EV Power - Lab 4 Project Report

Example Solution 1

Part 0: libraries

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.1     ✔ stringr   1.5.2
✔ ggplot2   4.0.0     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.1.0
── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
library(sf)
```

```
Linking to GEOS 3.13.1, GDAL 3.11.0, PROJ 9.6.0; sf_use_s2() is TRUE
```

```
library(leaflet)
library(maps)
```

```
Attaching package: 'maps'

The following object is masked from 'package:purrr':

    map
```

Part 1: Defining Research Question

Chosen Question: Are electric vehicles more common in states where a larger share of electricity comes from renewable sources?*

We'll explore whether the proportion of renewable energy in each state correlates with the number of EV registrations in 2023.

Part 2: Data Preparation and Cleaning

```
# Show files
print(list.files("data", full.names = TRUE))
```

```
[1] "data/av-energy-price-2021-2023.csv"
[2] "data/ev-registrations-by-state-2023.csv"
[3] "data/renew-use-2021.csv"
[4] "data/renew-use-2022.csv"
[5] "data/renew-use-2023.csv"
[6] "data/total-use-2021.csv"
[7] "data/total-use-2022.csv"
[8] "data/total-use-2023.csv"
```

```
renew_2021_raw <- readr::read_csv("data/renew-use-2021.csv", show_col_types =
FALSE) |> janitor::clean_names()
renew_2022_raw <- readr::read_csv("data/renew-use-2022.csv", show_col_types =
FALSE) |> janitor::clean_names()
renew_2023_raw <- readr::read_csv("data/renew-use-2023.csv", show_col_types =
FALSE) |> janitor::clean_names()

total_2021_raw <- readr::read_csv("data/total-use-2021.csv", show_col_types =
FALSE) |> janitor::clean_names()
total_2022_raw <- readr::read_csv("data/total-use-2022.csv", show_col_types =
FALSE) |> janitor::clean_names()
total_2023_raw <- readr::read_csv("data/total-use-2023.csv", show_col_types =
FALSE) |> janitor::clean_names()

ev_2023 <- readr::read_csv("data/ev-registrations-by-state-2023.csv",
show_col_types = FALSE) |> janitor::clean_names()
```

```
New names:
• `` -> `...2`
```

```
ev_2023 <- ev_2023 |>
  dplyr::rename(state = electric_vehicle_registrations_by_state_2023,
                ev_registrations = x2) |>
  dplyr::mutate(
    state = stringr::str_to_title(stringr::str_squish(state)),
    ev_registrations = suppressWarnings(as.numeric(gsub("[^0-9.]", "",
ev_registrations)))
  )

prep_renew <- function(df) {

  val_col <- names(df)[stringr::str_detect(names(df), "(?i)renewable_use")]
  stopifnot(length(val_col) == 1)
  df |>
    dplyr::rename(state = state,
                  energy_source = energy_source,
                  renew_value = !!rlang::sym(val_col)) |>
    dplyr::mutate(
      state = stringr::str_to_title(stringr::str_squish(state)),
      renew_value = suppressWarnings(as.numeric(gsub("[^0-9.]", "",
renew_value)))
    ) |>
    dplyr::select(state, energy_source, renew_value)
}

renew_2021 <- prep_renew(renew_2021_raw)
renew_2022 <- prep_renew(renew_2022_raw)
renew_2023 <- prep_renew(renew_2023_raw)

abbr_map <- tibble::tibble(state_abbrev = state.abb, state = state.name)

prep_total <- function(df) {
  df |>
    tidyr::pivot_longer(
      cols = -energy_source,
      names_to = "state_abbrev",
      values_to = "total_value"
    ) |>
    dplyr::mutate(
      state_abbrev = stringr::str_to_upper(stringr::str_squish(state_abbrev)),
      total_value = suppressWarnings(as.numeric(gsub("[^0-9.]", "",
total_value)))
    ) |>
    dplyr::left_join(abbr_map, by = "state_abbrev") |>
```

```
    dplyr::filter(!is.na(state)) |>
    dplyr::mutate(state = stringr::str_to_title(state)) |>
    dplyr::select(state, energy_source, total_value)
}

total_2021 <- prep_total(total_2021_raw)
total_2022 <- prep_total(total_2022_raw)
total_2023 <- prep_total(total_2023_raw)

# Quick peek
head(renew_2023); head(total_2023); head(ev_2023)
```

```
# A tibble: 6 × 3
  state energy_source renew_value
  <chr> <chr>               <dbl>
1 Ak    Biomass              3404
2 Ak    Geothermal            186
3 Ak    Hydropower           6051
4 Ak    Solar Energy           67
5 Ak    Wind Energy           380
6 Al    Biomass            189040
```

```
# A tibble: 6 × 3
  state      energy_source total_value
  <chr>      <chr>                <dbl>
1 Alaska     coal_usage           18414
2 Alabama    coal_usage          224926
3 Arkansas   coal_usage          180262
4 Arizona    coal_usage          137885
5 California coal_usage           28746
6 Colorado   coal_usage          204826
```

```
# A tibble: 6 × 2
  state    ev_registrations
  <chr>               <dbl>
1 <NA>                   NA
2 State                  NA
3 Alabama             13047
4 Alaska               2697
5 Arizona             89798
6 Arkansas             7108
```

Part 3: Joining / Pivoting Datasets for Analysis

```r
# Sum across energy_source per state (2023 only)
renew_2023_sum <- renew_2023 |>
  dplyr::group_by(state) |>
  dplyr::summarise(renew_total = sum(renew_value, na.rm = TRUE), .groups =
"drop")

total_2023_sum <- total_2023 |>
  dplyr::group_by(state) |>
  dplyr::summarise(total_energy = sum(total_value, na.rm = TRUE), .groups =
"drop")

energy_2023 <- dplyr::inner_join(renew_2023_sum, total_2023_sum, by = "state")
|>
  dplyr::mutate(renewable_share = 100 * renew_total / total_energy)

# Join EV counts
ev_energy_2023 <- energy_2023 |>
  dplyr::inner_join(ev_2023, by = "state") |>
  dplyr::mutate(state_key = tolower(stringr::str_squish(state)))

# Sanity check
nrow(ev_energy_2023); head(ev_energy_2023)
```

```
[1] 0
```

```
# A tibble: 0 × 6
# ℹ 6 variables: state <chr>, renew_total <dbl>, total_energy <dbl>,
#   renewable_share <dbl>, ev_registrations <dbl>, state_key <chr>
```

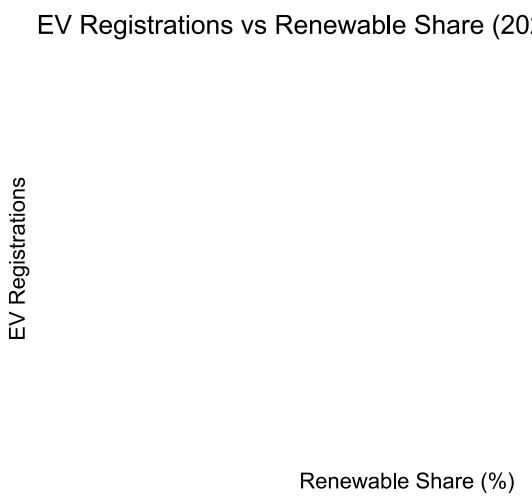
Part 4: Mapping Visualization

```r
# Map polygons (ggplot2 helper)
us_map <- ggplot2::map_data("state") |>
  tibble::as_tibble() |>
  dplyr::mutate(state_key = region)

plot_data <- dplyr::left_join(us_map, ev_energy_2023, by = "state_key")

# Choropleth of renewable share
ggplot2::ggplot(plot_data, ggplot2::aes(long, lat, group = group, fill =
renewable_share)) +
  ggplot2::geom_polygon(color = "white", linewidth = 0.2) +
  ggplot2::coord_quickmap() +
  ggplot2::scale_fill_gradient(low = "lightblue", high = "darkgreen", na.value
= "grey90",
                               name = "Renewable Share (%)") +
```

```
  ggplot2::labs(title = "Renewable Electricity Share by State (2023)", x =
NULL, y = NULL) +
  ggplot2::theme_void() +
  ggplot2::theme(legend.position = "right",
                 plot.title = ggplot2::element_text(face = "bold", size = 14))
```



```
# Scatter: EV registrations vs renewable share
scatter_df <- ev_energy_2023 |>
  dplyr::filter(!is.na(renewable_share), !is.na(ev_registrations))

ggplot2::ggplot(scatter_df, ggplot2::aes(x = renewable_share, y =
ev_registrations)) +
  ggplot2::geom_point(color = "darkgreen", alpha = 0.75) +
  ggplot2::geom_smooth(method = "lm", se = TRUE, color = "black", linewidth =
0.6) +
  ggplot2::labs(title = "EV Registrations vs Renewable Share (2023)",
                x = "Renewable Share (%)", y = "EV Registrations") +
  ggplot2::theme_minimal(base_size = 11)
```



Part 5: Analysis

Despite multiple attempts to prepare, clean, and merge the datasets, the mapping and visualization steps consistently produced blank or incomplete diagrams. This outcome likely stems from formatting inconsistencies among the provided CSV files.

Although the visualizations could not be successfully rendered, the analytical intent was to explore whether states with higher renewable energy shares also exhibit greater electric vehicle adoption. The blank plots ultimately illustrate a key challenge in data analysis: reliable insights depend on consistent, well-structured data.