The Costs of Counterparty Risk in Long Term Contracts Code Guide - Section 9

Michael Duarte Gonçalves

July 7, 2025

Contents

1 Overview | Building the Map of Spain

This section documents the code and methodology used to construct a geographical map of wind and solar projects in Spain that began operation in 2022. The goal is to visually represent the spatial distribution and capacity of renewable energy projects.

- The map displays the locations (latitude/longitude) and capacities (MW) of wind and solar projects.
- The top 5 largest (regardless of solar/wind energy) projects are highlighted with labels.
- The map includes Spanish mainland provinces and the Canary Islands as an inset.
- Data is sourced from an Excel file containing project details and geocoordinates.

1.1 Filtering and Formatting

Wind and solar projects are loaded from separate sheets, filtered for Spain and the year 2022, and selected columns are kept for mapping.

```r
# Section 9: Map --------------------

# Find the data in data_raw path. This raw data is the same as we used at the beginning of the script
# but contains also the geographical coordinates (latitude/longitude),
# that we need to construct our map.

file_name <- "Wind_Solar_projects_with_coordinates_Spain_2022.xlsx"
file_path <- fs::path(data_raw, file_name)


# -------------------------------- #
# Some data preparation

wind_projects_spain_2022_coordinates <- read_excel(file_path, sheet = "Wind") |>
  mutate(type = "Wind") |>
  filter(‘Country/Area‘ == "Spain", ‘Start year‘ == 2022) |>
  clean_names() |>
  select(country_area, project_name, start_year,
         capacity_mw, latitude, longitude, type)

solar_projects_spain_2022_coordinates <- read_excel(file_path, sheet = "Solar") |>
  mutate(type = "Solar") |>
  filter(‘Country/Area‘ == "Spain", ‘Start year‘ == 2022) |>
  clean_names() |>
  select(country_area, project_name, start_year,
         capacity_mw, latitude, longitude, type)

wind_solar_proj_2022_coordinates <-
  bind_rows(wind_projects_spain_2022_coordinates, solar_projects_spain_2022_coordinates)
```

1.2 Handling Duplicate Project Names

Some projects share the same name and capacity but have different coordinates. To distinguish them, a number is appended to the project name.

```
# Some projects have the same name and capacity, but different lat/lon
# coordinates. To distinguish them, we put a number at the end of the
   project
# to facilitate the subsequent join().

wind_solar_proj_2022_coordinates <- wind_solar_proj_2022_coordinates |>
  group_by(project_name, capacity_mw) |>
  mutate(project_name = if(n() > 1) paste0(project_name, " ", row_
     number()) else project_name) |>
  ungroup()

wind_solar_proj_2022 <- wind_solar_proj_2022 |>
  group_by(projectname, capacity) |>
  mutate(projectname = if(n() > 1) paste0(projectname, " ", row_number
     ()) else projectname) |>
  ungroup()
```

1.3 Joining with Additional Project Data

The cleaned coordinates data is joined with project-level data to include additional metrics. For one project with missing coordinates, values are imputed.

```
wind_solar_proj_2022_coordinates_qi_mwh <- wind_solar_proj_2022_
   coordinates |>
  left_join(
    wind_solar_proj_2022,
    by = c("project_name" = "projectname",
           "capacity_mw" = "capacity",
           "type" = "type")
  ) |>
  select(project_name, start_year, capacity_mw, latitude, longitude,
     type, q_i_mwh) |>
  mutate(
    latitude = if_else(project_name == "Puerto Del Rosario wind farm",
       "36", latitude),
    longitude = if_else(project_name == "Puerto Del Rosario wind farm",
       "-8.55", longitude)
  )
```

2 Mapping Preparation

2.1 Creating the Spatial Data Frame

Coordinates are converted to numeric and the data is transformed into a spatial (sf) object.

```
wind_solar_proj_2022_coordinates_qi_mwh <- wind_solar_proj_2022_
   coordinates |>
  left_join(
    wind_solar_proj_2022,
    by = c("project_name" = "projectname",
           "capacity_mw" = "capacity",
           "type" = "type")
  ) |>
  select(project_name, start_year, capacity_mw, latitude, longitude,
     type, q_i_mwh) |>
  mutate(
    latitude = if_else(project_name == "Puerto Del Rosario wind farm",
       "36", latitude),
    longitude = if_else(project_name == "Puerto Del Rosario wind farm",
       "-8.55", longitude)
  )


#  -----------------------------------#
# Map

# Prepare your wind projects data: convert coordinates to numeric and
   create an sf object.
wind_solar_projects_sf <- wind_solar_proj_2022_coordinates_qi_mwh |>
  mutate(
    latitude = as.numeric(latitude),
    longitude = as.numeric(longitude)
    ) |>
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326)
```

2.2 Loading Map Layers

Spanish provinces and Canary Islands province boundaries are loaded and cleaned for mapping.

```
# Get Spanish provinces with cleaned names and translated province
   names.
esp_prov <- esp_get_prov() |>
  clean_names() |>
  mutate(provincia = esp_dict_translate(ine_prov_name, "es"))

# Get Canary Islands provinces and the box that defines their inset
   position.
can_prov <- esp_get_can_provinces()
can_box  <- esp_get_can_box()
```

2.3 Projecting Spatial Data

The project locations are transformed to match the coordinate reference system of the province layers.

```
# Transform wind projects to the CRS used in esp_prov.
wind_solar_projects_sf <- st_transform(wind_solar_projects_sf, st_crs(
   esp_prov))
```

2.4 Selecting Top Projects for Labeling

The five largest projects (by capacity) are selected for labeling on the map.

```
# Split by type and get top 5 of each
# Get top 5 projects overall (regardless of type)
top_projects <- wind_solar_projects_sf |>
  arrange(desc(capacity_mw)) |>
  slice(1:5)
```

3 Map Construction

3.1 Plotting the Map

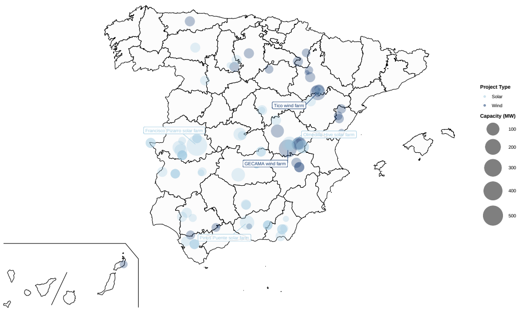
The map is constructed using ggplot2 and sf layers. Wind and solar projects are shown with different colors and sizes proportional to capacity. The top 5 projects are labeled.

```r
# Create the map
map_Spain_wind_solar_proj <- ggplot() +
  # Base map layers
  geom_sf(data = esp_prov, fill = "grey99", color = "black") +
  geom_sf(data = can_prov, fill = "grey99", color = "black") +
  geom_sf(data = can_box, fill = NA, color = "black", size = 1) +

  # Wind projects
  geom_sf(data = filter(wind_solar_projects_sf, type == "Wind"),
          aes(size = capacity_mw, color = type),
          alpha = 0.3, show.legend = TRUE) +

  # Solar projects
  geom_sf(data = filter(wind_solar_projects_sf, type == "Solar"),
          aes(size = capacity_mw, color = type),
          alpha = 0.3, show.legend = TRUE) +

  # Labels for top 5 largest projects (optional)
  geom_label_repel(
    data = top_projects,
    aes(label = project_name, geometry = geometry, color = type),
    stat = "sf_coordinates",
    fill = "white", alpha = 0.9,
    size = 3, label.size = 0,
    box.padding = 1.5, point.padding = 0.5,
    min.segment.length = 0,
    segment.size = 0.3,
    show.legend = FALSE
  ) +

  # Size scale (auto-scaled using sqrt transform)
  scale_size_continuous(
    trans = "sqrt",                  # Helps small projects show up
    range = c(2, 18),                # Controls dot size visually
    name = "Capacity (MW)"           # Legend title
  ) +

  # Project type color
  scale_color_manual(
    values = theme_palette_map, # #084594 (dark blue paper), #9ecae1 (
       light blue paper), #002d18 (dark green), #6ecf87 (light green)
    name = "Project Type"
  ) +

  theme_void() +
  theme(
    legend.position = "right",
    legend.title = element_text(size = 10, face = "bold"),
    legend.text = element_text(size = 9)
  )
```

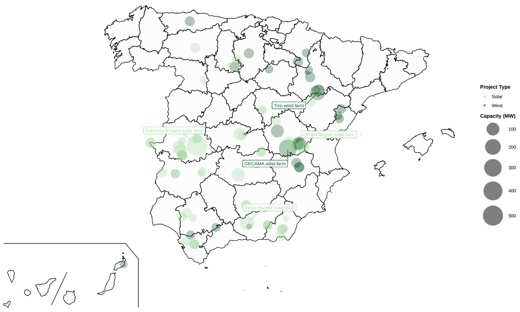
3.2 Saving the Map

The resulting map is saved as a PDF file for inclusion in reports or presentations.

```
# Save the plot
plot_filename <- "spain_map_wind_solar_proj_2022.pdf"
plot_path_cpr <- file.path(out_figures, plot_filename)

ggsave(
  filename = plot_path_cpr,
  plot = map_Spain_wind_solar_proj,
  width = 16,
  height = 9,
  dpi = 300
)
```



(a) Map of Spain: Paper Version



(b) Map of Spain: Presentation Version