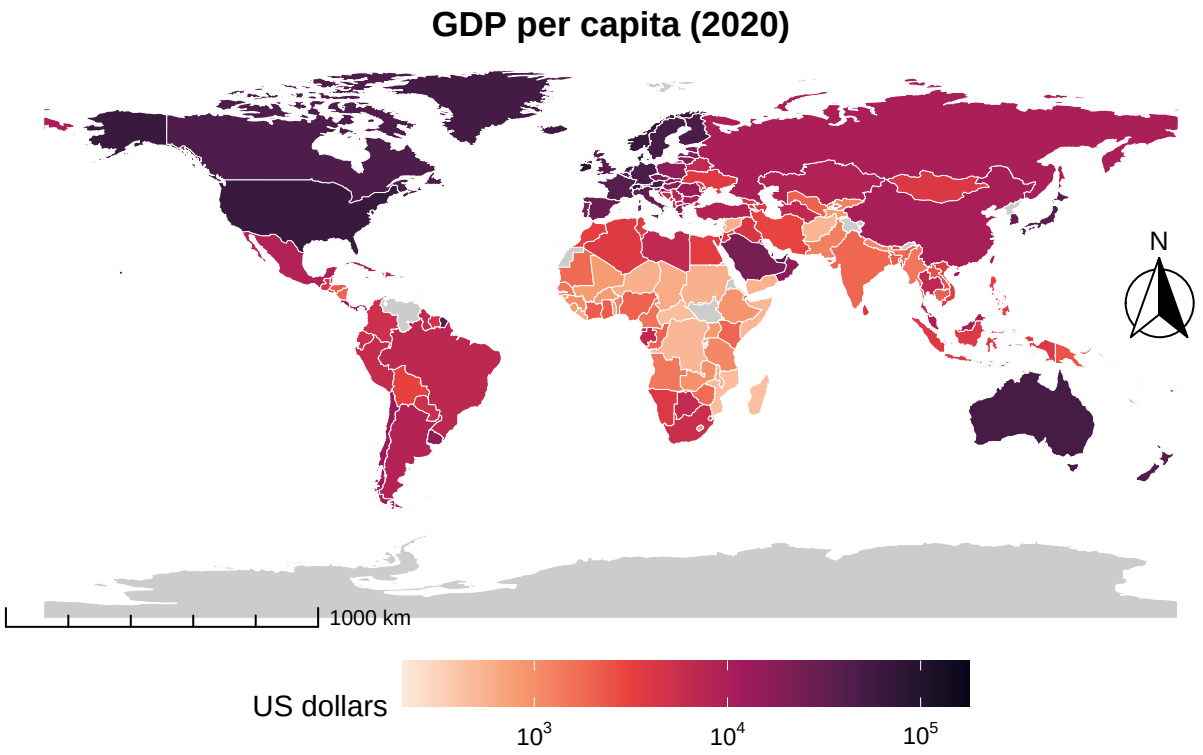
Adding Other Map Elements

Adding scale bars and north arrow using the `ggspatial` package:

```
library(ggspatial)

map_plot_elements <- world_data_loaded |>
  ggplot() +
  geom_sf(
    mapping = aes(fill = gdp_per_capita),
    color = "white",
    linewidth = 0.1
  ) +
  scale_fill_viridis_c(
    option = "rocket",
    direction = -1, # reverse pallete, lighter colors -> lower value
    name = "US dollars", # legend title
    na.value = "grey80",
    trans = "log10", # apply log transformation bec. density is highly skewed
    labels = scales::label_log(digits = 2)
  ) +
  # ADD GGSPATIAL LAYERS
  ggspatial::annotation_scale(
    location = "bl",
    width_hint = 0.3,
    style = "ticks"
  ) +
  ggspatial::annotation_north_arrow(
    location = "tr",
```

```
    which_north = "true",
    pad_x = unit(0.1, "in"),
    pad_y = unit(1, "in"),
    style = ggspatial::north_arrow_fancy_orienteering
  ) +
  # Add labels and theme
  labs(title = "GDP per capita (2020)") +
  theme_void() +
  theme(
    legend.position = "bottom",
    legend.key.width = unit(1.5, "cm"),
    plot.title = element_text(hjust = 0.5, face = "bold") # Center title
  )


map_plot_elements
```

```
## Scale on map varies by more than 10%, scale bar may be inaccurate
```




```
# Save the previous map
assign("plot_original", map_plot_viridis, envir = .GlobalEnv)
```

Project Exercise: Mapping African Indicators

Create a choropleth map showing a different indicator for African countries. Goal is to create a static map showing Life Expectancy (`LifeExp`) for African countries in the latest year available in the `gapminder` data, using `ggplot2`. Choose an appropriate sequential color palette and include a title, legend, and caption.

Step 1: Load the packages:

```
pacman::p_load(sf, tidyverse, rnaturalearth, gapminder, countrycode, viridis)
```

Step 2: Prepare the data.

Getting the polygons for African countries:

```
africa_sf <- rnaturalearth::ne_countries(
  scale = "medium",
  continent = "Africa",
  returnclass = "sf"
) |>
  select(name, iso_a3 = adm0_a3, geometry)
```

Filtering `gapminder` data for the latest year:

```
# glimpse(gapminder)
latest_year <- max(gapminder$year)

gapminder_latest <- gapminder |>
  filter(year == latest_year) |>
  select(country, lifeExp)
```

Getting the country codes from `countrycode`:

```
gapminder_latest <- gapminder_latest |>
  mutate(
    iso_a3 = countrycode::countrycode(
      country, origin = "country.name", destination = "iso3c"
    )
  )
```

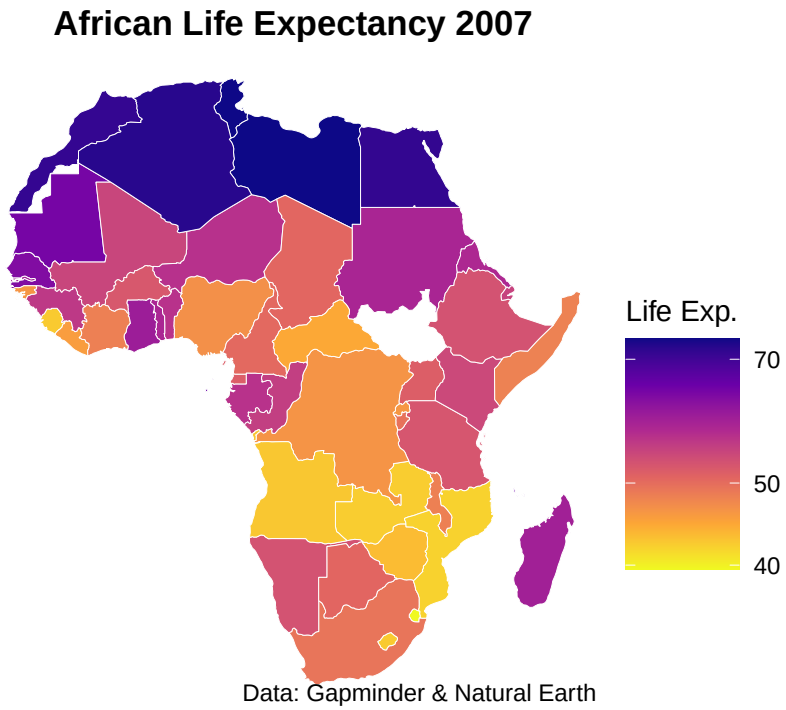
Left-joining `gapminder_latest` with `africa_sf`:

```
africa_life_exp_sf <- africa_sf |>
  left_join(gapminder_latest, by = "iso_a3") |>
  filter(!is.na(lifeExp))

africa_life_exp_sf
```

```
## Simple feature collection with 50 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -17.53564 ymin: -46.96289 xmax: 51.39023 ymax: 37.34038
## Geodetic CRS:  WGS 84
## First 10 features:
##            name iso_a3      country lifeExp                       geometry
## 1      Zimbabwe    ZWE     Zimbabwe  43.487 MULTIPOLYGON (((31.28789 -2...
## 2        Zambia    ZMB       Zambia  42.384 MULTIPOLYGON (((30.39609 -1...
## 3        Uganda    UGA       Uganda  51.542 MULTIPOLYGON (((33.90322 -1...
## 4       Tunisia    TUN      Tunisia  73.923 MULTIPOLYGON (((11.50459 33...
## 5          Togo    TGO         Togo  58.420 MULTIPOLYGON (((0.9004883 1...
## 6      Tanzania    TZA     Tanzania  52.517 MULTIPOLYGON (((39.49648 -6...
## 7      eSwatini    SWZ    Swaziland  39.613 MULTIPOLYGON (((31.94824 -2...
## 8         Sudan    SDN        Sudan  58.556 MULTIPOLYGON (((34.07812 9....
## 9  South Africa    ZAF South Africa  49.339 MULTIPOLYGON (((29.36484 -2...
## 10      Somalia    SOM      Somalia  48.159 MULTIPOLYGON (((41.53271 -1...
```

Creating the plot:

```
africa_map_plot <- africa_life_exp_sf |>
  ggplot() +
  geom_sf(
    mapping = aes(fill = lifeExp),
    color = "white",
    linewidth = 0.1
  ) +
  scale_fill_viridis_c(
    option = "plasma",
    direction = -1,
    trans = "log10",
    name = "Life Exp.",
    na.value = "grey80"
  ) +
  labs(
    title = paste("African Life Expectancy", latest_year),
    caption = "Data: Gapminder & Natural Earth"
  ) +
  theme_void() +
  theme(
    legend.position.inside = c(0.15, 0.8),
    legend.key.width = unit(1.5, "cm"),
    plot.title = element_text(hjust = 0.5, face = "bold"), # Center title
    plot.caption = element_text(hjust = 0.95, vjust = 30)
  )

africa_map_plot
```



Saving the plot:

```r
ggsave(
  "africa_life_exp.png",
  africa_map_plot,
  width = 7,
  height = 8,
  dpi = 600
)
```