April 2019 - Challenge

This month's challenge is from Reda Kebbaj. (Thanks!)

Find nine different prime numbers that can be placed in a 3x3 square in such a way that the average of every row, column, and diagonal is also a prime number.

Solution:

Below are squares for sizes 3, 4, and 5 which contain distinct primes having every row, column, and diagonal averages equal to a prime number. The various averages are rendered in red for each row, column, and diagonal. The diagonal averages are given for the diagonal followed by the anti- diagonal.

A natural conjecture is that such squares exist for every n.

Minizinc constraint code is included below.

n = 3

23	197	83	**101**
11	5	17	**11**
53	101	149	**101**
29	**101**	**83**	**59, 47**

n = 4

41	53	199	23	**79**
19	17	13	43	**23**
83	11	7	47	**37**
5	131	193	3	**83**
37	**53**	**103**	**29**	**17, 13**

n = 5

29	127	79	47	23	61
73	17	103	19	53	53
107	101	3	41	43	59
89	13	61	11	31	41
7	97	59	37	5	41
61	71	61	31	31	13, 13

```
include "globals.mzn";
include “alldifferent.mzn";

set of int: P = {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109,
113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199};

set of int: Primes = {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107,
109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241,
251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389,  };

% array[1..25] of var P: s;
% var P: r1;
% var P: r2;
% var P: r3;
% var P: r4;
% var P: r5;

% var P: c1;
% var P: c2;
% var P: c3;
% var P: c4;
% var P: c5;


% var P: d1;
% var P: d2;

% constraint (s[1]+s[2]+s[3] +s[4] +s[5] == 5*r1);
% constraint (s[6]+s[7]+s[8]+s[9]+s[10] == 5*r2);
% constraint (s[11]+s[12]+s[13] +s[14]+s[15] == 5*r3);
% constraint (s[16]+s[17]+s[18]+s[19]+s[20] == 5*r4);
% constraint (s[21]+s[22]+s[23]+s[24]+s[25] == 5*r4);

% constraint (s[1]+s[6]+s[11] +s[16] +s[21] == 5*c1);
% constraint (s[2]+s[7]+s[12]+s[17]+s[22] == 5*c2);
% constraint (s[3]+s[8]+s[13] +s[18]+s[23] == 5*c3);
% constraint (s[4]+s[9]+s[14]+s[19]+s[24] == 5*c4);
% constraint (s[5]+s[10]+s[15]+s[20]+s[25] == 5*c4);
% constraint (s[1]+s[7]+s[13] +s[19] + s[25]== 5*d1);
% constraint (s[5]+s[9]+s[13] +s[17] + s[21] == 5*d2);

%array[1..16] of var P: s;
array[1..9] of var P: s;
var P: a;
var P: b;
var P: c;

var P: d;
var P: e;
var P: f;

var P: g;
var P: h;

% var P: a;
% var P: b;
% var P: c;
% var P: d;

% var P: e;
% var P: f;
% var P: g;
% var P: h;

% var P: i;
% var P: j;

% constraint (s[1]+s[2]+s[3] +s[4] == 4*a);
% constraint (s[5]+s[6]+s[7] +s[8] == 4*b);
% constraint (s[9]+s[10]+s[11] +s[12] == 4*c);
% constraint (s[13]+s[14]+s[15] +s[16] == 4*d);

% constraint (s[1]+s[5]+s[9] +s[13] == 4*e);
% constraint (s[2]+s[6]+s[10] +s[14] == 4*f);
% constraint (s[3]+s[7]+s[11] +s[15] == 4*g);
% constraint (s[4]+s[8]+s[12] +s[16] == 4*h);

% constraint (s[1]+s[6]+s[11] +s[16] == 4*i);
% constraint (s[4]+s[7]+s[10] +s[13] == 4*j);

% constraint (s[7]+s[8]+s[9] == 3*c);

% constraint (s[1]+s[4]+s[7] == 3*d);
% constraint (s[2]+s[5]+s[8] == 3*e);
% constraint (s[3]+s[6]+s[9] == 3*f);

% constraint (s[1]+s[5]+s[9] == 3*g);
% constraint (s[3]+s[5]+s[7] == 3*h);
% constraint (s[1]+s[2]+s[3]+s[4]+s[5]+s[6]+s[7]+s[8]+s[9] +s[1]+s[4]+s[7] + s[2]+s[5]+s[8] + s[3]+s[6]+s[9] +s[1]+s[5]+s[9] +s[3]+s[5]+s[7] == 24*a);


constraint (s[1]+s[2]+s[3] == 3*a);
constraint (s[4]+s[5]+s[6] == 3*b);
constraint (s[7]+s[8]+s[9] == 3*c);

constraint (s[1]+s[4]+s[7] == 3*d);
constraint (s[2]+s[5]+s[8] == 3*e);
constraint (s[3]+s[6]+s[9] == 3*f);

constraint (s[1]+s[5]+s[9] == 3*g);
constraint (s[3]+s[5]+s[7] == 3*h);

% array[1..9] of var Primes: s;
% var Primes: a;
% var Primes: b;
% var Primes: c;

% var Primes: d;
% var Primes: e;
% var Primes: f;

% var Primes: g;
% var Primes: h;

% constraint (s[1]+s[2]+s[3] == 3*a);
% constraint (s[4]+s[5]+s[6] == 3*b);
% constraint (s[7]+s[8]+s[9] == 3*c);

% constraint (s[1]+s[4]+s[7] == 3*d);
% constraint (s[2]+s[5]+s[8] == 3*e);
% constraint (s[3]+s[6]+s[9] == 3*f);

% constraint (s[3]+s[6]+s[9] == 3*g);
% constraint (s[3]+s[6]+s[9] == 3*h);

constraint alldifferent(s);

solve satisfy;
```

Minizinc output:

```
s = array1d(1..9 ,[23, 113, 83, 11, 5, 17, 53, 41, 29]);
a = 73;
b = 11;
c = 41;
d = 29;
e = 53;
f = 43;
g = 19;
h = 47;
```