

# Package ‘codecountR’

September 27, 2023

**Title** Counting Codes in a Text and Preparing Data for Analysis

**Version** 0.0.3.1

**Description** Data analysis frequently requires coding, in particular when data are collected by interviews, by observations or even by questionnaires. Therefore, code counting and data preparation are necessary phases to carry out the analysis. Thus, the analysts will wish to count the codes inserted in a text (tokenization and counting of a list of pre-established codes) and to carry out the preparation of the data (feature scaling min-max normalization, Zscore, Box and Cox transformation, Yeo and Johnson transformation, median absolute deviation). For Box and Cox (1964) <<https://www.jstor.org/stable/2984418>> transformation, optimal Lambda is calculated by log-likelihood. The optimal lambda for Yeo and Johnson (2000) <[doi:10.1093/biomet/87.4.954](https://doi.org/10.1093/biomet/87.4.954)> transformation is calculated by correlation coefficient test inspired of Lye (1993) <[doi:10.1139/193-101](https://doi.org/10.1139/193-101)>. Median absolute deviation is calculated on the basis of Leys et al (1993) <[doi:10.1016/j.jesp.2013.03.013](https://doi.org/10.1016/j.jesp.2013.03.013)>. Package for educational purposes.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** stats

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Philippe Cohard [aut, cre]

**Maintainer** Philippe Cohard <[p.cohard@laposte.net](mailto:p.cohard@laposte.net)>

**Repository** CRAN

**Date/Publication** 2023-09-27 08:30:10 UTC

## R topics documented:

analysCodesList . . . . .	2
BoxAndCox . . . . .	3
codeCount . . . . .	3

loadCodes . . . . .	4
medAbsDev . . . . .	4
normMinMax . . . . .	5
piBlom . . . . .	6
subCalcBoxAndCox . . . . .	6
subCalcYeoAndJohnson . . . . .	7
tokenization . . . . .	7
YeoAndJohnson . . . . .	8
zScore . . . . .	9

**Index** **10**

---

analysCodesList	<i>analysCodesList</i>
-----------------	------------------------

---

**Description**

analysCodesList

**Usage**

```
analysCodesList(dataS, codesLis)
```

**Arguments**

dataS	a character
codesLis	a character

**Value**

a list

**Examples**

```
codes=list("@essai@", "@test@")
data = "this is an example @essai@, a bit long @essai@ text"
Result=analysCodesList(data,codes)
Result
```

---

BoxAndCox	<i>BoxAndCox</i>
-----------	------------------

---

**Description**

BoxAndCox

**Usage**

```
BoxAndCox(rawVect, minLambda)
```

**Arguments**

rawVect	a vector
minLambda	a number

**Value**

a list

**Examples**

```
vec=rlnorm(100, log(3), log(3))
BandC=BoxAndCox(vec, -3)
BandC
BAC=unlist(BandC$par)
BAC
rawVectBCFinal=unlist(subCalcBoxAndCox(vec, BandC$par))
```

---

codeCount	<i>codeCount</i>
-----------	------------------

---

**Description**

codeCount

**Usage**

```
codeCount(dataSet, code)
```

**Arguments**

dataSet	a character
code	a character

**Value**

a number

**Examples**

```
data = "this is an example @essai@"
codeCount(data, "@essai@") #number of lines containing the chain
```

---

loadCodes	<i>loadCodes</i>
-----------	------------------

---

**Description**

loadCodes

**Usage**

```
loadCodes(txtFile)
```

**Arguments**

txtFile      a character

**Value**

a list

**Examples**

```
theFile =system.file("codesList.txt", package = "codecountR")
data=loadCodes(theFile)
```

---

medAbsDev	<i>medAbsDev</i>
-----------	------------------

---

**Description**

medAbsDev

**Usage**

```
medAbsDev(nameDframe)
```

**Arguments**

nameDframe      a data.frame

**Value**

a data.frame

**Examples**

```
vec=rnorm(100, mean=10, sd=1)
vec2=rnorm(100, mean=15, sd=1)
df= data.frame(vec,vec2)
colnames(df) = c("vec", "vec2")
MAD=medAbsDev(df)
MAD
```

---

normMinMax

*normMinMax*

---

**Description**

normMinMax

**Usage**

```
normMinMax(nameDframe)
```

**Arguments**

nameDframe      a data.frame

**Value**

a data.frame

**Examples**

```
j=c(10,14,56,30,58,78,99,1)
k=c(10,12,14,16,18,20,22,24)
x=data.frame(j,k)
xMinMax=normMinMax(x)
xMinMax
```

piBlom                      *piBlom*

---

**Description**

piBlom

**Usage**

```
piBlom(nameVect, sSize)
```

**Arguments**

nameVect	a vector
sSize	a number

**Value**

a vector

**Examples**

```
vec=c(1,2,3)  
piBlom(vec, length(vec))
```

---

subCalcBoxAndCox            *subCalcBoxAndCox*

---

**Description**

subCalcBoxAndCox

**Usage**

```
subCalcBoxAndCox(sortedVect, actualLambda)
```

**Arguments**

sortedVect	a vector
actualLambda	a number

**Value**

a vector

**Examples**

```
vec=rlnorm(100, log(3), log(3))  
BandC=subCalcBoxAndCox(vec, -3)
```

---

subCalcYeoAndJohnson    *subCalcYeoAndJohnson*

---

**Description**

subCalcYeoAndJohnson

**Usage**

```
subCalcYeoAndJohnson(Vect, lambda)
```

**Arguments**

Vect	a vector
lambda	a number

**Value**

a vector

**Examples**

```
vec=rlnorm(100, log(3), log(3))  
subCalcYeoAndJohnson(vec, -3)  
#only one lamba value tested
```

---

tokenization    *tokenization*

---

**Description**

tokenization

**Usage**

```
tokenization(txtFile)
```

**Arguments**

txtFile	a character
---------	-------------

**Value**

a list

**Examples**

```
theFile =system.file("ExText.txt", package = "codecountR")
data=tokenization(theFile)
```

---

YeoAndJohnson

*YeoAndJohnson*

---

**Description**

YeoAndJohnson

**Usage**

```
YeoAndJohnson(rawVect, minLambda)
```

**Arguments**

rawVect	a vector
minLambda	a number

**Value**

a list

**Examples**

```
vec=rlnorm(100, log(3), log(3))
YandJ=YeoAndJohnson(vec, -3)
YandJ
YAJ=unlist(YandJ$par)
rawVectYJFinal=unlist(subCalcYeoAndJohnson(vec, YandJ$par))
```



---

zScore	<i>zScore</i>
--------	---------------

---

**Description**

zScore

**Usage**

zScore(namedDframe)

**Arguments**

namedDframe      a data.frame

**Value**

a data.frame

**Examples**

```
j=c(10,14,56,30,58,78,99,1)
k=c(10,12,14,16,18,20,22,24)
x=data.frame(j,k)
xZsc=zScore(x)
xZsc
```

# Index

analysCodesList, [2](#)  
BoxAndCox, [3](#)  
codeCount, [3](#)  
loadCodes, [4](#)  
medAbsDev, [4](#)  
normMinMax, [5](#)  
piBlom, [6](#)  
subCalcBoxAndCox, [6](#)  
subCalcYeoAndJohnson, [7](#)  
tokenization, [7](#)  
YeoAndJohnson, [8](#)  
zScore, [9](#)