

# Package ‘BetaBit’

May 6, 2026

**Title** Mini Games from Adventures of Beta and Bit

**Version** 2.2

**Description** Three games: proton, frequon and regression. Each one is a console-based data-crunching game for younger and older data scientists.

Act as a data-hacker and find Slawomir Pietraszko's credentials to the Proton server.

In proton you have to solve four data-based puzzles to find the login and password.

There are many ways to solve these puzzles. You may use loops, data filtering, ordering, aggregation or other tools.

Only basics knowledge of R is required to play the game, yet the more functions you know, the more approaches you can try.

In frequon you will help to perform statistical cryptanalytic attack on a corpus of ciphered messages.

This time seven sub-tasks are pushing the bar much higher. Do you accept the challenge?

In regression you will test your modeling skills in a series of eight sub-tasks.

Try only if ANOVA is your close friend.

It's a part of Beta and Bit project.

You will find more about the Beta and Bit project at <https://github.com/BetaAndBit/Charts>.

**Depends** R (>= 4.0.0)

**Imports** digest

**Suggests** XML, sp, raster, DALEX, ggplot2, ggthemes

**License** GPL-2

**LazyData** true

**Encoding** UTF-8

**URL** <https://github.com/BetaAndBit/Charts>

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Przemyslaw Biecek [aut, cre],  
Witold Chodor [trl],  
Katarzyna Fak [aut],  
Tomasz Zoltak [aut],  
Foundation SmarterPoland.pl [cph]

**Maintainer** Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-08-24 20:40:06 UTC

## Contents

bash_history . . . . .	2
dataFSW . . . . .	3
employees . . . . .	3
EnglishLetterFrequency . . . . .	4
food . . . . .	4
frequon . . . . .	5
FSW . . . . .	6
guns . . . . .	7
logs . . . . .	7
proton . . . . .	7
read_gpx . . . . .	8
regression . . . . .	9
top100passwords . . . . .	10
top100commonWords . . . . .	10
varLabels . . . . .	11
wikiquotes . . . . .	11
<b>Index</b>	<b>12</b>

---

bash_history	<i>The history of recently executed commands.</i>
--------------	---

---

## Description

The character vector of recently executed commands. Each element of the vector consists of command's name and command's arguments separated with a space.

## Usage

```
data(bash_history)
```

## Format

a character vector with 19913 elements.

---

dataFSW

*The data from the study of Polish upper-secondary schools students.*

---

### Description

It was conducted firstly in the same time as PISA 2009 study, with use of the same cognitive tests and questionnaires as in PISA 2009, but on a different group of students: first grade students of upper-secondary schools (in Poland most of the students in a regular PISA sample attends lower-secondary schools). The students who participated in the first wave of the study were followed in the 2nd grade of upper-secondary school within the research program *Our further study and work (Nasza Dalsza Nauka i Praca)*. Both studies were conducted by the Institute of Philosophy and Sociology Polish Academy of Sciences.

### Format

data frame: 3796 obs. of 54 variables

### Details

**The original data was changed a little, to better fit the purpose of the game.**

---

employees

*The database with employees of Faculty of Electronics and Information Technology of Warsaw University of Technology.*

---

### Description

The dataset describing names, surnames and faculty employees' logins. Note that it is an artificial dataset that imitates real database. The subsequent columns in this dataset describe:

- name. The name of an employee.
- surname. The surname of an employee.
- login. The login of an employee on the Proton server.

### Format

a data frame with 541 rows and three columns.

---

EnglishLetterFrequency

*The vector of letter frequencies in English.*

---

### Description

The vector with frequencies of 26 English letters. It is sorted by the frequency of usage. May be used to refine the transliteration.

### Usage

```
data(EnglishLetterFrequency)
```

### Format

a named vector with 26 elements.

---

food

*The food data from Rijksinstituut voor Volksgezondheid en Milieu*

---

### Description

The Dutch institute Rijksinstituut voor Volksgezondheid en Milieu has compiled a database with ingredient information on more than 2,200 food products. The data in this package is only a processed fraction of the huge and very interesting NEVO database available at <https://www.rivm.nl/documenten/nevo-online-versie>.

### Format

data frame: 2207 obs. of 9 variables

### Details

The preprocessed data can be used to reproduce the charts from the book Wykresy od kuchni (Chart runners) <https://github.com/BetaAndBit/Charts>

Note that data frames food, food\_max, food\_mini, food\_all have product names in English, while food\_pl, food\_max\_pl, food\_mini\_pl, food\_all\_pl have product names in Polish.

**Examples**

```

library("ggplot2")
head(food)

library("ggthemes")
ggplot(data = food, aes(x = Energy)) +
  geom_histogram(color = "white") +
  facet_wrap(~Group) +
  labs(title = "Energy value of the products", subtitle = "per 100 g",
       x = "Energy value", y = "Number") +
  theme_economist()

ggplot(data = food_mini, aes(x = Energy)) +
  geom_histogram(color = "white") +
  facet_wrap(~Group) +
  labs(title = "Energy value of the products", subtitle = "per 100 g",
       x = "Energy value", y = "Number") +
  theme_economist()

ggplot(data = food, aes(x = Protein, y = Fats,
                       color = Group, size = Energy)) +
  geom_point() +
  scale_color_brewer(type = "qual", palette = "Dark2") +
  labs(title = "Share of protein and fats", subtitle = "per 100 g",
       y = "Fats [g]", x = "Protein [g]") +
  theme_gdocs()

ggplot(data = food, aes(x = Group, y = Energy)) +
  geom_rug(sides = "l") +
  geom_violin(scale = "width", aes(fill = Group)) +
  geom_text(data = food_max, aes(label = Name),
           hjust = 0, vjust = 0, color = "blue4") +
  geom_boxplot(width = 0.2, coef = 100) +
  coord_flip() +
  labs(title = "Energy value distribution", subtitle = "per 100 g") +
  theme_gdocs() + theme(legend.position = "none")

```

---

frequon

*The Frequon (Frequency Analysis) Game*


---

**Description**

The frequon function is used for solving problems in the data-based game „The Frequon Game“.

**Usage**

```

frequon(...)

```

**Arguments**

... frequon function is called by different arguments, which vary depending on a problem that Bit is trying to solve. See `Details` in order to learn more about the list of possible arguments.

**Details**

Every time when some additional hints are needed one should add `hint=TRUE` argument to the `frequon` function.

In this game you are in contact with a group of people that are going to stop terrorists. You can communicate with them through `frequon` function.

In each call add `subject` parameter that will indicate which message you are answering. Add `content` parameter. It's value should match the request.

„The Frequon Game” is a free of charge, educational project of the SmarterPoland.pl Foundation.

**Author(s)**

- Katarzyna Fak - the idea and the implementation,
- Przemyslaw Biecek - comments and the integration with the ‘BetaBit’ package.

**Examples**

```
frequon()
frequon(hint=TRUE)
```

---

 FSW

*The data from the study of Polish upper-secondary schools students.*

---

**Description**

It was conducted firstly in the same time as PISA 2009 study, with use of the same cognitive tests and questionnaires as in PISA 2009, but on a different group of students: first grade students of upper-secondary schools (in Poland most of the students in a regular PISA sample attends lower-secondary schools). The students who participated in the first wave of the study were followed in the 2nd grade of upper-secondary school within the research program *Our further study and work (Nasza Dalsza Nauka i Praca)*. Both studies were conducted by the Institute of Philosophy and Sociology Polish Academy of Sciences.

**Format**

data frame: 3796 obs. of 54 variables

**Details**

**The original data was changed a little, to better fit the purpose of the game.**

---

guns	<i>The three messages to be decoded.</i>
------	--

---

**Description**

The messages to be decoded in the game ‘freuon()’. How to access it? You have to figure out this by yourself.

---

logs	<i>The history of logs into the Proton server</i>
------	---

---

**Description**

The dataset describing the history of logs: who, from where and when logged into the Proton server. The subsequent columns in this dataset describe:

- login. The login of the user which logs into the Proton server.
- host. The IP address of the computer, from which the log into the Proton server was detected.
- date. The date of log into the Proton server. Rows are sorted by this column.

**Usage**

```
data(logs)
```

**Format**

a data frame with 59366 rows and 3 columns.

---

proton	<i>The Proton Game</i>
--------	------------------------

---

**Description**

The proton function is used for solving problems in the data-based game „The Proton Game”. Solve four data-based puzzles in order to crack into Pietraszko’s account!

**Usage**

```
proton(...)
```

**Arguments**

... proton function is called by different arguments, which vary depending on a problem that Bit is trying to solve. See `Details` in order to learn more about the list of possible arguments.

**Details**

Every time when some additional hints are needed one should add `hint=TRUE` argument to the `proton` function.

In order to get more information about a user on the Proton server one should pass `action = "login"`, `login="XYZ"` arguments to the `proton` function.

In order to log into the Proton server one should pass `action = "login"`, `login="XYZ"`, `password="ABC"` arguments to the `proton` function. If the password matches login, then one will receive a message about successful login.

In order to log into a server different from Proton one should pass `action = "server"`, `host="XYZ"` arguments to the `proton` function.

„The Proton Game” is a free of charge, educational project of the SmarterPoland.pl Foundation.

**Author(s)**

Przemyslaw Biecek, <przemyslaw.biecek@gmail.com>, SmarterPoland.pl Foundation.

**Examples**

```
proton()
proton(hint=TRUE)
```

---

read\_gpx

*Read GPX File*

---

**Description**

Reads data in GPX form. Examples of tourist routes saved in this format can be downloaded from [mapa-turystyczna.pl](http://mapa-turystyczna.pl).

**Usage**

```
read_gpx(path, name = NULL, uniform = TRUE, dx = 25, span = 0.1)
```

```
## S3 method for class 'gpx_file'
plot(x, ..., type = "profile", color = "magenta")
```

**Arguments**

<code>path</code>	path the the gpx file with information about the route
<code>name</code>	name of the route
<code>uniform</code>	if TRUE then route will be converted into a uniform grid of points
<code>dx</code>	if uniform is TRUE then dx is the grid size
<code>span</code>	if uniform is TRUE then span is smoothing parameter
<code>x</code>	routs to be plotted

...	other parameters
type	what should be plotted? 'profile' for profiles, 'difference' for derivative, 'box-plot' absolute derivative
color	names of colors for lines

**Author(s)**

Przemyslaw Biecek

---

regression                      *The Regression Game*


---

**Description**

The regression function is used for solving problems in the data-based game „The regression Game”.

**Usage**

```
regression(...)
```

**Arguments**

... regression function is called with different arguments, which vary depending on a problem that Beta and Bit are trying to solve. See `Details` in order to learn more about the list of possible arguments.

**Details**

Every time when some additional hints are needed one should add `hint = TRUE` or `techHint = TRUE` argument to the `regression` function. Technical hints will point out R packages and/or functions which might help you to solve the task while "normal" hints provide you with methodological advices.

In this game you are helping Professor Pearson. You can communicate with him through the `regression` function.

In each call include the `subject` parameter (indicating which task you are trying to answer) and the `content` parameter (providing information Professor Pearson is asking you for in a given task).

Data used in the game comes from the study of Polish upper-secondary schools first grade students. It was conducted together with the PISA 2009 study using the same cognitive tests and questionnaires as in PISA 2009 but on a different group of students (in Poland most of the students in a PISA sample attends lower-secondary schools). The students who participated in the first wave of the study were followed in the 2nd grade of upper-secondary school within the research program *Our further study and work (Nasza Dalsza Nauka i Praca)*. Both studies were conducted by the Institute of Philosophy and Sociology Polish Academy of Sciences. **The original data was changed a little, to better fit the purpose of the game.**

„The Regression Game” is a free of charge, educational project of the SmarterPoland.pl Foundation.

**Value**

Function returns one of three possible values:

- TRUE if you provided correct answer to a task,
- FALSE if you provided wrong answer to a task,
- NULL if function can't identify task you wanted to answer.

**Author(s)**

- Tomasz Zoltak - the idea and the implementation,
- Mateusz Zoltak - comments, contribution to hints,
- Zuzanna Brzozowska - proofreading,
- Przemyslaw Biecek - comments and the integration with the 'BetaBit' package.

**Examples**

```
regression()  
regression(hint = TRUE)  
regression(techHint = TRUE)
```

---

top1000passwords      *The vector of 1000 most popular passwords.*

---

**Description**

The character vector of 1000 most commonly used passwords. It is sorted by the frequency of password's usage. First passwords in the vector are the most frequently used.

**Format**

a character vector with 1000 elements.

---

top100commonWords      *The vector of 100 most common words in English.*

---

**Description**

The character vector of 100 most commonly used words in English. It is sorted by the frequency of usage. May be used to refine the transliterator.

**Usage**

```
data(top100commonWords)
```

**Format**

a character vector with 100 elements.

---

varLabels	<i>The data frame containng labels of the variables from dataDNiP and DNiP datasets.</i>
-----------	--

---

**Description**

The data frame containng labels of the variables from dataDNiP and DNiP datasets.

**Format**

data frame: 54 obs. of 2 variables

---

wikiquotes	<i>List with quotes in 18 languages.</i>
------------	--

---

**Description**

The named list with 18 languages. Based on <<https://wikiquote.org/>>.

**Usage**

```
data(wikiquotes)
```

**Format**

a named list with 18 elements.

# Index

## \* datasets

- bash\_history, 2
- dataFSW, 3
- employees, 3
- EnglishLetterFrequency, 4
- food, 4
  - food\_all (food), 4
  - food\_all\_pl (food), 4
  - food\_max (food), 4
  - food\_max\_pl (food), 4
  - food\_mini (food), 4
  - food\_mini\_pl (food), 4
  - food\_pl (food), 4
- FSW, 6
- guns, 7
  - and (guns), 7
  - lyo (guns), 7
  - pcs (guns), 7
  - pistoale (guns), 7
  - plot.gpx\_file (read\_gpx), 8
  - proton, 7
- logs, 7
- top1000passwords, 10
- top100commonWords, 10
- varLabels, 11
- wikiquotes, 11
- read\_gpx, 8
- regression, 9
- roses (guns), 7
- top1000passwords, 10
- top100commonWords, 10
- varLabels, 11
- wikiquotes, 11