Package ‘steemr’

January 16, 2019

Version  0.1.3
Date  2019-01-15
Title  A Tool for Processing Steem Data
Author  Peng Zhao <pzhao@pzhao.net>
Maintainer  Peng Zhao <pzhao@pzhao.net>
Depends  R (>= 3.1.0)
Imports  RCurl, XML, rlist, wordcloud, tm, zoo, openair, lattice, latticeExtra, RODBC, VennDiagram, beginr, RColorBrewer, blogdown, ggplot2, httr, knitr, mongolite, plyr, shiny, lubridate, rjson, stringi, stringr, purrr, data.table, htmltab

Suggests

Description  Steem is a blockchain-based social media platform (see <https://en.wikipedia.org/wiki/Steemit>). The Steem social activity data are saved in the Steem blockchain, the SteemDB database, the SteemSQL database, and so on. ‘steemr’ is an R package that downloads the Steem data from the SteemDB and SteemSQL servers, re-organizes the data in a user-friendly way, and visualizes the data for further analysis.

License  GPL-3 | file LICENSE

URL  https://github.com/pzhaonet/steemr

BugReports  https://github.com/pzhaonet/steemr/issues

RoxygenNote  6.1.1

NeedsCompilation  no

LazyData  true

Repository  CRAN

Date/Publication  2019-01-16 09:20:03 UTC
R topics documented:

- accountCount
- acnsub
- adailyf
- avotenot
- avotep
- bblog
- bmd
- chasplit
- clearferfing
- clear_title
- gaccounts
- gblog
- gcner
- gcomments
- gdelegation
- getAccount
- getAccountVotes
- getBlog
- getComments
- getDelegation
- getNodes
- getPost
- getPostsByTag
- getReplies
- getSteemProperties
- getTransactions
- getTrending
- getWitnesses
- gfollow
- gfolloweer
- gfollowing
- gid
- gidpostl
- gidposts
- gpost
- gposts
- gspmv
- gur
- gvote
- gvoteer
- idlink
- pcner
- pdate
- phist
- phour
- repcalc
**accountCount**

*Number of Accounts on Steem Blockchain*

**Description**

Get the count of the number of accounts on the Steem Blockchain

**Usage**

```plaintext
accountCount(node)
```

**Arguments**

*node*  
Optional Argument (Steem Node to Query)

**Value**

number of accounts

**Examples**

```plaintext
## Not run:
accountCount()

## End(Not run)
```
acnsub

Analysis of the CN sub categories.

Description

Analysis of the CN sub categories.

Usage

acnsub(from = Sys.Date() - 7, to = Sys.Date(), sql_con, if_plot = FALSE, top = 10)

Arguments

from A Date object or character in '2017-10-24' format of the beginning of the period
to A Date object or character in '2017-10-24' format of the end of the period
sql_con A SQL connection
if_plot A logic value of whether plot the time series
top A numeric value of the Top tags for plotting

Value

A figure showing the active cn sub tag

adailyf

Calculate the the daily frequency. adailyf means 'analysis of the daily frequency'.

Description

Calculate the the daily frequency. adailyf means 'analysis of the daily frequency'.

Usage

adailyf(mydata, datecol, if_plot = FALSE, col = "steelblue3", ylab = "Daily posts")

Arguments

mydata A data frame with a date column.
datecol A character string of the date column name. The date column could either be Date or Character (in '%Y-%m-%d' format).
if_plot A logic value of whether plot the time series
col A color for plotting
ylab A character string of the y label
**avotenot**

Value

A dataframe of the daily frequency

---

**avotenot**

*Find which followers have not voted a post yet. avotenot means 'analysis who has not voted yet'.*

---

**Description**

Find which followers have not voted a post yet. avotenot means 'analysis who has not voted yet'.

**Usage**

```r
avotenot(postlink = NA)
```

**Arguments**

- `postlink`: A character string of the link to a target post.

**Value**

A character vector of the name list of the followers who have not voted the target post yet.

**Examples**

```r
## Not run:
avotenot("cn@dapeng/steemit-markdown")
## End(Not run)
```

---

**avotep**

*Summary of the voters of a series of posts. avotep means 'analysis of the votes on the given posts retrieved with the appbase_api method.'*

---

**Description**

Summary of the voters of a series of posts. avotep means 'analysis of the votes on the given posts retrieved with the appbase_api method.'

**Usage**

```r
avotep(mypost = NA, if_plot = TRUE, top = 10)
```
bblog

Build a hugo blog site for a steem author

Description

Build a hugo blog site for a steem author

Usage

bblog(author = NA, post_df, dest_path = "blog", initial = FALSE, template = "xmin", post_df_source = c("appbase_api", "steemsql.com"), orginal_link = c("steemit.com", "cnsteem.com", "busy.org", "steemdb.com", "steem.com"), my_github = "your_name/your_repo")

Arguments

author author name without @
post_df A dataframe with the posts contents retrieved from SteemSQL
dest_path A character string
initial if initialize a site
template the hugo template
post_df_source A character string of the data frame source.
orginal_link A character string
my_github A character string of a github repo

Value

a blogdown-hugo web site
**Examples**

```r
## Not run:
bblog()
```

```r
## End(Not run)
```

---

**bmd**

`bmd()`

*build markdown files from the posts*

**Description**

build markdown files from the posts

**Usage**

```r
bmd(post_df = NA, dest_path = "blog", post_df_source = c("appbase_api", "steemsql.com"),
if_yaml = FALSE, original_link = c("steemit.com", "cnsteem.com", "busy.org",
"steemdb.com", "steemd.com"))
```

**Arguments**

- `post_df` A data frame of the posts.
- `dest_path` A character string of the destination path for the markdown files.
- `post_df_source` A character string of the data frame source.
- `if_yaml` A logical value of whether the markdown files contain yaml headers.
- `original_link` A character string vector of the original links to the posts.

**Value**

markdown files.

**Examples**

```r
## Not run:
bmd()
```

```r
## End(Not run)
```
charsplit

Split the characters in a data frame

Description
Split the characters in a data frame

Usage
charsplit(dataframe = NA)

Arguments
dataframe A column of a dataframe with characters to split

Value
A character vector

clearferfing
Clear the format of the 'follow' information from steemdb.com.

Description
Clear the format of the 'follow' information from steemdb.com.

Usage
clearferfing(x)

Arguments
x A character string of the follower information from steemdb.com.

Value
A character string of the formatted follower information
clear_title

Clear the characters in the titles

Description
Clear the characters in the titles

Usage
clear_title(title)

Arguments
title The post title

Value
A clear title

gaccounts
Get the Steem account information within a period from SteemSQL
gaccounts means 'get account information'.

Description
Get the Steem account information within a period from SteemSQL gaccounts means 'get account information'.

Usage
gaccounts(from = Sys.Date() - 7, to = Sys.Date(), select = c("name", "created", "post_count", "last_post"), sql_con = NA, ylab = "Daily New Accounts", if_plot = FALSE)

Arguments
from A Date object or character in '2017-10-24' format of the beginning of the period
to A Date object or character in '2017-10-24' format of the end of the period
select A character string vector of the column names
sql_con A SQL connection
ylab Label on the y-axis
if_plot A logic value of whether plot the time series
Value

A data frame of the account information with a figure

Examples

```r
## Not run:
gaccounts()

## End(Not run)
```

---

**gblog**  
*Rename the getBlog() function from the steemRdata package*

Description

Rename the getBlog() function from the steemRdata package

Usage

```r
gblog(id = NA)
```

Arguments

- **id**  
  A Steem ID

Value

A dataframe with the ID’s posts

Examples

```r
## Not run:
gblog()

## End(Not run)
```
**gcner**

*Get the CNer name list*

**Description**

Get the CNer name list

**Usage**

```
getNER(name = Sys.Date(), system = TRUE)
```

**Arguments**

- `name`: A Date or character string in `"Y-m-d"` format

**Value**

A data frame of the CNer

**Examples**

```
## Not run:
gcner()
```

```
## End(Not run)
```

---

**gcomments**

*Get the comment records of an Steem ID within a period from SteemSQL. gcomments means 'get comment information'.*

**Description**

Get the comment records of an Steem ID within a period from SteemSQL. gcomments means 'get comment information'.

**Usage**

```
gcomments(id = NA, from = Sys.Date() - 7, to = Sys.Date(), select = c("root_title", "root_comment", "created", "body"), sql_con, ylab = "Daily Comments", if_plot = FALSE)
```

---

---

---
gdelegation

Arguments

id
A character string of a Steem ID

from
A Date object or character in '2017-10-24' format of the beginning of the period

to
A Date object or character in '2017-10-24' format of the end of the period

select
A character string vector of the column names

sql_con
A SQL connection

ylab
Label on the y-axis

if_plot
A logic value of whether plot the time series

Value
A data frame of the comment information with a figure

Examples

## Not run:
gcomments()

## End(Not run)

---

**gdelegation**

Get the delegation information of a Steem ID from SteemSQL. gdelegation means 'get delegation information'.

Usage

gdelegation(id = NA, sql_con, if_plot = FALSE)

Arguments

id
A character string of a Steem ID

sql_con
A SQL connection

if_plot
A logic value of whether plot the time series

Value
A data frame of the delegation information with a figure
getAccount

**Examples**

```python
## Not run:
gdelegation()

## End(Not run)
```

---

**getAccount**

**Account Details**

**Description**

Get the details of a user account

**Usage**

```python
getAccount(username, node)
```

**Arguments**

- `username`: user name
- `node`: Optional Argument (Steem Node to Query)

**Value**

Account Details

**Examples**

```python
## Not run:
getAccount("eroche")

## End(Not run)
```

---

**getAccountVotes**

**Votes made by an Account**

**Description**

Get a list of all the votes made by an account

**Usage**

```python
getAccountVotes(user = "eroche", node)
```
getBlog

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Account to query</td>
</tr>
<tr>
<td>node</td>
<td>Optional Argument (Steem Node to Query)</td>
</tr>
</tbody>
</table>

**Value**

Data Table with list of votes and what they voted on

**Examples**

```
## Not run:
getAccountVotes("eroche")
getAccountVotes("eroche", node = "https://api.steemit.com")

## End(Not run)
```

---

**getBlog**

**Blog History**

**Description**

Get all main posts from a users blog (excluding Resteems). This function may take some time to process depending on the size of the blog history on an account

**Usage**

```
getBlog(username, node)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Username of Blog Author.</td>
</tr>
<tr>
<td>node</td>
<td>Optional Argument (Steem Node to Query)</td>
</tr>
</tbody>
</table>

**Value**

Data Table with Details of Blog Posts

**Examples**

```
## Not run:
getBlog("eroche")

## End(Not run)
```
**getComments**  
*Comment List*

**Description**
Get a list of all the comments from a users blog. This function may take some time to process depending on the size of the transaction history on an account.

**Usage**
```
getComments(username, node)
```

**Arguments**
- `username`  
  Username of Comment Author.
- `node`  
  Optional Argument (Steem Node to Query)

**Value**
```
Data.Table with Details of Comments Posts
```

**Examples**
```
## Not run:
getComments("eroche")

## End(Not run)
```

---

**getDelegation**  
*Delegations Made by an Account*

**Description**
Delegations Made by an Account

**Usage**
```
getDelegation(user = "eroche", node)
```

**Arguments**
- `user`  
  Account To Query
- `node`  
  Optional Argument (Steem Node to Query)
Value

Data Table with list of delegations

Examples

```r
## Not run:
getDelegation("eroche")

## End(Not run)
```

---

table

<table>
<thead>
<tr>
<th>getNodes</th>
<th>List of Steem RPC nodes</th>
</tr>
</thead>
</table>

Description

The functions in steemRdata use public RPC nodes. This function returns a list of possible nodes which can be specified in the function calls.

Usage

```r
getNodes()
```

Value

List of nodes

Examples

```r
## Not run:
getNodes()

## End(Not run)
```

---

table

<table>
<thead>
<tr>
<th>getPost</th>
<th>Details of a Steem Post</th>
</tr>
</thead>
</table>

Description

Get details of a post, specified by the unique link (username and permlink)

Usage

```r
ggetPost(username, permlink, node)
```
getPostsByTag

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Username of Post Author.</td>
</tr>
<tr>
<td>permalink</td>
<td>Permalink of Post</td>
</tr>
<tr>
<td>node</td>
<td>Optional Argument (Steem Node to Query)</td>
</tr>
</tbody>
</table>

**Value**

List with Details of Post

**Examples**

```
## Not run:
getPost("eroche", "time-series-with-r")
getPost("eroche", "time-series-with-r", "https://api.steemit.com")

## End(Not run)
```

---

**getPostsByTag**

Recent Posts containing a particular tag.

**Description**

Get posts using a specific tag in Chronological order

**Usage**

```
getPostsByTag(tag = "steem", limit = 1, node)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>tag to search.</td>
</tr>
<tr>
<td>limit</td>
<td>number of items to return</td>
</tr>
<tr>
<td>node</td>
<td>Optional Argument (Steem Node to Query)</td>
</tr>
</tbody>
</table>

**Value**

Data.Table with Details of Posts

**Examples**

```
## Not run:
getPostsByTag("letseat", 1)

## End(Not run)
```
getReplies  Replies Made to a Post

Description
Replies Made to a Post

Usage
getReplies(user, permlink, node)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Account To Query</td>
</tr>
<tr>
<td>permlink</td>
<td>Permlink of post to query</td>
</tr>
<tr>
<td>node</td>
<td>Optional Argument (Steem Node to Query)</td>
</tr>
</tbody>
</table>

Value
Data Table with list of replies made to specified post

Examples

```r
# Not run:
getReplies("eroche", "data-wrangling-with-r")
```

getSteemProperties  Steem Properties

Description
Get details about the latest state of the Steem Blockchain.

Usage
getSteemProperties(node)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Optional Argument (Steem Node to Query)</td>
</tr>
</tbody>
</table>

Value
steem properties
**getTransactions**

### Examples

```r
## Not run:
g取得了SteemProperties()

## End(Not run)
```

<table>
<thead>
<tr>
<th>getTransactions</th>
<th>Transactions made on an account</th>
</tr>
</thead>
</table>

### Description

This function may take some time to process depending on the size of the transaction history on an account.

### Usage

```r
g取得了Transactions(user, n, node)
```

### Arguments

- **user**: Account To Query
- **n**: Number of Transactions since beginning of account history
- **node**: node

### Value

The First n of transactions on an account

### Examples

```r
## Not run:
g取得了transactions("eroche", 100)

## End(Not run)
```
**getTrending**  
*Current Trending Posts*

**Description**  
Current Trending Posts

**Usage**
getTrending(tag = "", limit = 100, node)

**Arguments**
- **tag**  
  Tag to Filter
- **limit**  
  Number of Posts to Return
- **node**  
  Optional Argument (Steem Node to Query)

**Value**
List with Details of Posts

**Examples**
```bash
## Not run:
getTrending()  # Get first 100 Trending Posts
gTTrending(20)  # Get Top 10 Trending Posts
gTTrending(20, node = "https://api.steemit.com")

## End(Not run)
```

**getWitnesses**  
*Current Steem Witnesses and Rank*

**Description**  
Not all witnesses returned with this function will be active.

**Usage**
getWitnesses(limit = 1000, node)

**Arguments**
- **limit**  
  Number of Witnesses to Return
- **node**  
  Optional Argument (Steem Node to Query)

Get details of the witnesses
**gfollow**

**Value**

List with Details of Witnesses

**Examples**

```r
## Not run:
gfollow()  # Get first 1000 Witnesses
gfollow(20)  # Get Top 20 Witnesses
gfollow(20, node = "https://api.steemit.com")

## End(Not run)
```

---

**gfollow**

Get a name list of an ID’s followers and following. gfollow means ‘get an ID’s follower information’.

**Description**

Get a name list of an ID’s followers and following. gfollow means ‘get an ID’s follower information’.

**Usage**

```r
gfollow(id = NA, method = c("steemdb.com", "steemsql.com", "steemdata.com"), sql_con)
```

**Arguments**

- `id` A character string of a Steem ID without ‘@’.
- `method` A character string of the Steem data server to connect.
- `sql_con` A connection to the SteemSQL server.

**Value**

A name list of an ID’s followers and following

**Examples**

```r
## Not run:
gfollow("dapeng")

## End(Not run)
```
### gfollower

*A list of an id’s followers from steemdb.com. gfollower means 'get the follower information'.*

#### Description

A list of an id’s followers from steemdb.com. gfollower means 'get the follower information'.

#### Usage

```r
gfollower(id = NA)
```

#### Arguments

- **id**: A character string of a Steem ID without '@'.

#### Value

- character a dataframe of an ID’s followers’ info

#### Examples

```r
# Not run:
gfollower("dapeng")
# End(Not run)
```

### gfollowing

*The id list of an id’s following from steemdb.com. gfollowing means 'get the following information'.*

#### Description

The id list of an id’s following from steemdb.com. gfollowing means 'get the following information'.

#### Usage

```r
gfollowing(id = NA)
```

#### Arguments

- **id**: A character string of a Steem ID without '@'.

#### Value

- A dataframe of an ID’s following info
**Examples**

```r
## Not run:
gid("dapeng")

## End(Not run)
```

---

`gid`  
*Get an ID’s detailed info. gid means ’get an ID’s information’.*

**Description**

Get an ID’s detailed info. gid means ’get an ID’s information’.

**Usage**

```r
gid(id = NA, method = c("appbase_api", "steemdb.com", "steemsql.com", "steemdata.com"),
    sql_con)
```

**Arguments**

- `id`  
  A character string of a Steem ID without '@'.

- `method`  
  A character string of the Steem data server to connect.

- `sql_con`  
  A connection to the SteemSQL server.

**Value**

A list or a dataframe of an ID’s detailed info

**Examples**

```r
## Not run:
gid()

## End(Not run)
```
**sidpostl**

*Obtain an ID’s post hyperlinks. sidpostl means 'get an ID’s post links'.*

**Description**

Obtain an ID’s post hyperlinks. sidpostl means 'get an ID’s post links'.

**Usage**

```r
sidpostl(id = NA, method = c("steemdb.com", "steemsql.com", "steemdata.com"), sql_con,
          post_number = 3, site = "steemit.com")
```

**Arguments**

- **id**: A character string of a Steem ID without '@'.
- **method**: A character string of the Steem data server to connect.
- **sql_con**: A connection to the SteemSQL server.
- **post_number**: A numeric value or NA. The number of the latest posts to be obtained. If NA, the 100 latest posts will be processed.
- **site**: A character string of the site of the steem web UI.

**Value**

A character string vector of an ID’s post hyperlinks.

**Examples**

```r
## Not run:
sidpostl()

## End(Not run)
```

**sidposts**

*Obtain an ID’s post detailed info from steemdb.com. 'sidposts' means 'get an ID’s posts'.*

**Description**

Obtain an ID’s post detailed info from steemdb.com. 'sidposts' means 'get an ID’s posts'.

**Usage**

```r
sidposts(id = NA, method = c("steemdb.com", "steemsql.com", "appbase_api", "steemdata.com"), sql_con, post_number = NA)
```
Arguments

- **id**: A character string of a Steem ID without '@'.
- **method**: A character string of the Steem data server to connect.
- **sql_con**: A connection to the SteemSQL server.
- **post_number**: A numeric value or NA. The number of the latest posts to be obtained. If NA, all the posts will be processed.

Value

A data frame of an ID's post detailed info.

Examples

```r
## Not run:
gidposts()
## End(Not run)
```

---

**gpost**

Get the complete info of a single given post. *gpost* means 'get a post'.

Description

Get the complete info of a single given post. *gpost* means 'get a post'.

Usage

```r
gpost(postlink = NA, method = c("steemdb.com", "steemsql.com"), sql_con, selected = FALSE, newline = FALSE, oldcolname)
```

Arguments

- **postlink**: A character of the link to a post
- **method**: A character string of the Steem data server to connect.
- **sql_con**: A connection to the SteemSQL server.
- **selected**: A logic value of whether return only selected info
- **newline**: A logic value of whether rbind the returned dataframe with an existing one
- **oldcolname**: A character string. If newline == TRUE, the returned dataframe is ordered according to oldcolname

Value

A dataframe
Examples

```r
## Not run:
gposts("cn/@dapeng/steemit-markdown")

## End(Not run)
```

---

**gposts**

Get the detailed information of given posts. *gposts* means 'get posts'.

Description

Get the detailed information of given posts. *gposts* means 'get posts'.

Usage

```r
gposts(postlinks = NA, method = c("steemdb.com", "steemsql.com", "steemdata.com"), sql_con)
```

Arguments

- `postlinks` A character string of hyperlinks to target posts
- `method` A character string of the Steem data server to connect.
- `sql_con` A connection to the SteemSQL server.

Value

A dataframe of the detailed information of the given posts

Examples

```r
## Not run:
gposts(c("cn/@dapeng/xuer-sale", "utopian-io/@dapeng/steemg-four-more"))

## End(Not run)
```
### gspmv

**Get the value of Steem per Mvest**

#### Description
Get the value of Steem per Mvest

#### Usage

```r
gspmv()
```

#### Value
A numeric value of the Steem per MVest

#### Examples

## Not run:
```r
gspmv()
```

## End(Not run)

### gur

**Get the utopian review data from utopian.rocks api**

#### Description
Get the utopian review data from utopian.rocks api

#### Usage

```r
gur(id = NA)
```

#### Arguments

- `id` the steeian’s ID to query

#### Value

a dataframe

#### Examples

## Not run:
```r
gur(NULL)
```

## End(Not run)
gvotep

*Description*

A post’s vote report based on steemdb.com. `gvotep` means 'get the vote information of a post.'

*Usage*

```r
gvotep(postlink = NA)
```

*Arguments*

- `postlink`: A character string of the link of a post

*Value*

A dataframe of a post’s voter information

*Examples*

```r
## Not run:
gvotep()

## End(Not run)
```

gvoter

*Description*

Get the vote information of given IDs from SteemSQL. `gvoter` means 'get the voter activities.'

*Usage*

```r
gvoter(voters = NA, from = Sys.Date() - 7, to = Sys.Date(), select = "*", sql_con = NA, if_plot = FALSE)
```

*Arguments*

- `voters`: A character vector of given Steem IDs.
- `from`: A character string of the starting date. in 'Y-m-d' format.
- `to`: A character string of the ending date. in 'Y-m-d' format.
- `select`: A character vector of the selected columns in the SteemSQL query.
- `sql_con`: A connection to the SteemSQL server.
- `if_plot`: A logic value of whether plot the daily votes.
idlink

Value
A list (and a diagram) of the voter report.

Examples
```r
## Not run:
gvoter(voters = NA)

## End(Not run)
```

idlink  Convert an id from a character to html hyperlink

Description
Convert an id from a character to html hyperlink

Usage
```
idlink(id = NA)
```

Arguments
```
id  A character string of a Steem ID without '@'.
```

Value
A character string of the hyperlink to the Steem id.

pcner  Plot function for the Shiny app scner

Description
Plot function for the Shiny app scner

Usage
```
pcner(i, sliderplot, whwechat, zh)
```

Arguments
```
i  A numeric indicator
sliderplot  The ID of the sliderplot
whwechat  The data from of the cner
zh  The Chinese dictionary
```
pdate

Plot a time series with x as Date.

Description

Plot a time series with x as Date.

Usage

pdate(x, y, myylab = "", mylegend = "", mycol = "darkgreen", myxlim = NULL, myylim = NULL)

Arguments

x A Date vector.
y A numeric vector.
myylab A character string of the y label.
mylegend A character string of the legend.
mycol A color.
myxlim A Date vector of the date range of x.
myylim A Date vector of the date range of y.

Value

A time series diagram.

phist

Plot a histogram of a vector.

Description

Plot a histogram of a vector.

Usage

phist(data = rnorm(1000), mybreaks = "Sturges", myylim = NULL, myylim = NULL, eightlines = TRUE, eightdigit = 0, eightcex = 0.8, eightcolors = c("red", "darkgreen", "blue", "black", "purple", "gold")[c(1, 2, 3, 2, 1, 6, 6, 5, 4, 5)], mylegend = "", myylab = "", return_df = FALSE, myfreq = FALSE, show_n = TRUE, show_skewness = TRUE, show_density = FALSE, show_normline = FALSE, show_mean = FALSE)
**Arguments**

- **data**: A numeric vector to plot.
- **mybreaks**: Breaks of the plotted bars.
- **myxlim**: x limit.
- **myylim**: y limit.
- **eightlines**: Whether to draw the eight lines.
- **eightdigit**: The digit number of the figures displayed on the diagram.
- **eightcex**: The character size of the figures.
- **eightcolors**: The colors of the eight lines.
- **mylegend**: The legend.
- **myxlab**: The x label.
- **return_df**: Whether return the data frame of summary.
- **myfreq**: Whether display the frequency.
- **show_n**: Whether show the sample number.
- **show_skewness**: Whether show the skewness test.
- **show_density**: Whether show the density.
- **show_normline**: Whether show the line of the normal distribution.
- **show_mean**: Whether show the mean value.

**Value**

A histogram diagram.

---

**phour**

*Hour rose plot. Copied and modified from the openair package. phour() plotted a 24-hour clock, indicating the active hours on the basis of a time column in a data frame. It is highly customized on the basis of the openair::windrose() function.*

---

**Description**

Hour rose plot. Copied and modified from the openair package. phour() plotted a 24-hour clock, indicating the active hours on the basis of a time column in a data frame. It is highly customized on the basis of the openair::windrose() function.

**Usage**

```r
phour(my_df = NA, col_time = "created", ws = "ws", wd = "hour360", ws2 = NA, wd2 = NA, ws.int = 30, angle = 1, type = "default", cols = "default", grid.line = NULL, width = 1, seg = 0.9, auto.text = TRUE, breaks = 4, offset = 10, paddle = FALSE, key.header = NULL, key.footer = "(SBD)", key.position = "right", key = FALSE, dig.lab = 5, statistic = "prop.count", pollutant = NULL, cust_labels = c(0, 6, 12, 18), annotate = FALSE, border = NA, quantile_line = TRUE, ...)
```
Arguments

my_df  
A data frame containing the fields ws and wd.

col_time  
the column name for the time stamp

ws  
A character string of the name of the column representing the radium of the circular sectors in the diagram. It can be SBD payout, votes, comments, etc..

wd  
A character string of the name of the column representing the hour of the day.

ws2  
The user can supply a second set of data with which the first can be compared.

wd2  
see ws2.

ws.int  
A numeric vector of the ws interval. Default is 2.

angle  
The hour spokes. Other potentially useful angle is 3 (hours).

type  
type determines how the data are split i.e. conditioned, and then plotted. It can be 'season', 'year', 'weekday'... The default is will produce a single plot using the entire data. It can also be a numeric or factor vector.

cols  
Colours for plotting. 'default', 'increment', 'heat', 'jet', 'hue' and user defined, such as c("yellow", "green", "blue", "black").

grid.line  
Grid line interval. NULL in default. It can also be a numeric value like 10, or a list like list(value = 10, lty = 5, col = "purple").

width  
The adjustment factor for width of payout intervals. For example, width = 1.5 will make the paddle width 1.5 times wider. For paddle = TRUE.

seg  
The width of the segments. 0.5 will produce segments 0.5 * angle.

auto.text  
A logical value of whether formatting the names and units automatically in the titles and axis labels

breaks  
A numeric vector of the number of break points for payouts. 4 by default, which generates the break points 2, 4, 6, 8 SBD for ws.int default of 2 SBD. It can also be c(0, 1, 10, 100), which breaks the data into segments <1, 1-10, 10-100, >100.

offset  
A numeric value (default 10) of the size of the 'hole' in the middle of the plot, expressed as a percentage of the polar axis scale.

paddle  
A logic value. TRUE means the 'paddle' style spokes, and FALSE means the 'wedge' style spokes.

key.header  
A character string of additional text above the scale key.

key.footer  
A character string of additional text below the scale key.

key.position  
A character string of the location of the scale key. 'top', 'right', 'bottom' and 'left'.

key  
Fine control of the scale key

dig.lab  
A numeric value of the significant digits at which scientific number formatting is used in break point.

statistic  
A character string of the statistic to be applied.

- 'prop.count' (default) sizes bins according to the proportion of the frequency of the records,
- 'prop.mean' sizes bins according to their relative contribution to the mean,
repcalc

**Description**

Calculate the reputation of an ID

**Usage**

repcalc(rep)

**Arguments**

- `rep` A numeric value of the raw reputation.

**Value**

- A numeric value of the real reputation

---

- `'abs.count'` provides the absolute count of records in each bin.

- `pollutant` Alternative data series to be sampled.

- `cust_labels` A numeric vector displayed as the customed labels

- `annotate` A logic value or a character string.
  - TRUE: the percentage calm and mean values are printed in each panel together with a description of the statistic below the plot.
  - " ": only the stastic is below the plot.
  - Custom annotations may be added by setting value to `c("annotation 1", "annotation 2")`.

- `border` A character string of the border colour for shaded areas.

- `quantile_line` whether to display the quantile line

- `...` other parameters

**Value**

A figure with the active hour rose

**Examples**

```r
## Not run:
phour()

## End(Not run)
```

---

```
repcalc
  Calculate the reputation of an ID
```

---

## Not run:
phour()

## End(Not run)
### sscner

**Description**

A shiny app to display and analyze the CNers. *sscner* means shiny app for CNers.

**Usage**

```r
sscner()
```

**Value**

A shiny app which can be displayed in a web browser.

---

### sscner_ui

**Description**

UI for the Shiny app sscner display and analysis.

**Usage**

```r
sscner_ui()
```

**Value**

A UI function

---

### sfollow

**Description**

A shiny app to display and analyze the followers a given ID. *sfollow* means shiny app for followers.

**Usage**

```r
sfollow()
```
**sfollow_server**

**Value**

a shinyapp which can be displayed in a web browser.

**Examples**

```r
t# Not run:
sfollow()

t# End(Not run)
```

---

**sfollow_server**  
*Server for the Shiny app sfollow display and analysis*

**Description**

Server for the Shiny app sfollow display and analysis

**Usage**

`sfollow_server(input, output, session)`

**Arguments**

- `input`  
The input of the server.
- `output`  
The output of the server.
- `session`  
The session of the server.

**Value**

A server function

---

**sfollow_ui**  
*UI for the Shiny app sfollow()*

**Description**

UI for the Shiny app sfollow()

**Usage**

`sfollow_ui()`

**Value**

A UI function
skewness

*Calculate the skewness*

**Description**

Calculate the skewness

**Usage**

skewness(x)

**Arguments**

- **x**: The data for calculation.

**Value**

The skewness

---

sposts

*A shiny app to display and analyze the posts of a given ID. sposts means shiny app for posts.*

**Description**

A shiny app to display and analyze the posts of a given ID. sposts means shiny app for posts.

**Usage**

sposts()

**Value**

A shiny app which can be displayed in a web browser.
**sposts_server**  
*Server for the Shiny app sposts display and analysis*

**Description**

Server for the Shiny app sposts display and analysis

**Usage**

`sposts_server(input, output, session)`

**Arguments**

- **input**: The input of the server.
- **output**: The output of the server.
- **session**: The session of the server.

---

**sposts_ui**  
*UI for the Shiny app sposts display and analysis*

**Description**

UI for the Shiny app sposts display and analysis

**Usage**

`sposts_ui()`

**Value**

A UI function
ssql

*Open Connections to the SteemSQL server. `ssql` means 'send to sql'.*

**Description**

Open Connections to the SteemSQL server. `ssql` means 'send to sql'.

**Usage**

```r
ssql(uid = NA, pwd = NA)
```

**Arguments**

- `uid` A character string of the SteemSQL ID.
- `pwd` A character string of the SteemSQL password.

**Value**

A connection to the SteemSQL server.

**Examples**

```r
## Not run:
ssql()

## End(Not run)
```

tag_of_post

*Retrieve tags from json_str of a post*

**Description**

Retrieve tags from json_str of a post

**Usage**

```r
tag_of_post(post_json_str)
```

**Arguments**

- `post_json_str` json string with tag information

**Value**

- `tags`
unitconvert  

Concert unit from GV, MV, kV to numeric values.

Description
Concert unit from GV, MV, kV to numeric values.

Usage
unitconvert(x)

Arguments
x  
the unit GV, MV, kV to convert

Value
the numeric multiplier.

whale  

Calculate the level of and ID

Description
Calculate the level of and ID

Usage
whale(x)

Arguments
x  
the Raw SP value of an ID

Value
the level of the ID
xatf

Adapt the x axis to the time range

Description
Adapt the x axis to the time range

Usage
xatf(x)

Arguments
x the time range

Value
the optimized label
Index

accountCount, 3
acnsub, 4
adailyf, 4
avotenot, 5
avotep, 5

bblog, 6
bmd, 7

charsplit, 8
clear_title, 9
clearferfing, 8
gaccounts, 9
gblog, 10
gcner, 11
gcomments, 11
gdelegation, 12
getAccount, 13
getAccountVotes, 13
getBlog, 14
getComments, 15
getDelegation, 15
gNodes, 16
getPost, 16
getPostsByTag, 17
getReplies, 18
getSteemProperties, 18
getTransactions, 19
getTrending, 20
ggetWitnesses, 20
gfollow, 21
gfollower, 22
gfollowing, 22
gid, 23
gidpostl, 24
gidposts, 24
gpost, 25
gposts, 26
gspmv, 27
gur, 27
gvotep, 28
gvoter, 28
idlink, 29
pcner, 29
pdate, 30
phist, 30
phour, 31
repcalc, 33
scner, 34
scner_ui, 34
sfollow, 34
sfollow_server, 35
sfollow_ui, 35
skewness, 36
sposts, 36
sposts_server, 37
sposts_ui, 37
ssql, 38
tag_of_post, 38
unitconvert, 39
whale, 39
xatf, 40