Crafting Crossword Puzzles with SAS Arrays

Breaking Norms and Making Code Fun Again

Mayur Jadhav

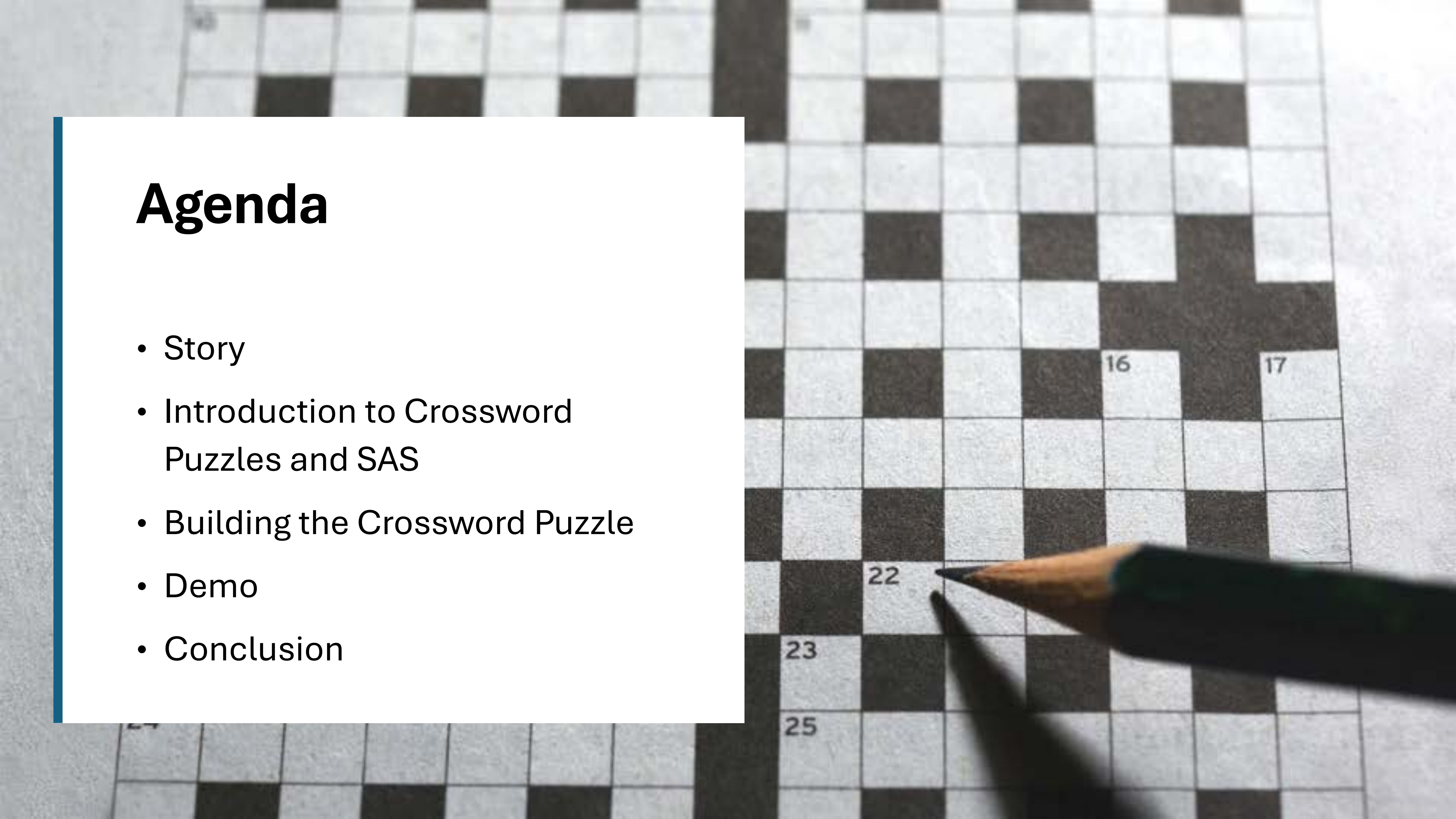
Posten Bring AS

Senior Advisor, BI Developer

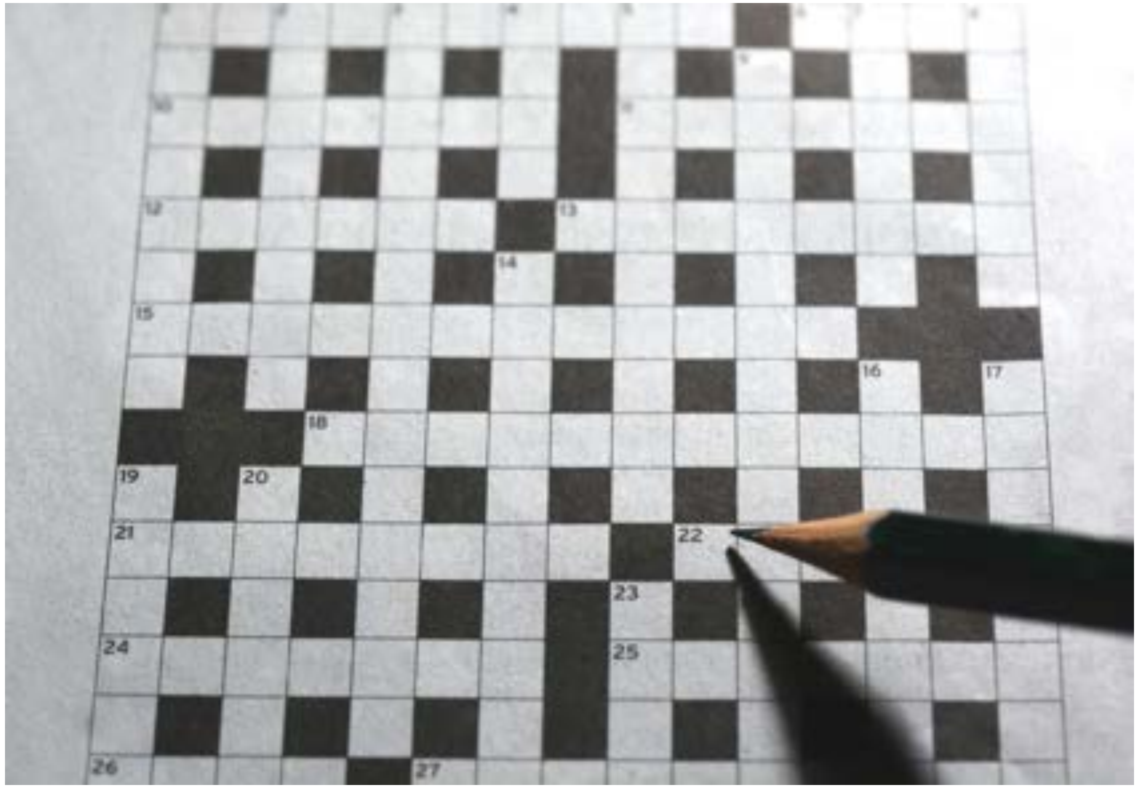
Executive MBA 2025

Agenda

- Story
- Introduction to Crossword Puzzles and SAS
- Building the Crossword Puzzle
- Demo
- Conclusion

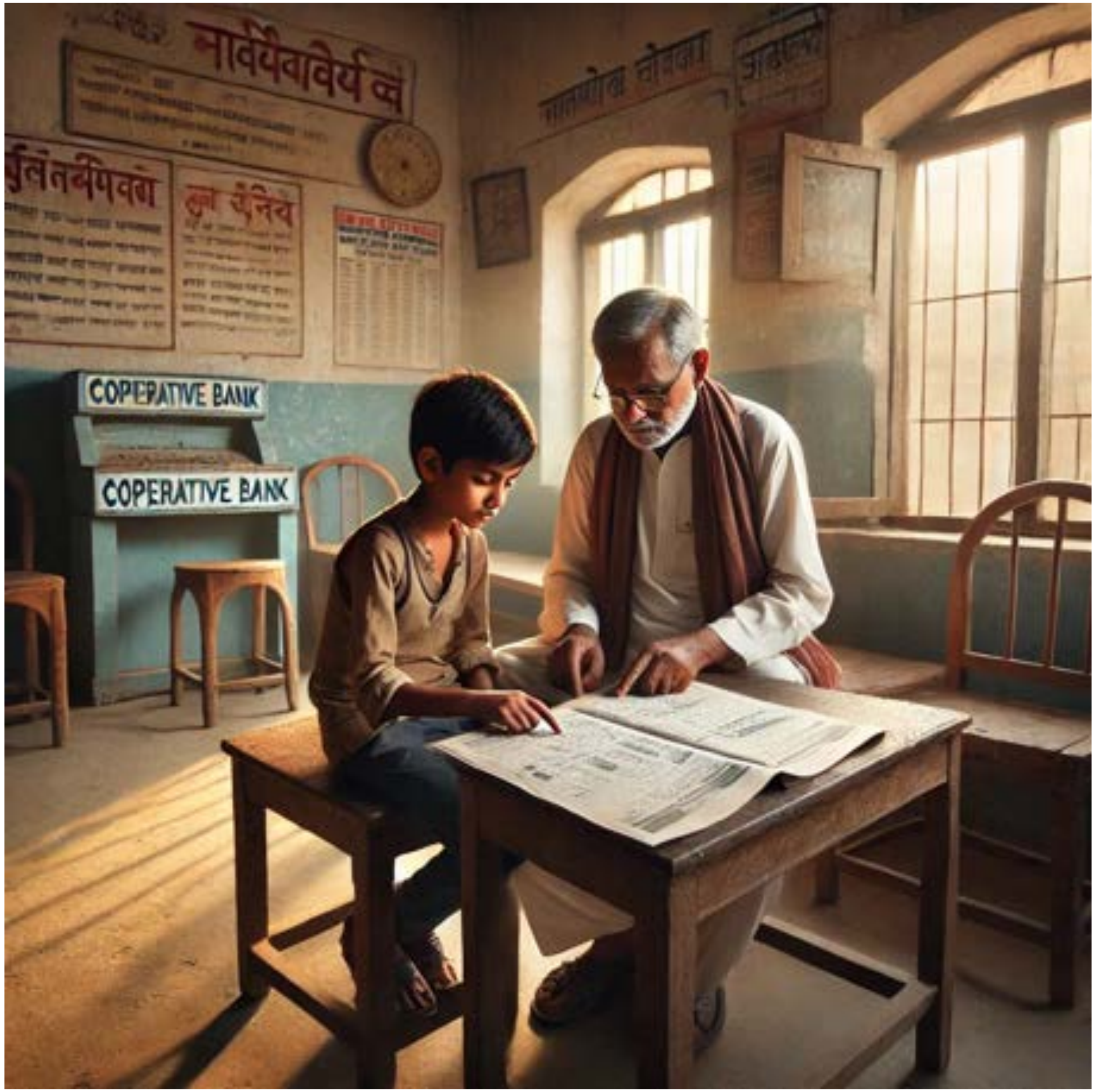


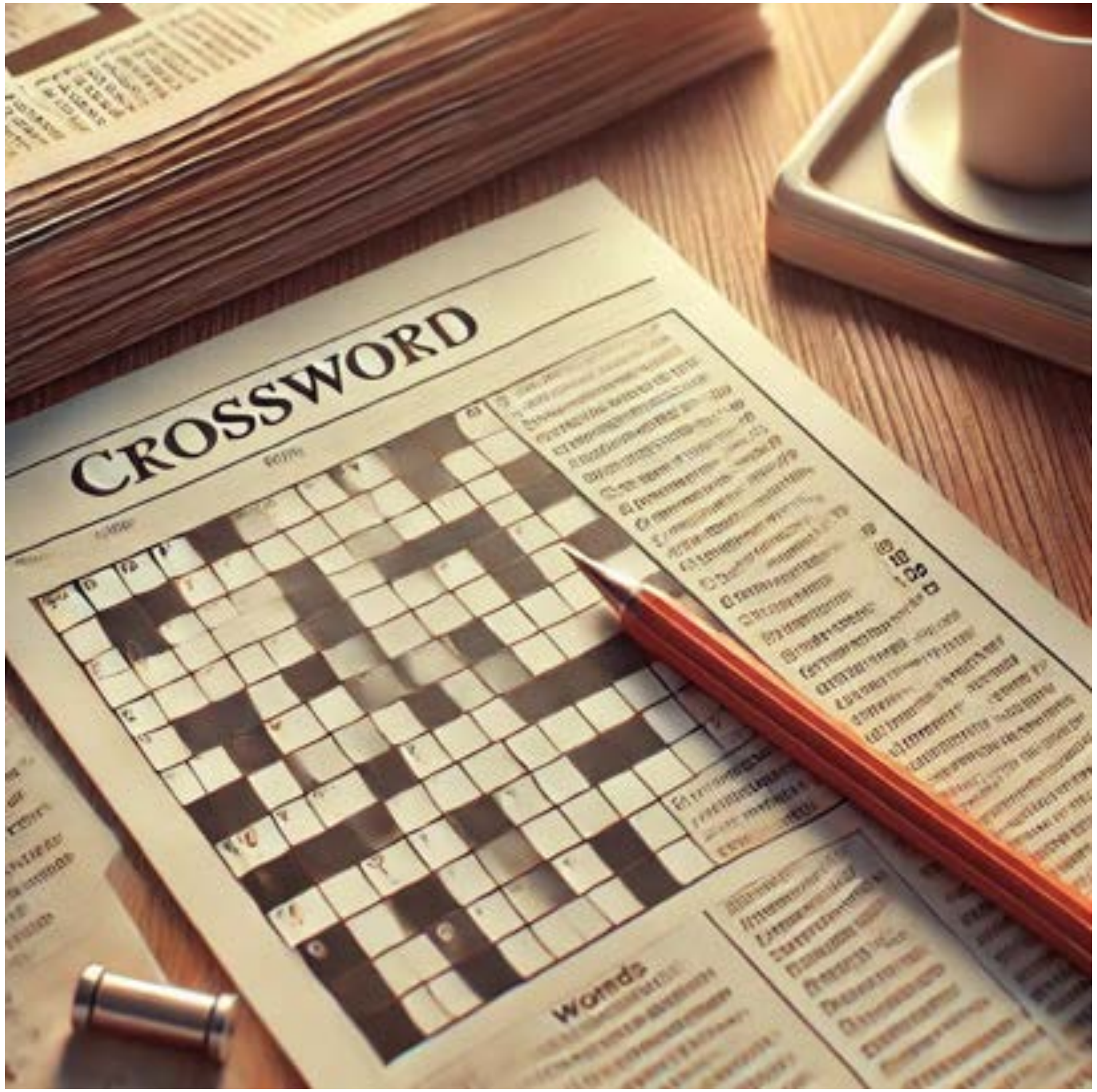
The Game



about 20 years back

about 40 years back


COPERATIVE BANK
COPERATIVE BANK


CROSSWORD
Words

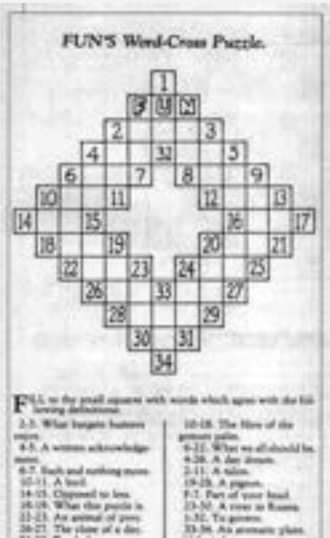
History timeline

THIS DAY IN HISTORY

DEC 21ST

1913

The first Crossword Puzzle published at NYW


FUN'S Word-Cross Puzzle.

1924

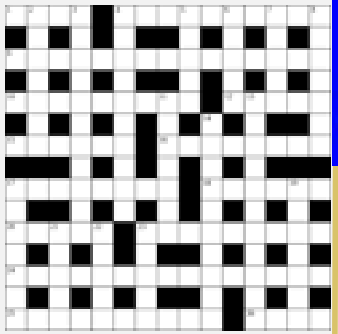
First Crossword Puzzle Book

1942

Crossword Puzzles Enter Newspapers – NYT published its first daily puzzle

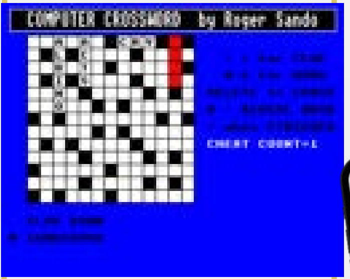
1950s

The Rise of Cryptic Crosswords



1970s-1980s

Digital Crosswords



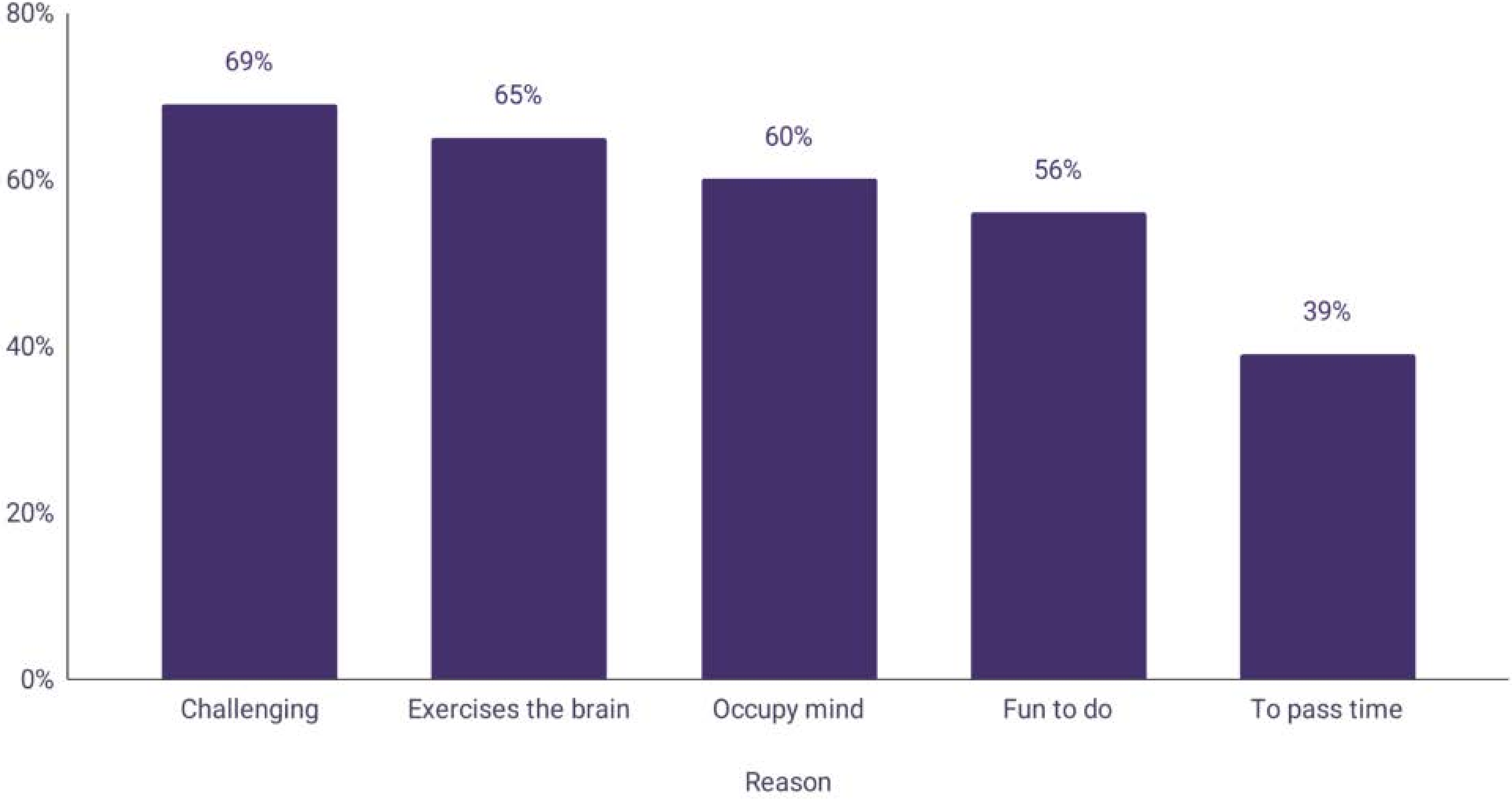
2000s

Crosswords Go Online

2021

Crossword Collaboration with AI

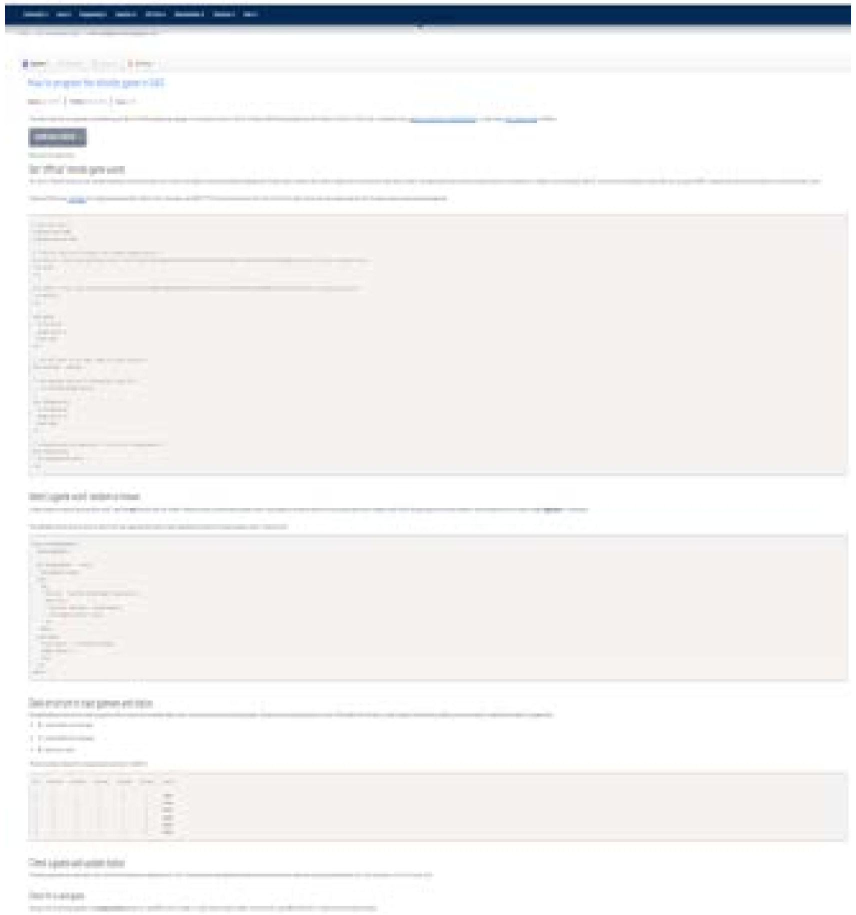
Top 5 reasons for people to play crosswords

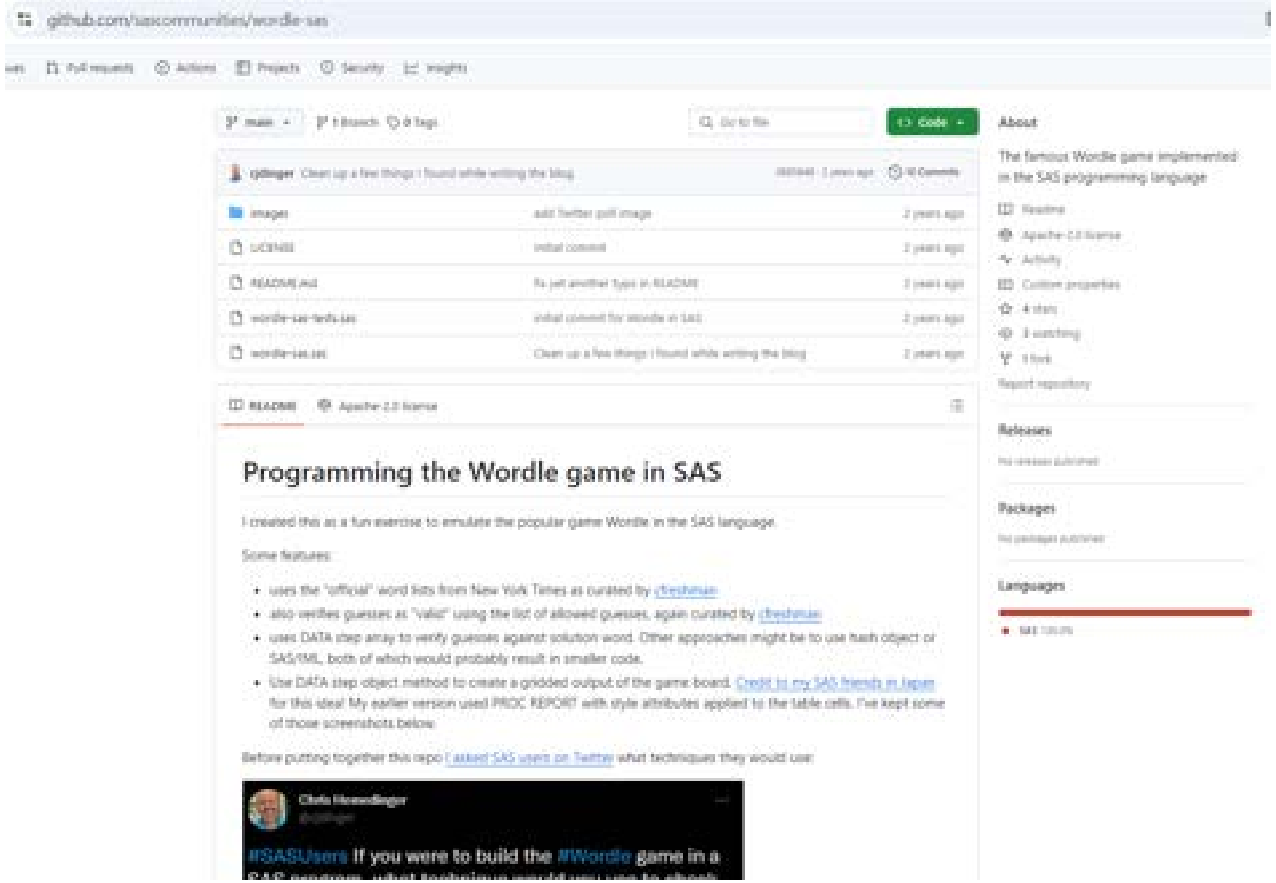


https://wordsrated.com/benefits-of-crossword-puzzles/

The Spark of an Idea

How to program the Wordle game in SAS





github.com/sascommunities/wordle-sas

cjdinger	Clean up a few things I found while writing the blog	10 Commits
images	add Twitter poll image	2 years ago
LICENSE	Initial commit	2 years ago
README.md	fix yet another typo in README	2 years ago
wordle-sas-tests.sas	initial commit for Wordle in SAS	2 years ago
wordle-sas.sas	Clean up a few things I found while writing the blog	2 years ago

About

The famous Wordle game implemented in the SAS programming language

- Readme
- Apache-2.0 license
- Activity
- Custom properties
- 4 stars
- 3 watching
- 1 fork

Report repository

Releases

No releases published

Packages

No packages published

Languages

- SAS 100.0%

Programming the Wordle game in SAS

I created this as a fun exercise to emulate the popular game Wordle in the SAS language.

Some features:

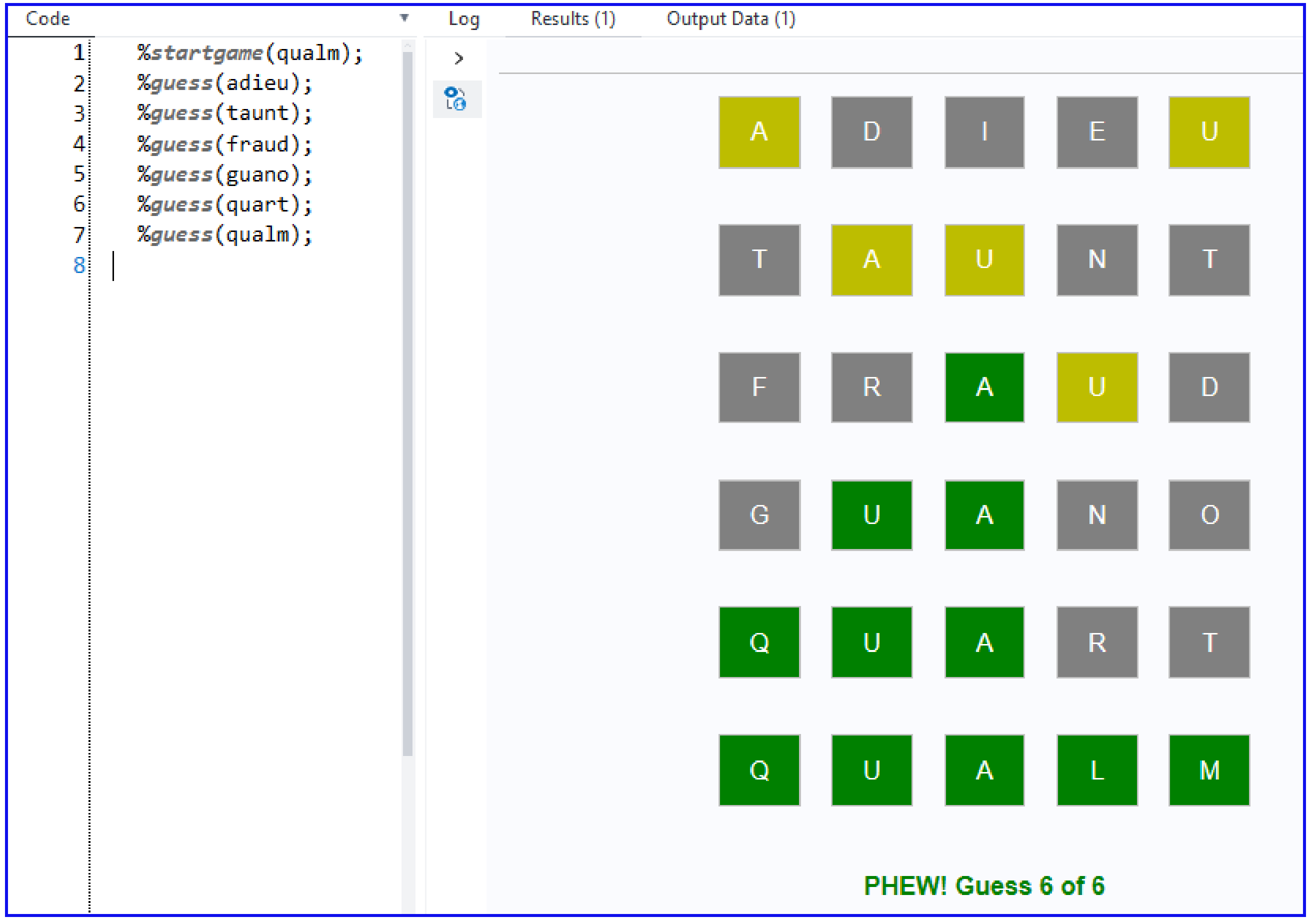
- uses the "official" word lists from New York Times as curated by cfreshman
- also verifies guesses as "valid" using the list of allowed guesses, again curated by cfreshman
- uses DATA step array to verify guesses against solution word. Other approaches might be to use hash object or SAS/IML, both of which would probably result in smaller code.
- Use DATA step object method to create a gridded output of the game board. Credit to my SAS friends in Japan for this idea! My earlier version used PROC REPORT with style attributes applied to the table cells. I've kept some of those screenshots below.

Before putting together this repo I asked SAS users on Twitter what techniques they would use:

Chris Hemedinger

#SASUsers If you were to build the #Wordle game in a

- https://communities.sas.com/t5/SAS-Communities-Library/How-to-program-the-Wordle-game-in-SAS/ta-p/802783
- https://github.com/sascommunities/wordle-sas



```
%startgame(qualm);
%guess(adieu);
%guess(taunt);
%guess(fraud);
%guess(guano);
%guess(quart);
%guess(qualm);
```

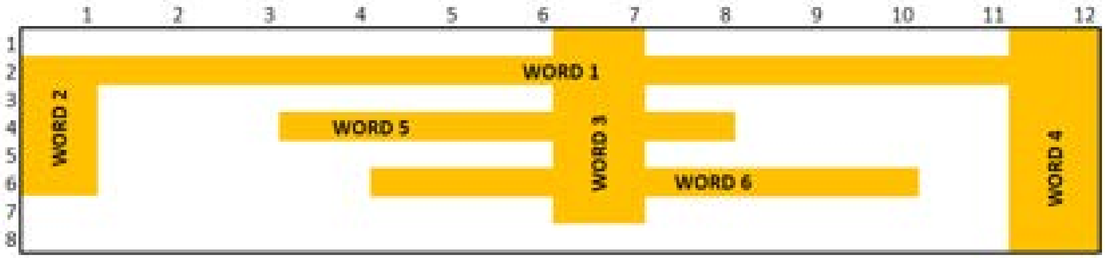
```
A D I E U
T A U N T
F R A U D
G U A N O
Q U A R T
Q U A L M

PHEW! Guess 6 of 6
```

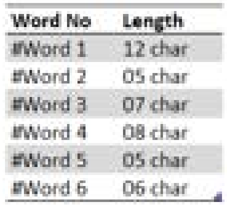
Wordle game by Chris

Building the Puzzle in SAS: Layout

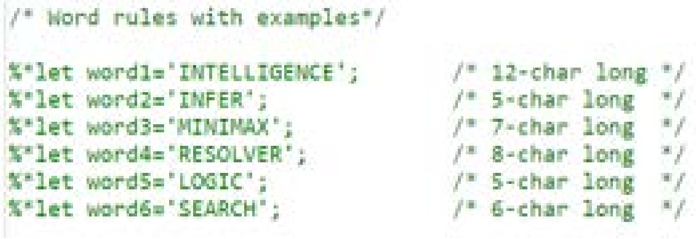
My Game, My Rules

12x8 Grid



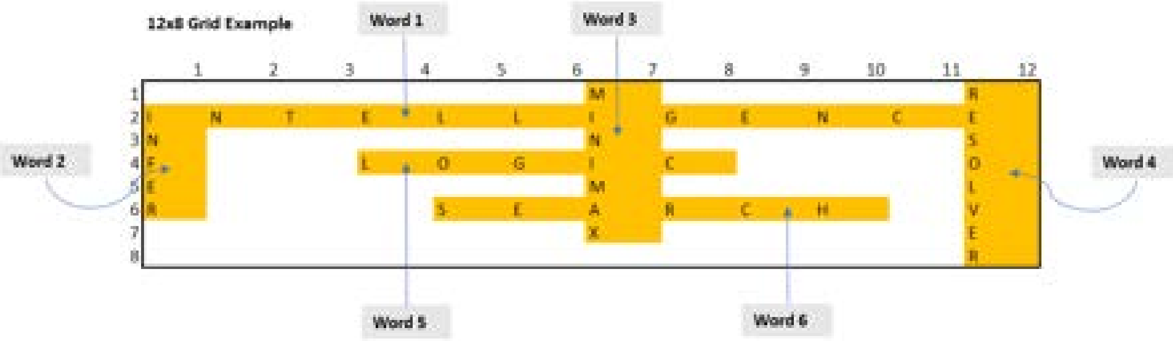


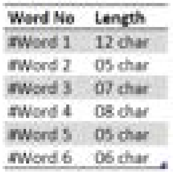
Word No	Length
#Word 1	12 char
#Word 2	05 char
#Word 3	07 char
#Word 4	08 char
#Word 5	05 char
#Word 6	06 char



```
/* Word rules with examples*/

%*let word1='INTELLIGENCE';       /* 12-char long */
%*let word2='INFER';              /* 5-char long  */
%*let word3='MINIMAX';            /* 7-char long  */
%*let word4='RESOLVER';           /* 8-char long  */
%*let word5='LOGIC';              /* 5-char long  */
%*let word6='SEARCH';             /* 6-char long  */
```

12x8 Grid Example

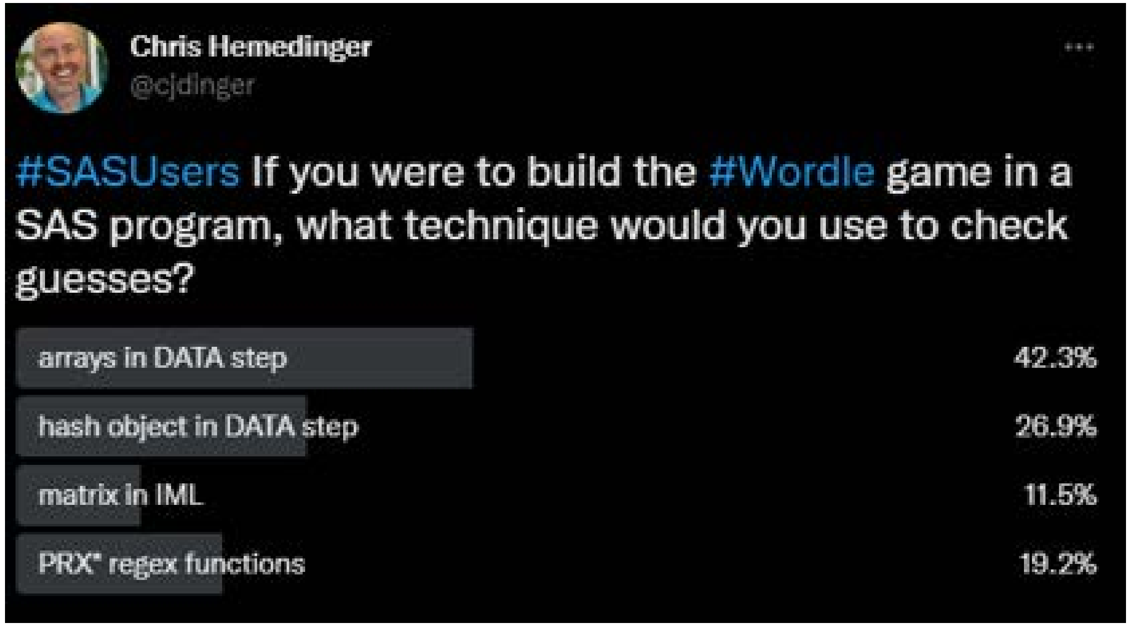




Word No	Length
#Word 1	12 char
#Word 2	05 char
#Word 3	07 char
#Word 4	08 char
#Word 5	05 char
#Word 6	06 char

Techniques for Building Word Puzzles in SAS

- SAS Arrays
- Hash Objects
- Hash of Hashes (Nested Hash Tables)
- SAS Formats
- PROC IML (Interactive Matrix Language)



Chris Hemedinger
@cjdinger

#SASUsers If you were to build the #Wordle game in a SAS program, what technique would you use to check guesses?

arrays in DATA step 42.3%

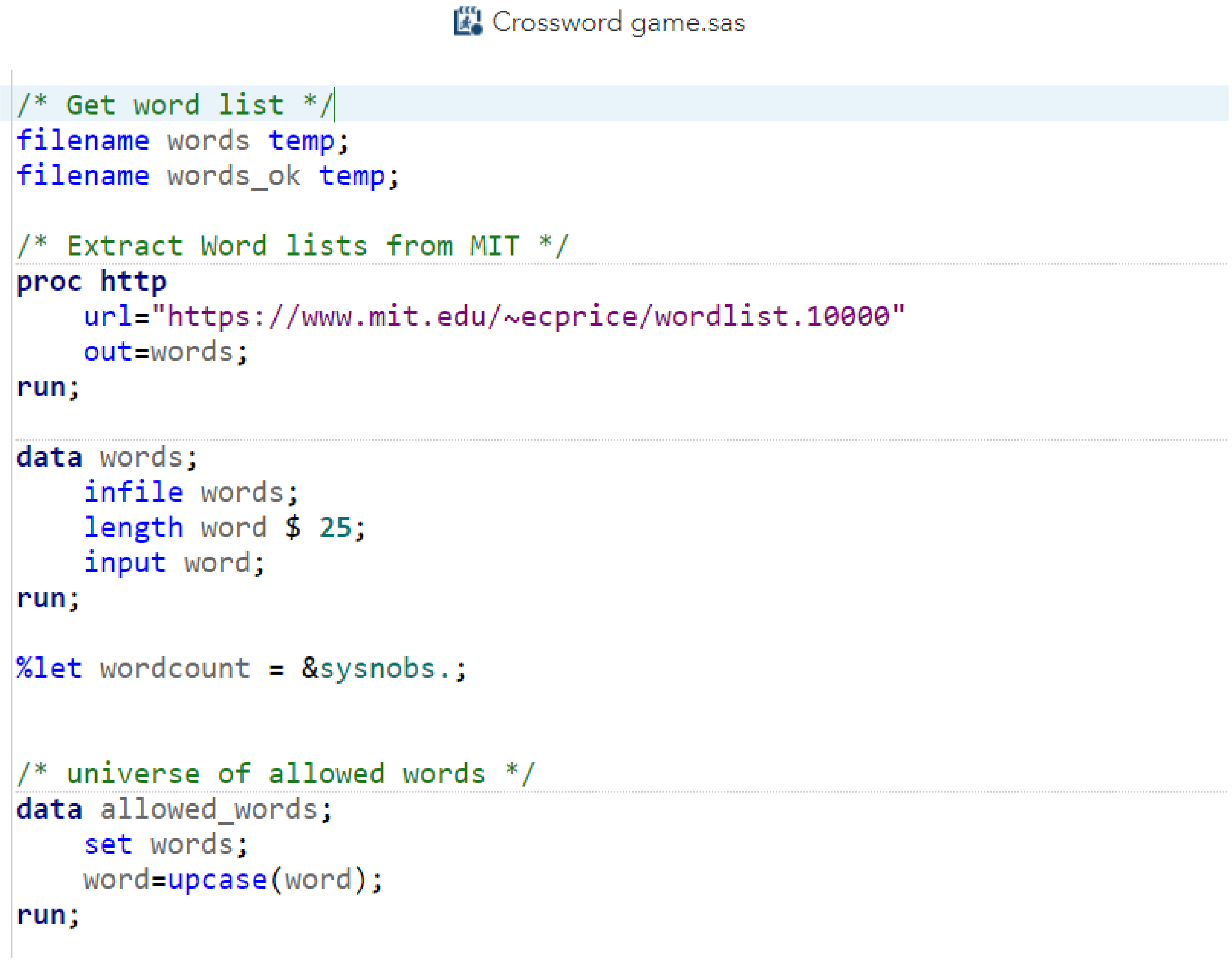
hash object in DATA step 26.9%

matrix in IML 11.5%

PRX* regex functions 19.2%

Approach:

1. Data-Driven Word Lists
2. SAS Arrays for Grid Layout
3. Iterative Loops for Word Placement
4. Character Manipulation with SAS Functions
5. Macro Programming for Dynamic Puzzles



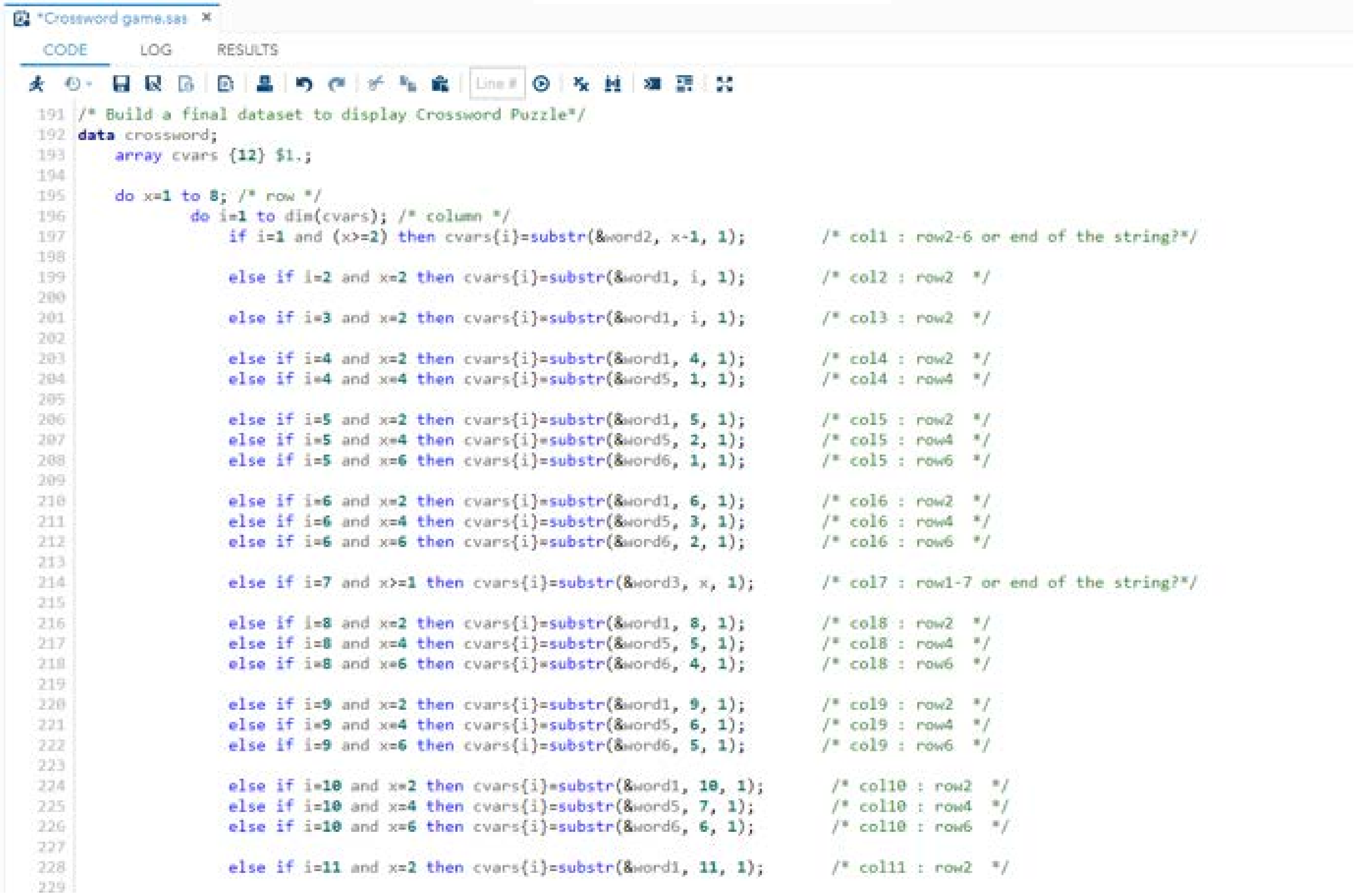
Crossword game.sas

```
/* Get word list */
filename words temp;
filename words_ok temp;

/* Extract Word lists from MIT */
proc http
    url="https://www.mit.edu/~ecprice/wordlist.10000"
    out=words;
run;

data words;
    infile words;
    length word $ 25;
    input word;
run;

%let wordcount = &sysnobs.;


/* universe of allowed words */
data allowed_words;
    set words;
    word=upcase(word);
run;
```

Word List

https://www.mit.edu/~ecprice/wordlist.10000

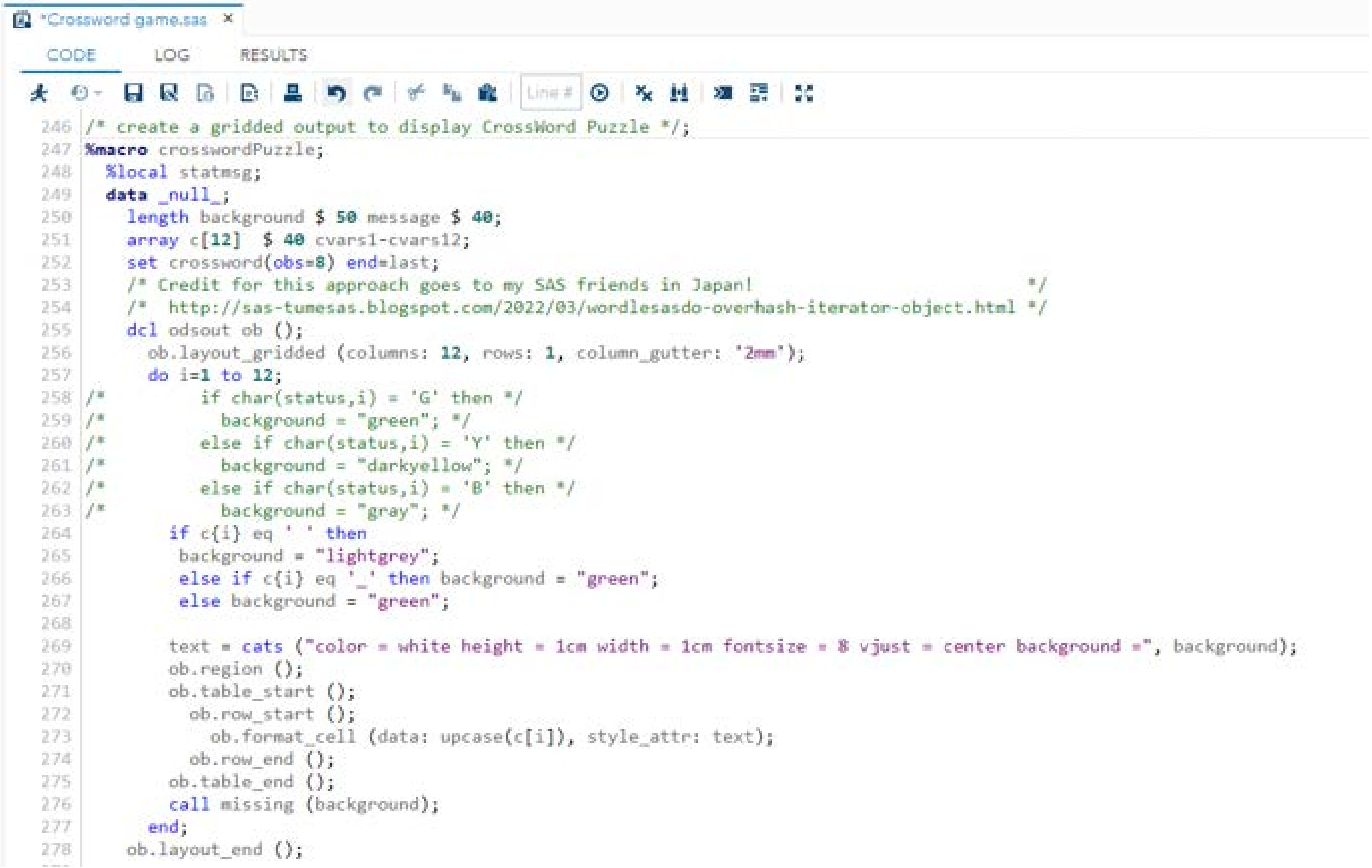
Crossword game.sas



```
/* Build a final dataset to display Crossword Puzzle*/
data crossword;
    array cvars {12} $1.;

    do x=1 to 8; /* row */
            do i=1 to dim(cvars); /* column */
                if i=1 and (x>=2) then cvars{i}=substr(&word2, x-1, 1);        /* col1 : row2-6 or end of the string?*/

                else if i=2 and x=2 then cvars{i}=substr(&word1, i, 1);        /* col2 : row2  */

                else if i=3 and x=2 then cvars{i}=substr(&word1, i, 1);        /* col3 : row2  */

                else if i=4 and x=2 then cvars{i}=substr(&word1, 4, 1);        /* col4 : row2  */
                else if i=4 and x=4 then cvars{i}=substr(&word5, 1, 1);        /* col4 : row4  */

                else if i=5 and x=2 then cvars{i}=substr(&word1, 5, 1);        /* col5 : row2  */
                else if i=5 and x=4 then cvars{i}=substr(&word5, 2, 1);        /* col5 : row4  */
                else if i=5 and x=6 then cvars{i}=substr(&word6, 1, 1);        /* col5 : row6  */

                else if i=6 and x=2 then cvars{i}=substr(&word1, 6, 1);        /* col6 : row2  */
                else if i=6 and x=4 then cvars{i}=substr(&word5, 3, 1);        /* col6 : row4  */
                else if i=6 and x=6 then cvars{i}=substr(&word6, 2, 1);        /* col6 : row6  */

                else if i=7 and x>=1 then cvars{i}=substr(&word3, x, 1);       /* col7 : row1-7 or end of the string?*/

                else if i=8 and x=2 then cvars{i}=substr(&word1, 8, 1);        /* col8 : row2  */
                else if i=8 and x=4 then cvars{i}=substr(&word5, 5, 1);        /* col8 : row4  */
                else if i=8 and x=6 then cvars{i}=substr(&word6, 4, 1);        /* col8 : row6  */

                else if i=9 and x=2 then cvars{i}=substr(&word1, 9, 1);        /* col9 : row2  */
                else if i=9 and x=4 then cvars{i}=substr(&word5, 6, 1);        /* col9 : row4  */
                else if i=9 and x=6 then cvars{i}=substr(&word6, 5, 1);        /* col9 : row6  */

                else if i=10 and x=2 then cvars{i}=substr(&word1, 10, 1);       /* col10 : row2  */
                else if i=10 and x=4 then cvars{i}=substr(&word5, 7, 1);        /* col10 : row4  */
                else if i=10 and x=6 then cvars{i}=substr(&word6, 6, 1);        /* col10 : row6  */

                else if i=11 and x=2 then cvars{i}=substr(&word1, 11, 1);       /* col11 : row2  */

```

SAS Array and iterative loop for Grid Layout

Crossword game.sas



```
/* create a gridded output to display CrossWord Puzzle */;
%macro crosswordPuzzle;
  %local statmsg;
  data _null_;
    length background $ 50 message $ 40;
    array c[12]  $ 40 cvars1-cvars12;
    set crossword(obs=8) end=last;
    /* Credit for this approach goes to my SAS friends in Japan!                          */
    /*  http://sas-tumesas.blogspot.com/2022/03/wordlesasdo-overhash-iterator-object.html */
    dcl odsout ob ();
      ob.layout_gridded (columns: 12, rows: 1, column_gutter: '2mm');
      do i=1 to 12;
/*         if char(status,i) = 'G' then */
/*           background = "green"; */
/*         else if char(status,i) = 'Y' then */
/*           background = "darkyellow"; */
/*         else if char(status,i) = 'B' then */
/*           background = "gray"; */
        if c{i} eq ' ' then
         background = "lightgrey";
         else if c{i} eq '_' then background = "green";
         else background = "green";

        text = cats ("color = white height = 1cm width = 1cm fontsize = 8 vjust = center background =", background);
        ob.region ();
        ob.table_start ();
          ob.row_start ();
            ob.format_cell (data: upcase(c[i]), style_attr: text);
          ob.row_end ();
        ob.table_end ();
        call missing (background);
      end;
    ob.layout_end ();
```

Display Crossword puzzle

Beyond Puzzles – Building Skills

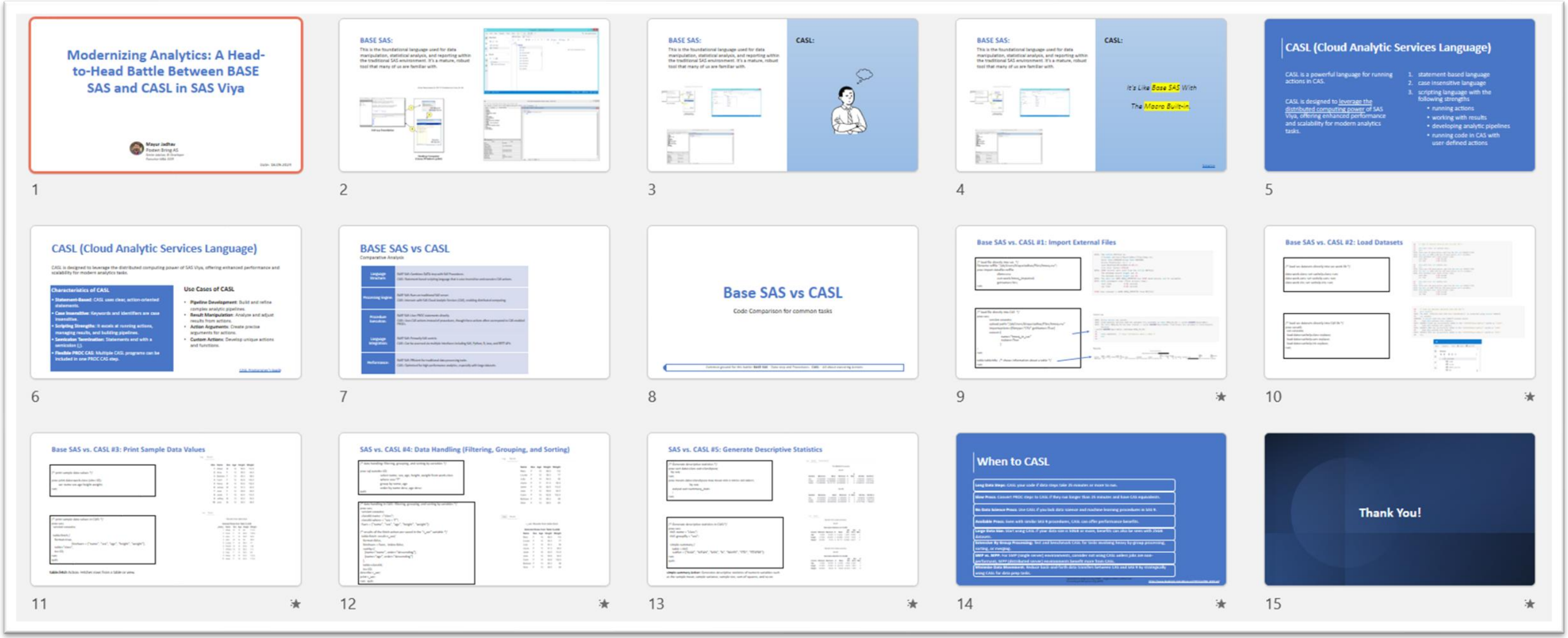
1. **Enhances Programming and Problem-Solving Skills**
   - Strengthens logic, algorithms, and debugging
   - Improves optimization and efficiency
2. **Improves Knowledge of Data Structures and Algorithms**
   - Mastery of structures like Trie, Hashmaps, arrays
   - String manipulation and word search optimization
3. **Boosts Creativity & Innovation**
   - Custom game design & UI/UX
   - Explore AI & Natural Language Processing (NLP)
4. **Portfolio & Career Building**
   - Showcases coding skills for job opportunities
   - Monetization potential through game distribution
5. **Cross-Platform & Continuous Learning**
   - Develop for mobile, web, or desktop
   - Learn new frameworks, languages, or APIs
6. **Community Engagement & Social Impact**
   - Contribute to open-source projects
   - Create educational tools and games
7. **Personal Satisfaction & Fun**
   - Combines creativity and technical learning
   - Rewarding and engaging project

“Play is the highest form of research.”

– Unknown

Building games isn’t just fun; it’s a hands-on, **creative way to learn and grow**.

BASE SAS vs CASL



1 Modernizing Analytics: A Head-to-Head Battle Between BASE SAS and CASL in SAS Viya

Mayur Jadhav

Posten Bring AS

2 BASE SAS:

This is the foundational language used for data manipulation, statistical analysis, and reporting within the traditional SAS environment. It's a mature, robust tool that many of us are familiar with.

3 BASE SAS:

This is the foundational language used for data manipulation, statistical analysis, and reporting within the traditional SAS environment. It's a mature, robust tool that many of us are familiar with.

CASL:

4 BASE SAS:

This is the foundational language used for data manipulation, statistical analysis, and reporting within the traditional SAS environment. It's a mature, robust tool that many of us are familiar with.

CASL:

It's Like Base SAS With

The Macro Built-in.

5 CASL (Cloud Analytic Services Language)

CASL is a powerful language for running actions in CAS.

CASL is designed to leverage the distributed computing power of SAS Viya, offering enhanced performance and scalability for modern analytics tasks.

1. statement-based language
2. case insensitive language
3. scripting language with the following strengths
   - running actions
   - working with results
   - developing analytic pipelines
   - running code in CAS with user-defined actions

6 CASL (Cloud Analytic Services Language)

CASL is designed to leverage the distributed computing power of SAS Viya, offering enhanced performance and scalability for modern analytics tasks.

Characteristics of CASL

- Statement-Based: CASL uses clear, action-oriented statements.
- Case Insensitive: Keywords and identifiers are case insensitive.
- Scripting Strengths: It excels at running actions, managing results, and building pipelines.
- Semicolon Termination: Statements end with a semicolon (;).
- Flexible PROC CAS: Multiple CASL programs can be included in one PROC CAS step.

Use Cases of CASL

- Pipeline Development: Build and refine complex analytic pipelines.
- Result Manipulation: Analyze and adjust results from actions.
- Action Arguments: Create precise arguments for actions.
- Custom Actions: Develop unique actions and functions.

7 BASE SAS vs CASL

Comparative Analysis

8 Base SAS vs CASL

Code Comparison for common tasks

9 Base SAS vs. CASL #1: Import External Files

10 Base SAS vs. CASL #2: Load Datasets

11 Base SAS vs. CASL #3: Print Sample Data Values

12 SAS vs. CASL #4: Data Handling (Filtering, Grouping, and Sorting)

13 SAS vs. CASL #5: Generate Descriptive Statistics

14 When to CASL

15 Thank You!

https://communities.sas.com/t5/SAS-Community-Nordic/FANS-nettverksm%C3%B8te-SAS-Data-Engineering-programmering-data/td-p/930400

Thank You!