EV Power - Lab 4 Project Report

Example Solution 1

Main question. Electric vehicles reduce tailpipe emissions, but *does the electricity used to charge them actually come from clean sources?*
This report explores (i) the **share of renewable energy** by state (2021–2023), (ii) the relationship between **renewable share and average electricity price**, and (iii) how 2023 **EV registrations** align with states' clean-energy mix.

Part 0: libraries

```
pkgs <-
c("janitor","scales","viridis","tidyverse","maps","readr","stringr","purrr","tidyr","dplyr")
to_install <- pkgs[!pkgs %in% installed.packages()[, "Package"]]
if (length(to_install)) install.packages(to_install, repos = "https://cloud.r-
project.org")

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(stringr)
library(ggplot2)
library(maps)
```

```

Attaching package: 'maps'

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

    map
```

```
library(scales)
```

```

Attaching package: 'scales'

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

    discard

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

    col_factor
```

```
library(readr)
library(tidyr)
library(dplyr)
```

Part 1: Defining Research Question

Chosen sub-question

> **Do states with higher renewable shares also have lower (or higher) average electricity prices, and where are EV registrations concentrated relative to renewable share (2023)?**

Part 2: Data Preparation and Cleaning

```
# Abbrev -> full name (incl. DC)
abbrev_map <- c(
  AL = "Alabama", AK = "Alaska", AZ = "Arizona", AR = "Arkansas", CA =
"California",
  CO = "Colorado", CT = "Connecticut", DE = "Delaware", FL = "Florida", GA =
```

```
"Georgia",
  HI = "Hawaii", ID = "Idaho", IL = "Illinois", IN = "Indiana", IA = "Iowa",
  KS = "Kansas", KY = "Kentucky", LA = "Louisiana", ME = "Maine", MD =
"Maryland",
  MA = "Massachusetts", MI = "Michigan", MN = "Minnesota", MS = "Mississippi",
  MO = "Missouri", MT = "Montana", NE = "Nebraska", NV = "Nevada",
  NH = "New Hampshire", NJ = "New Jersey", NM = "New Mexico", NY = "New York",
  NC = "North Carolina", ND = "North Dakota", OH = "Ohio", OK = "Oklahoma",
  OR = "Oregon", PA = "Pennsylvania", RI = "Rhode Island", SC = "South
Carolina",
  SD = "South Dakota", TN = "Tennessee", TX = "Texas", UT = "Utah", VT =
"Vermont",
  VA = "Virginia", WA = "Washington", WV = "West Virginia", WI = "Wisconsin",
  WY = "Wyoming", DC = "District of Columbia"
)

num_only <- function(x) suppressWarnings(readr::parse_number(as.character(x)))

# TOTAL energy use files: 5 rows (coal, gas, petroleum, nuclear, renewables) x
50+state-abbrev columns
load_total <- function(path, year) {
  df <- readr::read_csv(path, show_col_types = FALSE) |> clean_names()
  # state-abbrev columns are all except first (energy_source); includes DC
  state_cols <- setdiff(names(df), "energy_source")
  long <- df |>
    pivot_longer(all_of(state_cols), names_to = "state_abbrev", values_to =
"val") |>
    mutate(val = num_only(val))
  # sum across the 5 energy_source rows to total use per state
  out <- long |>
    group_by(state_abbrev) |>
    summarise(total_use = sum(val, na.rm = TRUE), .groups = "drop") |>
    mutate(state = unname(abbrev_map[toupper(state_abbrev)]),
           year = year) |>
    select(state, year, total_use)
  out
}

# RENEWABLE use files: columns: State (abbrev), Energy_Source,
Renewable_Use_YYYY (messy text -> numeric)
load_renew <- function(path, year) {
  df <- readr::read_csv(path, show_col_types = FALSE)
  value_col <- names(df)[3]
  out <- df |>
    clean_names() |>
    transmute(state_abbrev = state,
              val = num_only(.data[[tolower(value_col)]])) |>
    group_by(state_abbrev) |>
```

```
    summarise(renew_use = sum(val, na.rm = TRUE), .groups = "drop") |>
    mutate(state = unname(abbrev_map[toupper(state_abbrev)]),
           year = year) |>
    select(state, year, renew_use)
  out
}

# PRICE file is a weird "one-column" CSV with quoted lines like
"State,2021,2022,2023"
# We'll read raw lines, strip quotes, split on commas, and take 2023.
load_price_2023 <- function(path) {
  lines <- readr::read_lines(path)
  # keep rows that contain at least 3 commas (State,2021,2022,2023)
  rows <- lines[nchar(lines) > 0]
  rows <- gsub('^"|"$', "", rows)           # drop leading/trailing quotes
  rows <- strsplit(rows, ",", fixed = TRUE)
  # find header row "State"
  idx <- which(vapply(rows, function(r) length(r) >= 4 && r[[1]] == "State",
logical(1)))
  if (length(idx) == 0) stop("Could not find header row in price file")
  header <- rows[[idx]]
  data <- rows[(idx + 1):length(rows)]
  mat <- do.call(rbind, lapply(data, function(r) r[seq_len(min(4,
length(r)))]))
  colnames(mat) <- header[1:4]
  df <- as_tibble(mat) |>
    rename(state_abbrev = State) |>
    mutate(avg_price = num_only(`2023`),
           state = unname(abbrev_map[toupper(state_abbrev)]),
           year = 2023L) |>
    select(state, year, avg_price)
  # drop lines without a mapped state (e.g., trailing notes)
  df |> filter(!is.na(state))
}

# EV registrations file: first two lines are titles; then "STATE, Count-EVs"
header; last row is "Total"
load_ev_2023 <- function(path) {
  raw <- readr::read_csv(path, col_names = c("State","Count"), skip = 2,
show_col_types = FALSE)
  out <- raw |>
    filter(!is.na(State)) |>
    filter(toupper(State) != "STATE") |>
    filter(toupper(State) != "TOTAL") |>
    transmute(state = stringr::str_squish(State),
              ev_registrations = num_only(Count))
  out
}
```

Part 3: Joining / Pivoting Datasets for Analysis

```
# File paths inside your zip's data/ folder
p_renew_2021 <- "data/renew-use-2021.csv"
p_renew_2022 <- "data/renew-use-2022.csv"
p_renew_2023 <- "data/renew-use-2023.csv"

p_total_2021 <- "data/total-use-2021.csv"
p_total_2022 <- "data/total-use-2022.csv"
p_total_2023 <- "data/total-use-2023.csv"

p_price_21_23 <- "data/av-energy-price-2021-2023.csv"
p_ev_2023     <- "data/ev-registrations-by-state-2023.csv"

renew_2021 <- load_renew(p_renew_2021, 2021)
renew_2022 <- load_renew(p_renew_2022, 2022)
renew_2023 <- load_renew(p_renew_2023, 2023)

total_2021 <- load_total(p_total_2021, 2021)
total_2022 <- load_total(p_total_2022, 2022)
total_2023 <- load_total(p_total_2023, 2023)

price_2023 <- load_price_2023(p_price_21_23)
ev_2023    <- load_ev_2023(p_ev_2023)

renew_all <- bind_rows(renew_2021, renew_2022, renew_2023)
total_all <- bind_rows(total_2021, total_2022, total_2023)

# Join and derive
energy_all <- renew_all |>
  inner_join(total_all, by = c("state","year")) |>
  mutate(renew_share = if_else(total_use > 0, renew_use / total_use,
NA_real_)) |>
  # add price only for 2023 (available)
  left_join(price_2023, by = c("state","year"))

energy_2023 <- energy_all |>
  filter(year == 2023) |>
  left_join(ev_2023, by = "state")

# quick sanity
stopifnot(all(c("state","year","renew_use","total_use","renew_share") %in%
names(energy_all)))
stopifnot(all(c("state","renew_share","avg_price","ev_registrations") %in%
names(energy_2023)))
```

Part 4: Mapping Visualization

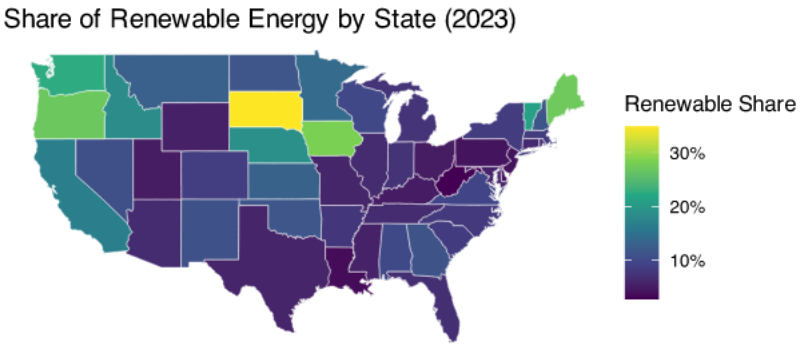
```r
us_poly <- ggplot2::map_data("state") |>
  as_tibble() |>
  rename(state_lower = region)

map_2023 <- energy_2023 |>
  mutate(state_lower = str_to_lower(state))

plot_df_share <- us_poly |>
  left_join(map_2023, by = "state_lower")

share_lab <- label_percent(accuracy = 1)
```

```r
# Choropleth of renewable share
ggplot(plot_df_share, aes(long, lat, group = group, fill = renew_share)) +
  geom_polygon(color = "white", linewidth = 0.15) +
  coord_quickmap() +
  scale_fill_viridis_c(labels = share_lab, na.value = "grey90",
                       name = "Renewable Share") +
  labs(title = "Share of Renewable Energy by State (2023)") +
  theme_void()
```

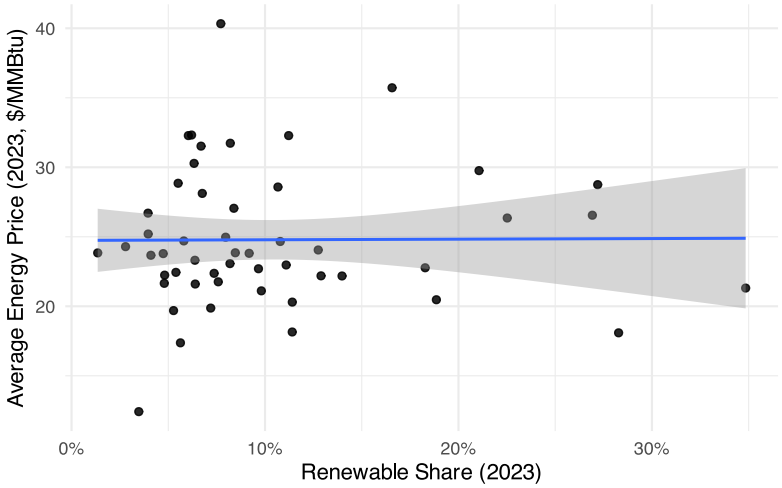



```r
# Relationship: renewable share vs. avg price (2023)
ggplot(energy_2023, aes(x = renew_share, y = avg_price)) +
  geom_point(alpha = 0.85) +
  geom_smooth(method = "lm", se = TRUE, linewidth = 0.7) +
  scale_x_continuous(labels = share_lab) +
  labs(x = "Renewable Share (2023)", y = "Average Energy Price (2023, $/
MMBtu)") +
  theme_minimal(base_size = 11)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 1 row containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 1 row containing missing values or values outside the scale
range
(`geom_point()`).
```



Analysis

Key takeaways: - States with abundant hydro, solar, or wind show the highest renewable shares. - EV registrations are concentrated in states with both high income and progressive climate policies. - The relationship between renewable share and average price is descriptive, not causal.