

## Disclaimer

I'm not a native speaker but I'm doing my best.

If I say something that may be interpreted as either negative or positive (e.g. a joke vs. an offense) I always, always, mean the positive.



# Express introduction to evExpress SAS package

大阪 SAS 勉強会 - 第 12 回 - talk 15



Bartosz Jabłoński  
(バートシュ ヤブウォンスキー)  
✉ [yabwon@gmail.com](mailto:yabwon@gmail.com)  
[linkedin.com/in/yabwon](https://linkedin.com/in/yabwon)

2026 年 1 月 28 日  
Osaka, Japan



Bart Jabłoński, PhD  
**yabwon** (2022)

Warsaw  
University  
of Technology

SASing since 2009



**Experis**  
Manpower Group  
[www.experis.com](http://www.experis.com)



**Takeda**  
Pharmaceutical  
[www.takeda.com](http://www.takeda.com)



Polish SAS  
Users Group  
[www.polsug.com](http://www.polsug.com)



## Problem Statement

SAS expressions are provided in the form of text strings, say:

"A and B or C",

"A \* B + C", or

"42 = A + B + C",

where A, B , and C are variables from some data set HAVE.

Evaluate those expressions using vales from HAVE.



## Problem Statement - Example

code	A	B
A + B	17	42
A = B	100	101
A or B	1	0



## Problem Statement - Example

code	A	B	result
A + B	17	42	
A = B	100	101	
A or B	1	0	

```
result = EVALUATE(code);
```



## Problem Statement - Example

code	A	B	result
A + B	17	42	59
A = B	100	101	
A or B	1	0	

```
result = EVALUATE(code);
```



## Problem Statement - Example

code	A	B	result
A + B	17	42	59
A = B	100	101	0
A or B	1	0	

```
result = EVALUATE(code);
```



## Problem Statement - Example

code	A	B	result
A + B	17	42	59
A = B	100	101	0
A or B	1	0	1

```
result = EVALUATE(code);
```



## Problem Statement - Example

code

A + B

A = B

A or B

Pharmaceuticals Industry Applications 263

**Parsing a Compound Boolean Condition to Evaluate Its Clauses  
Individually and in Logical Groups**

*Daniel M. Himmel, Merck & Co., Inc.*

E(code);



## Problem Statement - Example

code

A + B

A = B

A or B

Paper SM12

**Using the full power of SAS to resolve Algorithmic MedDRA SMQs**

Steve Prust, elderbrook solutions GmbH, Biberach an der Riß, Germany

**ABSTRACT**

This paper presents a method to resolve MedDRA algorithmic SMQs. The method involves using a user-written SAS function that allows data operations that might normally involve several DATA steps or PROCs to be performed in a single macro call (and within a DATA step) as well as hash tables and a little-used SAS MACRO interface. The method has also been extended to include weighted SMQs and flexible contemporaneous requirements and is a comprehensive solution to current algorithmic SMQ definitions. The resultant code demonstrates how diverse elements of the SAS language can be combined to produce powerful solutions.

E(code);



## Problem Statement - Example

[Yet another reason to love hash tables](#)  
Bart Jablonski <syabwon@GMAIL.COM>  
Thu, 4 Jul 2019 16:47:41 -0400  
[text/plain](#) (138 lines)

Hi SAS-Lers,

I had this cool discussion at the office today and the result was very interesting (at lest for me) code which I like to share.

Idea was as old as programming: you have a dataset which contain both data and selection conditions (i.e. if a condition is satisfied than output an observation). I know that keeping both data and logic in the same place is not very "good practice" but that was the setup.



Is there a SAS package  
for this job?

Of course there is!

The evExpress  
package..





*the way to share*

`github.com/yabwon/SAS_PACKAGES`

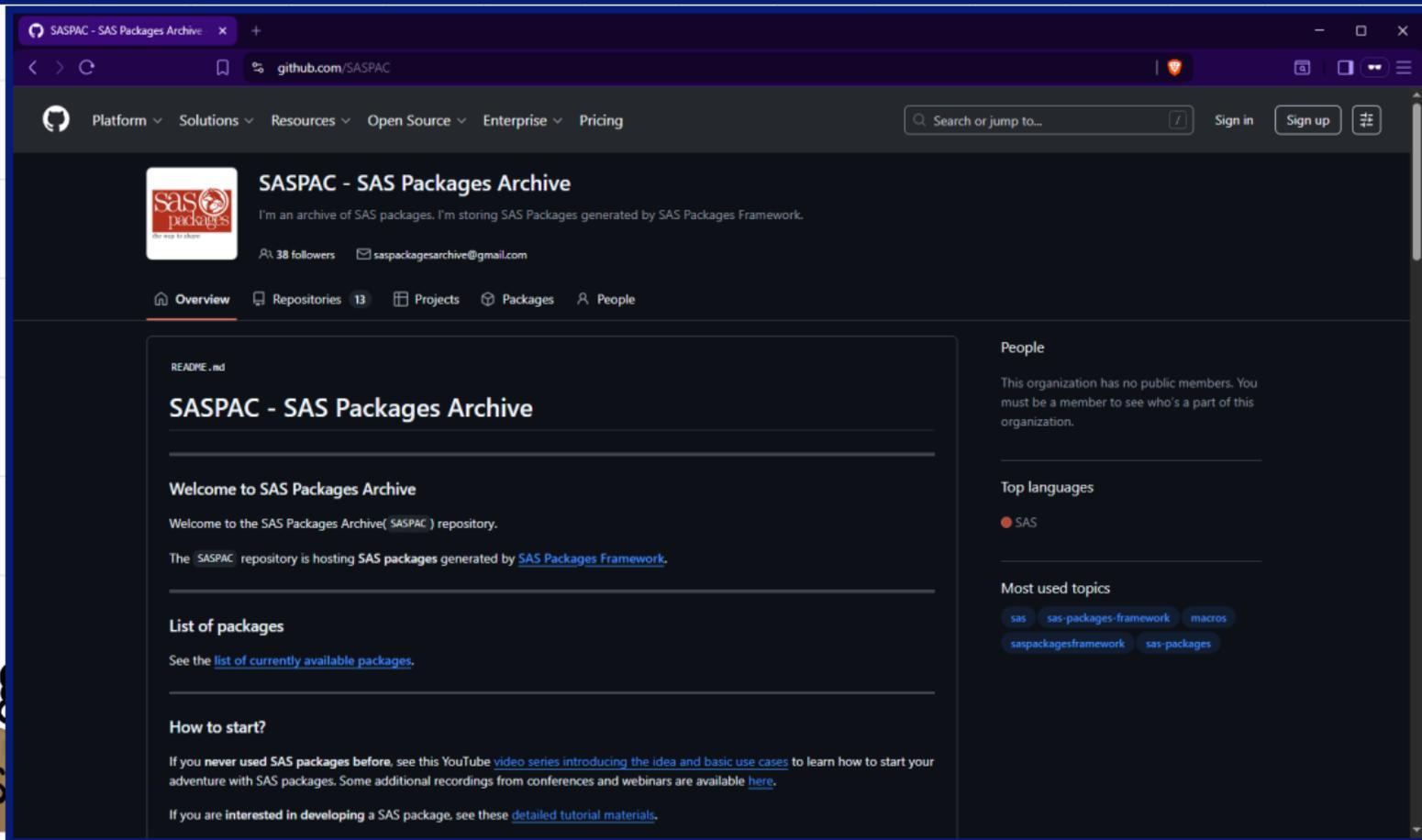




*the way to share*

[github.com/SASPAC/evexpress](https://github.com/SASPAC/evexpress)





The screenshot shows a web browser window displaying the GitHub repository page for "SASPAC - SAS Packages Archive". The browser's address bar shows "github.com/SASPAC". The repository page header includes navigation links for Platform, Solutions, Resources, Open Source, Enterprise, and Pricing, along with a search bar and "Sign in" / "Sign up" buttons. The repository profile shows a logo, the name "SASPAC - SAS Packages Archive", a description "I'm an archive of SAS packages. I'm storing SAS Packages generated by SAS Packages Framework.", 38 followers, and an email address "saspackagesarchive@gmail.com". Below the profile are tabs for Overview, Repositories (13), Projects, Packages, and People. The main content area displays the "README.md" file with the following text:

**SASPAC - SAS Packages Archive**

---

**Welcome to SAS Packages Archive**

Welcome to the SAS Packages Archive( `SASPAC` ) repository.

The `SASPAC` repository is hosting **SAS packages** generated by [SAS Packages Framework](#).

---

**List of packages**

See the [list of currently available packages](#).

---

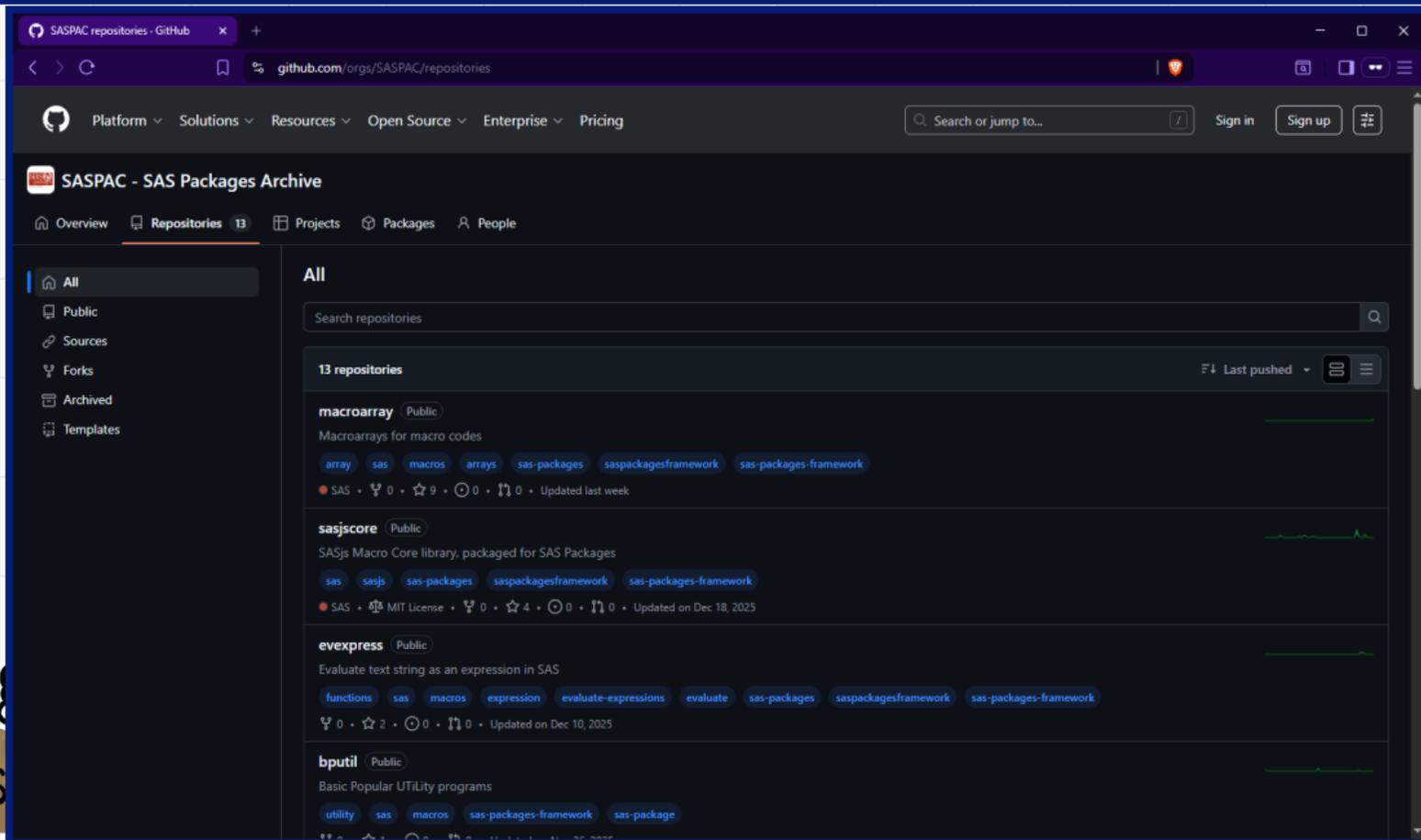
**How to start?**

If you **never used SAS packages before**, see this YouTube [video series introducing the idea and basic use cases](#) to learn how to start your adventure with SAS packages. Some additional recordings from conferences and webinars are available [here](#).

If you are interested in **developing a SAS package**, see these [detailed tutorial materials](#).

On the right side of the repository page, there are sections for "People" (stating no public members) and "Top languages" (showing SAS as the top language). Below that is a "Most used topics" section with tags for "sas", "sas-packages-framework", "macros", "saspackagesframework", and "sas-packages".





The screenshot shows the GitHub interface for the SASPAC repository. The browser address bar displays `github.com/orgs/SASPAC/repositories`. The page title is "SASPAC - SAS Packages Archive". The navigation menu includes "Platform", "Solutions", "Resources", "Open Source", "Enterprise", and "Pricing". A search bar is present with the text "Search or jump to...". The repository list is filtered to "All" and shows 13 repositories. The first repository is "macroarray", described as "Macroarrays for macro codes", with tags for "array", "sas", "macros", "arrays", "sas-packages", "saspackagesframework", and "sas-packages-framework". It has 9 stars and was updated last week. The second is "sasjscore", described as "SASjs Macro Core library, packaged for SAS Packages", with tags for "sas", "sasjs", "sas-packages", "saspackagesframework", and "sas-packages-framework". It has 4 stars and was updated on Dec 18, 2025. The third is "evexpress", described as "Evaluate text string as an expression in SAS", with tags for "functions", "sas", "macros", "expression", "evaluate-expressions", "evaluate", "sas-packages", "saspackagesframework", and "sas-packages-framework". It has 2 stars and was updated on Dec 10, 2025. The fourth is "bputil", described as "Basic Popular UTILITY programs", with tags for "utility", "sas", "macros", "sas-packages-framework", and "sas-package".

SASPAC repositories - GitHub

github.com/orgs/SASPAC/repositories

Platform Solutions Resources Open Source Enterprise Pricing

SASPAC - SAS Packages Archive

Overview Repositories 13 Projects Packages People

All

Search repositories

13 repositories Last pushed

**macroarray** Public

Macroarrays for macro codes

array sas macros arrays sas-packages saspackagesframework sas-packages-framework

SAS 0 9 0 0 0 Updated last week

**sasjscore** Public

SASjs Macro Core library, packaged for SAS Packages

sas sasjs sas-packages saspackagesframework sas-packages-framework

SAS MIT License 0 4 0 0 0 Updated on Dec 18, 2025

**evexpress** Public

Evaluate text string as an expression in SAS

functions sas macros expression evaluate-expressions evaluate sas-packages saspackagesframework sas-packages-framework

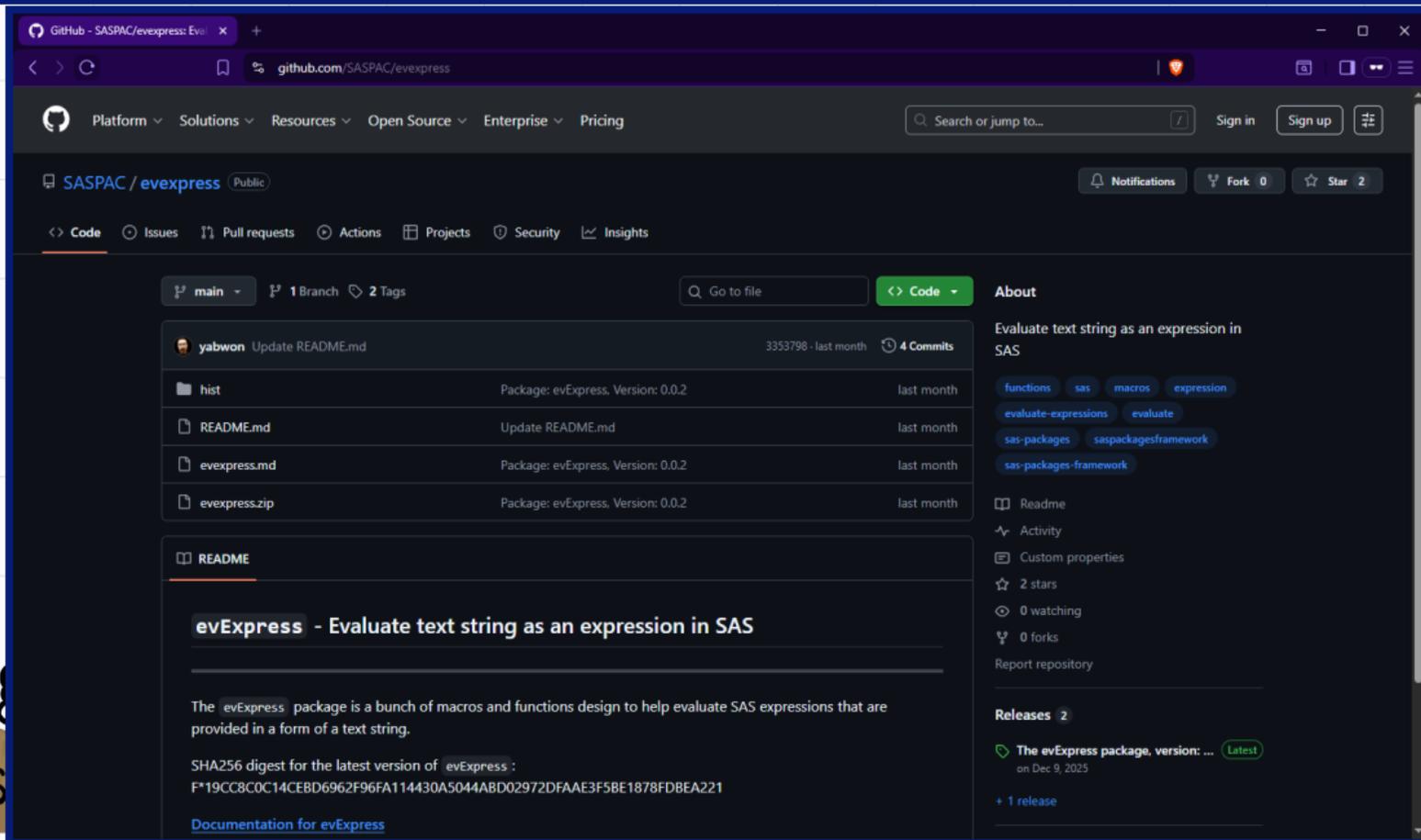
0 2 0 0 0 Updated on Dec 10, 2025

**bputil** Public

Basic Popular UTILITY programs

utility sas macros sas-packages-framework sas-package





GitHub - SASPAC/evexpress: Evaluate text string as an expression in SAS

Platform Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

SASPAC / evexpress Public

Notifications Fork 0 Star 2

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 2 Tags

Go to file Code

**yabwon** Update README.md 3353798 · last month 4 Commits

hist	Package: evExpress, Version: 0.0.2	last month
README.md	Update README.md	last month
evexpress.md	Package: evExpress, Version: 0.0.2	last month
evexpress.zip	Package: evExpress, Version: 0.0.2	last month

README

## evExpress - Evaluate text string as an expression in SAS

The `evExpress` package is a bunch of macros and functions design to help evaluate SAS expressions that are provided in a form of a text string.

SHA256 digest for the latest version of `evExpress` :

```
F*19CC8C0C14CEBD6962F96FA114430A5044ABD02972DFAAE3F5BE1878FDBEA221
```

[Documentation for evExpress](#)

About

Evaluate text string as an expression in SAS

functions sas macros expression

evaluate-expressions evaluate

sas-packages saspackagesframework

sas-packages-framework

Readme

Activity

Custom properties

2 stars

0 watching

0 forks

Report repository

Releases 2

The evExpress package, version: ... Latest

on Dec 9, 2025

+ 1 release

## The Code

```
%loadPackage(evExpress)
```



## The Code

```
%loadPackage(evExpress)
```

code	X	Y	C
(x > 1) and c="A"	3	11	A
x < 1 and (9 < y < 13)	-3	17	B
x < 1 or (81 < y*y)	3	11	C
(3 - sqrt(x+y))	5	11	D
12 + sum(of x--y)	3	9	E
sin(x)**2 + cos(x)**2	-3	11	F



## The Code

```
%loadPackage(evExpress)
```

code	X	Y	C
(x > 1) and c="A"	3	11	A
x < 1 and (9 < y < 13)	-3	17	B
x < 1 or (81 < y*y)	3	11	C
(3 - sqrt(x+y))	5	11	D
12 + sum(of x--y)	3	9	E
sin(x)**2 + cos(x)**2	-3	11	F

```
%evExpressDS(have,exp=code,want=work.want)
```



## The Code

```
%loadPackage(evExpress)
```

```
code
(x > 1) and c="A"
x < 1 and (9 < y < 13)
x < 1 or (81 < y*y)
(3 - sqrt(x+y))
12 + sum(of x--y)
sin(x)**2 + cos(x)**2
```

```
%evExpressDS(have, exp=code)
```

Data set: WORK.WANT

Obs	code	x	y	c	errorCheck	result
1	(x > 1) and c="A"	3	11	A	0	1
2	x < 1 and (9 < y < 13)	-3	17	B	0	0
3	x < 1 or (81 < y*y)	3	11	C	0	1
4	(3 - sqrt(x+y))	5	11	D	0	-1
5	12 + sum(of x--y)	3	9	E	0	24
6	sin(x)**2 + cos(x)**2	-3	11	F	0	1



## The Code

```
%loadPackage(evExpress)
```

code	X	Y	C
(x > 1) and c="A"	3	11	A
x < 1 and (9 < y < 13)	-3	17	B
x < 1 or (81 < y*y)	3	11	C
(3 - sqrt(x+y))	5	11	D
12 + sum(of x--y)	3	9	E
sin(x)**2 + cos(x)**2	-3	11	F

```
data want;  
  set have  
    curobs=curobs  
    indcname=indcname;  
  
  value = evExpress(code, indcname, curobs);  
run;
```



## The Code

```
%loadPackage(evExpress)
```

```
code
(x > 1) and c="A"
x < 1 and (9 < y < 13)
x < 1 or (81 < y*y)
(3 - sqrt(x+y))
12 + sum(of x--y)
sin(x)**2 + cos(x)**2
```

```
data want;
set have
curobs=curobs
indsname=indsname;
```

```
value = evExpress(code, indsname, curobs);
run;
```

Data set: WORK.WANT

Obs	code	x	y	c	value
1	(x > 1) and c="A"	3	11	A	1
2	x < 1 and (9 < y < 13)	-3	17	B	0
3	x < 1 or (81 < y*y)	3	11	C	1
4	(3 - sqrt(x+y))	5	11	D	-1
5	12 + sum(of x--y)	3	9	E	24
6	sin(x)**2 + cos(x)**2	-3	11	F	1



## The Code

```
%loadPackage(evExpress)
```

```
code
```

```
1 data want2;  
2   set have2  
3   curobs=curobs  
4   indcname=indcname;  
5  
6   value = evExpress(code, indcname, curobs);  
7 run;
```

```
NOTE: There were 6000 observations read from the data set WORK.HAVE2.
```

```
NOTE: The data set WORK.WANT2 has 6000 observations and 5 variables.
```

```
NOTE: DATA statement used (Total process time):
```

real time	18.85 seconds
user cpu time	8.73 seconds
system cpu time	9.48 seconds
memory	2925.09k
OS Memory	26876.00k

```
value = evExpress(code, indcname, curobs);  
run;
```



- Bartosz Jabłoński, *"SAS packages - the way to share (a how to) - extended"*, SAS GF 2020 Proceedings (updated version as of November 2025), [https://github.com/yabwon/SAS\\_PACKAGES/tree/master/SPF/Documentation](https://github.com/yabwon/SAS_PACKAGES/tree/master/SPF/Documentation)
- Bartosz Jabłoński, *evExpress packages documentation*, (as of January 2026), <https://github.com/SASPAC/evexpress>
- Rick Langston, *"Submitting SAS Code On The Side"*, SAS Global Forum 2013 Proceedings, 032-2013, <https://support.sas.com/resources/papers/proceedings13/032-2013.pdf>
- Quentin McMullen, *"A Close Look at How DOSUBL Handles Macro Variable Scope"*, SAS Global Forum 2020 Proceedings, 4958-2020, <https://support.sas.com/resources/papers/proceedings13/032-2013.pdf>
- Steve Prust, *"Using the full power of SAS to resolve Algorithmic MedDRA SMQs"*, PHUSE EU 2025 Proceedings, SM12, [https://www.lexjansen.com/phuse/2025/SM/PAP\\_SM12.pdf](https://www.lexjansen.com/phuse/2025/SM/PAP_SM12.pdf)
- Daniel M. Himmel, *"Parsing a Compound Boolean Condition to Evaluate Its Clauses Individually and in Logical Groups"*, NESUG 1990 Proceedings, <https://www.lexjansen.com/nesug/nesug90/NESUG90042.pdf>
- Bartosz Jabłoński et al., *"Yet another reason to love hash tables"*, SAS-L discussion group archive, 2019-07-04, <https://listserv.uga.edu/scripts/wa-UGA.exe?A2=ind1907A&L=SAS-L&P=R18571>



ありがとうございます

dziękuję bardzo

