weather

We are using `GSODR` package to download one year of daily weather data(temperature and precipitation).

```
# load required packages
library(GSODR)
library(ggplot2)
library(tidygeocoder)
```

```
# Download and Return weather station Data
stations_all <- get_inventory()
```

```
# Data Wrangling
weather_stations <- stations_all |>
  filter(CTRY == "AS") |> select(STNID, NAME, LAT, LON) |>
  rename(lat = LAT, lon = LON, stnid = STNID, stname = NAME) |>
  distinct(lat, lon, .keep_all = TRUE)
```

```
# save data
usethis::use_data(weather_stations, overwrite = TRUE)
```

```
# load data
load("../data/weather_stations.rda")
```

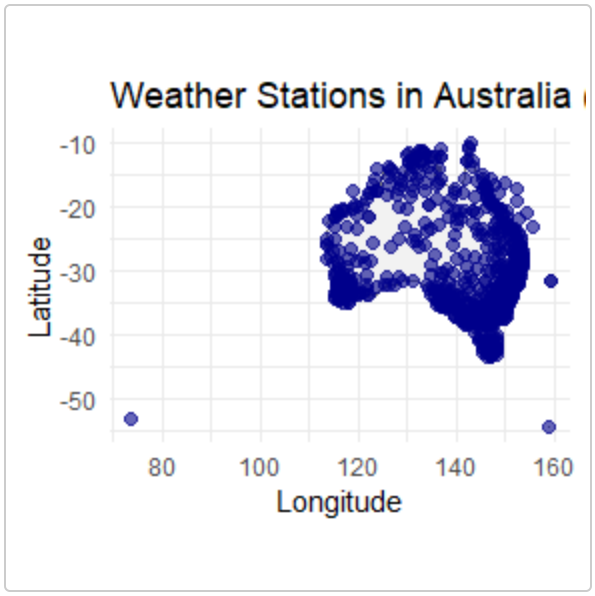
```
# print head of data
head(weather_stations, n = 2)
```

```
# A tibble: 6 x 7
#  stnid        stname                 lat   lon
#  <chr>        <chr>                <dbl> <dbl>
#1 941000-99999 KALUMBURU            -14.3  127.
#2 941020-99999 TROUGHTON ISLAND     -13.8  126.
#3 941030-99999 BROWSE ISLAND AWS    -14.1  124.
```

```
# Plotting the Map of the Weather Stations

ggplot(data = weather_stations) +
  borders("world", regions = "Australia", fill = "gray95", colour = "gray50") +
  geom_point(aes(x = lon, y = lat), color = "darkblue", alpha = 0.6, size = 2) +
  coord_fixed(1.3) +
  labs(
    title = "Weather Stations in Australia (2024)",
```

```
    x = "Longitude",
    y = "Latitude"
  ) +
  theme_minimal()
```



We added address column to the weather stations dataset

```
# Adding State Column and city column
# Reverse geocode using latitude and longitude
stations_with_location <- weather_stations |>
  reverse_geocode(lat = lat, long = lon, method = "osm")  # Uses OpenStreetMap
```

```
> head(stations_with_location, n = 3)
# A tibble: 3 x 7
#  stnid        stname              lat   lon address
#  <chr>        <chr>             <dbl> <dbl> <chr>
#1 941000-99999 KALUMBURU         -14.3  127. Drysdale River, Shire Of Wyndham-East Kimberley,
         Western Australia~
#2 941020-99999 TROUGHTON ISLAND  -13.8  126. Troughton Island Airport, Shire Of Wyndham-East
         Kimberley, Western~
```

Now we want to extract state and city name from address text.

```
# --- helper to extract city / state ---------------------------------------
parse_address <- function(addr) {
  tokens <- str_split(addr, ",")[[1]]          # split at commas
  tokens <- str_trim(tokens)                   # trim spaces
  tokens <- tokens[!(tokens == "Australia")]   # drop country
  tokens <- tokens[!str_detect(tokens, "^[0-9]+$")]  # drop pure numerics (postcodes)

  # State  = last token remaining
  # City   = first token remaining
  state <- tokens[length(tokens)]
  city  <- tokens[1]
```

```
  tibble(city = city, state = state)
}

# --- apply to tibble -----------------------------------------------------
stations_with_location <-
  stations_with_location %>%
  rowwise() %>%                                # operate row-by-row
  mutate(                                        # append new columns
    parsed = list(parse_address(address))
  ) %>%
  unnest(parsed) %>%                            # expand list-column
  ungroup()
```

```
> head(stations_with_location, n = 3)
# A tibble: 3 x 7
#  stnid        stname              lat   lon address                        city  state
#  <chr>        <chr>             <dbl> <dbl> <chr>                          <chr> <chr>
#1 941000-99999 KALUMBURU         -14.3  127. Drysdale River, Shire Of Wynd~ Drys~ West~
#2 941020-99999 TROUGHTON ISLAND  -13.8  126. Troughton Island Airport, Shi~ Trou~ West~
#3 941030-99999 BROWSE ISLAND AWS -14.1  124. Shire Of Wyndham-East Kimberl~ Shir~ West~
```

```
# save final data set
weather_stations <- stations_with_location
usethis::use_data(weather_stations, overwrite = TRUE)
```