Package 'geoPlot'
24 July 2012

Type: Package
Title: Geocodes street or ip addresses and plots them on a map.
Version: 2.0
Date: 2012-07-24
Author: Randall Shane, PhD
Maintainer: Randall Shane, PhD <rshane@basexvi.com>

Description: There are 4 primary functions and RgoogleMaps and rjson are dependencies:
1) Resolve street address to point and box coordinates.
2) Resolve ip address to point coordinates.
3) Plot resolved points on specified Google map.
4) Determine distance between 2 points in km or mi using haversine formula.

License: GPL-2 or later

LazyLoad: yes

R Topics Documented:

- geoPlot package
- addrListLookup
- degrees2radians
- ipListLookup
- geoAddress
- geoIP
- geoPlot
- haversine

geoPlot Package

The geoPlot package uses 2 dependencies during install. These dependencies are RgoogleMaps and rjson. use the following to install with all dependencies:

```
> install.packages("geoPlot",dependencies=TRUE)
```

The package has several functions and will determine latitude and longitude coordinates for either street addresses or ip addresses and plot them on an exported 640 X 640 Google map. There is also the ability to determine the distance between two points using the haversine formula.

addrListLookup

This function resolves a data frame of street addresses through the geoAddress function. The data frame must have the following field order:

id, address, city, state/province, postal, country

The id field is an arbitrary reference field and the only requirement is that the data frame contain 5 fields. This means that one of the fields can be left off or other fields added. The address fields will be concatenated before being processed through the Google maps API for resolution so as long as there are 4, functionality is preserved.

USAGE:

```
>geoAddresses <- addrListLookup(addresses)
```

The data frame containing the following elements:
id, address, latitude (point coordinate), longitude (point coordinate),

north box coordinate, south box coordinate, east box coordinate, west box coordinate.

---

degrees2radians

This is a reference for the haversine function. If you want to convert degrees to radians, simply substitute the degree.

USAGE:

```
>degrees2radians(180)

>x <- -79.34526
>degrees2radians(x)
```

---

ipListLookup

This function resolves a list of ip addresses through the geoAddress function.

USAGE:

```
>geoIPs <- ipListLookup(ips)
```

A data frame is returned containing the following elements:
ip, statusCode, latitude, longitude, statusMessage, countryCode, countryName,
regionName, cityName, zipCode, timeZone

---

geoAddress

The geoAddress function uses the Google maps API to resolve a single street addresses to point and box coordinates. The data must have the following field order:

      id, address, city, state/province, postal, country

The id field is an arbitrary reference field and the only requirement is that the data frame contain 5 fields. This means that one of the fields can be left off or other fields added. The address fields will be concatenated before being processed through the Google maps API for resolution so as long as there are 4, functionality is preserved.

USAGE

```
>geoAddress(c("001","202 South Central Avenue","Flagler Beach","FL","32136","US"))
```

The data frame containing the following elements:
id, address, latitude (point coordinate), longitude (point coordinate),
north box coordinate, south box coordinate, east box coordinate, west box coordinate.

---

geoIP

The geoIP function uses the InfoDB API to resolve an IPV4 addresses to point coordinates.
The input data must only possess the ip address.

```
> geoIP("38.122.8.198")
```

A data frame is returned containing the following elements:
ip, statusCode, latitude, longitude, statusMessage, countryCode, countryName,
regionName, cityName, zipCode, timeZone

geoPlot

geoPlot pulls a Google map excerpt in 640 X 640 resolution of the type specified and plots point coordinates on the map in the specified color. The available map types are roadmap, mobile, satellite, terrain, hybrid, mapmaker-roadmap and mapmaker-hybrid with a default of mobile. The zoom level of the map can be set from 1 (entire globe) to 25 (ant mound). I'm not actually sure what the highest zoom is - at some point the map just doesn't get any closer. The default value is 6. The color of the plotted point is selectable with a default of red and the regular color palette is available.

PATTERN: geoPlot(x,zoom=NUM,maptype="TYPE",color= "COL")

USAGE:

```
>geoPlot(x)
>geoPlot(x,zoom=4)
>geoPlot(x,zoom=4,maptype="rodamap")
>geoPlot(x,zoom=7,color="orange")
>geoPlot(x,zoom=5,maptype="satellite",color="yellow")
>geoPlot(x,zoom=6,maptype="hybrid")
>geoPlot(x,zoom=3,maptype="roadmap",color="blue")
>geoPlot(x,zoom=1,maptype="mapmaker-roadmap",color="green")
>geoPlot(x,zoom=2,maptype="mapmaker-hybrid",color="purple")
```

Ths output is a PNG file of the specified map with the plotted points.

haversine

The haversine formula determines the distance between 2 points on a globe. Simply feed the points either as numbers or reference and the result will be calculated in kilometers and miles.

PATTERN: haversine(xLat, xLon, yLat, yLon)

USAGE:

```
>haversine("28.54703","-81.30292","26.93422","-80.09421")
>haversine(geoAddresses[1,3],geoAddresses[1,4],geoAddresses[2,3],geoAddresses[2,4])
>haversine(geoIPs[1,3],geoIPs[1,4],geoIPs[2,3],geoIPs[2,4])
```

The resulting list contains the following fields:
  xLat, xLon, yLat, yLon, distKm, distMi

note

Enjoy the package. Please feel free to contact me with any questions or enhancement ideas at rshane@basexvi.com

Thanks!
Randall Shane, PhD