

Package ‘SensusR’

July 11, 2017

Type Package

Title Sensus Analytics

Version 2.2.0

Date 2017-07-11

Author Matthew S. Gerber

Maintainer Matthew S. Gerber <gerber.matthew@gmail.com>

Description Provides access and analytic functions for Sensus data.

License GPL-3

Copyright The Rector and Visitors of the University of Virginia

URL <https://github.com/MatthewGerber/sensus/wiki>

Imports jsonlite (>= 0.9.16), lubridate (>= 1.3.3), plyr (>= 1.8.3),
ggmap (>= 2.6.1), ggplot2 (>= 2.2.1), R.utils (>= 2.3.0),
openssl (>= 0.9.6)

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-07-11 16:51:03 UTC

R topics documented:

plot.AccelerometerDatum	2
plot.AltitudeDatum	3
plot.BatteryDatum	3
plot.CellTowerDatum	4
plot.CompassDatum	4
plot.LightDatum	5
plot.LocationDatum	5
plot.ScreenDatum	6
plot.SoundDatum	6
plot.SpeedDatum	7
plot.TelephonyDatum	8

plot.WlanDatum	8
sensus.decompress.json	9
sensus.decrypt.bin.files	9
sensus.get.all.timestamp.lags	10
sensus.get.timestamp.lags	11
sensus.plot.data.frequency.by.day	11
sensus.plot.lag.cdf	12
sensus.read.json	12
sensus.remove.device.id	13
sensus.sync.from.aws.s3	14
sensus.write.csv.files	14
sensus.write.rdata.files	15
SensusR	16
trim	16
trim.leading	17
trim.trailing	17
Index	18

`plot.AccelerometerDatum`
Plot accelerometer data.

Description

Plot accelerometer data.

Usage

```
## S3 method for class 'AccelerometerDatum'
plot(x, pch = ".", type = "l", ...)
```

Arguments

<code>x</code>	Accelerometer data.
<code>pch</code>	Plotting character.
<code>type</code>	Line type.
<code>...</code>	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$AccelerometerDatum)
```

plot.AltitudeDatum *Plot altitude data.*

Description

Plot altitude data.

Usage

```
## S3 method for class 'AltitudeDatum'  
plot(x, pch = ".", type = "l", ...)
```

Arguments

x	Altitude data.
pch	Plotting character.
type	Line type.
...	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")  
data = sensus.read.json(data.path)  
plot(data$AltitudeDatum)
```

plot.BatteryDatum *Plot battery data.*

Description

Plot battery data.

Usage

```
## S3 method for class 'BatteryDatum'  
plot(x, pch = ".", type = "l", ...)
```

Arguments

x	Battery data.
pch	Plotting character.
type	Line type.
...	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$BatteryDatum)
```

```
plot.CellTowerDatum    Plot cell tower data.
```

Description

Plot cell tower data.

Usage

```
## S3 method for class 'CellTowerDatum'
plot(x, ...)
```

Arguments

x	Cell tower data.
...	Other plotting arguments.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$CellTowerDatum)
```

```
plot.CompassDatum    Plot compass data.
```

Description

Plot compass data.

Usage

```
## S3 method for class 'CompassDatum'
plot(x, pch = ".", type = "l", ...)
```

Arguments

x	Compass data.
pch	Plotting character.
type	Line type.
...	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$CompassDatum)
```

plot.LightDatum *Plot light data.*

Description

Plot light data.

Usage

```
## S3 method for class 'LightDatum'
plot(x, pch = ".", type = "l", ...)
```

Arguments

x	Light data.
pch	Plotting character.
type	Line type.
...	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$LightDatum)
```

plot.LocationDatum *Plot location data.*

Description

Plot location data.

Usage

```
## S3 method for class 'LocationDatum'
plot(x, ...)
```

Arguments

x Location data.
 ... Arguments to pass to plotting routines. This can include two special arguments: qmap.args (passed to `qmap`) and geom.point.args (passed to `geom_point`).

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$LocationDatum)
```

plot.ScreenDatum *Plot screen data.*

Description

Plot screen data.

Usage

```
## S3 method for class 'ScreenDatum'
plot(x, ...)
```

Arguments

x Screen data.
 ... Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$ScreenDatum)
```

plot.SoundDatum *Plot sound data.*

Description

Plot sound data.

Usage

```
## S3 method for class 'SoundDatum'
plot(x, pch = ".", type = "l", ...)
```

Arguments

x	Sound data.
pch	Plotting character.
type	Line type.
...	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$SoundDatum)
```

<code>plot.SpeedDatum</code>	<i>Plot speed data.</i>
------------------------------	-------------------------

Description

Plot speed data.

Usage

```
## S3 method for class 'SpeedDatum'
plot(x, pch = ".", type = "l", ...)
```

Arguments

x	Speed data.
pch	Plotting character.
type	Line type.
...	Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(data$SpeedDatum)
```

plot.TelephonyDatum *Plot telephony data.*

Description

Plot telephony data.

Usage

```
## S3 method for class 'TelephonyDatum'  
plot(x, ...)
```

Arguments

x Telephony data.
... Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")  
data = sensus.read.json(data.path)  
plot(data$TelephonyDatum)
```

plot.WlanDatum *Plot WLAN data.*

Description

Plot WLAN data.

Usage

```
## S3 method for class 'WlanDatum'  
plot(x, ...)
```

Arguments

x WLAN data.
... Other plotting parameters.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")  
data = sensus.read.json(data.path)  
plot(data$WlanDatum)
```

```
sensus.decompress.json
```

Decompresses JSON files downloaded from AWS S3.

Description

Decompresses JSON files downloaded from AWS S3.

Usage

```
sensus.decompress.json(local.path)
```

Arguments

`local.path` Path to location on local machine.

Value

None

Examples

```
# sensus.decompress.json("~/Desktop/data")
```

```
sensus.decrypt.bin.files
```

Decrypts Sensus .bin files that were encrypted using asymmetric public/private key encryption.

Description

Decrypts Sensus .bin files that were encrypted using asymmetric public/private key encryption.

Usage

```
sensus.decrypt.bin.files(data.path, is.directory = TRUE, recursive = TRUE,  
  rsa.private.key.path, rsa.private.key.password = askpass,  
  replace.files = TRUE)
```

Arguments

`data.path` Path to Sensus .bin data (either a file or a directory).
`is.directory` Whether or not the path is a directory.
`recursive` Whether or not to read files recursively from directory indicated by path.
`rsa.private.key.path` Path to RSA private key generated using OpenSSL.
`rsa.private.key.password` Password used to decrypt the RSA private key.
`replace.files` Whether or not to delete .bin files after they have been decrypted.

Value

None

Examples

```

# data.path = system.file("extdata", "example_data", package="SensusR")
# sensus.decrypt.bin.files(data.path = data.path,
#                           rsa.private.key.path = "/path/to/private.pem",
#                           replace.files = FALSE)
  
```

```
sensus.get.all.timestamp.lags
```

Get timestamp lags for a Sensus data frame.

Description

Get timestamp lags for a Sensus data frame.

Usage

```
sensus.get.all.timestamp.lags(data)
```

Arguments

`data` Data to plot lags for (e.g., the result of `read.sensus.json`).

Value

List of lags organized by datum type.

Examples

```

data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
lags = sensus.get.all.timestamp.lags(data)
plot(lags[["AccelerometerDatum"]])
  
```

```
sensus.get.timestamp.lags
```

Get timestamp lags for a Sensus datum.

Description

Get timestamp lags for a Sensus datum.

Usage

```
sensus.get.timestamp.lags(datum)
```

Arguments

datum One element of a Sensus data frame (e.g., data\$CompassDatum).

Value

List of lags.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
plot(sensus.get.timestamp.lags(data$AccelerometerDatum))
```

```
sensus.plot.data.frequency.by.day
```

Plot data frequency by day.

Description

Plot data frequency by day.

Usage

```
sensus.plot.data.frequency.by.day(datum, xlab = "Study Day",
  ylab = "Data Frequency", main = "Data Frequency")
```

Arguments

datum Data frame for a single datum.
xlab Label for x-axis.
ylab Label for y-axis.
main Label for plot.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
sensus.plot.data.frequency.by.day(data$AccelerometerDatum)
```

```
sensus.plot.lag.cdf      Plot the CDF of inter-reading time lags.
```

Description

Plot the CDF of inter-reading time lags.

Usage

```
sensus.plot.lag.cdf(datum, xlim = c(0, 1),
  xlab = "Inter-reading time (seconds)", ylab = "Percentile",
  main = paste("Inter-reading times (n=", nrow(datum), ")", sep = ""))
```

Arguments

datum	Data frame for a single datum.
xlim	Limits for the x-axis.
xlab	Label for x-axis.
ylab	Label for y-axis.
main	Label for plot.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
sensus.plot.lag.cdf(data$AccelerometerDatum)
```

```
sensus.read.json      Read JSON-formatted Sensus data.
```

Description

Read JSON-formatted Sensus data.

Usage

```
sensus.read.json(data.path, is.directory = TRUE, recursive = TRUE,
  convert.to.local.timezone = TRUE, local.timezone = Sys.timezone())
```

Arguments

<code>data.path</code>	Path to Sensus JSON data (either a file or a directory).
<code>is.directory</code>	Whether or not the path is a directory.
<code>recursive</code>	Whether or not to read files recursively from directory indicated by path.
<code>convert.to.local.timezone</code>	Whether or not to convert timestamps to the local timezone.
<code>local.timezone</code>	If converting timestamps to local timesonze, the local timezone to use.

Value

All data, listed by type.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
```

```
sensus.remove.device.id
```

Removes all data associated with a device ID from a data collection.

Description

Removes all data associated with a device ID from a data collection.

Usage

```
sensus.remove.device.id(datum, device.id)
```

Arguments

<code>datum</code>	Data collection to process.
<code>device.id</code>	Device ID to remove.

Value

Data without a particular device ID.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
filtered.data = sensus.remove.device.id(data$AccelerometerDatum, "a448s0df98f")
```

```
sensus.sync.from.aws.s3
```

Synchronizes data from Amazon S3 to a local path.

Description

Synchronizes data from Amazon S3 to a local path.

Usage

```
sensus.sync.from.aws.s3(s3.path, profile = "default",
  local.path = tempfile(), aws.path = "/usr/local/bin/aws", delete = TRUE,
  decompress = TRUE)
```

Arguments

s3.path	Path within S3. This can be a prefix (partial path).
profile	AWS credentials profile to use for authentication.
local.path	Path to location on local machine.
aws.path	Path to AWS client.
delete	Whether or not to delete local files that are not present in the S3 path.
decompress	Whether or not to decompress any gzip files after downloading them.

Value

Local path to location of downloaded data.

Examples

```
# data.path = sensus.sync.from.aws.s3("s3://bucket/path/to/data", local.path = "~/Desktop/data")
```

```
sensus.write.csv.files
```

Write data to CSV files.

Description

Write data to CSV files.

Usage

```
sensus.write.csv.files(data, directory, file.name.prefix = "")
```

Arguments

`data` Data to write, as read using [sensus.read.json](#).
`directory` Directory to write CSV files to. Will be created if it does not exist.
`file.name.prefix` Prefix to add to the generated file names.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
# sensus.write.csv.files(data, directory = "/path/to/directory")
```

```
sensus.write.rdata.files
```

Write data to rdata files.

Description

Write data to rdata files.

Usage

```
sensus.write.rdata.files(data, directory, file.name.prefix = "")
```

Arguments

`data` Data to write, as read using [sensus.read.json](#).
`directory` Directory to write CSV files to. Will be created if it does not exist.
`file.name.prefix` Prefix to add to the generated file names.

Examples

```
data.path = system.file("extdata", "example_data", package="SensusR")
data = sensus.read.json(data.path)
# sensus.write.csv.files(data, directory = "/path/to/directory")
```

SensusR

SensusR: Sensus Analytics

Description

Provides access and analytic functions for Sensus data. More information can be found at the following URL:

Details

<https://github.com/predictive-technology-laboratory/sensus/wiki>

SensusR functions

The SensusR functions handle reading, cleaning, plotting, and otherwise analyzing data collected via the Sensus system.

trim

Trim leading and trailing white space from a string.

Description

Trim leading and trailing white space from a string.

Usage

```
trim(x)
```

Arguments

x String to trim.

Value

Result of trimming.

Examples

```
trim(" asdf ")
```

trim.leading	<i>Trim leading white space from a string.</i>
--------------	--

Description

Trim leading white space from a string.

Usage

```
trim.leading(x)
```

Arguments

x	String to trim.
---	-----------------

Value

Result of trimming.

Examples

```
trim.leading(" asdfasdf")
```

trim.trailing	<i>Trim trailing white space from a string.</i>
---------------	---

Description

Trim trailing white space from a string.

Usage

```
trim.trailing(x)
```

Arguments

x	String to trim.
---	-----------------

Value

Result of trimming.

Examples

```
trim.trailing("asdfasdf ")
```

Index

`geom_point`, [6](#)

`plot.AccelerometerDatum`, [2](#)

`plot.AltitudeDatum`, [3](#)

`plot.BatteryDatum`, [3](#)

`plot.CellTowerDatum`, [4](#)

`plot.CompassDatum`, [4](#)

`plot.LightDatum`, [5](#)

`plot.LocationDatum`, [5](#)

`plot.ScreenDatum`, [6](#)

`plot.SoundDatum`, [6](#)

`plot.SpeedDatum`, [7](#)

`plot.TelephonyDatum`, [8](#)

`plot.WlanDatum`, [8](#)

`qmap`, [6](#)

`sensus.decompress.json`, [9](#)

`sensus.decrypt.bin.files`, [9](#)

`sensus.get.all.timestamp.lags`, [10](#)

`sensus.get.timestamp.lags`, [11](#)

`sensus.plot.data.frequency.by.day`, [11](#)

`sensus.plot.lag.cdf`, [12](#)

`sensus.read.json`, [12](#), [15](#)

`sensus.remove.device.id`, [13](#)

`sensus.sync.from.aws.s3`, [14](#)

`sensus.write.csv.files`, [14](#)

`sensus.write.rdata.files`, [15](#)

`SensusR`, [16](#)

`SensusR-package (SensusR)`, [16](#)

`trim`, [16](#)

`trim.leading`, [17](#)

`trim.trailing`, [17](#)