Assignment 7 Solutions: Spatial Data I

Part 1: Exploring Spatial Data with sf

Applied Quantitative Methods II, UC3M

Spring 2026

Contents


```r
library(sf)
library(spData)
library(dplyr)
library(ggplot2)

data(world)
```

1. Inspecting an `sf` object

a) Inspect the structure of the `world` dataset:

```r
class(world)
```

```
## [1] "sf"         "tbl_df"     "tbl"        "data.frame"
```

```r
names(world)
```

```
##  [1] "iso_a2"    "name_long" "continent" "region_un" "subregion" "type"
##  [7] "area_km2"  "pop"       "lifeExp"   "gdpPercap" "geom"
```

```r
nrow(world)
```

```
## [1] 177
```

`class(world)` returns both `"sf"` and `"data.frame"`: an sf object is a regular data frame augmented with an extra geometry column (of class `sfc`) that stores the spatial shapes (polygons, points, lines). The geometry column is "sticky" — standard dplyr operations (filter, mutate, select) retain it automatically, so spatial attributes travel with the data without any extra effort.

b) Check the coordinate reference system:

```r
st_crs(world)
```

```
## Coordinate Reference System:
##   User input: EPSG:4326
##   wkt:
## GEOGCRS["WGS 84",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["World."],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]
```

The dataset uses EPSG:4326 (WGS84 — World Geodetic System 1984). WGS84 is the global standard coordinate system used by GPS and most web mapping tools. Coordinates are expressed in decimal degrees of longitude (east-west) and latitude (north-south), making it suitable for global datasets where a common datum is needed across all regions.

c) Inspect geometry types:

```r
unique(st_geometry_type(world))
```

```
## [1] MULTIPOLYGON
## 18 Levels: GEOMETRY POINT LINESTRING POLYGON MULTIPOINT ... TRIANGLE
```

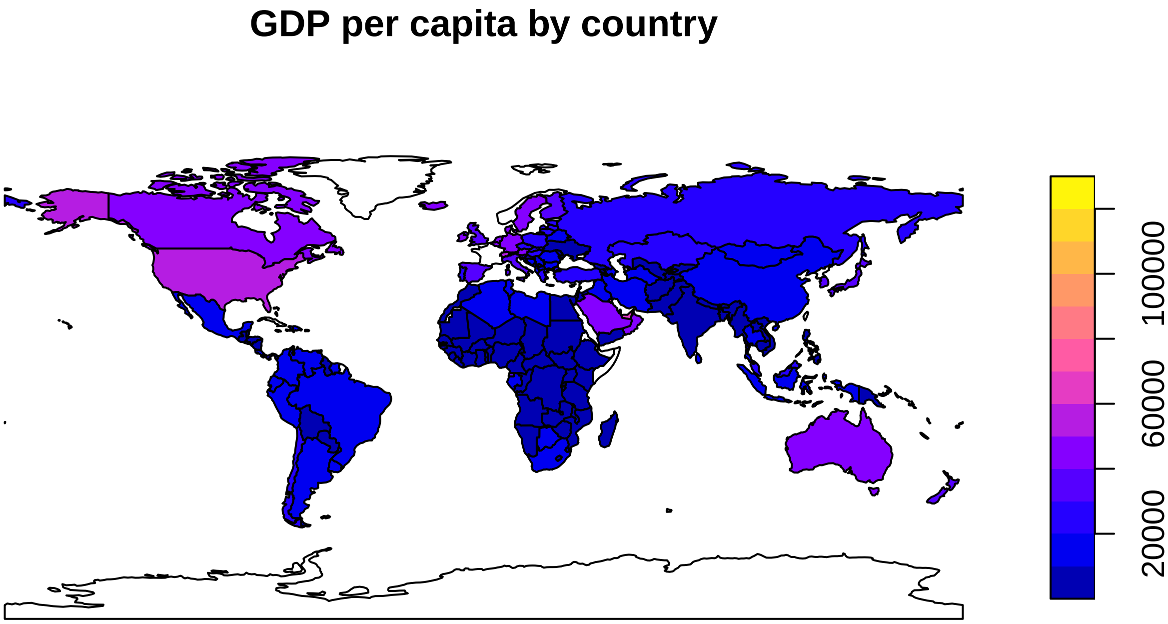
The geometry type is `MULTIPOLYGON`. A MULTIPOLYGON is a collection of one or more polygons treated as a single geographic feature. Countries require multiple polygons when their territory is not a single contiguous land mass — for example, the United States includes Alaska and Hawaii as separate polygons, and France includes overseas territories such as Martinique and Guadeloupe in the Caribbean.

d) Quick base-R map of GDP per capita:

```r
pdf("world_gdp_base.pdf")
plot(world["gdpPercap"])
dev.off()
```

```
## pdf
##   2
```

```r
# Display inline as well
plot(world["gdpPercap"], main = "GDP per capita by country")
```



The map reveals a sharp global inequality pattern. Western and Northern Europe, North America, and Australia/New Zealand appear as the wealthiest regions (dark end of the scale). Sub-Saharan Africa and parts of South and Southeast Asia occupy the lowest end. East Asia shows intermediate-to-high values, reflecting rapid economic growth in countries such as South Korea and Japan.

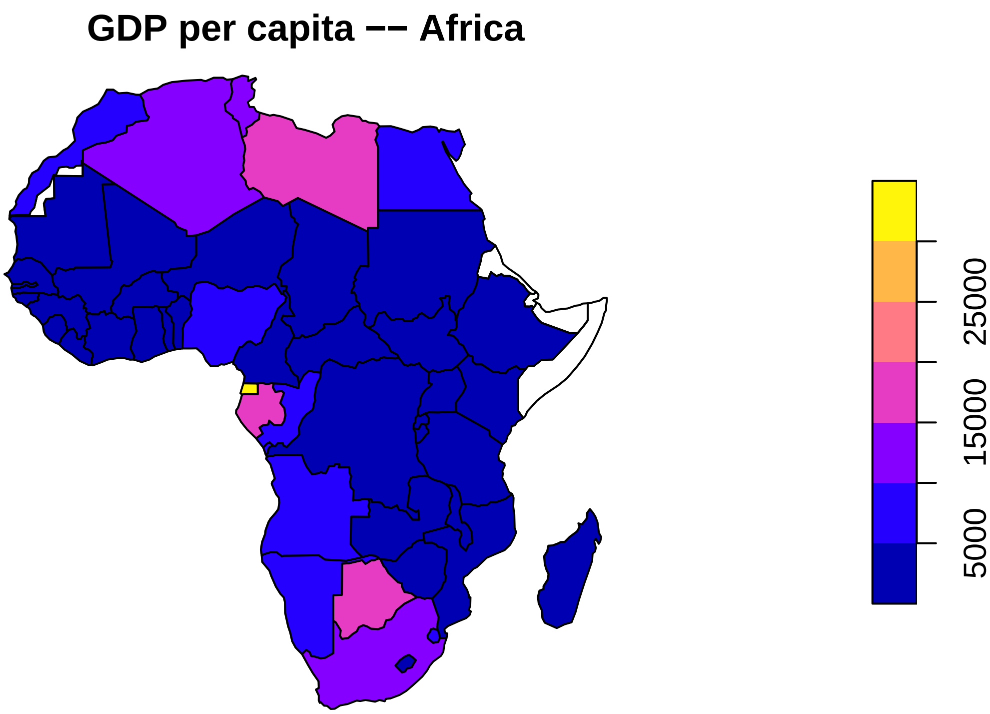
2. Attribute operations

a) Filter to African countries:

```
africa = filter(world, continent == "Africa")
nrow(africa)
```

```
## [1] 51
```

```
plot(africa["gdpPercap"], main = "GDP per capita -- Africa")
```



The dataset contains 51 African countries. The UN recognizes 54 sovereign African states, so this count is slightly below expectations and likely reflects missing data or the exclusion of very small territories from the `spData` world polygon dataset.

b) Add `pop_millions` and summarise GDP per capita by continent:

```
world = world %>%
  mutate(pop_millions = pop / 1e6)

gdp_by_continent = world %>%
  group_by(continent) %>%
  summarise(mean_gdpPercap = mean(gdpPercap, na.rm = TRUE))

print(st_drop_geometry(gdp_by_continent))
```

```
## # A tibble: 8 x 2
##   continent               mean_gdpPercap
## * <chr>                            <dbl>
## 1 Africa                           5042.
## 2 Antarctica                        NaN
## 3 Asia                            20026.
## 4 Europe                          29451.
## 5 North America                   18384.
## 6 Oceania                         15828.
## 7 Seven seas (open ocean)           NaN
## 8 South America                   13762.
```

When `summarise()` is called on a grouped `sf` object, it **unions** the geometries within each group and retains the resulting geometry column. To obtain a plain data frame without spatial information, use `st_drop_geometry()` before or after the summary step. This avoids carrying unneeded geometry through purely tabular analyses.

c) Top 5 African countries by GDP per capita:

```
africa_sorted = africa %>%
  arrange(desc(gdpPercap)) %>%
  select(name_long, gdpPercap)

print(head(st_drop_geometry(africa_sorted), 5))
```

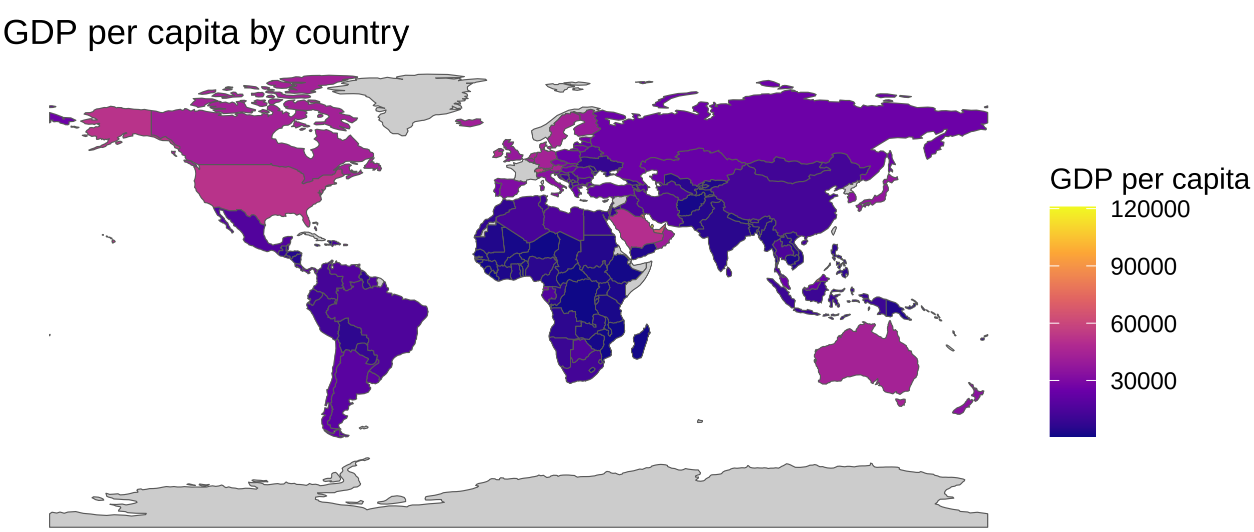
```
## # A tibble: 5 x 2
##   name_long         gdpPercap
##   <chr>                 <dbl>
## 1 Equatorial Guinea    31543.
## 2 Gabon                16679.
## 3 Libya                16372.
## 4 Botswana             15915.
## 5 Algeria              13483.
```

The five African countries with the highest GDP per capita in this dataset are shown above. Equatorial Guinea ranks high due to its oil revenues relative to a small population; Gabon and Libya are also oil-dependent economies; Botswana benefits from diamond exports and relatively strong institutions; the fifth position is typically taken by a North African economy (Mauritius or Algeria depending on the dataset vintage).

3. Simple visualization with `ggplot2`

a) Choropleth map of world GDP per capita:

```
ggplot(world) +
  geom_sf(aes(fill = gdpPercap)) +
  scale_fill_viridis_c(option = "plasma", na.value = "grey80",
                       name = "GDP per capita") +
  theme_void() +
  labs(title = "GDP per capita by country")
```

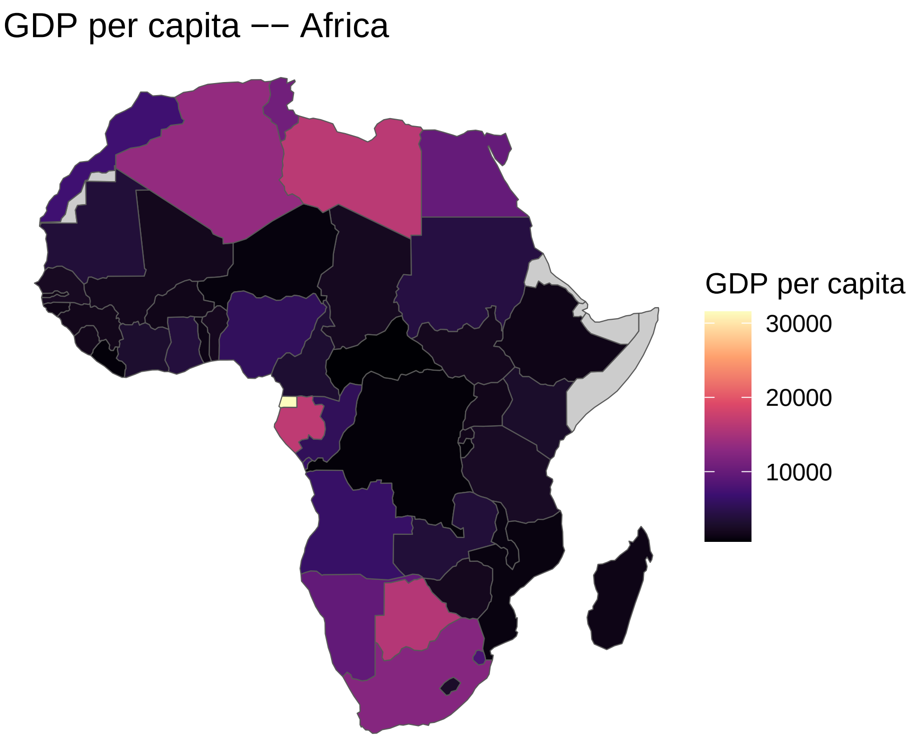



```
ggsave("world_gdp.pdf", width = 10, height = 5)
```

The geographic pattern mirrors what the base-R map showed. Western Europe, North America, and Oceania stand out as the wealthiest cluster. East Asia shows a gradient from high (Japan, South Korea) to middle (China). Sub-Saharan Africa and South Asia concentrate the lowest values, with a few exceptions (e.g., Equatorial Guinea's oil wealth).

b) Africa map with magma palette:

```
ggplot(africa) +
  geom_sf(aes(fill = gdpPercap)) +
  scale_fill_viridis_c(option = "magma", na.value = "grey80",
                       name = "GDP per capita") +
  theme_void() +
  labs(title = "GDP per capita -- Africa")
```



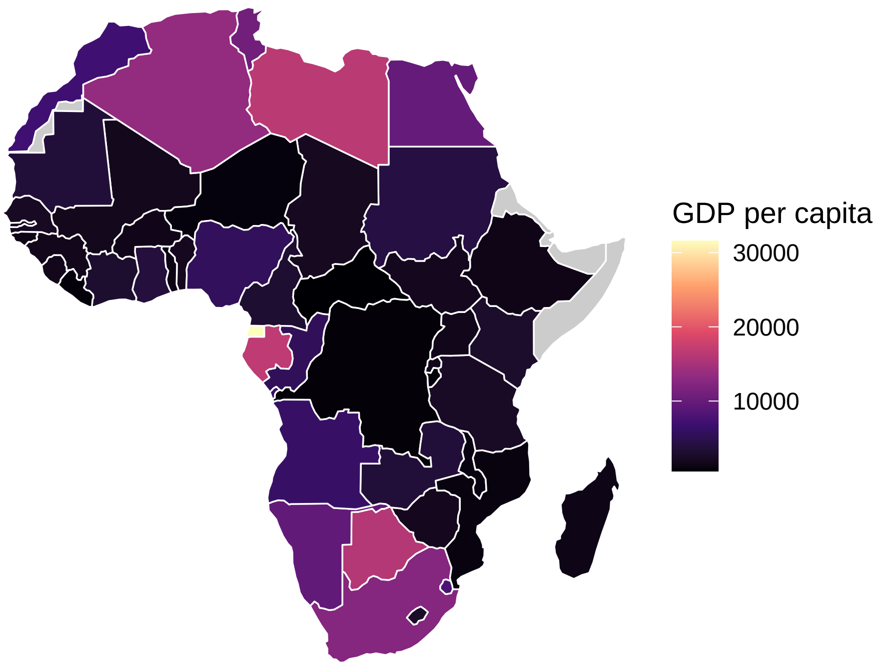

```
ggsave("africa_gdp.pdf", width = 7, height = 6)
```

Within Africa, there is substantial variation. A cluster of relatively wealthier countries appears in North Africa (Egypt, Libya, Tunisia) and in Southern Africa (Botswana, South Africa, Namibia). Central and West Africa (with the exception of oil-rich Equatorial Guinea and Gabon) display the lowest values, reflecting low diversification and persistent structural poverty.

c) Africa map with country borders:

```
ggplot(africa) +
  geom_sf(aes(fill = gdpPercap), color = "white", linewidth = 0.3) +
  scale_fill_viridis_c(option = "magma", na.value = "grey80",
                       name = "GDP per capita") +
  theme_void() +
  labs(title = "GDP per capita -- Africa (with borders)")
```

GDP per capita −− Africa (with borders)




```r
ggsave("africa_gdp_borders.pdf", width = 7, height = 6)
```

Adding white country borders significantly improves readability, especially for smaller countries where adjacent fill colours alone make it hard to distinguish units. The thin white lines demarcate each country without competing visually with the fill scale, making it easier to identify specific countries of interest and to compare neighbours.