

# Package ‘geohash’

August 29, 2016

**Type** Package

**Title** Tools for Geohash Creation and Manipulation

**Version** 0.1.2

**Date** 2016-06-14

**Author** Oliver Keyes [aut, cre], G. Nagel [cph]

**Maintainer** Oliver Keyes <ironholds@gmail.com>

**Description** Provides tools to encode lat/long pairs into geohashes, decode those geohashes, and identify their neighbours.

**License** MIT + file LICENSE

**URL** <https://github.com/Ironholds/geohash/>

**BugReports** <https://github.com/Ironholds/geohash/issues>

**LazyData** TRUE

**LinkingTo** Rcpp

**Imports** Rcpp

**RoxygenNote** 5.0.1

**Suggests** testthat, knitr, rmarkdown, olctools

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2016-06-15 00:59:44

## R topics documented:

|                         |          |
|-------------------------|----------|
| geohash . . . . .       | 2        |
| gh_decode . . . . .     | 2        |
| gh_encode . . . . .     | 3        |
| gh_neighbours . . . . . | 3        |
| <b>Index</b>            | <b>5</b> |

---

|         |                                |
|---------|--------------------------------|
| geohash | <i>Tools for handling URLs</i> |
|---------|--------------------------------|

---

**Description**

The geohash package provides tools to encode lat/long pairs into geohashes, decode those geohashes, and identify neighbours their neighbours.

**See Also**

the [package vignette](#).

---

|           |                         |
|-----------|-------------------------|
| gh_decode | <i>Decode Geohashes</i> |
|-----------|-------------------------|

---

**Description**

gh\_decode takes geohashes and turns them back into latitude/longitude pairs, with an associated margin of error for each value.

**Usage**

```
gh_decode(hashes)
```

**Arguments**

hashes            a character vector of geohashes.

**Value**

a data.frame of four columns; "lat", "lng", "lat\_error" and "lng\_error"

**See Also**

[gh\\_encode](#) for generating geohashes, and [gh\\_neighbours](#) for identifying the neighbouring hash boxes to a geohash.

**Examples**

```
# A simple example:
gh_encode(lat = 42.60498046875, lng = -5.60302734375, precision = 5)
#[1] "ezs42"

gh_decode("ezs42")
# lat      lng      lat_error lng_error
# 42.60498 -5.603027 0.02197266 0.02197266
```

---

|           |   |
|-----------|---|
| gh_encode | <i>Encode Latitude and Longitude Values</i> |
|-----------|---|

---

### Description

[gh\\_encode](#) generates geohashes of a given precision from vectors of latitude and longitude values.

### Usage

```
gh_encode(lats, lngs, precision)
```

### Arguments

|           |  |
|-----------|--|
| lats      | a numeric vector of latitude values.   |
| lngs      | a numeric vector of longitude values. Must be the same length as lat.  |
| precision | an integer representing the precision the hashes should have. This should be between 1 and 10; if the precision requested is outside that range, it will use a default precision of 6. |

### Value

a character vector of hashes, the same length as lat and lng, with NA values where one of the equivalent lat/lng pair was NA.

### See Also

[gh\\_decode](#), for taking geohashes and turning them back into coordinates, and [gh\\_neighbours](#) for retrieving the neighbouring hashes to a particular hash.

### Examples

```
# A simple example:  
gh_encode(lat = 42.60498046875, lng = -5.60302734375, precision = 5)  
#[1] "ezs42"
```

---

|               |                                    |
|---------------|------------------------------------|
| gh_neighbours | <i>Get neighbours to geohashes</i> |
|---------------|------------------------------------|

---

### Description

Geohashes are calculated using a fixed-sized box, which means it's easy to take a geohash and extract the neighbouring boxes around it, in each direction. These functions either extract all neighbours, or individual neighbours, depending on your preference.

**Usage**

```
gh_neighbours(hashes)
```

```
north(hashes)
```

```
northeast(hashes)
```

```
east(hashes)
```

```
southeast(hashes)
```

```
south(hashes)
```

```
southwest(hashes)
```

```
west(hashes)
```

```
northwest(hashes)
```

**Arguments**

hashes            a vector of geohashes, which can be computed with [gh\\_encode](#)

**See Also**

[gh\\_encode](#) for generating hashes and [gh\\_decode](#) for resolving them into latitude and longitude pairs.

**Examples**

```
#Get a single neighbours  
north("ezs42")
```

```
#Get all neighbours!  
gh_neighbours("ezs42")
```

# Index

east (gh\_neighbours), 3

geohash, 2

geohash-package (geohash), 2

gh\_decode, 2, 3, 4

gh\_encode, 2, 3, 3, 4

gh\_neighbours, 2, 3, 3

neighbours (gh\_neighbours), 3

north (gh\_neighbours), 3

northeast (gh\_neighbours), 3

northwest (gh\_neighbours), 3

south (gh\_neighbours), 3

southeast (gh\_neighbours), 3

southwest (gh\_neighbours), 3

west (gh\_neighbours), 3