

# EV Power - Lab 4 Project Report

## Example Solution 1

### Part 0: libraries

```
#| label: setup #| include: false library(tidyverse) library(janitor) library(stringr) library(readr) library(ggplot2) library(scales) library(maps)
```

### Part 1: Defining Research Question

Chosen Question: Do states with a higher share of renewable energy usage also have more EV registrations (2023)?

### Part 2: Data Preparation and Cleaning

```
suppressPackageStartupMessages({  
  library(tidyverse)  
  library(janitor)  
  library(readr)  
  library(stringr)  
  library(dplyr)  
})
```

Warning: package 'janitor' was built under R version 4.3.3

```
fp <- list(  
  renew_2021 = "data/renew-use-2021.csv",  
  renew_2022 = "data/renew-use-2022.csv",  
  renew_2023 = "data/renew-use-2023.csv",
```

```

total_2021 = "data/total-use-2021.csv",
total_2022 = "data/total-use-2022.csv",
total_2023 = "data/total-use-2023.csv",
price_all  = "data/av-energy-price-2021-2023.csv",
ev_2023    = "data/ev-registrations-by-state-2023.csv"
)

renew_2021 <- read_csv(fp$renew_2021, show_col_types = FALSE) %>% clean_names()
renew_2022 <- read_csv(fp$renew_2022, show_col_types = FALSE) %>% clean_names()
renew_2023 <- read_csv(fp$renew_2023, show_col_types = FALSE) %>% clean_names()
total_2021 <- read_csv(fp$total_2021, show_col_types = FALSE) %>% clean_names()
total_2022 <- read_csv(fp$total_2022, show_col_types = FALSE) %>% clean_names()
total_2023 <- read_csv(fp$total_2023, show_col_types = FALSE) %>% clean_names()
price_all  <- read_csv(fp$price_all , show_col_types = FALSE) %>% clean_names()
ev_2023    <- read_csv(fp$ev_2023 , show_col_types = FALSE) %>% clean_names()

```

New names:

```
* `` -> `...2`
```

```

to_full_state <- function(ab) {
  ab <- tolower(ab)
  look <- setNames(tolower(state.name), tolower(state.abb))
  out <- unname(look[ab])
  out[is.na(out) & ab == "dc"] <- "district of columbia"
  ifelse(is.na(out), ab, out)
}

reshape_if_wide <- function(df, value_name = "value") {
  nm <- names(df)
  abbrev_cols <- c(tolower(state.abb), "dc")
  state_cols <- intersect(nm, abbrev_cols)
  if (length(state_cols) >= 30) {
    df %>%
      pivot_longer(all_of(state_cols), names_to = "state_abbr", values_to = value_name) %>%
      mutate(state = to_full_state(state_abbr), state = str_to_lower(state)) %>%
      select(-state_abbr, -any_of("us"))
  } else {
    df
  }
}

```

```

standardize_state_col <- function(df) {
  candidates <- c("state", "state_name", "st", "abbr", "region", "statecode")
  nm <- names(df)
  hit <- candidates[candidates %in% nm]
  if (!length(hit)) stop("No state-like column found. Columns are: ", paste(nm, collapse=" "))
  if (hit[1] != "state") df <- rename(df, state = all_of(hit[1]))
  df %>% mutate(state = str_to_lower(str_squish(as.character(state))))
}

numify <- function(x) suppressWarnings(as.numeric(x))

renew_2021 <- reshape_if_wide(renew_2021, value_name = "renew_use")
renew_2022 <- reshape_if_wide(renew_2022, value_name = "renew_use")
renew_2023 <- reshape_if_wide(renew_2023, value_name = "renew_use")
total_2021 <- reshape_if_wide(total_2021, value_name = "total_use")
total_2022 <- reshape_if_wide(total_2022, value_name = "total_use")
total_2023 <- reshape_if_wide(total_2023, value_name = "total_use")

if ("state" %in% names(renew_2021)) renew_2021 <- standardize_state_col(renew_2021)
if ("state" %in% names(renew_2022)) renew_2022 <- standardize_state_col(renew_2022)
if ("state" %in% names(renew_2023)) renew_2023 <- standardize_state_col(renew_2023)
if ("state" %in% names(total_2021)) total_2021 <- standardize_state_col(total_2021)
if ("state" %in% names(total_2022)) total_2022 <- standardize_state_col(total_2022)
if ("state" %in% names(total_2023)) total_2023 <- standardize_state_col(total_2023)

price_all <- if (!"state" %in% names(price_all)) {
  reshape_if_wide(price_all, value_name = "price_per_kwh")
} else price_all %>% standardize_state_col()

ev_2023 <- if (!"state" %in% names(ev_2023)) {
  reshape_if_wide(ev_2023, value_name = "ev_count")
} else ev_2023 %>% standardize_state_col()

for (nm in c("renew_use", "total_use", "price", "price_per_kwh", "ev_count")) {
  for (dfn in c("renew_2021", "renew_2022", "renew_2023", "total_2021", "total_2022", "total_2023")) {
    if (exists(dfn) && nm %in% names(get(dfn))) {
      tmp <- get(dfn); tmp[[nm]] <- numify(tmp[[nm]]); assign(dfn, tmp)
    }
  }
}

```

```

if (!("year" %in% names(renew_2021))) renew_2021 <- renew_2021 %>% mutate(year = 2021)
if (!("year" %in% names(renew_2022))) renew_2022 <- renew_2022 %>% mutate(year = 2022)
if (!("year" %in% names(renew_2023))) renew_2023 <- renew_2023 %>% mutate(year = 2023)
if (!("year" %in% names(total_2021))) total_2021 <- total_2021 %>% mutate(year = 2021)
if (!("year" %in% names(total_2022))) total_2022 <- total_2022 %>% mutate(year = 2022)
if (!("year" %in% names(total_2023))) total_2023 <- total_2023 %>% mutate(year = 2023)

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

```

*## \*\*Part 3: Joining / Pivoting Datasets for Analysis\*\**

```

library(dplyr)
library(tidyr)
library(stringr)

```

```

numify <- function(x) suppressWarnings(as.numeric(gsub("[^0-9eE+\\-\\.]", "", as.character(x))))
pick_val <- function(df, prefs = character()) {
  hit <- prefs[prefs %in% names(df)]
  if (length(hit) >= 1) return(hit[1])
  cand <- setdiff(names(df), c("state", "year", "energy_source"))
  cand_num <- cand[sapply(df[cand], function(x) any(!is.na(numify(x)))) ]
  if (length(cand_num) == 0) stop("No numeric value column found; check Part 2 outputs.")
  cand_num[1]
}

```

```

renew_val_col <- pick_val(renew_all, c("renew_use", "value", "amount"))
total_val_col <- pick_val(total_all, c("total_use", "value", "amount"))

```

```

renew_by_state_year <- renew_all %>%
  mutate(val = numify(.data[[renew_val_col]])) %>%
  group_by(state, year) %>%
  summarise(renew_use = sum(val, na.rm = TRUE), .groups = "drop")

```

```

total_by_state_year <- total_all %>%
  mutate(val = numify(.data[[total_val_col]])) %>%
  group_by(state, year) %>%
  summarise(total_use = sum(val, na.rm = TRUE), .groups = "drop")

```

```

energy_joined <- renew_by_state_year %>%

```

```

inner_join(total_by_state_year, by = c("state", "year")) %>%
mutate(
  renew_share = ifelse(total_use > 0, (renew_use/total_use) * 100, NA_real_)
)

ev_val_col <- if ("ev_count" %in% names(ev_2023)) "ev_count" else
  if ("registrations" %in% names(ev_2023)) "registrations" else
  if ("value" %in% names(ev_2023)) "value" else NA_character_

energy_ev_2023 <- energy_joined %>%
  filter(year == 2023) %>%
  left_join(
    {
      if (!is.na(ev_val_col)) {
        ev_2023 %>%
          mutate(ev_count = numify(.data[[ev_val_col]])) %>%
          select(state, ev_count)
      } else {
        tibble(state = unique(energy_joined$state[energy_joined$year==2023]), ev_count = NA)
      }
    },
    by = "state"
  ) %>%
  mutate(
    ev_per_total_energy = ifelse(total_use > 0, ev_count/total_use, NA_real_)
  ) %>%
  select(state, year, renew_use, total_use, renew_share, ev_count, ev_per_total_energy)

renew_summary <- energy_joined %>%
  group_by(year) %>%
  summarise(
    avg_renew_share = mean(renew_share, na.rm = TRUE),
    median_renew_share = median(renew_share, na.rm = TRUE),
    min_renew_share = min(renew_share, na.rm = TRUE),
    max_renew_share = max(renew_share, na.rm = TRUE),
    .groups = "drop"
  )

```

Warning: There were 2 warnings in `summarise()`.

The first warning was:

i In argument: `min\_renew\_share = min(renew\_share, na.rm = TRUE)`.  
Caused by warning in `min()`:

! no non-missing arguments to min; returning Inf  
i Run `dplyr::last\_dplyr\_warnings()` to see the 1 remaining warning.

```
renew_change <- energy_joined %>%  
  filter(year %in% c(2021, 2023)) %>%  
  group_by(state) %>%  
  summarise(  
    y2021 = suppressWarnings(as.numeric(first(renew_share[year == 2021]))),  
    y2023 = suppressWarnings(as.numeric(first(renew_share[year == 2023])))  
  ) %>%  
  mutate(change_21_23 = y2023 - y2021) %>%  
  arrange(desc(change_21_23))
```

*## \*\*Part 4: Mapping Visualization\*\**

```
library(ggplot2)  
library(scales)
```

Warning: package 'scales' was built under R version 4.3.3

Attaching package: 'scales'

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

discard

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

col\_factor

```
library(ggrepel)
```

Warning: package 'ggrepel' was built under R version 4.3.3

```
library(maps) # built-in US polygons
```

Warning: package 'maps' was built under R version 4.3.3

Attaching package: 'maps'

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

map

```
library(dplyr)
library(stringr)

states_map <- map_data("state") %>%
  as_tibble() %>%
  rename(region = region) %>%
  mutate(region = str_to_lower(region))

map_df <- states_map %>%
  left_join(
    energy_ev_2023 %>%
    transmute(region = state, renew_share),
    by = "region"
  )

ggplot(map_df, aes(long, lat, group = group, fill = renew_share)) +
  geom_polygon(color = "white", linewidth = 0.2) +
  coord_fixed(1.3) +
  scale_fill_viridis_c(
    labels = label_percent(scale = 1),
    na.value = "grey90",
    name = "% Renewable"
  ) +
  labs(
    title = "Share of Renewable Energy by State (2023)",
    caption = "Data: course-provided CSVs (2021-2023)."
  ) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    panel.grid = element_blank(),
```

```
legend.position = "right"  
)
```

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>



Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>

Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x\$label), x\$x, x\$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>
```

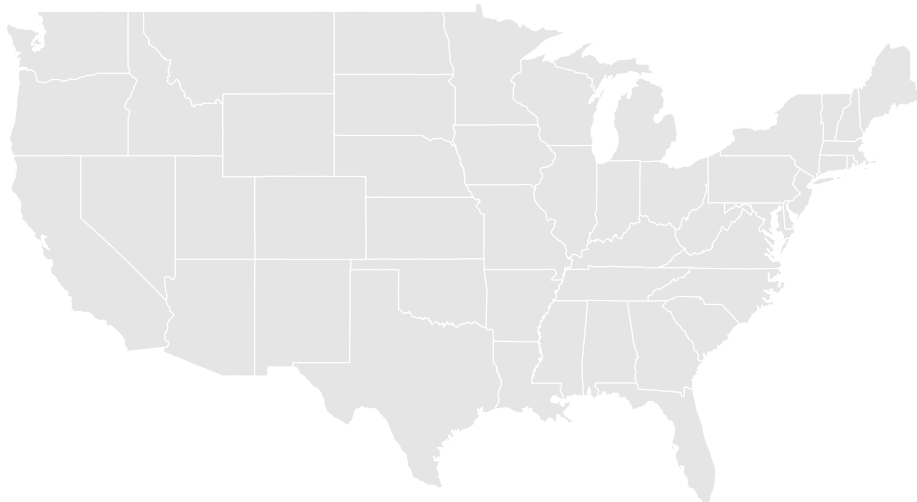
```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <e2>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :  
conversion failure on 'Data: course-provided CSVs (2021-2023).' in  
'mbcsToSbcs': dot substituted for <93>
```

## Share of Renewable Energy by State (2023)



Data: course-provided CSVs (2021...2023).

```
stopifnot(all(c("renew_share", "ev_count") %in% names(energy_ev_2023)))

ggplot(energy_ev_2023, aes(x = renew_share, y = ev_count, label = str_to_title(state))) +
  geom_point(alpha = 0.85) +
  ggrepel::geom_text_repel(size = 3, max.overlaps = 20) +
  scale_x_continuous(labels = label_percent(scale = 1)) +
  scale_y_continuous(labels = label_comma()) +
  labs(
    title = "Are EVs Concentrated in Cleaner-Grid States? (2023)",
    x = "Renewable Energy Share (%)",
    y = "EV Registrations (count)"
  ) +
  theme_minimal()
```

