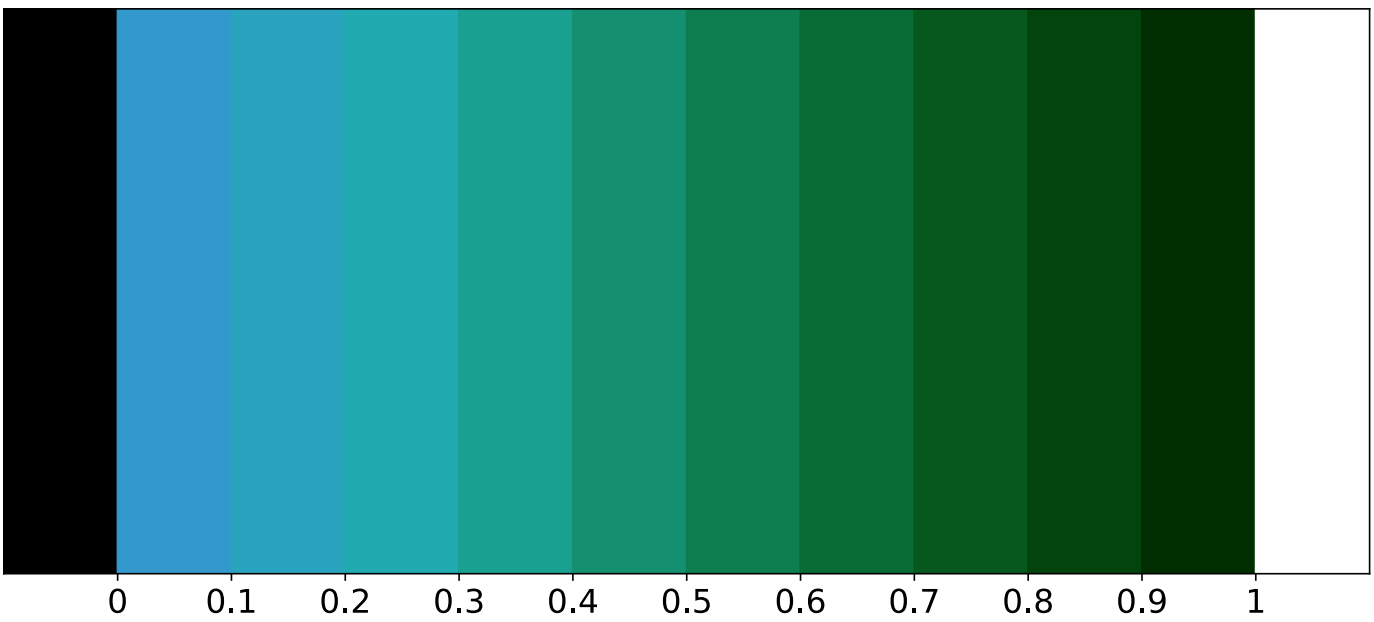
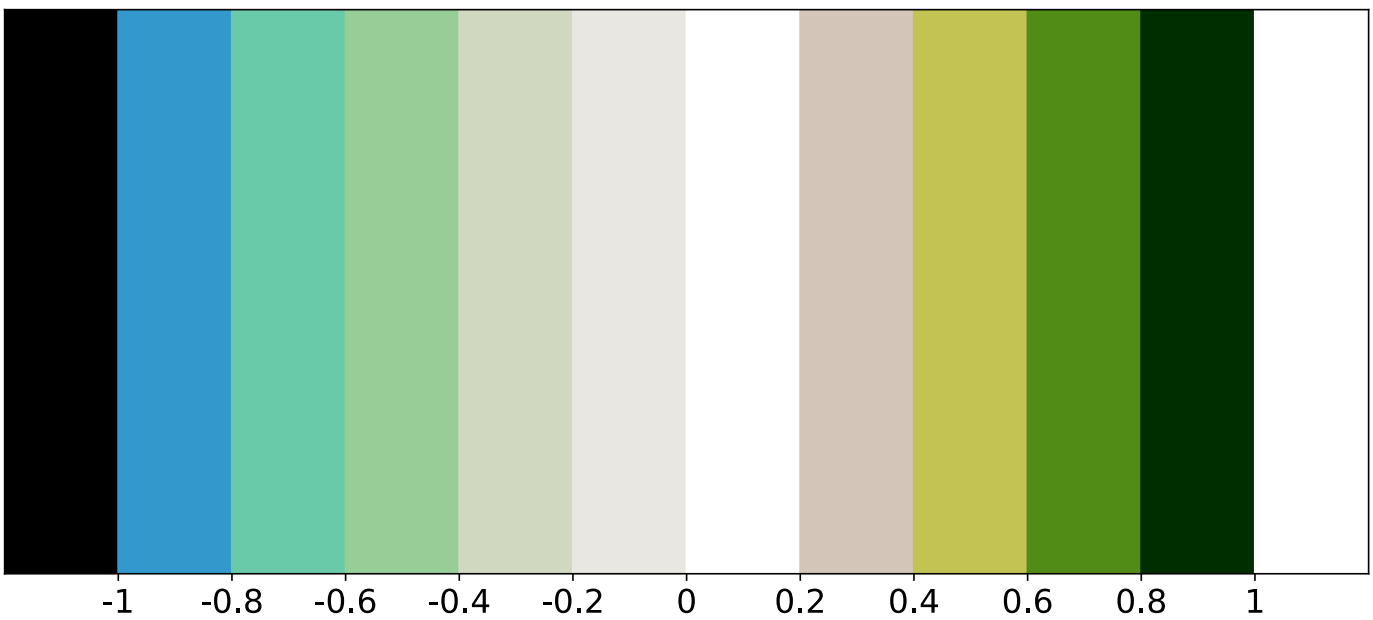
```
Visualisations:
  Example:
    make: 1
    vals: [0,1]
    colours: [[0.2, 0.6, 0.8], '012E01']
    floor:
    ceiling:
    n_steps:
    ramp_flag: 1
    steps_flag: 0
    show: 1
```



An example of creating a colour ramp from the run card input
By default, there are 10 steps in a colour ramp
By default, the floor and ceiling values are black and white respectively
The palette accepts both RGB lists (either 0-1 or 0-255) and hex strings

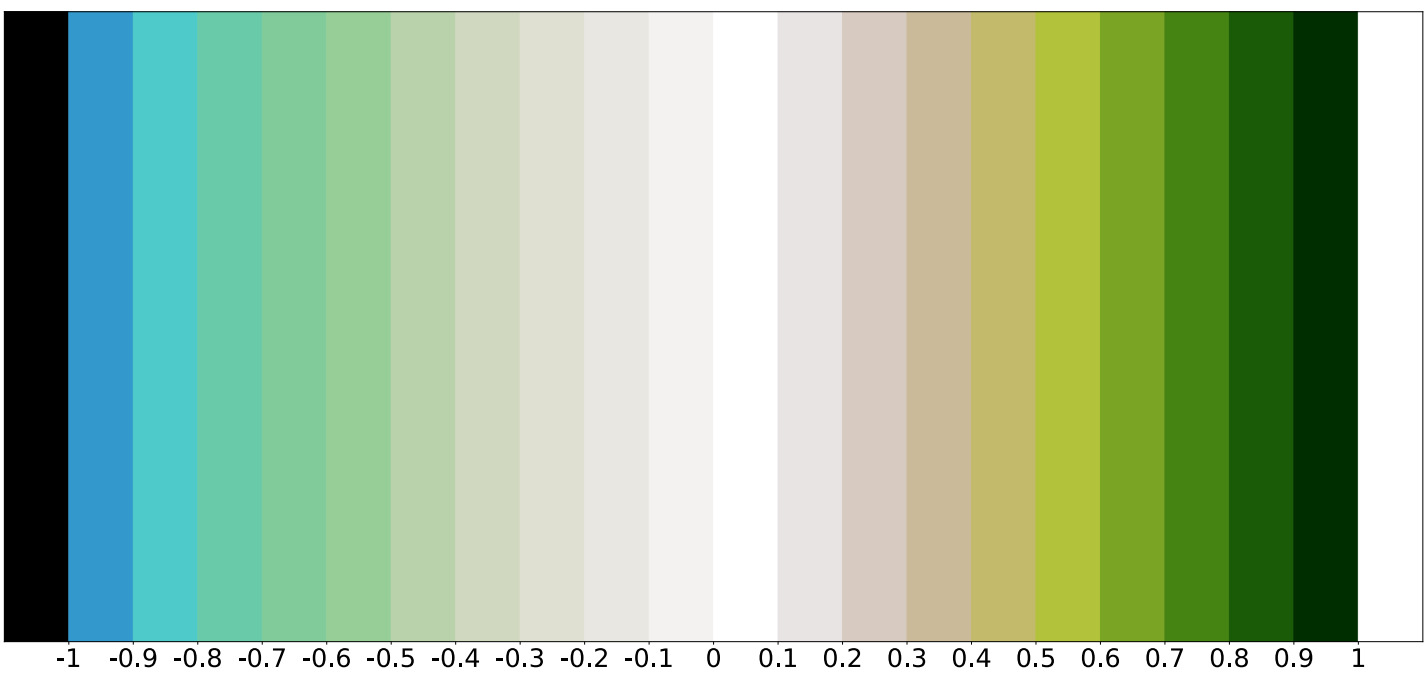
```
Visualisations:
  Example:
    make: 1
    vals: [-1,1]
    colours: [[0.2, 0.6, 0.8], [255, 255,
255], '012E01']
    floor:
    ceiling:
    n_steps:
    ramp_flag: 1
    steps_flag: 0
    show: 1
```



Example of a simple colour ramp visualisation, with three colours

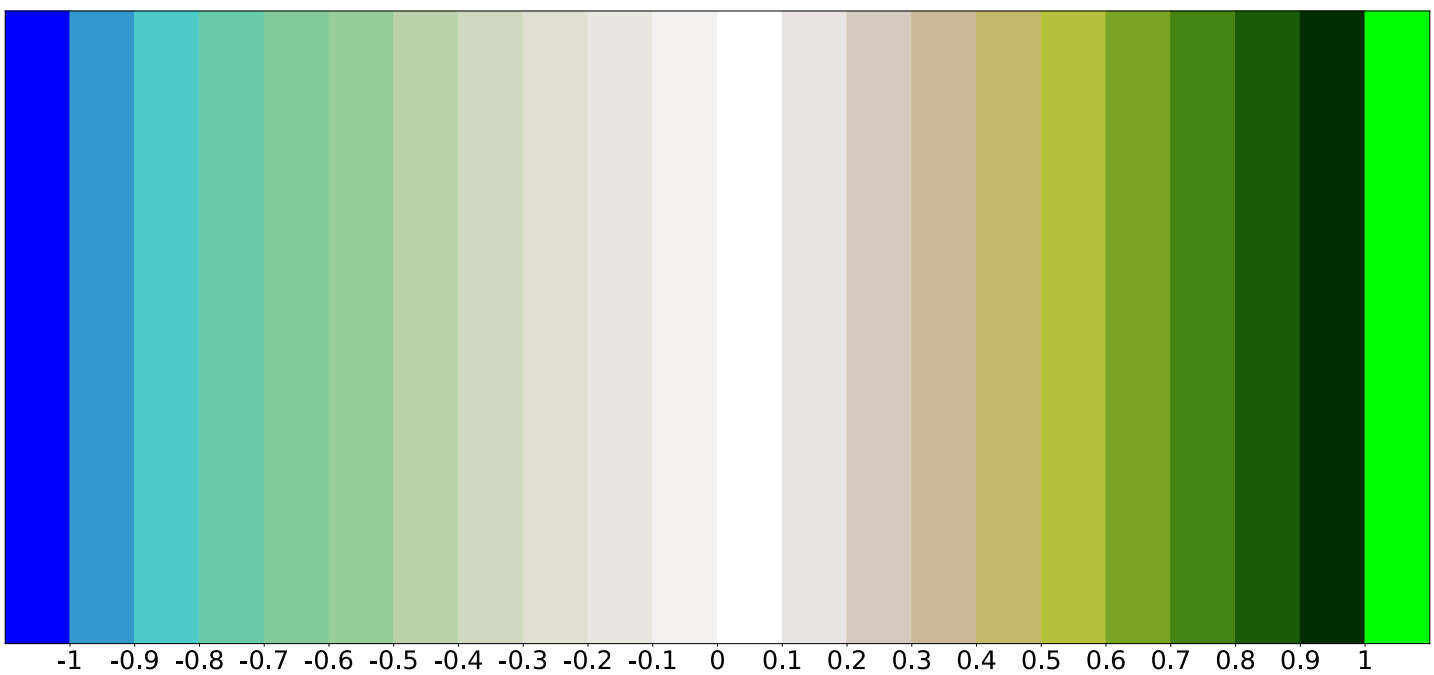
Any number of colours can be passed to the ramp, and a warning will be raised if there is a low number of intermediate steps in a ramp

```
Visualisations:
  Example:
    make: 1
    vals: [-1,1]
    colours: [[0.2, 0.6, 0.8], [255,
255, 255], '012E01']
    floor:
    ceiling:
    n_steps: 20
    ramp_flag: 1
    steps_flag: 0
    show: 1
```



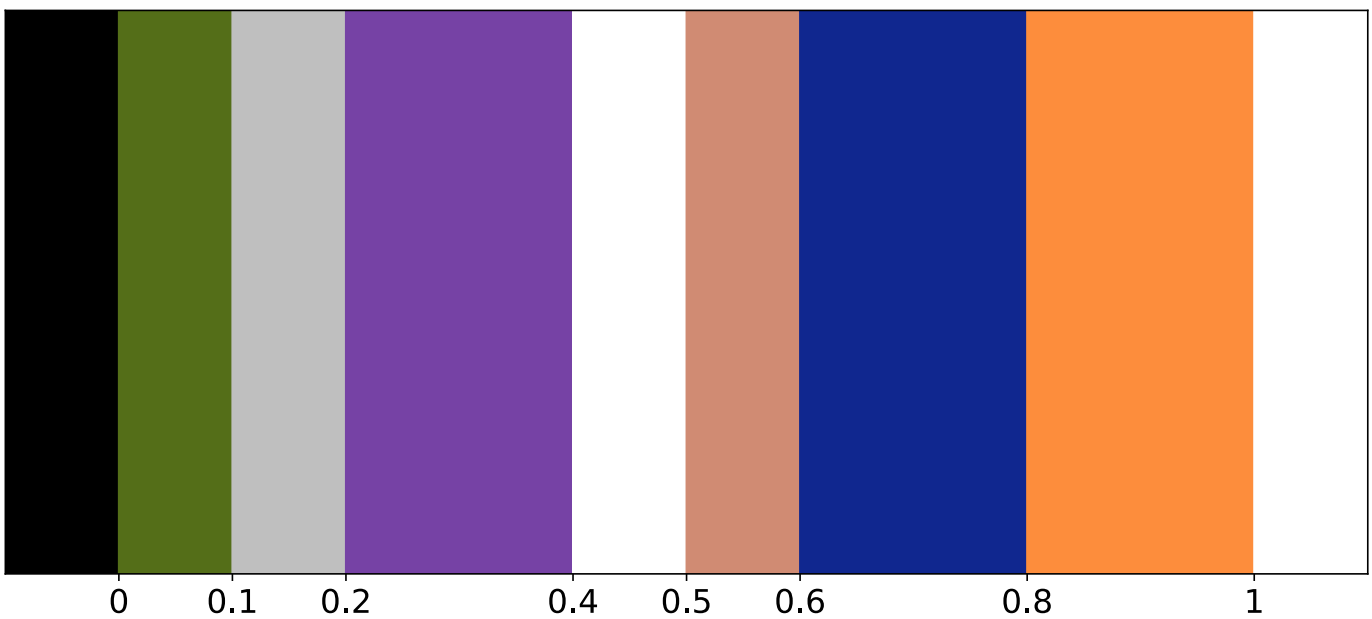
Increasing the number of steps in the ramp naturally makes it smoother, fine graining the palette to a greater extent

```
Visualisations:
  Example:
    make: 1
    vals: [-1,1]
    colours: [[0.2, 0.6, 0.8], [255, 255,
255], '012E01']
    floor: '#00f'
    ceiling: '#0f0'
    n_steps: 20
    ramp_flag: 1
    steps_flag: 0
    show: 1
```



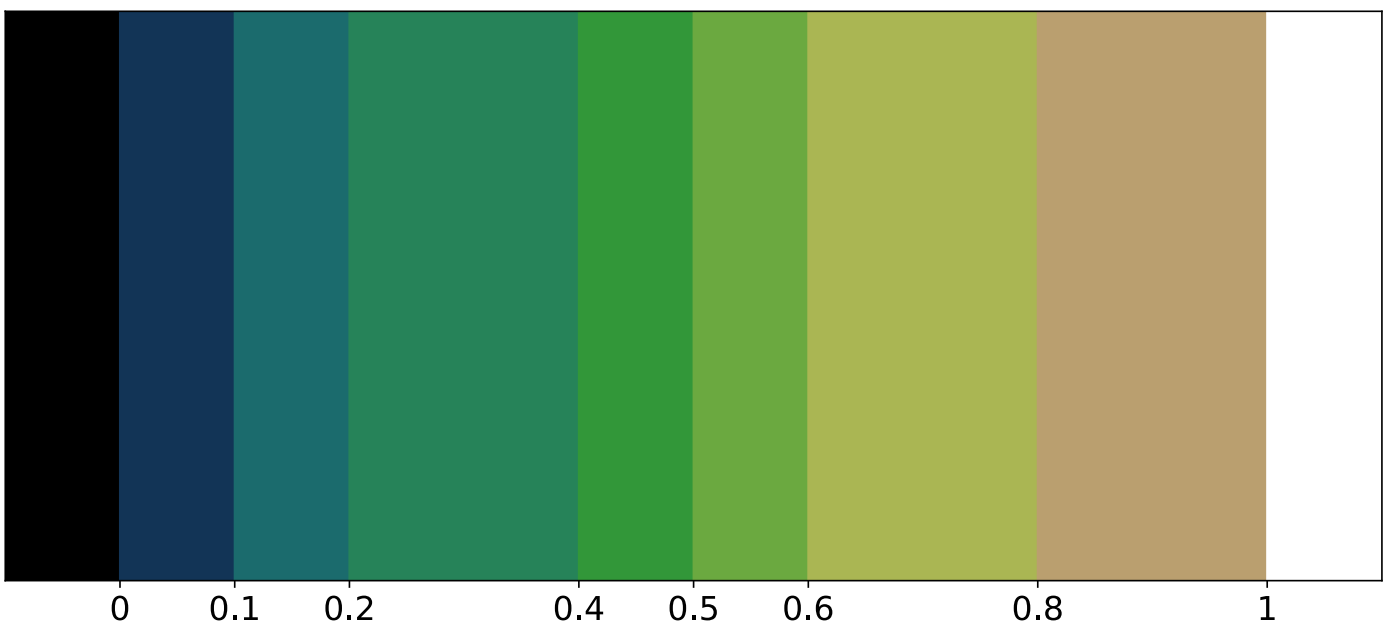
The floor and ceiling colours can be set manually if desired – a pixel with a value below vis_min (here -1) will be coloured pure blue, a pixel with a value above vis_max (here 1) will be coloured pure green

```
Visualisations:
  Example:
    make: 1
    vals: [0, 0.1, 0.2, 0.4, 0.5, 0.6,
0.8, 1]
    colours: ['546e18', 'bfbfbf',
'7642a5', 'fff', 'd08b73', '10278f',
'fd8d3c’]
    floor:
    ceiling:
    n_steps:
    ramp_flag: 0
    steps_flag: 1
    show: 1
```



A series of uneven steps can also be used to generate a visualization that can be fine grained in some numerical range, and coarse grained in others, with each range coloured with a specified colour

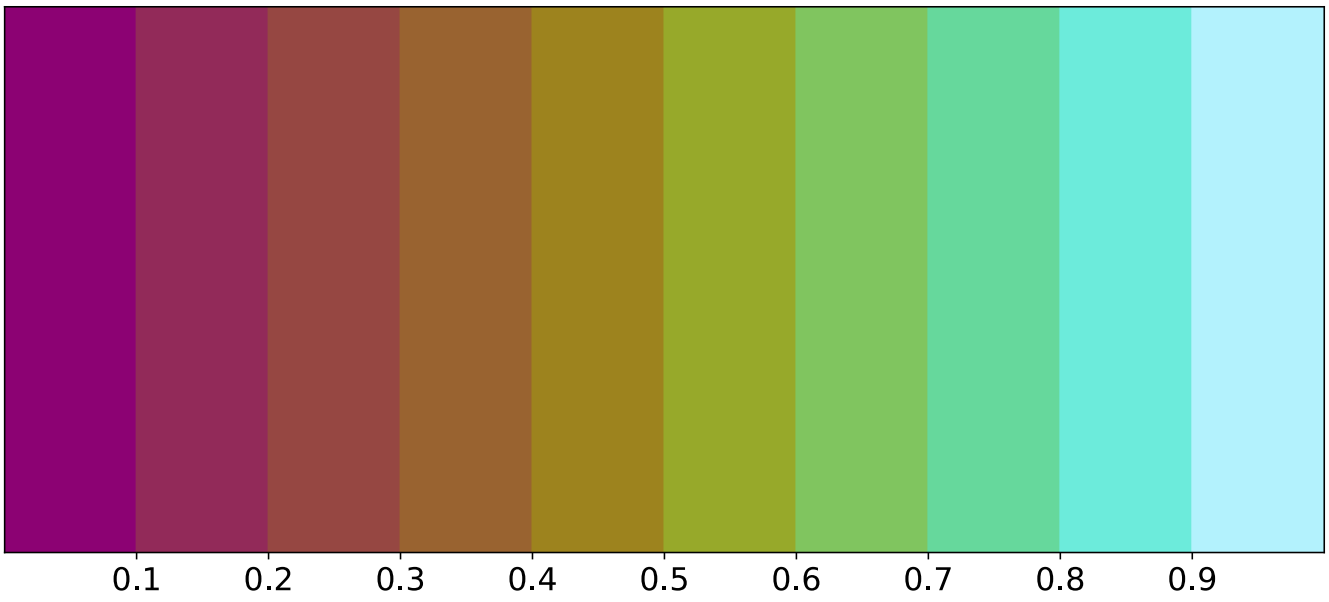
```
Visualisations:
  Example:
    make: 1
    vals: [0, 0.1, 0.2, 0.4, 0.5, 0.6,
0.8, 1]
    colours: ['123456', 'ba9f6f’]
    floor:
    ceiling:
    n_steps:
    ramp_flag: 1
    steps_flag: 1
    show: 1
```



The two methods can be combined, giving a smooth colour ramp over uneven intervals

Again, this can handle arbitrarily many colours

```
Visualisations:
  Example:
    vis_min: 0.1
    vis_max: 0.9
    palette: 'hawaii_10'
    make: 0
    show: 1
```



Pre-defined palettes, stored in the Palettes.yml, can also be loaded in and applied to a given scale

This is equivalent to

```
Visualisations:
  Example:
    make: 1
    vals: [0.1, 0.9]
    colours: 'hawaii_10'
    floor:
    ceiling:
    n_steps:
    ramp_flag: 0
    steps_flag: 0
    show: 1
```