Package 'qrcode'

August 29, 2016

Type Package

Title QRcode Generator for R

Version 0.1.1

Date 2015-08-13

Author Victor Teh

Maintainer Victor Teh <victorteh@gmail.com>

Description Create QRcode in R.

License GPL-3

Depends R (>= 3.0.0)

Imports R.utils, stringr, stats, utils

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2015-08-23 23:27:47

R topics documented:

DataStringBinary *Function to convert input data string to binary polynomial*

Description

Convert input data string to binary polynomial

Usage

```
DataStringBinary(dataString, qrInfo)
```

Arguments

dataString	input data string.
qrInfo	dataframe that store all the required info to generate qrcode.

ECgenerator *Error correction code generator Generate error correction code based on the input polynomial.*

Description

Error correction code generator Generate error correction code based on the input polynomial.

Usage

```
ECgenerator(GenPoly, DataPoly, DCWordCount, ECWordCount)
```

Arguments

GenPoly	generated polynomial to calculate error correction code word
DataPoly	input data polynomial
DCWordCount	data code word count
ECWordCount	error code word count

Value

Error code word polynomial

formatPolyGen *Function to calculate and generate format polynomial*

Description

Function to calculate and generate format polynomial

Usage

```
formatPolyGen(formatString, polyString)
```

Arguments

formatString	QRcode format binary string
polyString	polynomial to create ECL for formatString

polynomialGenerator *Function to generate polynomial*

Description

Function to generate polynomial

Usage

```
polynomialGenerator(ECcount)
```

Arguments

ECcount	error correction code word count

Value

polynomail to generate Error correction code

qrCodeSpec *QRcode specifications and requirements.*

Description

List of different versions of QRcode specification and requirements. For more details can refer to QRcode standard.

Usage

```
data(qrCodeSpec)
```

Format

```
'data.frame': 160 obs. of  11 variables:
 $ Version       : int  1 1 1 1 2 2 2 2 3 3 ...
 $ ECL           : Factor w/ 4 levels "H","L","M","Q": 2 3 4 1 2 3 4 1 2 3 ...
 $ Numeric       : int  41 34 27 17 77 63 48 34 127 101 ...
 $ Alphanumeric  : int  25 20 16 10 47 38 29 20 77 61 ...
 $ Byte          : int  17 14 11 7 32 26 20 14 53 42 ...
 $ Dcword        : int  19 16 13 9 34 28 22 16 55 44 ...
 $ ECwordPerBlock: int  7 10 13 17 10 16 22 28 15 26 ...
 $ Grp1          : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DCinGrp1      : int  19 16 13 9 34 28 22 16 55 44 ...
 $ Grp2          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ DCinGrp2      : int  0 0 0 0 0 0 0 0 0 0 ...
```

Details

- Version. QRcode version.
- ECL. Error Correction Level. Consisted of 4 level L,M,Q and H.
- Numeric. Number of numeric character supported by the given version and ECL.
- Alphanumeric. Number of alphabet and numeric character supported by the given version and ECL.
- Byte.Number of byte supported by the given version and ECL.
- Dcword. Data code word count.
- ECwordPerBlock. Error correction word count per block.
- Grp1. Number of block in group 1.
- DCinGrp1. Number of data code word in each group 1.
- Grp2. Number of block in group 2.
- DCinGrp2. Number of data code word in each group 2.

qrcode_gen *QRcode generator*

Description

Create QRcode in R. Capable to generate all variant of QRcode, version 1 to 40 and Error correct level of "L","M","Q" and " H". Not all reader in market can support all QRcode version, qrcode_gen has a software limit to version 10 which is tested working in most reader.

Usage

```
qrcode_gen(dataString, ErrorCorrectionLevel = "L", dataOutput = FALSE,
  plotQRcode = TRUE, wColor = "White", bColor = "black", mask = 1,
  softLimitFlag = TRUE)
```

Arguments

dataString	input string for the QRcode
ErrorCorrectionLevel	Error Correction Level. The available options are "L","M","Q" and " H". Default value as "L"
dataOutput	option to export data as matrix. Default value is FALSE.
plotQRcode	option to plot QRcode. Default value is TRUE.
wColor	color of the white module(white squre) in QRcode. Default value "white".
bColor	color pf the black module(black squre) in QRcode. Default value "black".
mask	mask for QRcode to increase decodability. Available value is 0-7.
softLimitFlag	flag to limit the QRcode version to 10. Default value TRUE.

Value

A matrix that represent the QRcode. 1 as black module and 0 as white module.

Examples

```
qrcode_gen('www.r-project.org')

 #User may change the color of the module
 qrcode_gen('www.r-project.org',bColor='Green3')
```

qrFillUpMatrix *Function to fill up the data bits*

Description

Fill up the predefined QRcode matrix with the input binary string.

Usage

```
qrFillUpMatrix(allBinary, data, version)
```

Arguments

allBinary	all data in binary in character format.
data	matrix data created by qrFillUpMatrix
version	version of the QRcode.

Value

matrix filled up with the data bits

qrInitMatrix *Function to initialize QRcode in matrix for different version*

Description

Create a basic structre of QRcode in matrix format. Each element in QRcode will be marked as different value.

Usage

```
qrInitMatrix(version)
```

Arguments

version	version number of the target QRcode

qrInterleave *Function to interleave the Data Code and Error Correction Core*

Description

Function to interleave the Data Code and Error Correction Core

Usage

```
qrInterleave(poly, dataPoly, qrInfo)
```

Arguments

poly	error correction code word polynomial
dataPoly	input data code word polynomial
qrInfo	dataframe that store all the required info to generate QRcode. Via qrVersionInfo

Value

Interleaved polynomial readied to fill up the QRcode matrix

qrMask *Apply mask to the QRcode matrix*

Description

Apply mask to the QRcode matrix

Usage

```
qrMask(data, qrInfo, mask)
```

Arguments

data	QRcode matrix
qrInfo	dataframe that store all the required info to generate QRcode. Via qrVersionInfo
mask	mask for QRcode to increase decodability. Available value is 0-7.

Details

QRcode stardard specify 8 masks as listed below.

- M0, (row + column)
- M1, (row)
- M2, (column)
- M3, (row + column)
- M4, (row
- M5, ((row * column)
- M6, (((row * column)
- M7, (((row + column)

`qrVersionInfo` *Function to identify the version of the QRcode based on input string*

Description

Function to identify the version of the QRcode based on input string

Usage

```
qrVersionInfo(dataString, ECLevel = "L")
```

Arguments

`dataString`	dataString is the input string
`ECLevel`	Error Correction Level. In QRcode standard, the are 4 levels \"L\",\"M\",\"Q\" and \"H\" which represent 7%, 15%, 20% and 30% data recovery capability.

Value

1 row dataframe that include all required info to generate QRcode.

versionPolyGen *Function to calculate and generate version polynomial*

Description

Function to calculate and generate version polynomial

Usage

```
versionPolyGen(versionString, polyString)
```

Arguments

versionString	version in binary string
polyString	polynomial in binary string, specified in the stardard to calculate version ECL.

Value

version polynomial.

Index