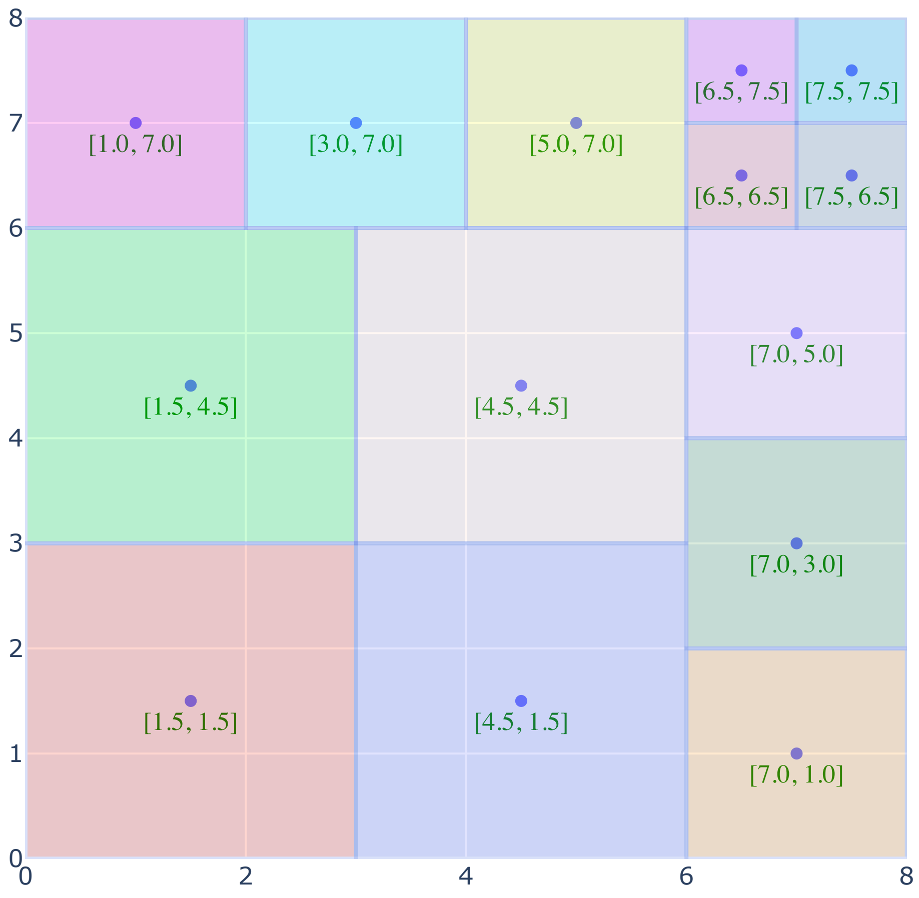
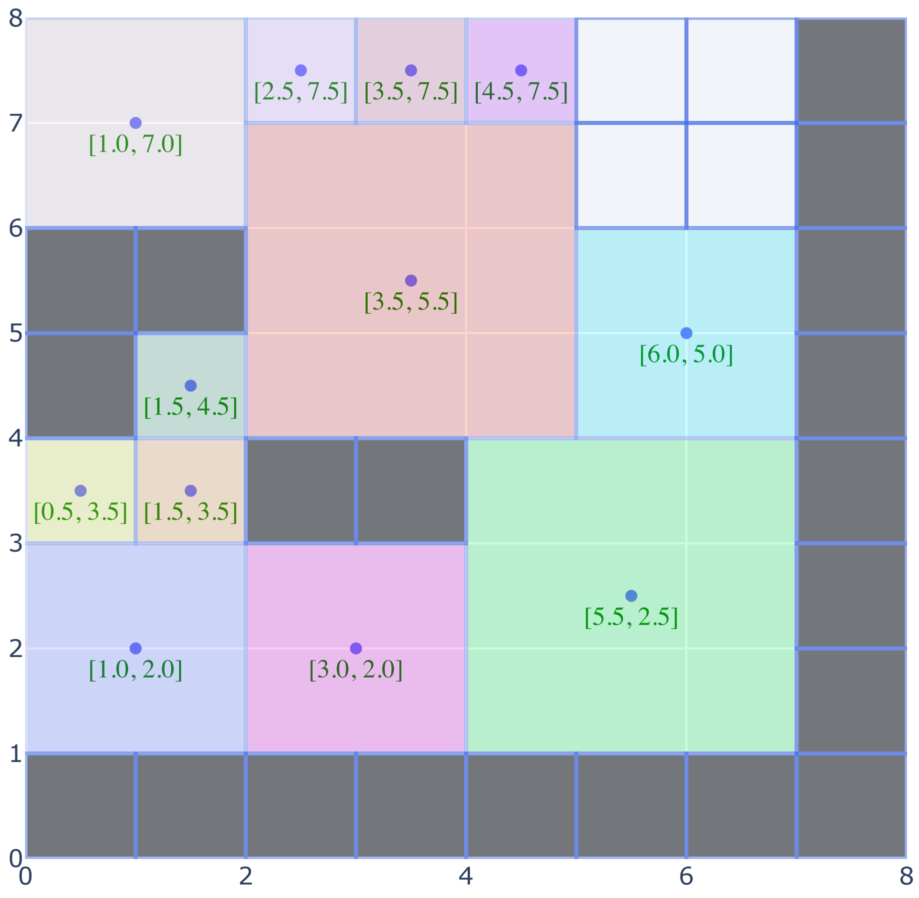
```
In [ ]: #!/usr/bin/env python
        # -*- coding: utf-8 -*-
        """
        Simple OOP based python square packing implementation.

        @Author: bmetenko
        @Date: 22May2022
        @Links: https://github.com/bmetenko
        """
        ######################################################################

        import pandas as pd
        import numpy as np
        import plotly.graph_objects as go
        import plotly.io as pio
        from squares import Square, SquareCanvas, check_bounds

        # pio.renderers.default = "plotly_mimetype+notebook"
        pio.renderers.default = "svg"

        ######################################################################

        list_square_radii_1 = [3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1]
        list_squares_1 = [Square(radius) for radius in list_square_radii_1]

        B = SquareCanvas(max_bound=8, contents=list_squares_1)

        separator = "-------------\n"
        print(f"{B.contents=}\n"
        f"{separator}{B.frame}\n"
        f"{separator}{B.y_list=}\n"
        f"{separator}{B.x_list=}\n"
        f"{separator}{B.center_list=}"
        f"{separator}")

        B.generate_plotly()

        ######################################################################
        custom_frame = \
        np.array([
            [-1.0, 0.0, 0.0, 0.0,-1.0,-1.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0, 0.0, 0.0,-1.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0,-1.0, 0.0, 0.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0,-1.0, 0.0, 0.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
            [-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0],
        ])

        list_square_radii_2 = [3, 3, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1]
        list_squares_2 = [Square(i) for i in list_square_radii_2]
        C = SquareCanvas(contents=list_squares_2, frame_override=custom_frame)
        print(f"{separator}{C.frame}\n{separator}{C.center_list=}\n{separator}")
        C.generate_plotly()

        ######################################################################
```

```
B.contents=[Square3::ctr@[1.5, 1.5], Square3::ctr@[1.5, 4.5], Square3::ctr@[4.
5, 1.5], Square3::ctr@[4.5, 4.5], Square2::ctr@[1.0, 7.0], Square2::ctr@[3.0,
7.0], Square2::ctr@[5.0, 7.0], Square2::ctr@[7.0, 1.0], Square2::ctr@[7.0, 3.
0], Square2::ctr@[7.0, 5.0], Square1::ctr@[6.5, 6.5], Square1::ctr@[6.5, 7.5],
Square1::ctr@[7.5, 6.5], Square1::ctr@[7.5, 7.5]]
------------
[[ 1  1  1  2  2  2  5  5]
 [ 1  1  1  2  2  2  5  5]
 [ 1  1  1  2  2  2  6  6]
 [ 3  3  3  4  4  4  6  6]
 [ 3  3  3  4  4  4  7  7]
 [ 3  3  3  4  4  4  7  7]
 [ 8  8  9  9 10 10 11 12]
 [ 8  8  9  9 10 10 13 14]]
------------
B.y_list=[1.5, 4.5, 1.5, 4.5, 7.0, 7.0, 7.0, 1.0, 3.0, 5.0, 6.5, 7.5, 6.5, 7.
5]
------------
B.x_list=[1.5, 1.5, 4.5, 4.5, 1.0, 3.0, 5.0, 7.0, 7.0, 7.0, 6.5, 6.5, 7.5, 7.
5]
------------
B.center_list=['[1.5, 1.5]', '[1.5, 4.5]', '[4.5, 1.5]', '[4.5, 4.5]', '[1.0,
7.0]', '[3.0, 7.0]', '[5.0, 7.0]', '[7.0, 1.0]', '[7.0, 3.0]', '[7.0, 5.0]',
'[6.5, 6.5]', '[6.5, 7.5]', '[7.5, 6.5]', '[7.5, 7.5]']------------
```

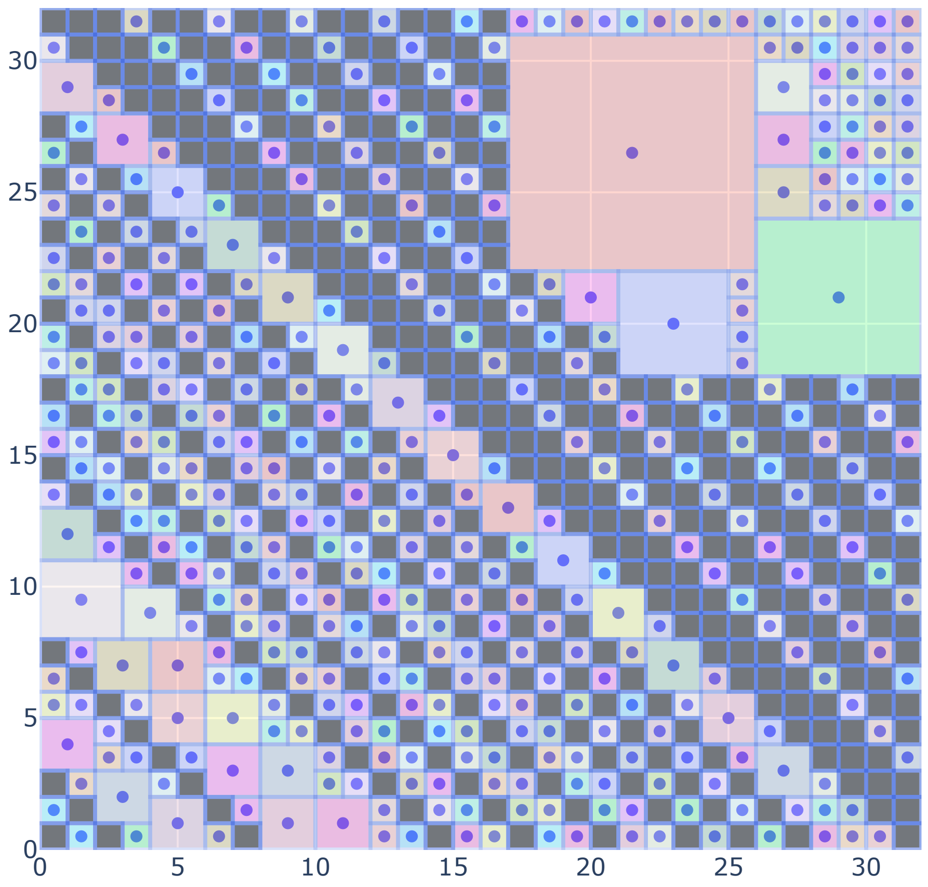


```
------------
[[-1.  3.  3.  7. -1. -1.  4.  4.]
 [-1.  3.  3.  8.  9. -1.  4.  4.]
 [-1.  5.  5. -1.  1.  1.  1. 10.]
 [-1.  5.  5. -1.  1.  1.  1. 11.]
 [-1.  2.  2.  2.  1.  1.  1. 12.]
 [-1.  2.  2.  2.  6.  6.  0.  0.]
 [-1.  2.  2.  2.  6.  6.  0.  0.]
 [-1. -1. -1. -1. -1. -1. -1. -1.]]
------------
C.center_list=['[3.5, 5.5]', '[5.5, 2.5]', '[1.0, 2.0]', '[1.0, 7.0]', '[3.0,
2.0]', '[6.0, 5.0]', '[0.5, 3.5]', '[1.5, 3.5]', '[1.5, 4.5]', '[2.5, 7.5]',
'[3.5, 7.5]', '[4.5, 7.5]']
------------
```



In [ ]:
```
import PIL.Image
I = np.asarray(PIL.Image.open('bw_sprite1.png')).astype(int)
where_not0 = np.where(I != 0)
where_0 = np.where(I == 0)

I[where_0] = 0
I[where_not0] = -1
print(I)

list_square_radii_3 = [9, 6, 4, 3, 2, 1, 1, *[1,2]*30, *[1]*342]
list_squares_3 = [Square(i) for i in list_square_radii_3]
D = SquareCanvas(contents=list_squares_3, frame_override=I)
a = D.frame
print(f"{D.frame}")
D.generate_plotly(show_text=False)
```

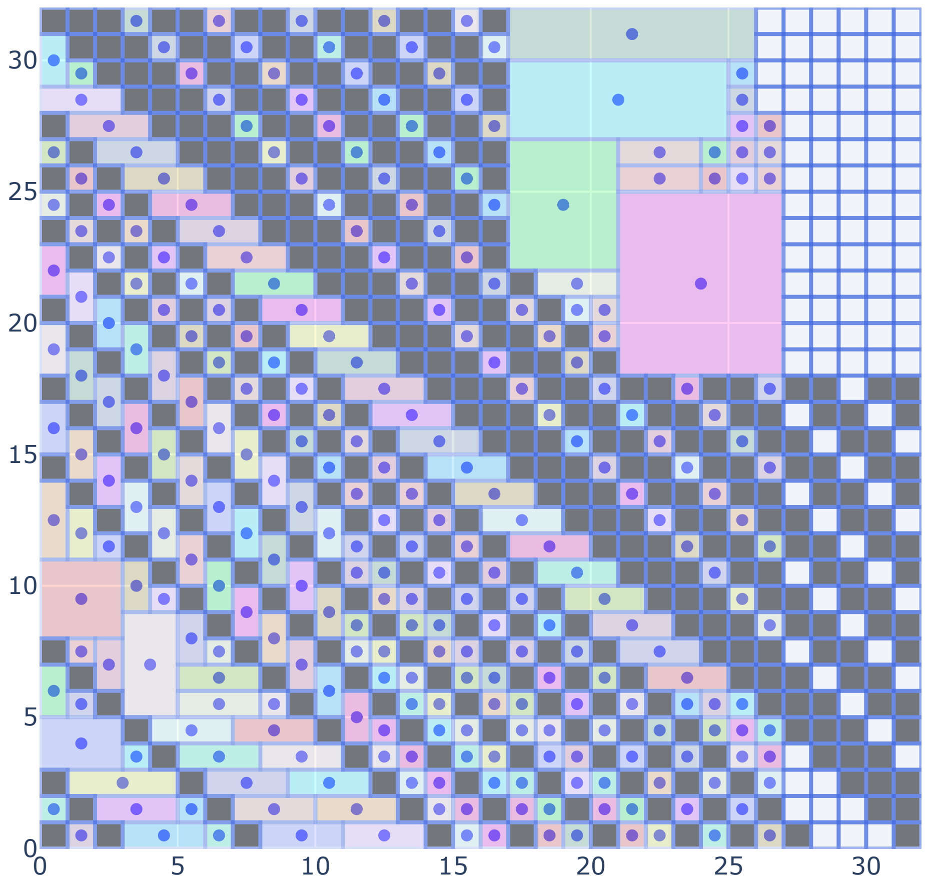
```
[[-1  0 -1 ...  0  0 -1]
 [ 0 -1  0 ...  0 -1 -1]
 [-1  0  0 ... -1 -1 -1]
 ...
 [ 0  0 -1 ...  0  0  0]
 [ 0 -1 -1 ...  0  0  0]
 [-1 -1 -1 ...  0  0  0]]
[[ -1   6  -1 ...  11  28  -1]
 [ 30  -1  32 ...  11  -1  -1]
 [ -1  13  13 ...  -1  -1  -1]
 ...
 [368 369  -1 ... 380 381 382]
 [383  -1  -1 ... 394 395 396]
 [ -1  -1  -1 ... 407 408 409]]
```



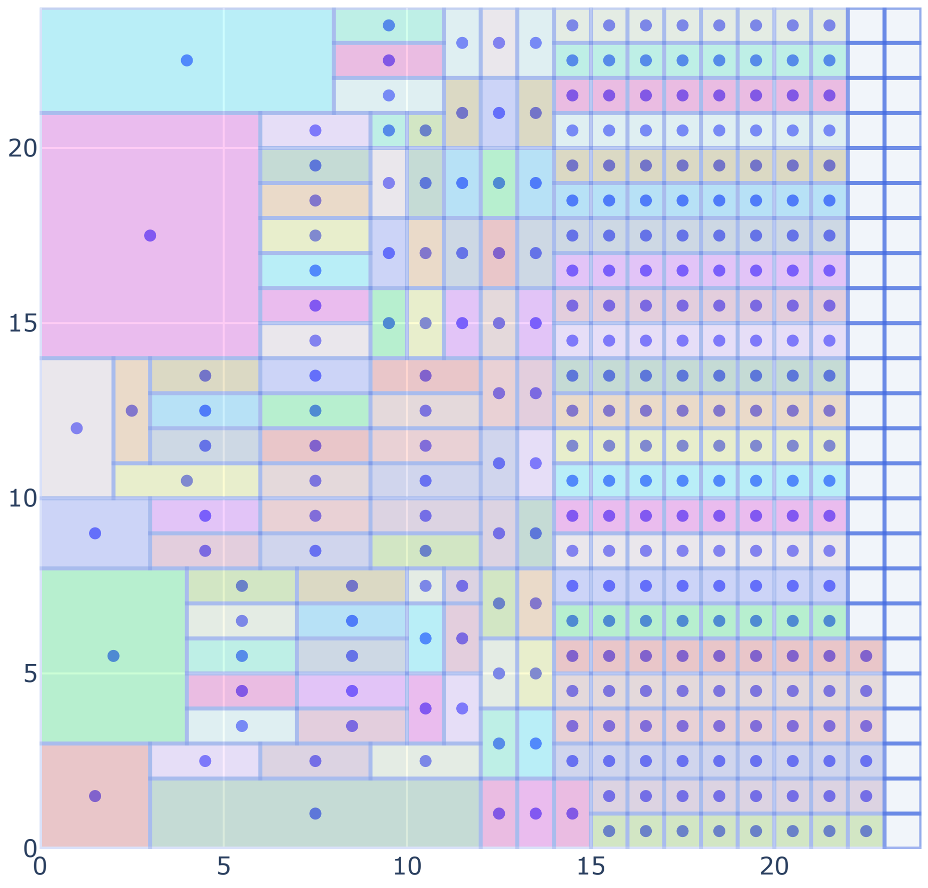
In [ ]:
```python
from squares import Rect
E = Rect(5, 4).rotate(90)
print(f"{E.area=}, {E.center=}, \n{E.coordinates=}")
```

```
E.area=20.0, E.center=[2.0, 2.5],
E.coordinates=[array([0., 0.]), array([0., 5.]), array([4., 0.]), array([4.,
5.])]
```

```
In [ ]: I = np.asarray(PIL.Image.open('bw_sprite1.png')).astype(int)

        # transform black and white
        where_not0 = np.where(I != 0)
        where_0 = np.where(I == 0)

        I[where_0] = 0
        I[where_not0] = -1

        x_list = [3, 4, 3, 2, 6, 8, 4, 1, 9, *[3]*40, *[1]*40, *[1]*200]
        y_list = [3, 5, 2, 4, 7, 3, 1, 3, 2, *[1]*40, *[2]*40, *[1]*200]

        F = SquareCanvas(contents=[Rect(length=i, width=j) for i, j in zip(x_list, y_li
        F.generate_plotly(show_text=False)
```

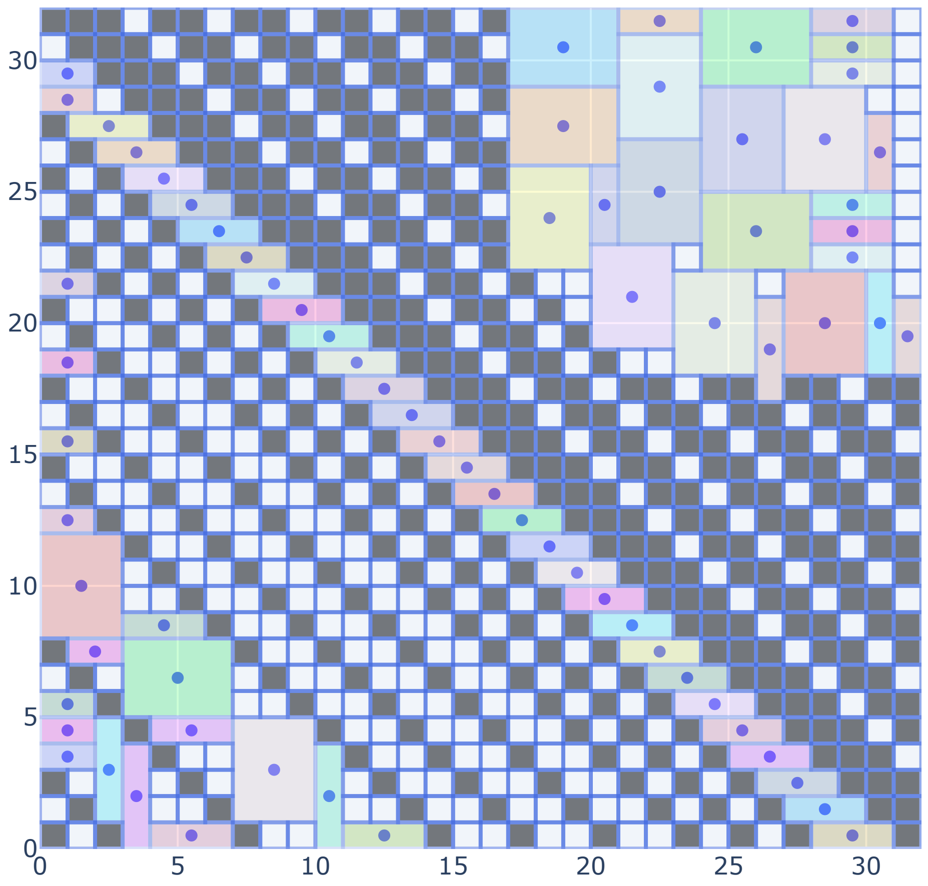


```
In [ ]: F = SquareCanvas(contents=[Rect(length=i, width=j) for i, j in zip(x_list, y_li
        F.generate_plotly(show_text=False)
```



```
In [ ]: I = np.asarray(PIL.Image.open('bw_sprite1.png')).astype(int)

        # transform black and white
        where_not0 = np.where(I != 0)
        where_0 = np.where(I == 0)

        I[where_0] = 0
        I[where_not0] = -1

        zip_populate = zip([4,3, 1, 4, 1, 4] * 5 + [1,1,1] * 14, [3, 4, 2, 3, 2, 1] * 5

        F = SquareCanvas(
            contents=[
                Rect(length=i, width=j).rotate(90)
                for i, j
                in zip_populate
                ],
            max_bound = 24,
            allow_rotation=True,
            frame_override=I
            )
        F.generate_plotly(show_text=False)
```

```
I = np.asarray(PIL.Image.open('bw_sprite1.png')).astype(int)

# transform black and white
where_not0 = np.where(I != 0)
where_0 = np.where(I == 0)

I[where_0] = 0
I[where_not0] = -1

zip_populate = zip([4,3, 1, 4, 1, 4] * 5 + [1,1,1] * 14, [3, 4, 2, 3, 2, 1] * 5

F = SquareCanvas(
    contents=[
        Rect(length=i, width=j).rotate(90)
        for i, j
        in zip_populate
        ],
    max_bound = 24,
    allow_rotation=False,
    frame_override=I
    )
F.generate_plotly(show_text=False)
```



0
5
10
15
20
25
30
0
5
10
15
20
25
30

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
/Users/bohdanmetenko/Code/R_projects/SquarePacking/pySquares.ipynb Cell 6' in
<cell line: 35>()
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=30'>31</a> I[where_not0] = -1
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=32'>33</a> zip_populate = zip([4,3, 1,
 4, 1, 4] * 5 + [1,1,1] * 14, [3, 4, 2, 3, 2, 1] * 5 + [3,3,3] * 14)
---> <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=34'>35</a> F = SquareCanvas(
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=35'>36</a>     contents=[
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=36'>37</a>         Rect(length=i, width
=j).rotate(90)
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=37'>38</a>         for i, j
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=38'>39</a>         in zip_populate
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=39'>40</a>         ],
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=40'>41</a>     max_bound = 24,
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=41'>42</a>     allow_rotation=False,
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=42'>43</a>     frame_override=I
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=43'>44</a>     )
     <a href='vscode-notebook-cell:/Users/bohdanmetenko/Code/R_projects/Square
Packing/pySquares.ipynb#ch0000005?line=44'>45</a> F.generate_plotly(show_text=
False)

File ~/Code/R_projects/SquarePacking/squares.py:173, in SquareCanvas.__init__
(self, max_bound, contents, frame_override, validate, allow_rotation)
    170 self.rotation = allow_rotation
    172 for sq in contents:
--> 173     self.add_contents(sq)
    175 if validate:
    176     self.check_all_filled(contents)

File ~/Code/R_projects/SquarePacking/squares.py:211, in SquareCanvas.add_conte
nts(self, sq)
    208                         self.frame[x0][y0] = int(len(self._contents))
    210 if not placed:
--> 211     raise IndexError("Not all placed...")

IndexError: Not all placed...
```

Above error is expected since these rectangles would not fit when there is no rotation allowed.