Package: Rvoterdistance (via r-universe)

May 29, 2026

Type Package

Title Voter Distance to Polling Locations and Geographic Boundaries

Version 2.1.0

Description Calculates the distance between each voter in a voter file (given lat/long coordinates or sf point geometries) and multiple polling or vote-by-mail drop box locations. Returns nearest location, k-nearest locations, or all locations within a distance threshold. Also computes minimum distance from voters to geographic boundaries such as rivers, state borders, or district lines provided as 'sf' line or polygon geometries. Core computation uses the Haversine formula and spherical cross-track distance implemented in C++ via 'Rcpp'.

License GPL (>= 2)

URL https://github.com/lorenc5/Rvoterdistance

BugReports https://github.com/lorenc5/Rvoterdistance/issues

Encoding UTF-8

LazyData true

Depends R (>= 4.0)

Imports Rcpp (>= 1.0.0)

Suggests sf, ggplot2, rdrobust, testthat (>= 3.0.0), knitr, rmarkdown

LinkingTo Rcpp

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://lorenc5.r-universe.dev

Date/Publication 2026-05-29 18:00:16 UTC

RemoteUrl https://github.com/lorenc5/rvoterdistance

RemoteRef HEAD

RemoteSha 7627243b8eff87ece4245aff58832f3c65d318b7

Contents

`dbox` *King County ballot drop box locations*

Description

A data frame of ballot drop box locations in King County, Washington.

Usage

```
dbox
```

Format

A data frame with columns including `lat` and `long`.

Source

King County Elections

`dist_km` *Calculate minimum distance in kilometers*

Description

Given lat/lon vectors for voters and locations, returns the minimum Haversine distance in kilometers for each voter to the nearest location.

Usage

```
dist_km(lat1, lon1, lat2, lon2)
```

Arguments

lat1	Numeric vector of voter latitudes.
lon1	Numeric vector of voter longitudes.
lat2	Numeric vector of location latitudes.
lon2	Numeric vector of location longitudes.

Value

Numeric vector of minimum distances in kilometers.

Examples

```
data(meck_ev)
d <- dist_km(
  voter_meck$lat, voter_meck$long,
  early_meck$lat, early_meck$long
)
summary(d)
```

---

dist_mile *Calculate minimum distance in miles*

Description

Given lat/lon vectors for voters and locations, returns the minimum Haversine distance in miles for each voter to the nearest location.

Usage

```
dist_mile(lat1, lon1, lat2, lon2)
```

Arguments

lat1	Numeric vector of voter latitudes.
lon1	Numeric vector of voter longitudes.
lat2	Numeric vector of location latitudes.
lon2	Numeric vector of location longitudes.

Value

Numeric vector of minimum distances in miles.

Examples

```
data(meck_ev)
d <- dist_mile(
  voter_meck$lat, voter_meck$long,
  early_meck$lat, early_meck$long
)
summary(d)
```

`dist_to_boundary` *Distance from voters to a geographic boundary*

Description

Computes the minimum great-circle distance from each voter to the nearest point on a boundary line or polygon edge. The boundary can represent a river, state border, or any other geographic feature provided as an `sf` geometry object.

Usage

```
dist_to_boundary(
  voters,
  boundary,
  voter_coords = NULL,
  units = c("km", "miles", "meters"),
  progress = TRUE
)
```

Arguments

`voters`	A data frame, matrix, or `sf` POINT object containing voter locations. See `nearest_location()` for details on input formats.
`boundary`	An `sf` or `sfc` object with LINESTRING, MULTILINESTRING, POLYGON, or MULTIPOLYGON geometry. Will be transformed to WGS-84 (EPSG:4326) if needed.
`voter_coords`	Character vector of length 2 giving the column names for latitude and longitude in `voters` (e.g., `c("lat", "lon")`). Required if `voters` is a data frame; ignored for `sf` objects.
`units`	One of `"km"` (default), `"miles"`, or `"meters"`.
`progress`	Logical; show progress messages? Default `TRUE`.

Details

For polygon inputs the distance is measured to the polygon's **boundary** (perimeter), not to its interior. A point inside the polygon returns the positive distance to the nearest edge.

Core computation uses the spherical cross-track distance formula implemented in C++ for performance, with bounding-box pruning to skip distant segments.

Value

Numeric vector of distances (one per voter) in the requested units.

Examples

```
## Not run:
library(sf)
# Create a simple north-south boundary line
border <- st_sf(
  geometry = st_sfc(
    st_linestring(matrix(c(-109.05, 31.33, -109.05, 37.0),
      ncol = 2, byrow = TRUE
    )),
    crs = 4326
  )
)

voters <- data.frame(lat = c(35.08, 32.0), lon = c(-106.65, -108.5))
dist_to_boundary(voters, border, voter_coords = c("lat", "lon"))

## End(Not run)
```

---

early_meck *Mecklenburg County early voting locations*

Description

A data frame of early voting locations in Mecklenburg County, North Carolina.

Usage

```
early_meck
```

Format

A data frame with columns including `lat` and `long`.

Source

Mecklenburg County Board of Elections

haversine *Haversine distance between two points*

Description

Compute the Haversine (great-circle) distance between a single pair of lat/lon coordinates.

Usage

```
haversine(lat1, lon1, lat2, lon2, units = c("meters", "km", "miles"))
```

Arguments

lat1	Latitude of point 1 (degrees).
lon1	Longitude of point 1 (degrees).
lat2	Latitude of point 2 (degrees).
lon2	Longitude of point 2 (degrees).
units	One of "meters", "km", or "miles". Default "meters".

Value

Numeric scalar distance in the specified units.

Examples

```
# New York to London
haversine(40.7128, -74.0060, 51.5074, -0.1278, units = "km")
```

king_geo *King County voter sample with geocoded addresses*

Description

A sample of geocoded voter records from King County, Washington, including latitude and longitude of residential addresses.

Usage

```
king_geo
```

Format

A data frame with columns including Residence_Addresses_Latitude and Residence_Addresses_Longitude.

Source

King County voter file (anonymized sample)

`nearest_location` *Find nearest polling locations for each voter*

Description

Calculates the distance between each voter and a set of polling/drop box locations using the Haversine formula. Can return the single nearest location, the k nearest, or all locations within a distance threshold.

Usage

```
nearest_location(
  voters,
  locations,
  voter_coords = NULL,
  location_coords = NULL,
  k = 1L,
  max_dist = NULL,
  units = c("km", "miles", "meters"),
  append_data = TRUE,
  progress = FALSE
)
```

Arguments

`voters`	A data frame, matrix, or `sf` POINT object containing voter locations. If a data frame or matrix, must contain lat/lon columns specified by `voter_coords`.
`locations`	A data frame, matrix, or `sf` POINT object containing polling/drop box locations. If a data frame or matrix, must contain lat/lon columns specified by `location_coords`.
`voter_coords`	Character vector of length 2: `c("lat_col", "lon_col")` identifying the latitude and longitude columns in `voters`. Ignored if `voters` is an `sf` object.
`location_coords`	Character vector of length 2: `c("lat_col", "lon_col")` identifying the latitude and longitude columns in `locations`. Ignored if `locations` is an `sf` object.
`k`	Integer. Number of nearest locations to return per voter. Default `1`.
`max_dist`	Numeric or `NULL`. If not `NULL`, return all locations within this distance of each voter. Units controlled by `units`. Overrides `k`.
`units`	Character. One of `"km"`, `"miles"`, or `"meters"`. Default `"km"`.
`append_data`	Logical. If `TRUE` (default), include voter and matched location columns in the output. When `k = 1`, also appends the matched location row.
`progress`	Logical. If `TRUE`, print progress for large computations. Default `FALSE`.

Value

A data frame. If k = 1 and max_dist is NULL: one row per voter with distance columns (distance_m, distance_km, distance_miles). If k > 1 or max_dist is not NULL: one row per voter-location pair with a rank column.

Examples

```
data(meck_ev)

# Nearest single location for each voter
result <- nearest_location(voter_meck, early_meck,
  voter_coords = c("lat", "long"),
  location_coords = c("lat", "long")
)
head(result)

# 3 nearest locations per voter
result_k3 <- nearest_location(voter_meck, early_meck,
  voter_coords = c("lat", "long"),
  location_coords = c("lat", "long"),
  k = 3
)
head(result_k3)

# All locations within 10 km
result_10km <- nearest_location(voter_meck, early_meck,
  voter_coords = c("lat", "long"),
  location_coords = c("lat", "long"),
  max_dist = 10, units = "km"
)
head(result_10km)
```

---

voter_meck *Mecklenburg County voter sample with geocoded addresses*

Description

A sample of geocoded voter records from Mecklenburg County, North Carolina, including latitude and longitude.

Usage

```
voter_meck
```

Format

A data frame with columns including lat and long.

Source

Mecklenburg County voter file (anonymized sample)

Index