Hillforts Primer

An Analysis of the Atlas of Hillforts of Britain and Ireland

Part 5

Mike Middleton
https://orcid.org/0000-0001-5813-6347

Version 1.0, March 2024.
This research was begun in March 2022.

Part 1: Name, Admin & Location Data

Colab Notebook: Live code (Must be logged into Google. Select Google Colaboratory, at the top of the screen, if page opens as raw code)
HTML: Read only

Part 2: Management & Landscape

Colab Notebook: Live code
HTML: Read only

Part 3: Boundary & Dating

Colab Notebook: Live code
HTML: Read only

Part 4: Investigations & Interior

Colab Notebook: Live code
HTML: Read only

Part 5: Entrance, Enclosing & Annex

Colab Notebook: Live code
HTML: Read only

- Entrance Data
- Enclosing Data
- Annex Data
- Reference Data
- Acknowledgements

Appendix 1: Hypotheses Testing the Alignment of Hillforts with an Area of 21 Hectares or More

Colab Notebook: Live code
HTML: Read only

Appendix 2: Classification Northwest

Colab Notebook: Live code
HTML: Read only

User Settings

Pre-processed data and images are available for download (without the need to run the code in these files) here: https://github.com/MikeDairsie/Hillforts-Primer.

To review only confirmed hillforts (see Part 1: Status, Data Reliability), download, save images or to change the background image to show the topography, first save a copy of this document into your Google Drive folder. Once saved, change, confirmed_only, download_data, save_images and/or show_topography to **True** in the code blocks below, **Save** and then select **Runtime>Run all** in the main menu above to rerun the code. If selected, running the code will initiate the download and saving of files. Each document will download a number of data packages and you may be prompted to **allow** multiple downloads. Be patient, downloads may take a little time after the document has finished running. Note that each part of the Hillforts Primer is independent and the download, save_image and show_topography variables will need to be enabled in each document, if this functionality is required. Also note that saving images will activate the Google Drive folder and this will request the user to **allow** access. Selecting show_topography will change the background image to a colour topographic map. It should also be noted that, if set to True, this view will only show the distribution of the data selected. It will not show the overall distribution as a grey background layer as is seen when using the simple coastal outlines.

In [ ]:
```
confirmed_only = False
```

In [ ]:
```
download_data = False
```

In [ ]:
```
save_images = False
```

In [ ]:
```
show_topography = False
```

Bypass Code Setup

The initial sections of all the Hillforts Primer documents set up the coding environment and define functions used to plot, reprocess and save the data. If you would like to bypass the setup, please use the following link:

Go to Review Data Part 5.

Source Data

The Atlas of Hillforts of Britain and Ireland data is made available under the licence, Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). This allows for redistribution, sharing and transformation of the data, as long as the results are credited and made available under the same licence conditions.

The data was downloaded from The Atlas of Hillforts of Britain and Ireland website as a csv file (comma separated values) and saved onto the author's GitHub repository thus enabling the data to be used by this document.

Lock, G. and Ralston, I. 2017. Atlas of Hillforts of Britain and Ireland. [ONLINE] Available at: https://hillforts.arch.ox.ac.uk
Rest services: https://maps.arch.ox.ac.uk/server/rest/services/hillforts/Atlas_of_Hillforts/MapServer
Licence: https://creativecommons.org/licenses/by-sa/4.0/
Help: https://hillforts.arch.ox.ac.uk/assets/help.pdf
Data Structure: https://maps.arch.ox.ac.uk/assets/data.html
Hillforts: Britain, Ireland and the Nearer Continent (Sample):
https://www.archaeopress.com/ArchaeopressShop/DMS/A72C523E8B6742ED97BA86470E747C69/9781789692266-sample.pdf

Map outlines made with Natural Earth. Free vector and raster map data @ naturalearthdata.com.

Reload Data and Python Functions

This study is split over multiple documents. Each file needs to be configured and have the source data imported. As this section does not focus on the assessment of the data it is minimised to facilitate the documents readability.

Python Modules and Code Setup

The Python imports enable the Hillforts Atlas data to be analysed and mapped within this document. The Python code can be run on demand, (see: User Settings). This means that as new research becomes available, the source for this document can be updated to a revised copy of the Atlas data and the impact of that research can be reviewed using the same code and graphic output. The Hillforts Atlas is a baseline and this document is a tool that can be used to assess the impact new research is making in this area.

In [ ]:
```
import sys
print(f'Python: {sys.version}')

import sklearn
print(f'Scikit-Learn: {sklearn.__version__}')

import pandas as pd
print(f'pandas: {pd.__version__}')

import numpy as np
print(f'numpy: {np.__version__}')

%matplotlib inline
import matplotlib
print(f'matplotlib: {matplotlib.__version__}')
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.patches as mpatches
import matplotlib.patches as patches
from matplotlib.cbook import boxplot_stats
from matplotlib.lines import Line2D
import matplotlib.cm as cm

import seaborn as sns
print(f'seaborn: {sns.__version__}')
sns.set(style="whitegrid")

import scipy
print(f'scipy: {scipy.__version__}')
from scipy import stats
from scipy.stats import gaussian_kde

import os
import collections
import math
import random
import PIL
import urllib
random.seed(42)
# A random seed is used to ensure that the random numbers created are the
#same for each run of this document.

from slugify import slugify

# Import Google colab tools to access Drive
from google.colab import drive
```

```
Python: 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
Scikit-Learn: 1.2.2
pandas: 1.5.3
numpy: 1.25.2
matplotlib: 3.7.1
seaborn: 0.13.1
scipy: 1.11.4
```

Ref: https://www.python.org/

Ref: https://scikit-learn.org/stable/

Ref: https://pandas.pydata.org/docs/

Ref: https://numpy.org/doc/stable/

Ref: https://matplotlib.org/

Ref: https://seaborn.pydata.org/

Ref: https://docs.scipy.org/doc/scipy/index.html

Ref: https://pypi.org/project/python-slugify/

In [ ]:
```
# # Ensure Python is ≥3.7
# import sys
# assert sys.version_info >= (3, 7)
# print(f'Python: {sys.version}')

# # Ensure Scikit-Learn is ≥1.0.2
# import sklearn
# assert sklearn.__version__ >= "1.0.2"
# print(f'Scikit-Learn: {sklearn.__version__}')

# # Ensure Pandas is ≥1.3.5
# import pandas as pd
# assert pd.__version__ >= "1.3.5"
# print(f'pandas: {pd.__version__}')
```

```
# # Ensure Numpy is ≥1.21.6
# import numpy as np
# assert np.__version__ >= "1.21.6"
# print(f'numpy: {np.__version__}')

# # Ensure matplotlib is ≥3.2.2
# %matplotlib inline
# import matplotlib
# assert matplotlib.__version__ >= "3.2.2"
# print(f'matplotlib: {matplotlib.__version__}')
# import matplotlib.pyplot as plt
# import matplotlib.cm as cm
# import matplotlib.patches as mpatches
# from matplotlib.cbook import boxplot_stats
# from matplotlib.lines import Line2D

# # Ensure Seaborn is ≥0.11.2
# import seaborn as sns
# assert sns.__version__ >= "0.11.2"
# print(f'seaborn: {sns.__version__}')
# sns.set(style="whitegrid")

# # Ensure Scipy is ≥1.4.1
# import scipy
# assert scipy.__version__ >= "1.4.1"
# print(f'scipy: {scipy.__version__}')
# from scipy import stats
# from scipy.stats import gaussian_kde

# # Import Python libraries
# import os
# import collections
# from slugify import slugify

# # Import Google colab tools to access Drive
# from google.colab import drive
```

Plot Figures and Maps functions

The following functions will be used to plot data later in the document.

In [ ]:
```
def show_records(plt, plot_data):
    text_colour = 'k'
    if show_topography == True:
        text_colour = 'w'
    plt.annotate(str(len(plot_data))+' records', xy=(-1180000, 6420000), \
                 xycoords='data', ha='left', color=text_colour)
```

In [ ]:
```
def get_backgrounds():
    if show_topography == True:
        backgrounds = ["hillforts-topo-01.png",
                    "hillforts-topo-north.png",
                    "hillforts-topo-northwest-plus.png",
                    "hillforts-topo-northwest-minus.png",
                    "hillforts-topo-northeast.png",
                    "hillforts-topo-south.png",
                    "hillforts-topo-south-plus.png",
                    "hillforts-topo-ireland.png",
                    "hillforts-topo-ireland-north.png",
                    "hillforts-topo-ireland-south.png"]
    else:
        backgrounds = ["hillforts-outline-01.png",
                    "hillforts-outline-north.png",
                    "hillforts-outline-northwest-plus.png",
                    "hillforts-outline-northwest-minus.png",
                    "hillforts-outline-northeast.png",
                    "hillforts-outline-south.png",
                    "hillforts-outline-south-plus.png",
                    "hillforts-outline-ireland.png",
                    "hillforts-outline-ireland-north.png",
                    "hillforts-outline-ireland-south.png"]
    return backgrounds
```

In [ ]:
```
def get_bounds():
    bounds = [[-1200000,220000,6400000,8700000],
    [-1200000,220000,7000000,8700000],
    [-1200000,-480000,7000000,8200000],
    [-900000,-480000,7100000,8200000],
    [-520000, 0,7000000,8700000],
    [-800000,220000,6400000,7100000],
    [-1200000,220000,6400000,7100000],
    [-1200000,-600000,6650000,7450000],
    [-1200000,-600000,7050000,7450000],
```

```
    [-1200000,-600000,6650000,7080000]]
    return bounds
```

In [ ]:
```
def show_background(plt, ax, location=""):
    backgrounds = get_backgrounds()
    bounds = get_bounds()
    folder = "https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/hillforts-topo/"
    #"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/
    #hillforts-topo/"

    if location == "n":
        background = os.path.join(folder, backgrounds[1])
        bounds = bounds[1]
    elif location == "nw+":
        background = os.path.join(folder, backgrounds[2])
        bounds = bounds[2]
    elif location == "nw-":
        background = os.path.join(folder, backgrounds[3])
        bounds = bounds[3]
    elif location == "ne":
        background = os.path.join(folder, backgrounds[4])
        bounds = bounds[4]
    elif location == "s":
        background = os.path.join(folder, backgrounds[5])
        bounds = bounds[5]
    elif location == "s+":
        background = os.path.join(folder, backgrounds[6])
        bounds = bounds[6]
    elif location == "i":
        background = os.path.join(folder, backgrounds[7])
        bounds = bounds[7]
    elif location == "in":
        background = os.path.join(folder, backgrounds[8])
        bounds = bounds[8]
    elif location == "is":
        background = os.path.join(folder, backgrounds[9])
        bounds = bounds[9]
    else:
        background = os.path.join(folder, backgrounds[0])
        bounds = bounds[0]

    img = np.array(PIL.Image.open(urllib.request.urlopen(background)))
    ax.imshow(img, extent=bounds)
```

In [ ]:
```
def get_counts(data):
    data_counts = []
    for col in data.columns:
        count = len(data[data[col] == 'Yes'])
        data_counts.append(count)
    return data_counts
```

In [ ]:
```
def add_annotation_plot(ax):
    ax.annotate("Middleton, M. 2024, Hillforts Primer", size='small', \
                color='grey', xy=(0.01, 0.01), xycoords='figure fraction', \
                horizontalalignment = 'left')
    ax.annotate("Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk", \
                size='small', color='grey', xy=(0.99, 0.01), \
                xycoords='figure fraction', horizontalalignment = 'right')
```

In [ ]:
```
def add_annotation_l_xy(ax):
    ax.annotate("Middleton, M. 2024, Hillforts Primer", size='small', \
                color='grey', xy=(0.01, 0.035), xycoords='figure fraction', \
                horizontalalignment = 'left')
    ax.annotate("Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk", \
                size='small', color='grey', xy=(0.99, 0.035), \
                xycoords='figure fraction', horizontalalignment = 'right')
```

In [ ]:
```
def plot_bar_chart(data, split_pos, x_label, y_label, title, clip=False):
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    x_data = data.columns
    x_data = [x.split("_")[split_pos:] for x in x_data]
    x_data_new = []
    for l in x_data :
        txt =  ""
        for part in l:
            txt += "_" + part
        x_data_new.append(txt[1:])
    if clip:
        x_data_new = x_data_new[:-1]
        new_data = data.copy()
        data = new_data.drop(['Dating_Date_Unknown'], axis=1)
    y_data = get_counts(data)
    ax.bar(x_data_new,y_data)
```

```
    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
    add_annotation_plot(ax)
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
```

In [ ]:
```
def plot_bar_chart_using_two_tables(x_data, y_data, x_label, y_label, title):
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    ax.bar(x_data,y_data)
    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
    add_annotation_plot(ax)
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
```

In [ ]:
```
def plot_bar_chart_numeric(data, split_pos, x_label, y_label, title, \
                           n_bins, extra=''):
    new_data = data.copy()
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    data[x_label].plot(kind='hist', bins = n_bins)
    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
    add_annotation_plot(ax)
    title = f'{title} {extra}'
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
```

In [ ]:
```
def get_bins(data, bins_count):
    data_range = data.max() - data.min()
    print(bins_count)
    if bins_count != None:
        x_bins = [x for x in range(data.min(), data.max(), bins_count)]
        n_bins = len(x_bins)
    else:
        n_bins = int(data_range)
        if n_bins < 10:
            multi = 10
            while n_bins< 10:
                multi *= 10
                n_bins = int(data_range * multi)
        elif n_bins > 100:
            n_bins = int(data_range)/10

    return n_bins
```

In [ ]:
```
def plot_histogram(data, x_label, title, bins_count = None):
    n_bins = get_bins(data, bins_count)
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    ax.set_xlabel(x_label)
    ax.set_ylabel('Count')
    plt.ticklabel_format(style='plain')
    plt.hist(data, bins=n_bins)
    plt.title(get_print_title(title))
    add_annotation_plot(ax)
    save_fig(title)
    plt.show()
```

In [ ]:
```
def plot_continuous(data, x_lable, title):
    fig = plt.figure(figsize=(12,8))
    ax = fig.add_axes([0,0,1,1])
    ax.set_xlabel(x_lable)
    plt.plot(data, linewidth=4)
    plt.ticklabel_format(style='plain')
    plt.title(get_print_title(title))
    add_annotation_plot(ax)
    save_fig(title)
    plt.show()
```

In [ ]:
```
def plot_data_range(data, feature, o="v"):
    fig = plt.figure(figsize=(12,8))
    ax = fig.add_axes([0,0,1,1])
    ax.set_xlabel(feature)
    add_annotation_plot(ax)
    plt.title(get_print_title(feature + " Range"))
    plt.ticklabel_format(style='plain')
    if o == "v":
        sns.boxplot(data=data, orient="v", whis=[2.2, 97.8])
```

```
    else:
        sns.boxplot(data=data, orient="h", whis=[2.2, 97.8])
    save_fig(feature + " Range")
    plt.show()

    bp = boxplot_stats(data, whis=[2.2, 97.8])

    low = bp[0].get('whislo')
    q1 = bp[0].get('q1')
    median =  bp[0].get('med')
    q3 = bp[0].get('q3')
    high = bp[0].get('whishi')

    return [low, q1, median, q3, high]
```

In [ ]:
```
def plot_data_range_plus(data, feature, o="v"):
    fig = plt.figure(figsize=(12,8))
    ax = fig.add_axes([0,0,1,1])
    ax.set_xlabel(feature)
    add_annotation_plot(ax)
    plt.title(get_print_title(feature + " Range (Outlier Steps)"))
    plt.ticklabel_format(style='plain')
    if o == "v":
        sns.boxplot(data=data, orient="v", whis=[2.2, 97.8])
    else:
        sns.boxplot(data=data, orient="h", whis=[2.2, 97.8])

    # Add annotation lines
    x = [24, 24, 54, 54]
    y = [-0.05, -0.075, -0.075, -0.05]
    x1 = [54, 54, 84, 84]
    y1 = [-0.1, -0.125, -0.125, -0.1]
    x2 = [84, 84, 114, 114]
    y2 = [-0.05, -0.075, -0.075, -0.05]

    line_1 = plt.plot(x,y)
    line_2 = plt.plot(x1,y1)
    line_3 = plt.plot(x2,y2)

    # Add annotation text
    text_kwargs = dict(ha='center', va='center', fontsize=16, color='k')
    plt.text(39, -0.1, '30 Ha', **text_kwargs)
    plt.text(69, -0.1, '30 Ha', **text_kwargs)
    plt.text(99, -0.1, '30 Ha', **text_kwargs)

    save_fig(feature + " Range")
    plt.show()

    return
```

In [ ]:
```
def location_XY_plot():
    plt.ticklabel_format(style='plain')
    plt.xlim(-1200000,220000)
    plt.ylim(6400000,8700000)
    add_annotation_l_xy(plt)
```

In [ ]:
```
def add_grey(region=''):
    if show_topography == False:
        # plots all the hillforts as a grey background
        loc = location_data.copy()
        if region == 's':
            loc = loc[loc['Location_Y'] < 8000000].copy()
            loc = loc[loc['Location_X'] > -710000].copy()
        elif region == 'ne':
            loc = loc[loc['Location_Y'] < 8000000].copy()
            loc = loc[loc['Location_X'] > -800000].copy()

        plt.scatter(loc['Location_X'], loc['Location_Y'], c='Silver')
```

In [ ]:
```
def plot_over_grey_numeric(merged_data, a_type, title, extra="", \
                           inner=False, fringe=False, \
                           oxford=False, swindon=False):
    plot_data = merged_data
    fig, ax = plt.subplots(figsize=(14.2 * 0.66, 23.0 * 0.66))
    show_background(plt, ax)
    location_XY_plot()
    add_grey()
    patches = add_oxford_swindon(oxford,swindon)
    plt.scatter(plot_data['Location_X'], plot_data['Location_Y'], c='Red')
    if fringe:
        f_for_legend = add_21Ha_fringe()
        patches.append(f_for_legend)
    if inner:
        i_for_legend = add_21Ha_line()
```

```
        patches.append(i_for_legend)
    show_records(plt, plot_data)
    plt.legend(loc='upper left', handles= patches)
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
    print(f'{round(((len(plot_data)/4147)*100), 2)}%')
```

In [ ]:
```
def plot_over_grey_boundary(merged_data, a_type, boundary_type):
    plot_data = merged_data[merged_data[a_type] == boundary_type]
    fig, ax = plt.subplots(figsize=(9.47, 15.33))
    show_background(plt, ax)
    location_XY_plot()
    add_grey(region='')
    plt.scatter(plot_data['Location_X'], plot_data['Location_Y'], c='Red')
    show_records(plt, plot_data)
    plt.title(get_print_title('Boundary_Type: ' + boundary_type))
    save_fig('Boundary_Type_' + boundary_type)
    plt.show()
```

In [ ]:
```
def add_21Ha_line():
    x_values = \
     [-367969, -344171, -263690, -194654, -130542, -119597, -162994, -265052]#, -304545]
    y_values = \
     [7019842, 6944572, 6850593, 6779602, 6735058, 6710127, 6684152, 6663609]#, 6611780]
    plt.plot(x_values, y_values, 'k', ls='-', lw=15, alpha=0.25, label = \
             '≥ 21 Ha Line')
    add_to_legend = Line2D([0], [0], color='k', lw=15, alpha=0.25, label = \
                           '≥ 21 Ha Line')
    return add_to_legend
```

In [ ]:
```
def add_21Ha_fringe():
    x_values = \
     [-367969,-126771,29679,-42657,-248650,-304545,-423647,-584307,-367969]
    y_values = \
     [7019842,6847138,6671658,6596650,6554366,6611780,6662041,6752378,7019842]
    plt.plot(x_values, y_values, 'k', ls=':', lw=5, alpha=0.45, label = \
             '≥ 21 Ha Fringe')
    add_to_legend = Line2D([0], [0], color='k', ls=':', lw=5, alpha=0.45, \
                           label = '≥ 21 Ha Fringe')
    return add_to_legend
```

In [ ]:
```
def plot_density_over_grey(data, data_type, extra='', inner=False, fringe=False):
    new_data = data.copy()
    new_data = new_data.drop(['Density'], axis=1)
    new_data = add_density(new_data)
    fig, ax = plt.subplots(figsize=((14.2 * 0.66)+2.4, 23.0 * 0.66))
    show_background(plt, ax)
    location_XY_plot()
    add_grey()
    plt.scatter(new_data['Location_X'], new_data['Location_Y'], \
                c=new_data['Density'], cmap=cm.rainbow, s=25)
    if fringe:
        add_21Ha_fringe()
    if inner:
        add_21Ha_line()
        plt.legend(loc='lower left')
    plt.colorbar(label='Density')
    title = f'Density - {data_type} {extra}'
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
```

In [ ]:
```
def plot_density_over_grey_three(data_low, data_iqr, data_high, title, \
                                 extra='', inner=False, fringe=False):
    new_data_low = data_low.copy()
    new_data_low = new_data_low.drop(['Density'], axis=1)
    new_data_low = add_density(new_data_low)

    new_data_iqr = data_iqr.copy()
    new_data_iqr = new_data_iqr.drop(['Density'], axis=1)
    new_data_iqr = add_density(new_data_iqr)

    new_data_high = data_high.copy()
    new_data_high = new_data_high.drop(['Density'], axis=1)
    new_data_high = add_density(new_data_high)

    fig, ax = plt.subplots(1, 3)
    fig.set_figheight(7)
    fig.set_figwidth(15)

    bounds = get_bounds()
    folder = "https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/hillforts-topo/"
    #"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/
```

```
    #hillforts-topo/"
    background = os.path.join(folder, "hillforts-bw-02.png")
    bounds = bounds[0]
    img = np.array(PIL.Image.open(urllib.request.urlopen(background)))
    ax[0].imshow(img, extent=bounds)
    ax[1].imshow(img, extent=bounds)
    ax[2].imshow(img, extent=bounds)

    ax[0].scatter(new_data_low['Location_X'], new_data_low['Location_Y'],\
                  c=new_data_low['Density'], cmap=cm.rainbow, s=25)
    ax[1].scatter(new_data_iqr['Location_X'], new_data_iqr['Location_Y'], \
                  c=new_data_iqr['Density'], cmap=cm.rainbow, s=25)
    ax[2].scatter(new_data_high['Location_X'], new_data_high['Location_Y'], \
                  c=new_data_high['Density'], cmap=cm.rainbow, s=25)

    ax[0].get_yaxis().set_visible(False)
    ax[1].get_yaxis().set_visible(False)
    ax[2].get_yaxis().set_visible(False)

    ax[0].get_xaxis().set_visible(False)
    ax[1].get_xaxis().set_visible(False)
    ax[2].get_xaxis().set_visible(False)

    ax[0].set_title("1st Quarter (Tiny Hillforts)")
    ax[1].set_title("IQR (Small to Medium Hillforts)")
    ax[2].set_title("4th Quarter (Large Hillforts)")

    fig.suptitle(get_print_title(title), y=1.08)
    ax[0].annotate("Middleton, M. 2024, Hillforts Primer", size='small', \
                   color='grey', xy=(0, -0.1), xycoords='axes fraction', \
                   horizontalalignment = 'left')
    ax[2].annotate("Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk", \
                   size='small', color='grey', xy=(1, -0.1), \
                   xycoords='axes fraction', horizontalalignment = 'right')
    save_fig(title)
    plt.show()
```

In [ ]:

```
def plot_density_over_grey_four(data_1, data_2, data_3, data_4, \
                                title, extra='', inner=False, fringe=False):
    new_data_1 = data_1.copy()
    new_data_1 = new_data_1.drop(['Density'], axis=1)
    new_data_1 = add_density(new_data_1)

    new_data_2 = data_2.copy()
    new_data_2 = new_data_2.drop(['Density'], axis=1)
    new_data_2 = add_density(new_data_2)

    new_data_3 = data_3.copy()
    new_data_3 = new_data_3.drop(['Density'], axis=1)
    new_data_3 = add_density(new_data_3)

    new_data_4 = data_4.copy()
    new_data_4 = new_data_4.drop(['Density'], axis=1)
    new_data_4 = add_density(new_data_4)

    fig, ax = plt.subplots(1, 4)
    fig.set_figheight(7)
    fig.set_figwidth(20)

    bounds = get_bounds()
    folder = "https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/hillforts-topo/"
    #"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/
    #hillforts-topo/"
    background = os.path.join(folder, "hillforts-bw-02.png")
    bounds = bounds[0]
    img = np.array(PIL.Image.open(urllib.request.urlopen(background)))
    ax[0].imshow(img, extent=bounds)
    ax[1].imshow(img, extent=bounds)
    ax[2].imshow(img, extent=bounds)
    ax[3].imshow(img, extent=bounds)

    ax[0].scatter(new_data_1['Location_X'], new_data_1['Location_Y'], \
                  c=new_data_1['Density'], cmap=cm.rainbow, s=25)
    ax[1].scatter(new_data_2['Location_X'], new_data_2['Location_Y'], \
                  c=new_data_2['Density'], cmap=cm.rainbow, s=25)
    ax[2].scatter(new_data_3['Location_X'], new_data_3['Location_Y'], \
                  c=new_data_3['Density'], cmap=cm.rainbow, s=25)
    ax[3].scatter(new_data_4['Location_X'], new_data_4['Location_Y'], \
                  c=new_data_4['Density'], cmap=cm.rainbow, s=25)

    ax[0].get_yaxis().set_visible(False)
    ax[1].get_yaxis().set_visible(False)
    ax[2].get_yaxis().set_visible(False)
    ax[3].get_yaxis().set_visible(False)
```

```python
    ax[0].get_xaxis().set_visible(False)
    ax[1].get_xaxis().set_visible(False)
    ax[2].get_xaxis().set_visible(False)
    ax[3].get_xaxis().set_visible(False)

    ax[0].set_title("NE")
    ax[1].set_title("SE")
    ax[2].set_title("SW")
    ax[3].set_title("NW")

    fig.suptitle(get_print_title(title), y=1.08)
    ax[0].annotate("Middleton, M. 2024, Hillforts Primer", size='small', \
                   color='grey', xy=(0, -0.1), xycoords='axes fraction', \
                   horizontalalignment = 'left')
    ax[3].annotate("Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk", \
                   size='small', color='grey', xy=(1, -0.1), \
                   xycoords='axes fraction', horizontalalignment = 'right')
    save_fig(title)
    plt.show()
```

In [ ]:

```python
def plot_density_over_grey_five(data_1, data_2, data_3, data_4, data_5, title, \
                                extra='', inner=False, fringe=False):
    new_data_1 = data_1.copy()
    new_data_1 = new_data_1.drop(['Density'], axis=1)
    new_data_1 = add_density(new_data_1)

    new_data_2 = data_2.copy()
    new_data_2 = new_data_2.drop(['Density'], axis=1)
    new_data_2 = add_density(new_data_2)

    new_data_3 = data_3.copy()
    new_data_3 = new_data_3.drop(['Density'], axis=1)
    new_data_3 = add_density(new_data_3)

    new_data_4 = data_4.copy()
    new_data_4 = new_data_4.drop(['Density'], axis=1)
    new_data_4 = add_density(new_data_4)

    new_data_5 = data_5.copy()
    new_data_5 = new_data_5.drop(['Density'], axis=1)
    new_data_5 = add_density(new_data_5)

    fig, ax = plt.subplots(1, 5)
    fig.set_figheight(7)
    fig.set_figwidth(24)

    bounds = get_bounds()
    folder = "https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/hillforts-topo/"
    #"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/
    #hillforts-topo/"
    background = os.path.join(folder, "hillforts-bw-02.png")
    bounds = bounds[0]
    img = np.array(PIL.Image.open(urllib.request.urlopen(background)))
    ax[0].imshow(img, extent=bounds)
    ax[1].imshow(img, extent=bounds)
    ax[2].imshow(img, extent=bounds)
    ax[3].imshow(img, extent=bounds)
    ax[4].imshow(img, extent=bounds)

    ax[0].scatter(new_data_1['Location_X'], new_data_1['Location_Y'], \
                  c=new_data_1['Density'], cmap=cm.rainbow, s=25)
    ax[1].scatter(new_data_2['Location_X'], new_data_2['Location_Y'], \
                  c=new_data_2['Density'], cmap=cm.rainbow, s=25)
    ax[2].scatter(new_data_3['Location_X'], new_data_3['Location_Y'], \
                  c=new_data_3['Density'], cmap=cm.rainbow, s=25)
    ax[3].scatter(new_data_4['Location_X'], new_data_4['Location_Y'], \
                  c=new_data_4['Density'], cmap=cm.rainbow, s=25)
    ax[4].scatter(new_data_5['Location_X'], new_data_5['Location_Y'], \
                  c=new_data_5['Density'], cmap=cm.rainbow, s=25)

    ax[0].get_yaxis().set_visible(False)
    ax[1].get_yaxis().set_visible(False)
    ax[2].get_yaxis().set_visible(False)
    ax[3].get_yaxis().set_visible(False)
    ax[4].get_yaxis().set_visible(False)

    ax[0].get_xaxis().set_visible(False)
    ax[1].get_xaxis().set_visible(False)
    ax[2].get_xaxis().set_visible(False)
    ax[3].get_xaxis().set_visible(False)
    ax[4].get_xaxis().set_visible(False)

    ax[0].set_title("0")
    ax[1].set_title("1")
    ax[2].set_title("2")
```

```python
    ax[3].set_title("3")
    ax[4].set_title("4")

    fig.suptitle(get_print_title(title), y=1.08)
    ax[0].annotate("Middleton, M. 2024, Hillforts Primer", size='small', \
                   color='grey', xy=(0, -0.1), xycoords='axes fraction', \
                   horizontalalignment = 'left')
    ax[4].annotate("Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk", \
                   size='small', color='grey', xy=(1, -0.1), \
                   xycoords='axes fraction', horizontalalignment = 'right')
    save_fig(title)
    plt.show()
```

In [ ]:

```python
def plot_density_over_grey_six(data_1, data_2, data_3, data_4, data_5, data_6, \
                               title, extra='', inner=False, fringe=False):
    new_data_1 = data_1.copy()
    new_data_1 = new_data_1.drop(['Density'], axis=1)
    new_data_1 = add_density(new_data_1)

    new_data_2 = data_2.copy()
    new_data_2 = new_data_2.drop(['Density'], axis=1)
    new_data_2 = add_density(new_data_2)

    new_data_3 = data_3.copy()
    new_data_3 = new_data_3.drop(['Density'], axis=1)
    new_data_3 = add_density(new_data_3)

    new_data_4 = data_4.copy()
    new_data_4 = new_data_4.drop(['Density'], axis=1)
    new_data_4 = add_density(new_data_4)

    new_data_5 = data_5.copy()
    new_data_5 = new_data_5.drop(['Density'], axis=1)
    new_data_5 = add_density(new_data_5)

    new_data_6 = data_6.copy()
    new_data_6 = new_data_6.drop(['Density'], axis=1)
    new_data_6 = add_density(new_data_6)

    fig, ax = plt.subplots(1, 6)
    fig.set_figheight(6)
    fig.set_figwidth(24)

    bounds = get_bounds()
    folder = "https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/hillforts-topo/"
    #"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/
    #hillforts-topo/"
    background = os.path.join(folder, "hillforts-bw-02.png")
    bounds = bounds[0]
    img = np.array(PIL.Image.open(urllib.request.urlopen(background)))
    ax[0].imshow(img, extent=bounds)
    ax[1].imshow(img, extent=bounds)
    ax[2].imshow(img, extent=bounds)
    ax[3].imshow(img, extent=bounds)
    ax[4].imshow(img, extent=bounds)
    ax[5].imshow(img, extent=bounds)

    ax[0].scatter(new_data_1['Location_X'], new_data_1['Location_Y'], \
                  c=new_data_1['Density'], cmap=cm.rainbow, s=25)
    ax[1].scatter(new_data_2['Location_X'], new_data_2['Location_Y'], \
                  c=new_data_2['Density'], cmap=cm.rainbow, s=25)
    ax[2].scatter(new_data_3['Location_X'], new_data_3['Location_Y'], \
                  c=new_data_3['Density'], cmap=cm.rainbow, s=25)
    ax[3].scatter(new_data_4['Location_X'], new_data_4['Location_Y'], \
                  c=new_data_4['Density'], cmap=cm.rainbow, s=25)
    ax[4].scatter(new_data_5['Location_X'], new_data_5['Location_Y'], \
                  c=new_data_5['Density'], cmap=cm.rainbow, s=25)
    ax[5].scatter(new_data_5['Location_X'], new_data_5['Location_Y'], \
                  c=new_data_5['Density'], cmap=cm.rainbow, s=25)

    ax[0].get_yaxis().set_visible(False)
    ax[1].get_yaxis().set_visible(False)
    ax[2].get_yaxis().set_visible(False)
    ax[3].get_yaxis().set_visible(False)
    ax[4].get_yaxis().set_visible(False)
    ax[5].get_yaxis().set_visible(False)

    ax[0].get_xaxis().set_visible(False)
    ax[1].get_xaxis().set_visible(False)
    ax[2].get_xaxis().set_visible(False)
    ax[3].get_xaxis().set_visible(False)
    ax[4].get_xaxis().set_visible(False)
    ax[5].get_xaxis().set_visible(False)

    ax[0].set_title("Part Univallate")
```

```
    ax[1].set_title("Univallate")
    ax[2].set_title("Part Bivallate")
    ax[3].set_title("Bivallate")
    ax[4].set_title("Part Multivallate")
    ax[5].set_title("Multivallate")

    fig.suptitle(get_print_title(title), y=1.08)
    ax[0].annotate("Middleton, M. 2024, Hillforts Primer", size='small', \
                   color='grey', xy=(0, -0.1), xycoords='axes fraction', \
                   horizontalalignment = 'left')
    ax[5].annotate("Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk", \
                   size='small', color='grey', xy=(1, -0.1), \
                   xycoords='axes fraction', horizontalalignment = 'right')
    save_fig(title)
    plt.show()
```

In [ ]:
```
def add_oxford_swindon(oxford=False,swindon=False):
    # plots a circle over Swindon & Oxford
    radius = 50
    marker_size = (2*radius)**2
    patches = []
    if oxford:
        plt.scatter(-144362,6758380, c='dodgerblue', s=marker_size, alpha=0.50)
        b_patch = mpatches.Patch(color='dodgerblue', label='Oxford orbit')
        patches.append(b_patch)
    if swindon:
        plt.scatter(-197416, 6721977, c='yellow', s=marker_size, alpha=0.50)
        y_patch = mpatches.Patch(color='yellow', label='Swindon orbit')
        patches.append(y_patch)
    return patches
```

In [ ]:
```
def plot_over_grey(merged_data, a_type, yes_no, extra="", inner=False, \
                   fringe=False, oxford=False,swindon=False,topo=False):
    # plots selected data over the grey dots. yes_no controlls filtering the
    #data for a positive or negative values.
    plot_data = merged_data[merged_data[a_type] == yes_no]
    fig, ax = plt.subplots(figsize=(14.2 * 0.66, 23.0 * 0.66))
    show_background(plt, ax)
    location_XY_plot()
    add_grey()
    patches = add_oxford_swindon(oxford,swindon)
    plt.scatter(plot_data['Location_X'], plot_data['Location_Y'], c='Red')
    if fringe:
        f_for_legend = add_21Ha_fringe()
        patches.append(f_for_legend)
    if inner:
        i_for_legend = add_21Ha_line()
        patches.append(i_for_legend)
    show_records(plt, plot_data)
    plt.legend(loc='upper left', handles= patches)
    plt.title(get_print_title(f'{a_type} {extra}'))
    save_fig(f'{a_type}_{extra}')
    plt.show()
    print(f'{round(((len(plot_data)/4147)*100), 2)}%')
    return plot_data
```

In [ ]:
```
def plot_type_values(data, data_type, title,extra=''):
    new_data = data.copy()
    fig, ax = plt.subplots(figsize=((14.2 * 0.66)+2.4, 23.0 * 0.66))
    show_background(plt, ax)
    location_XY_plot()
    add_grey()
    plt.scatter(new_data['Location_X'], new_data['Location_Y'], \
                c=new_data[data_type], cmap=cm.rainbow, s=25)
    plt.colorbar(label=data_type)
    title = f'{data_type} {extra}'
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
```

In [ ]:
```
def bespoke_plot(plt, title):
    add_annotation_plot(plt)
    plt.ticklabel_format(style='plain')
    plt.title(get_print_title(title))
    save_fig(title)
    plt.show()
```

In [ ]:
```
def add_cluster_split_lines(plt, ax, extra=None):
    x_min = -550000
    if extra == 'ireland':
        x_min = -1200000
    plt.vlines(x=[-500000], ymin=7070000, ymax=9000000, colors='r', \
               ls='-', lw=3)
    plt.hlines(y=[7070000], xmin=x_min, xmax=200000, colors='r', ls='-', lw=3)
```

```
    ax.annotate("N/S split", color='k', xy=(50000, 7090000), xycoords='data')
    ax.annotate("E/W split", color='k', xy=(-480000, 8660000), xycoords='data')
    ax.annotate("Irish Sea split", color='k', xy=(-1150000, 7510000), \
                xycoords='data')
    plot_line((-800000,6400000), (-550000,7070000))
    plot_line((-550000,7070000), (-666000,7440000))
    plot_line((-666000,7440000),(-900000,7500000))
```

In [ ]:
```
def plot_values(data, feature, title, extra=''):
    fig, ax = plt.subplots(figsize=((14.2 * 0.66)+2.4, 23.0 * 0.66))
    show_background(plt, ax)
    location_XY_plot()
    plt.scatter(data['Location_X'], data['Location_Y'], c=data[feature], \
                cmap=cm.rainbow, s=25)
    plt.colorbar(label=feature)
    title = f'{title} {extra}'
    plt.title(title)
    save_fig(title)
    plt.show()
```

In [ ]:
```
def plot_line(point1, point2):
    x_values = [point1[0], point2[0]]
    y_values = [point1[1], point2[1]]
    plt.plot(x_values, y_values, 'r', ls='0', lw=2, alpha=1)
```

In [ ]:
```
def density_scatter_lines(location_data, scatter_data, plot_title, \
                          inner=False, fringe=False):
    fig, ax = plt.subplots(figsize=((14.2 * 0.66)+2.0, 23.0 * 0.66))
    show_background(plt, ax)
    location_XY_plot()
    plt.scatter(location_data['Location_X'], location_data['Location_Y'], \
                c=location_data['Density_trans'], cmap=cm.rainbow, s=25)
    plt.colorbar(label='Density Transformed')
    if inner:
        add_21Ha_line()
    if fringe:
        add_21Ha_fringe()
    plt.scatter(scatter_data['Location_X'], scatter_data['Location_Y'], c='Red')
    plt.legend(loc='lower left')
    plt.title(get_print_title(plot_title))
    save_fig(plot_title)
    plt.show()
```

In [ ]:
```
def south_density_scatter_lines(location_data, scatter_data, plot_title, \
                                inner=False, fringe=False):
    fig, ax = plt.subplots(figsize=((6.73*1.5)+2.0, (4.62*1.5)))
    show_background(plt, ax, 's')
    plt.ticklabel_format(style='plain')
    plt.xlim(-800000,220000)
    plt.ylim(6400000,7100000)
    plt.scatter(location_data['Location_X'], location_data['Location_Y'], \
                c=location_data['Density_trans'], cmap=cm.rainbow, s=25)
    plt.colorbar(label='Density Transformed')
    if inner:
        add_21Ha_line()
    if fringe:
        add_21Ha_fringe()
    plt.scatter(scatter_data['Location_X'], scatter_data['Location_Y'], \
                c='red', s=60)
    add_annotation_plot(plt)
    plt.legend(loc='lower right')
    plt.title(get_print_title(plot_title))
    save_fig(plot_title)
    plt.show()
```

In [ ]:
```
def add_linear_south():
    x_values = [-115637,-286900]
    y_values = [6678188,6585812]
    xx_values = [-244249,-363049]
    yy_values = [6555133,6589612]
    xxx_values = [-392213,-363146]
    yyy_values = [6577365,6647256]
    x4_values = [-169664, -207084]
    y4_values = [6599254, 6615290]
    x5_values = [-238560,-200891]
    y5_values = [6668083,6637826]


    plt.plot(x_values, y_values, 'g', ls='-', lw=8, alpha=0.6, label = \
          'Poss. correlation to linear routes?')
    plt.plot(xx_values, yy_values, 'g', ls='-', lw=8, alpha=0.6)
    plt.plot(xxx_values, yyy_values, 'g', ls='-', lw=8, alpha=0.6)
    plt.plot(x4_values, y4_values, 'g', ls='-', lw=8, alpha=0.6)
    plt.plot(x5_values, y5_values, 'g', ls='-', lw=8, alpha=0.6)
```

```
In [ ]: def plot_over_grey_south(merged_data, a_type, yes_no, extra=""):
            # plots selected data over the grey dots. yes_no controlls filtering the
            # data for a positive or negative values.
            plot_data = merged_data[merged_data[a_type] == yes_no]
            fig, ax = plt.subplots(1,figsize=((6.73*1.5), (4.62*1.5)))
            show_background(plt, ax, 's')
            plt.ticklabel_format(style='plain')
            plt.xlim(-800000,220000)
            plt.ylim(6400000,7100000)
            add_annotation_l_xy(plt)
            add_grey('s')
            add_linear_south()
            plt.scatter(plot_data['Location_X'], plot_data['Location_Y'], c='Red')
            plt.legend(loc='lower right')
            plt.title(get_print_title(f'{a_type} {extra}'))
            save_fig(f'{a_type}_{extra}')
            plt.show()
            return plot_data
```

```
In [ ]: def plot_over_grey_north(merged_data, a_type, yes_no, extra="", anno=False):
            # plots selected data over the grey dots. yes_no controlls filtering the
            # data for a positive or negative values.
            plot_data = merged_data[merged_data[a_type] == yes_no]
            fig, ax = plt.subplots(1,figsize=((5.28*2), (5.28*2)))
            show_background(plt, ax, 'n')
            plt.ticklabel_format(style='plain')
            plt.xlim(-800000,0)
            plt.ylim(7200000,8000000)
            if anno == 'Stirling':
                plt.annotate('SC1514: Gillies Hill', xy=(-443187, 7578896), \
                             xycoords='data', ha='left', xytext=(-135, 110), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
                plt.annotate('SC3420: Morebattle Hill', xy=(-263209,7461125), \
                             xycoords='data', ha='left', xytext=(-90, -100), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
                plt.annotate('EN4374: Pike House Camp', xy=(-209365,7418590), \
                             xycoords='data', ha='left', xytext=(-10, 80), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
                plt.annotate('SC3900: Kilmurdie', xy=(-305133,7566913), \
                             xycoords='data', ha='left', xytext=(20, 50), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
            elif anno == 'Traprain':
                plt.annotate('SC3932: Traprain Law', xy=(-297708,7551155), \
                             xycoords='data', ha='left', xytext=(35, 80), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
                plt.annotate('SC3037: Law Hill', xy=(-372530,7642082), \
                             xycoords='data', ha='left', xytext=(-150, 12), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
                plt.annotate("SC3571: Kerr's Knowe", xy=(-367362,7485652), \
                             xycoords='data', ha='left', xytext=(-200, 0), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
                plt.annotate('SC3327: Eildon Hill North', xy=(-301491, 7476601), \
                             xycoords='data', ha='left', xytext=(35, 35), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
            elif anno == 'Kerr':
                plt.annotate("SC3571: Kerr's Knowe", xy=(-367362,7485652), \
                             xycoords='data', ha='left', xytext=(-200, 0), \
                             textcoords='offset points', \
                             arrowprops=dict(arrowstyle="->", color='k', lw=1, \
                                             connectionstyle="arc3,rad=-0.2"))
            add_annotation_l_xy(plt)
            add_grey('ne')
            plt.scatter(plot_data['Location_X'], plot_data['Location_Y'], c='Red')
            plt.title(get_print_title(f'{a_type} {extra}'))
            save_fig(f'{a_type}_{extra}')
            plt.show()
            return plot_data
```

In [ ]:
```
def get_proportions(date_set):
    total = sum(date_set) - date_set[-1]
    newset = []
    for entry in date_set[:-1]:
        newset.append(round(entry/total,2))
    return newset
```

In [ ]:
```
def plot_dates_by_region(nw,ne,ni,si,s, features):
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    x_data = nw[features].columns
    x_data = [x.split("_")[2:] for x in x_data][:-1]
    x_data_new = []
    for l in x_data:
        txt =  ""
        for part in l:
            txt += "_" + part
        x_data_new.append(txt[1:])

    set1_name = 'NW'
    set2_name = 'NE'
    set3_name = 'N Ireland'
    set4_name = 'S Ireland'
    set5_name = 'South'
    set1 = get_proportions(get_counts(nw[features]))
    set2 = get_proportions(get_counts(ne[features]))
    set3 = get_proportions(get_counts(ni[features]))
    set4 = get_proportions(get_counts(si[features]))
    set5 = get_proportions(get_counts(s[features]))

    X_axis = np.arange(len(x_data_new))

    budge = 0.25

    plt.bar(X_axis - 0.55 + budge, set1, 0.3, label = set1_name)
    plt.bar(X_axis - 0.4 + budge, set2, 0.3, label = set2_name)
    plt.bar(X_axis - 0.25 + budge, set3, 0.3, label = set3_name)
    plt.bar(X_axis - 0.1 + budge, set4, 0.3, label = set4_name)
    plt.bar(X_axis + 0.05 + budge, set5, 0.3, label = set5_name)

    plt.xticks(X_axis, x_data_new)
    plt.xlabel('Dating')
    plt.ylabel('Proportion of Total Dated Hillforts in Region')
    title = 'Proportions of Dated Hillforts by Region'
    plt.title(title)
    plt.legend()
    add_annotation_plot(ax)
    save_fig(title)
    plt.show()
```

In [ ]:
```
def get_pcent_list(old_list):
    pcnt_list = []
    total = sum(old_list)
    for item in old_list:
        pcnt_list.append(round(item/total,2))
    return pcnt_list
```

In [ ]:
```
def order_set(set_list, x_data, pcnt=False):
    new_list = []
    set_values = set_list.index.tolist()
    for val in x_data:
        if val in set_values:
            new_list.append(set_list.loc[[val]].values[0])
        else:
            new_list.append(0)
    if pcnt:
        new_list = get_pcent_list(new_list)
    return new_list
```

In [ ]:
```
def plot_feature_by_region(nw,ne,ni,si,s, feature, title, clip):
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    max_val = \
    int(max([nw[feature].max(),ne[feature].max(),\
             ni[feature].max(),si[feature].max(),s[feature].max()]))

    x_data = [x-1 for x in range(max_val+2)]

    set0_name = 'NW'
    set1_name = 'NE'
    set2_name = 'N Ireland'
    set3_name = 'S Ireland'
    set4_name = 'S'
```

```python
    set0 = nw[feature].value_counts()
    set1 = ne[feature].value_counts()
    set2 = ni[feature].value_counts()
    set3 = si[feature].value_counts()
    set4 = s[feature].value_counts()

    set0 = order_set(set0,x_data, True)[:clip]
    set1 = order_set(set1,x_data, True)[:clip]
    set2 = order_set(set2,x_data, True)[:clip]
    set3 = order_set(set3,x_data, True)[:clip]
    set4 = order_set(set4,x_data, True)[:clip]

    X_axis = np.arange(len(x_data[:clip]))

    budge = 0.2

    plt.bar(X_axis - 0.6 + budge, set0, 0.3, label = set0_name)
    plt.bar(X_axis - 0.45 + budge, set1, 0.3, label = set1_name)
    plt.bar(X_axis - 0.3 + budge, set2, 0.3, label = set2_name)
    plt.bar(X_axis - 0.15 + budge, set3, 0.3, label = set3_name)
    plt.bar(X_axis + 0 + budge, set4, 0.3, label = set4_name)

    plt.xticks(X_axis, x_data)
    plt.xlabel('Number')
    plt.ylabel('Percentage of regional total')
    plt.title(title)
    plt.legend()
    add_annotation_plot(ax)
    save_fig(title)
    plt.show()
```

In [ ]:
```python
def plot_quadrents(ramparts,ditches,ne,se,sw,nw):
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])
    #x_data = [x for x in range(11)]
    x_data = [x for x in range(8)]

    set0_name = 'Ramparts'
    set00_name = 'Ditches'
    set1_name = 'NE'
    set2_name = 'SE'
    set3_name = 'SW'
    set4_name = 'NW'
    set0 = ramparts['Enclosing_Max_Ramparts'].value_counts()
    set0 = order_set(set0,x_data)[:8]
    set00 = ditches['Enclosing_Ditches_Number'].value_counts()
    set00 = order_set(set00,x_data)[:8]
    set1 = ne['Enclosing_NE_Quadrant'].value_counts()
    set1 = order_set(set1,x_data)[:8]
    set2 = se['Enclosing_SE_Quadrant'].value_counts()
    set2 = order_set(set2,x_data)[:8]
    set3 = sw['Enclosing_SW_Quadrant'].value_counts()
    set3 = order_set(set3,x_data)[:8]
    set4 = nw['Enclosing_NW_Quadrant'].value_counts()
    set4 = order_set(set4,x_data)[:8]

    X_axis = np.arange(len(x_data[:8]))

    budge = 0.2

    plt.bar(X_axis - 0.6 + budge, set0, 0.2, label = set0_name)
    plt.bar(X_axis - 0.46 + budge, set00, 0.2, label = set00_name)
    plt.bar(X_axis - 0.32 + budge, set1, 0.2, label = set1_name)
    plt.bar(X_axis - 0.18 + budge, set2, 0.2, label = set2_name)
    plt.bar(X_axis - 0.04 + budge, set3, 0.2, label = set3_name)
    plt.bar(X_axis + 0.1 + budge, set4, 0.2, label = set4_name)

    plt.xticks(X_axis, x_data)
    plt.xlabel('Number')
    plt.ylabel('Count')
    title = 'Ditches, Ramparts and Quadrant by Number'
    plt.title(title)
    plt.legend()
    add_annotation_plot(ax)
    save_fig(title)
    plt.show()
```

In [ ]:
```python
def plot_regions(nw,ne,ni,si,s, features, xlabel, title, split_pos, \
                 yes_no, pcent=False):
    fig = plt.figure(figsize=(12,5))
    ax = fig.add_axes([0,0,1,1])

    x_data = features
    x_data = [x.split("_")[split_pos:] for x in x_data]
```

```python
    x_data_new = []
    for l in x_data:
        txt =  ""
        for part in l:
            txt += "_" + part
        x_data_new.append(txt[1:])

    set0_name = 'NW'
    set1_name = 'NE'
    set2_name = 'N Ireland'
    set3_name = 'S Ireland'
    set4_name = 'S'

    set0_list = []
    set1_list = []
    set2_list = []
    set3_list = []
    set4_list = []

    for feature in features:
        set0_list.append((nw[feature].values == yes_no).sum())
        set1_list.append((ne[feature].values == yes_no).sum())
        set2_list.append((ni[feature].values == yes_no).sum())
        set3_list.append((si[feature].values == yes_no).sum())
        set4_list.append((s[feature].values == yes_no).sum())

    set0 = set0_list
    set1 = set1_list
    set2 = set2_list
    set3 = set3_list
    set4 = set4_list

    if pcent:
        set0 = get_pcent_list(set0)
        set1 = get_pcent_list(set1)
        set2 = get_pcent_list(set2)
        set3 = get_pcent_list(set3)
        set4 = get_pcent_list(set4)

    X_axis = np.arange(len(x_data))

    budge = 0.3

    plt.bar(X_axis - 0.6 + budge, set0, 0.13, label = set0_name)
    plt.bar(X_axis - 0.45 + budge, set1, 0.13, label = set1_name)
    plt.bar(X_axis - 0.3 + budge, set2, 0.13, label = set2_name)
    plt.bar(X_axis - 0.15 + budge, set3, 0.13, label = set3_name)
    plt.bar(X_axis + 0 + budge, set4, 0.13, label = set4_name)

    plt.xticks(X_axis, x_data_new)
    plt.xlabel(xlabel)
    if pcent:
        plt.ylabel('Percentage of Regional Total')
    else:
        plt.ylabel('Count')
    plt.title(get_print_title(f'{title}'))
    plt.legend()
    add_annotation_plot(ax)
    save_fig(title)
    plt.show()
```

Review Data Functions

The following functions will be used to confirm that features are not lost or forgotten when splitting the data.

In [ ]:

```python
def test_numeric(data):
    temp_data = data.copy()
    columns = data.columns
    out_cols = ['Feature','Entries', 'Numeric', 'Non-Numeric', 'Null']
    feat, ent, num, non, nul = [],[],[],[],[]
    for col in columns:
        if temp_data[col].dtype == 'object':
            feat.append(col)
            temp_data[col+'_num'] = temp_data[col].str.isnumeric()
            entries = temp_data[col].notnull().sum()
            true_count = temp_data[col+'_num'][temp_data[col+'_num'] == \
                                                True].sum()
            null_count = temp_data[col].isna().sum()
            ent.append(entries)
            num.append(true_count)
            non.append(entries-true_count)
            nul.append(null_count)
        else:
            print(f'{col} {temp_data[col].dtype}')
```

```
    summary = pd.DataFrame(list(zip(feat, ent, num, non, nul)))
    summary.columns = out_cols
    return summary
```

In [ ]:
```
def find_duplicated(numeric_data, text_data, encodeable_data):
    d = False
    all_columns = \
    list(numeric_data.columns) + list(text_data.columns) + \
    list(encodeable_data.columns)
    duplicate = \
     [item for item, count in collections.Counter(all_columns).items() \
      if count > 1]
    if duplicate :
        print(f"There are duplicate features: {duplicate}")
        d = True
    return d
```

In [ ]:
```
def test_data_split(main_data, numeric_data, text_data, encodeable_data):
    m = False
    split_features = \
    list(numeric_data.columns) + list(text_data.columns) + \
    list(encodeable_data.columns)
    missing = list(set(main_data)-set(split_features))
    if missing:
        print(f"There are missing features: {missing}")
        m = True
    return m
```

In [ ]:
```
def review_data_split(main_data, numeric_data, text_data, \
                      encodeable_data = pd.DataFrame()):
    d = find_duplicated(numeric_data, text_data, encodeable_data)
    m = test_data_split(main_data, numeric_data, text_data, encodeable_data)
    if d != True and m != True:
        print("Data split good.")
```

In [ ]:
```
def find_duplicates(data):
    print(f'{data.count() - data.duplicated(keep=False).count()} duplicates.')
```

In [ ]:
```
def count_yes(data):
    total = 0
    for col in data.columns:
        count = len(data[data[col] == 'Yes'])
        total+= count
        print(f'{col}: {count}')
    print(f'Total yes count: {total}')
```

Null Value Functions

The following functions will be used to update null values.

In [ ]:
```
def fill_nan_with_minus_one(data, feature):
    new_data = data.copy()
    new_data[feature] = data[feature].fillna(-1)
    return new_data
```

In [ ]:
```
def fill_nan_with_NA(data, feature):
    new_data = data.copy()
    new_data[feature] = data[feature].fillna("NA")
    return new_data
```

In [ ]:
```
def test_numeric_value_in_feature(feature, value):
    test = feature.isin([-1]).sum()
    return test
```

In [ ]:
```
def test_catagorical_value_in_feature(dataframe, feature, value):
    test = dataframe[feature][dataframe[feature] == value].count()
    return test
```

In [ ]:
```
def test_cat_list_for_NA(dataframe, cat_list):
    for val in cat_list:
        print(val, test_catagorical_value_in_feature(dataframe, val,'NA'))
```

In [ ]:
```
def test_num_list_for_minus_one(dataframe, num_list):
    for val in num_list:
        feature = dataframe[val]
        print(val, test_numeric_value_in_feature(feature, -1))
```

In [ ]:
```
def update_cat_list_for_NA(dataframe, cat_list):
    new_data = dataframe.copy()
```

```
    for val in cat_list:
        new_data = fill_nan_with_NA(new_data, val)
    return new_data
```

In [ ]:
```
def update_num_list_for_minus_one(dataframe, cat_list):
    new_data = dataframe.copy()
    for val in cat_list:
        new_data = fill_nan_with_minus_one(new_data, val)
    return new_data
```

Reprocessing Functions

In [ ]:
```
def add_density(data):
    new_data = data.copy()
    xy = np.vstack([new_data['Location_X'],new_data['Location_Y']])
    new_data['Density'] = gaussian_kde(xy)(xy)
    return new_data
```

Save Image Functions

In [ ]:
```
# Set-up figure numbering
fig_no = 0
part = 'Part05'
IMAGES_PATH = r'/content/drive/My Drive/'
fig_list = pd.DataFrame(columns=['fig_no', 'file_name', 'title'])
topo_txt = ""
if show_topography:
    topo_txt = "-topo"
```

In [ ]:
```
# Remove unicode characters from file names
def get_file_name(title):
    file_name = slugify(title)
    return file_name
```

In [ ]:
```
# Remove underscore from figure titles
def get_print_title(title):
    title = title.replace("_", " ")
    title = title.replace("-", " ")
    title = title.replace(",", ";")
    return title
```

In [ ]:
```
# Format figure numbers to have three digits
def format_figno(no):
    length = len(str(no))
    fig_no = ''
    for i in range(3-length):
        fig_no = fig_no + '0'
    fig_no = fig_no + str(no)
    return fig_no
```

In [ ]:
```
# Mount Google Drive if figures to be saved
if save_images == True:
    drive.mount('/content/drive')
    os.getcwd()
else:
    pass
```

In [ ]:
```
def save_fig(fig_id, tight_layout=True, fig_extension="png", resolution=300):
    global fig_no
    global IMAGES_PATH
    if save_images:
        #IMAGES_PATH = r'/content/drive/My Drive/Colab Notebooks/Hillforts_Primer_Images/HP_Part_05_images/'
        fig_no+=1
        fig_no_txt = format_figno(fig_no)
        file_name = file_name = get_file_name(f'{part}_{fig_no_txt}')
        file_name = f'hillforts_primer_{file_name}{topo_txt}.{fig_extension}'
        fig_list.loc[len(fig_list)] = [fig_no, file_name, get_print_title(fig_id)]
        path = os.path.join(IMAGES_PATH, file_name)
        print("Saving figure", file_name)
        plt.tight_layout()
        plt.savefig(path, format=fig_extension, dpi=resolution,  \
                    bbox_inches='tight')
    else:
        pass
```

Load Data

The source csv file is loaded and the first two rows are displayed to confirm the load was successful. Note that, to the left, an index has been added automatically. This index will be used frequently when splitting and remerging data extracts.

In [ ]:
```
hillforts_csv = r"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/hillforts-atlas-source-data-cs
#"https://raw.githubusercontent.com/MikeDairsie/Hillforts-Primer/main/
#hillforts-atlas-source-data-csv/hillforts.csv"
hillforts_data = pd.read_csv(hillforts_csv, index_col=False)
pd.set_option('display.max_columns', None, 'display.max_rows', None)
hillforts_data.head(2)
```

```
<ipython-input-73-b2855eddaef9>:4: DtypeWarning: Columns (10,12,68,83,84,85,86,165,183) have mixed types. Specify dt
ype option on import or set low_memory=False.
  hillforts_data = pd.read_csv(hillforts_csv, index_col=False)
```

Out[ ]:

	OBJECTID	Main_Atlas_Number	Main_Country_Code	Main_Country	Main_Title_Name	Main_Site_Name	Main_Alt_Name	Main_Display_N
0	1	1	EN	England	EN0001 Aconbury Camp, Herefordshire	Aconbury Camp	Aconbury Beacon	Aconbury C Hereford (Aconbury Bea
1	2	2	EN	England	EN0002 Bach Camp, Herefordshire	Bach Camp	NaN	Bach C Hereford

Filter confirmed (if selected)

If confirmed_only is set to True in User Settings above, this will filter the source data so that it contains only confirmed forts.

In [ ]:
```
if confirmed_only == True:
    hillforts_data = \
    hillforts_data[hillforts_data['Status_Interpretation_Reliability'] == \
                   'Confirmed']
    print(f'Data filtered to contain only {len(hillforts_data)} confirmed hillforts.')
else:
    print(f'Using all {len(hillforts_data)} record in the Hillforts Atlas.')
```

```
Using all 4147 record in the Hillforts Atlas.
```

Download Function

In [ ]:
```
from google.colab import files
def download(data_list, filename, hf_data=hillforts_data):
    if download_data == True:
        name_and_number = \
        hf_data[['Main_Atlas_Number','Main_Display_Name']].copy()
        dl = name_and_number.copy()
        for pkg in data_list:
            if filename not in ['england', 'wales','scotland',\
                                'republic-of-ireland','norhtern-ireland', \
                                'isle-of-man', 'roi-ni', 'eng-wal-sco-iom']:
                if pkg.shape[0] == hillforts_data.shape[0]:
                    dl = pd.merge(dl, pkg, left_index=True, right_index=True)
            else:
                dl = data_list[0]
        dl = dl.replace('\r',' ', regex=True)
        dl = dl.replace('\n',' ', regex=True)
        fn = 'hillforts_primer_' + filename
        fn = get_file_name(fn)
        dl.to_csv(fn+'.csv', index=False)
        files.download(fn+'.csv')
    else:
        pass
```

Reload Name and Number

The Main Atlas Number and the Main Display Name are the primary uninqe reference identiriers in the data. With these, users can identify any record numerically and by name. Throughout this document, the data will be clipped into a number of sub-data packages. Where needed, these data extracts will be combined with Name and Number features to ensure the data can be understood and can, if needed, be concorded.

In [ ]:
```
name_and_number_features = ['Main_Atlas_Number','Main_Display_Name']
name_and_number = hillforts_data[name_and_number_features].copy()
name_and_number.head()
```

Out[ ]:

	Main_Atlas_Number	Main_Display_Name
0	1	Aconbury Camp, Herefordshire (Aconbury Beacon)
1	2	Bach Camp, Herefordshire
2	3	Backbury Camp, Herefordshire (Ethelbert's Camp)
3	4	Brandon Camp, Herefordshire
4	5	British Camp, Herefordshire (Herefordshire Bea...

Reload Location

In [ ]:
```
location_numeric_data_short_features = ['Location_X','Location_Y']
location_numeric_data_short = hillforts_data[location_numeric_data_short_features]
location_numeric_data_short = add_density(location_numeric_data_short)
location_numeric_data_short.head()
location_data = location_numeric_data_short.copy()
location_data.head()
```

Out[ ]:

	Location_X	Location_Y	Density
0	-303295	6798973	1.632859e-12
1	-296646	6843289	1.540172e-12
2	-289837	6808611	1.547729e-12
3	-320850	6862993	1.670548e-12
4	-261765	6810587	1.369981e-12

Reload Dating

In [ ]:
```
date_features = [
 'Dating_Date_Pre_1200BC',
 'Dating_Date_1200BC_800BC',
 'Dating_Date_800BC_400BC',
 'Dating_Date_400BC_AD50',
 'Dating_Date_AD50_AD400',
 'Dating_Date_AD400_AD800',
 'Dating_Date_Post_AD800',
 'Dating_Date_Unknown']

date_data = hillforts_data[date_features].copy()
date_data.head()
```

Out[ ]:

	Dating_Date_Pre_1200BC	Dating_Date_1200BC_800BC	Dating_Date_800BC_400BC	Dating_Date_400BC_AD50	Dating_Date_AD50_AD400	D
0	No	No	Yes	Yes	Yes	
1	No	No	No	No	No	
2	No	No	No	No	No	
3	No	No	No	No	Yes	
4	No	No	Yes	Yes	Yes	

Reload Regional Data Packages

See Cluster Data Packages in Part 1: Name, Admin & Location Data
https://colab.research.google.com/drive/1C7HcuLuGGhG8o4EGciS-XTAhxVs3MhX3?usp=sharing

In [ ]:
```
cluster_data = \
hillforts_data[['Location_X','Location_Y', 'Main_Country_Code']].copy()
cluster_data['Cluster'] = 'NA'
cluster_data['Cluster'].where(cluster_data['Main_Country_Code'] != \
                              'NI', 'I', inplace=True)
cluster_data['Cluster'].where(cluster_data['Main_Country_Code'] != \
                              'IR', 'I', inplace=True)

cluster_data['Cluster'] = np.where(
   (cluster_data['Cluster'] == 'I') & (cluster_data['Location_Y'] >= 7060000) , \
```

```
    'North Ireland', cluster_data['Cluster']
    )
north_ireland = cluster_data[cluster_data['Cluster'] == 'North Ireland'].copy()

cluster_data['Cluster'] = np.where(
    (cluster_data['Cluster'] == 'I') & (cluster_data['Location_Y'] < 7060000) , \
    'South Ireland', cluster_data['Cluster']
    )
south_ireland = cluster_data[cluster_data['Cluster'] == 'South Ireland'].copy()

cluster_data['Cluster'] = np.where(
    (cluster_data['Cluster'] == 'NA') & (cluster_data['Location_Y'] < 7070000) , \
    'South', cluster_data['Cluster']
    )
south = cluster_data[cluster_data['Cluster'] == 'South'].copy()

cluster_data['Cluster'] = np.where(
    (cluster_data['Cluster'] == \
     'NA') & (cluster_data['Location_Y'] >= 7070000) & \
      (cluster_data['Location_X'] >= -500000), 'Northeast', cluster_data['Cluster']
    )
north_east = cluster_data[cluster_data['Cluster'] == 'Northeast'].copy()

cluster_data['Cluster'] = np.where(
    (cluster_data['Cluster'] == \
     'NA') & (cluster_data['Location_Y'] >= 7070000) & \
      (cluster_data['Location_X'] < -500000), 'Northwest', cluster_data['Cluster']
    )
north_west = cluster_data[cluster_data['Cluster'] == 'Northwest'].copy()

temp_cluster_location_packages = [north_ireland, south_ireland, south, \
                                  north_east, north_west]

cluster_packages = []
for pkg in temp_cluster_location_packages:
    pkg = pkg.drop(['Main_Country_Code'], axis=1)
    cluster_packages.append(pkg)

north_ireland, south_ireland, south, north_east, north_west = \
cluster_packages[0], cluster_packages[1], cluster_packages[2], \
cluster_packages[3], cluster_packages[4]
```

Review Data Part 5

Entrance Data

Additional information relating to entrances is contained in an Entrances Table. This can be downloaded from the Hillforts Atlas Rest Service API here or this project's data store here. The Entrances Table has not been analysed as part of the Hillforts Primer at this time.

In [ ]:

```
entrance_features = [
 'Entrances_Breaks',
 'Entrances_Breaks_Comments',
 'Entrances_Original',
 'Entrances_Original_Comments',
 'Entrances_Guard_Chambers',
 'Entrances_Chevaux',
 'Entrances_Chevaux_Comments',
 'Entrances_Summary',
 'Related_Entrances']

entrance_data = hillforts_data[entrance_features].copy()
entrance_data.head()
```

Out[ ]:

	Entrances_Breaks	Entrances_Breaks_Comments	Entrances_Original	Entrances_Original_Comments	Entrances_Guard_Chambers	Entrances_(
0	6.0	Two original and four modern gaps.	2.0	Two original inturned entrances at SE and SW c...	No	
1	3.0	N entrance damaged by wagon access and possibl...	2.0	S entrance original, that on the NW possibly ...	No	
2	2.0	Entrances intact	2.0	Interesting inturn to N entrance	No	
3	3.0	Modern gap to the S.	2.0	Off-set entrance on the E. Possibly another to...	No	
4	6.0	Probable modern breaks not recorded.	6.0	Two entrances are from Phase I and four from P...	No	

There are null values in all but two entrance features.

In [ ]:
```
entrance_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 9 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Entrances_Breaks             3830 non-null   float64
 1   Entrances_Breaks_Comments    1193 non-null   object 
 2   Entrances_Original           3941 non-null   float64
 3   Entrances_Original_Comments  1126 non-null   object 
 4   Entrances_Guard_Chambers     4147 non-null   object 
 5   Entrances_Chevaux            4147 non-null   object 
 6   Entrances_Chevaux_Comments   77 non-null     object 
 7   Entrances_Summary            4132 non-null   object 
 8   Related_Entrances            2749 non-null   object 
dtypes: float64(2), object(7)
memory usage: 291.7+ KB
```

Entrance Numeric Data

There are two numeric features. Both contain null values that will be resolved below.

In [ ]:
```
entrance_numeric_features = [
 'Entrances_Breaks',
 'Entrances_Original']

entrance_numeric_data = entrance_data[entrance_numeric_features ].copy()
entrance_numeric_data.head()
```

Out[ ]:

	Entrances_Breaks	Entrances_Original
0	6.0	2.0
1	3.0	2.0
2	2.0	2.0
3	3.0	2.0
4	6.0	6.0

Entrance Numeric Data - Resolve Null Values

Test for -1.

In [ ]:
```
test_num_list_for_minus_one(entrance_numeric_data, entrance_numeric_features)
```

```
Entrances_Breaks 0
Entrances_Original 0
```

Replace null with -1.

In [ ]:
```
entrance_numeric_data = \
update_num_list_for_minus_one(entrance_numeric_data, entrance_numeric_features)
entrance_numeric_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 2 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Entrances_Breaks     4147 non-null   float64
 1   Entrances_Original   4147 non-null   float64
dtypes: float64(2)
memory usage: 64.9 KB
```

Entrances Breaks Data Plotted

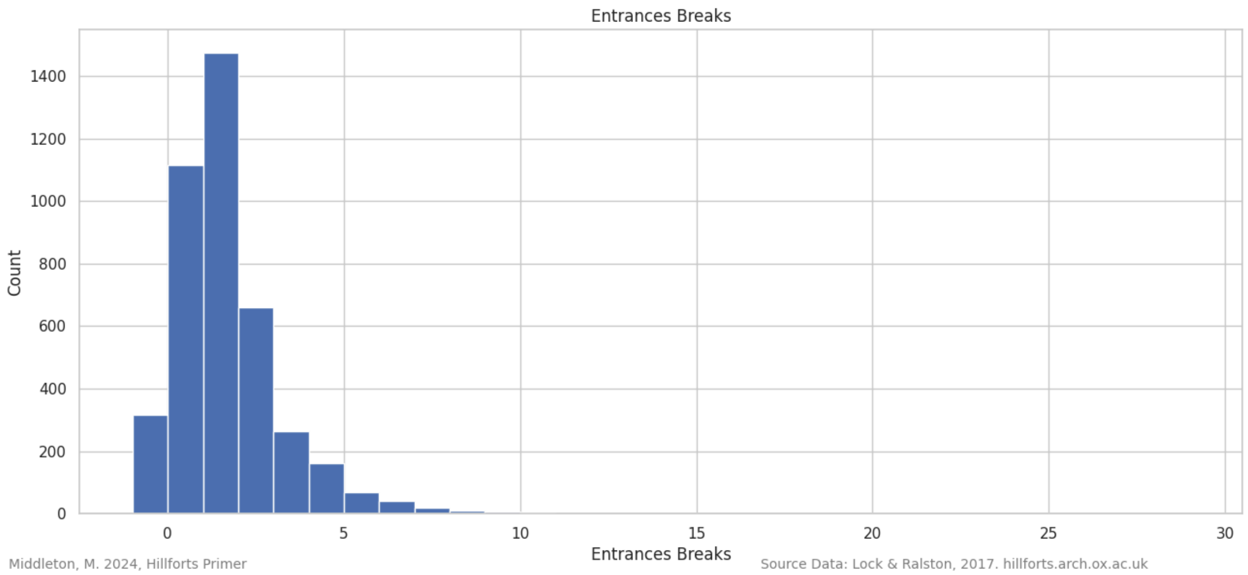
Entrance breaks has a long tail of outliers. Most hillforts (90.26%) have five entrances or less. 78.3% have two entrances or less.

In [ ]:
```python
entrance_numeric_data['Entrances_Breaks'].value_counts().sort_index()
```

Out[ ]:
```
-1.0      317
 0.0     1114
 1.0     1474
 2.0      661
 3.0      264
 4.0      162
 5.0       68
 6.0       42
 7.0       19
 8.0       10
 9.0        7
 10.0       4
 11.0       1
 15.0       1
 16.0       1
 19.0       1
 29.0       1
Name: Entrances_Breaks, dtype: int64
```

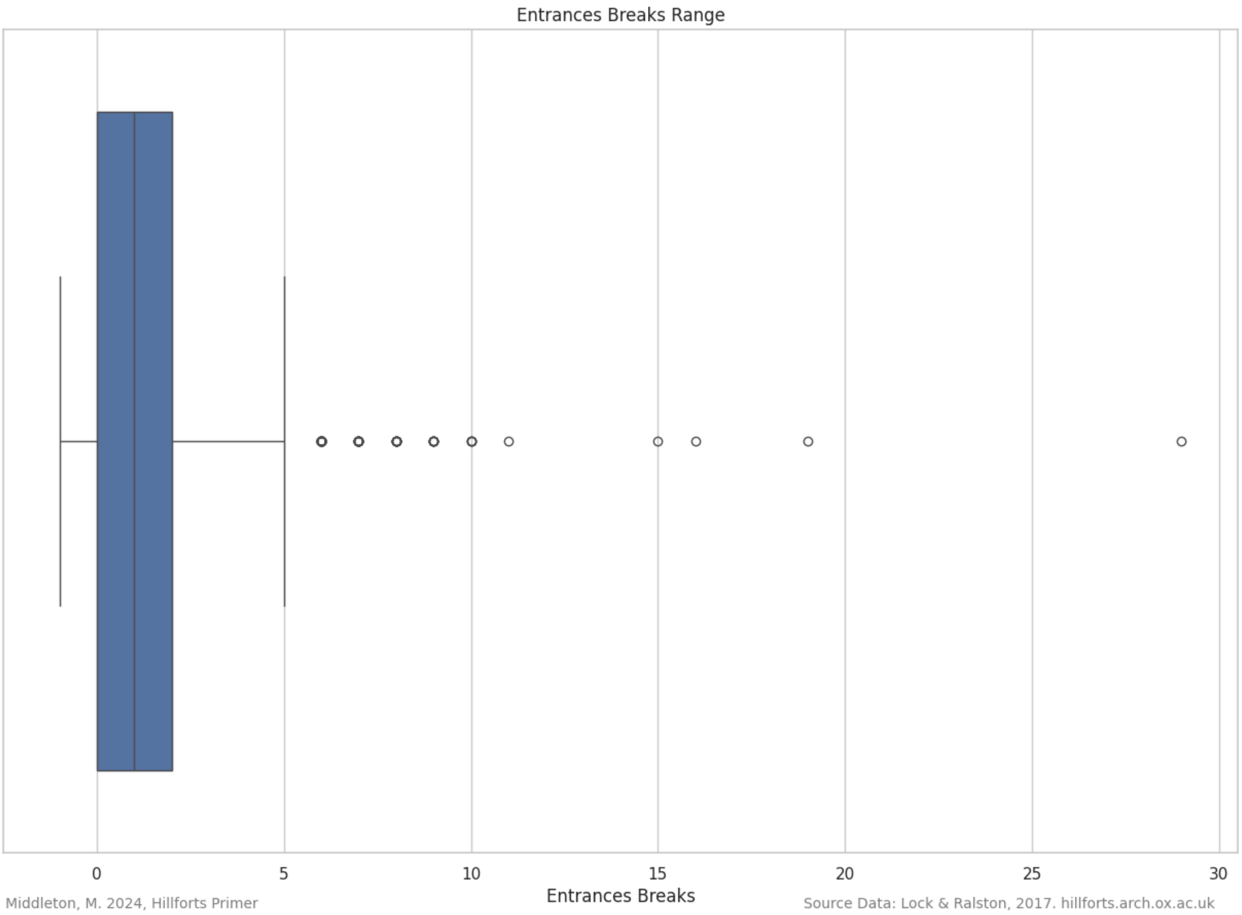
In [ ]:
```python
plot_histogram(entrance_numeric_data['Entrances_Breaks'], 'Entrances Breaks', \
               'Entrances Breaks')
```

None



Outliers range from 6 to 29 entrances.

In [ ]:
```python
entrances_breaks_data = \
plot_data_range(entrance_numeric_data['Entrances_Breaks'], \
                'Entrances Breaks', "h")
```



In [ ]:
```
entrances_breaks_data
```

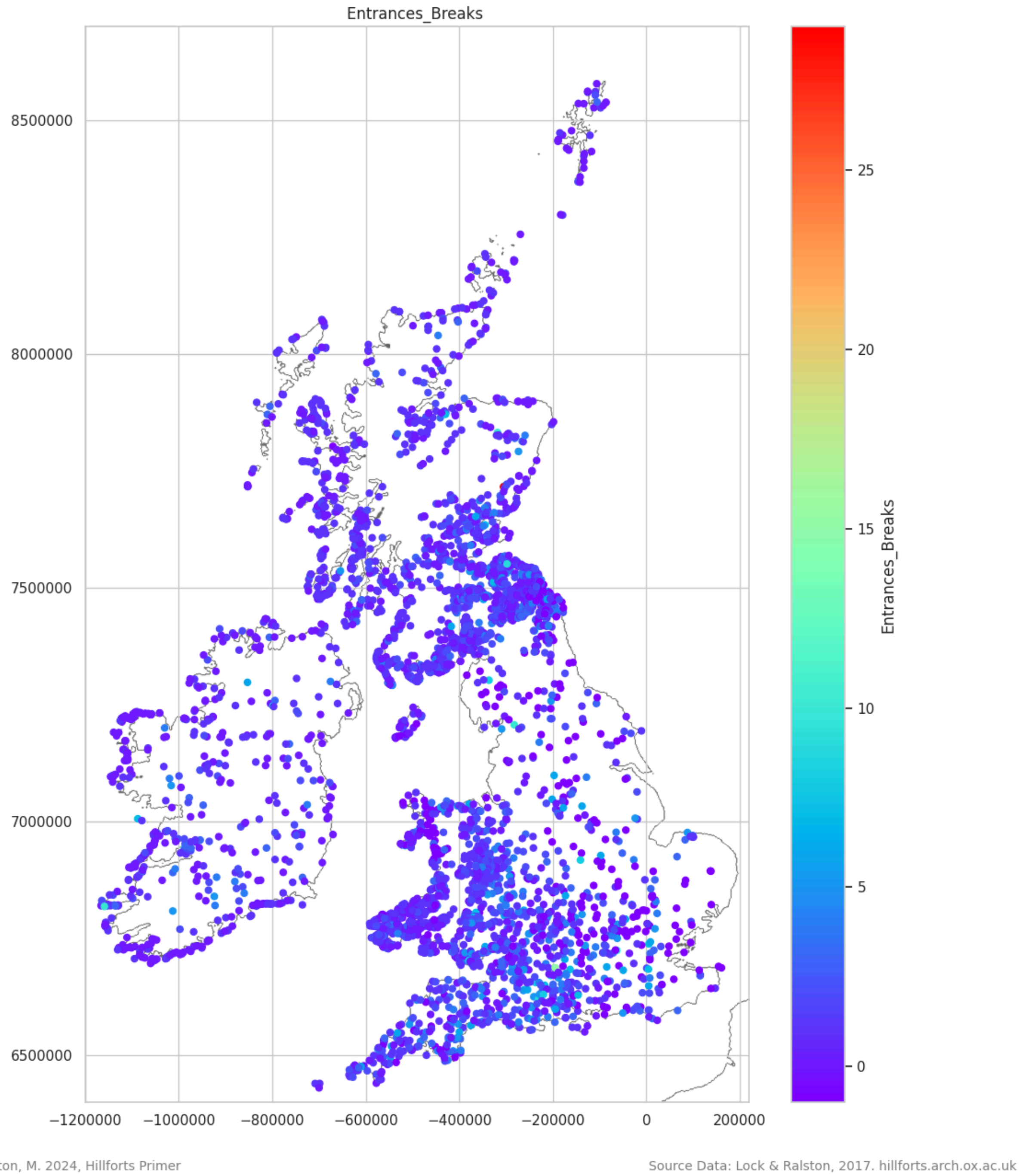
Out[ ]:
```
[-1.0, 0.0, 1.0, 2.0, 5.0]
```

Entrance Breaks Mapped

The high concentration of hillforts with five entrances or less, and the long tail up to 29 entrances, causes the plot of entrance breaks to lack clarity. The options are to reproject the data using a boxcox projection or to split the data into ranges. In this case, splitting the data using quartile ranges, will plot meaningful groupings while reducing the plot range of each figure. This will improve the clarity of each map.

In [ ]:
```
location_entrance_data = \
pd.merge(location_numeric_data_short, entrance_numeric_data, \
         left_index=True, right_index=True)
```

In [ ]:
```
plot_values(location_entrance_data, 'Entrances_Breaks', 'Entrances_Breaks')
```



Entrance Breaks Interquartile Range (Mid 50%) Distribution Mapped

Most hillforts have zero to two entrances (78.3%). All coastal forts in Ireland, most in the west and north of Scotland and most in the Welsh uplands fall in this range. The Northeast and the South both have a large number of hillforts that fall within this range.

In [ ]:

```
eb_iqr = \
location_entrance_data[location_entrance_data['Entrances_Breaks'].\
                       between(0,2)].copy()
eb_upper_q = \
location_entrance_data[location_entrance_data['Entrances_Breaks'].\
                       between(2.1,5)].copy()
eb_out = \
location_entrance_data[location_entrance_data['Entrances_Breaks']>5].copy()
```

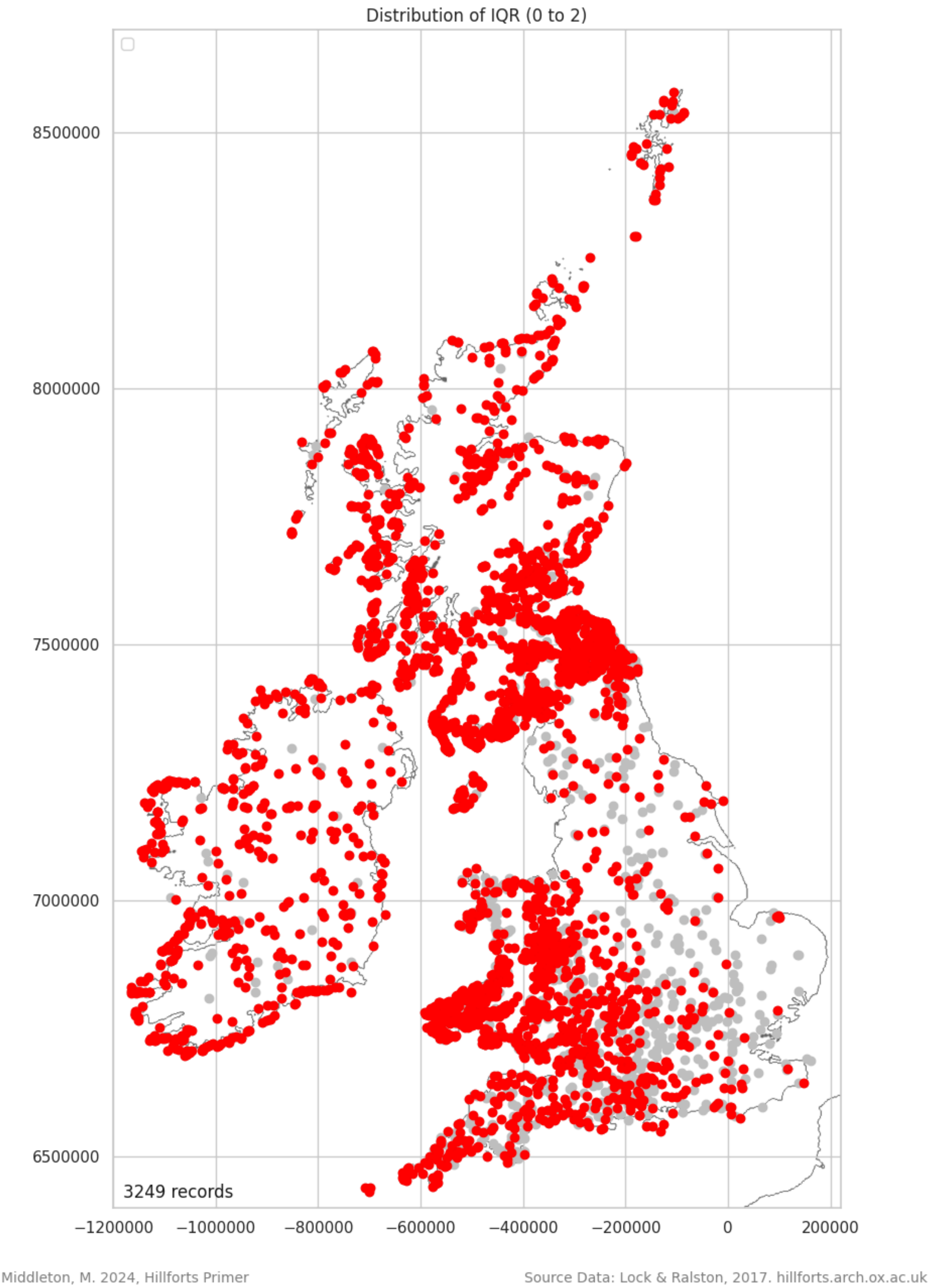
In [ ]:

```
print(f'{round(len(eb_iqr)/len(location_entrance_data)*100, 2)}% of hillforts have two entrances or less (IQR).')
```

```
78.35% of hillforts have two entrances or less (IQR).
```

In [ ]:

```
plot_over_grey_numeric(eb_iqr, 'Entrances_Breaks', 'Distribution of IQR (0 to 2)')
```

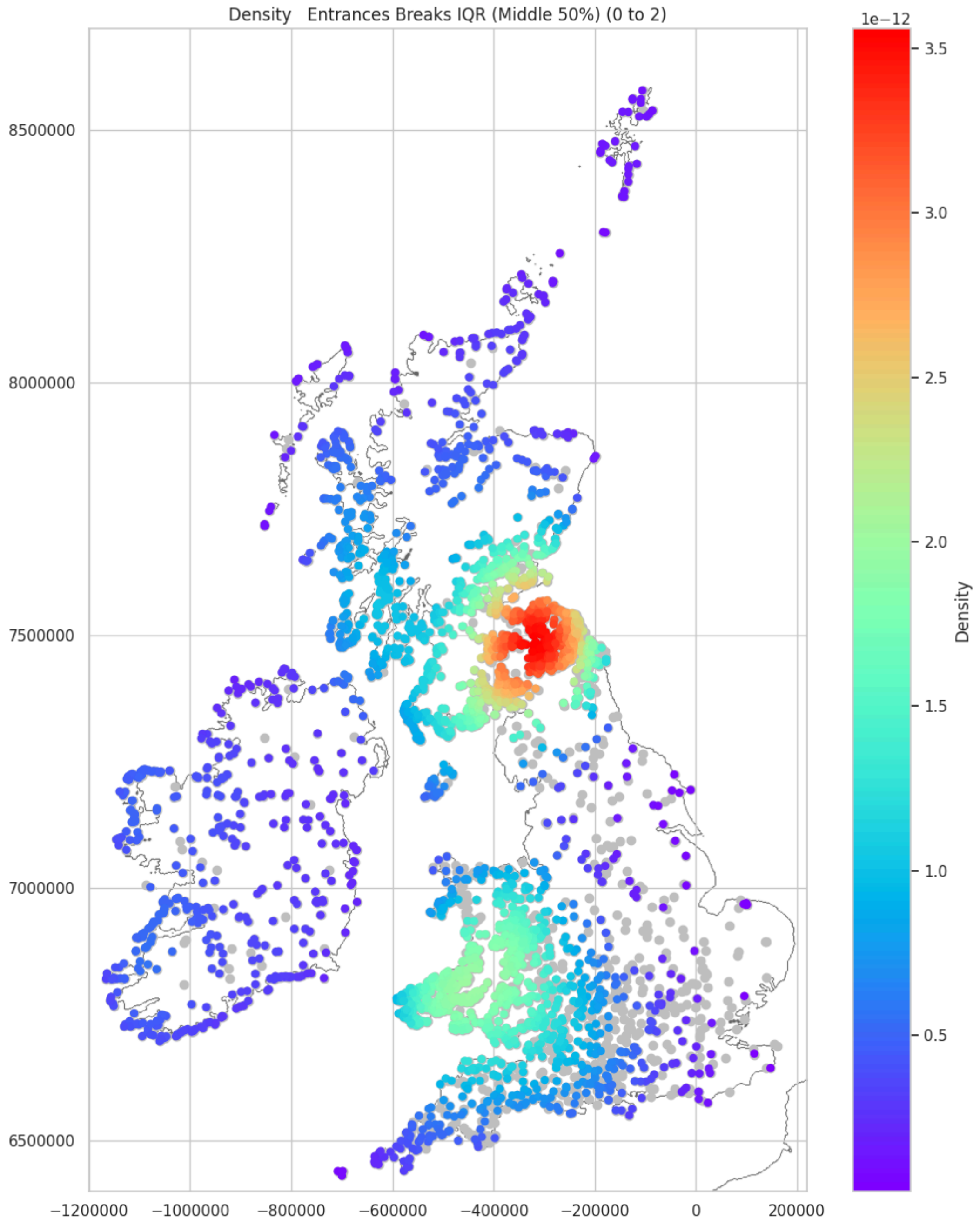


78.35%

Entrance Breaks Interquartile Range (Mid 50%) Density Mapped

This range has a high representation of hillforts from all regions meaning the distribution clusters, seen when plotting the location data in, Part 1: Density Data Transformed Mapped, are replicated in this subset of the Entrance Breaks data.

```
In [ ]: plot_density_over_grey(eb_iqr, 'Entrances_Breaks', 'IQR (Middle 50%) (0 to 2)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

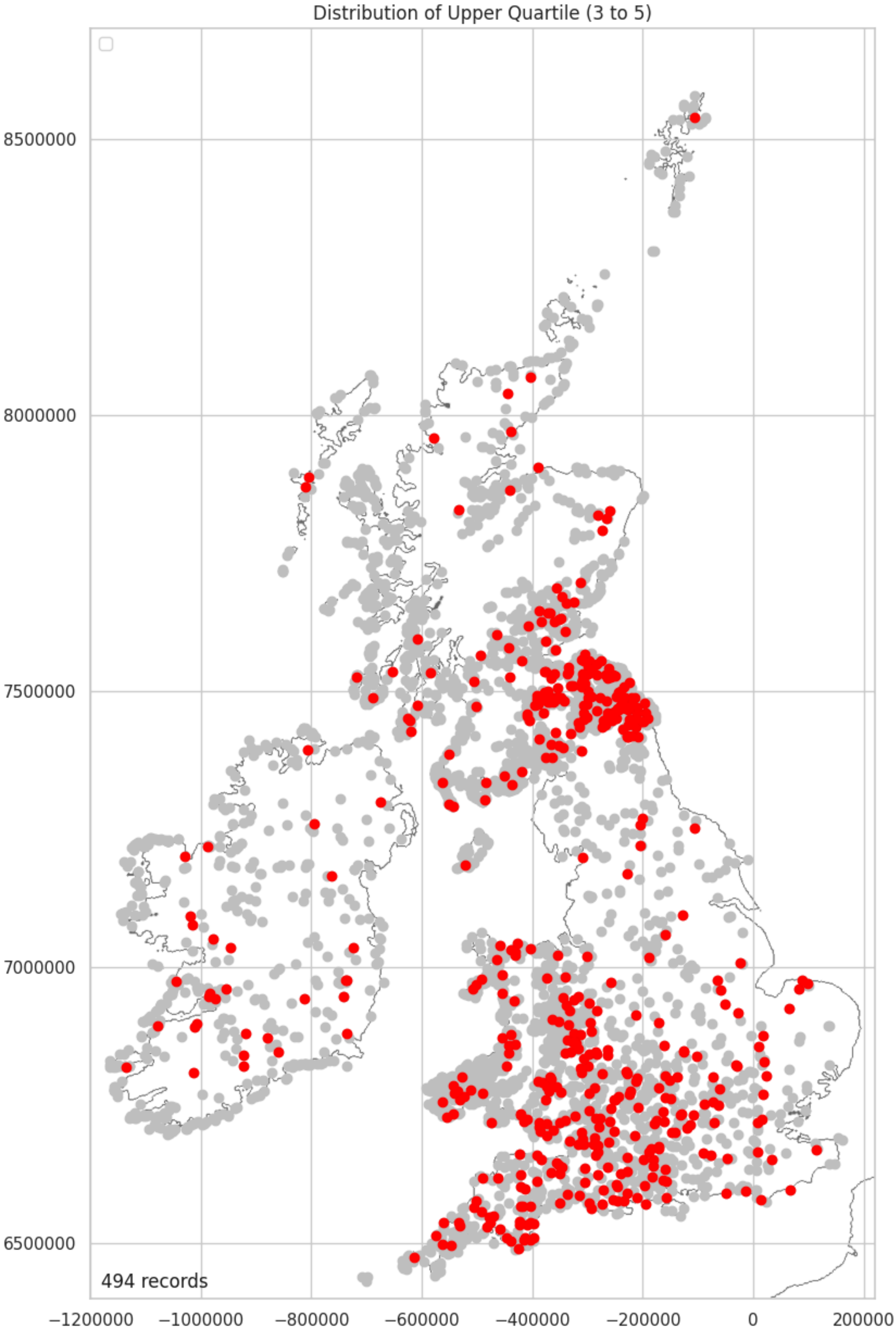
Entrance Breaks Upper Quartile Distribution Mapped

Only 11.9% of hillforts have three to five entrances.

In [ ]:
```
print(f'{round(len(eb_upper_q)/len(location_entrance_data)*100,2)} have three to five entrances (Upper quartile).')
```

```
11.91 have three to five entrances (Upper quartile).
```

In [ ]:
```
plot_over_grey_numeric(eb_upper_q, 'Entrances_Breaks', \
                       'Distribution of Upper Quartile (3 to 5)')
```



Middleton, M. 2024, Hillforts Primer

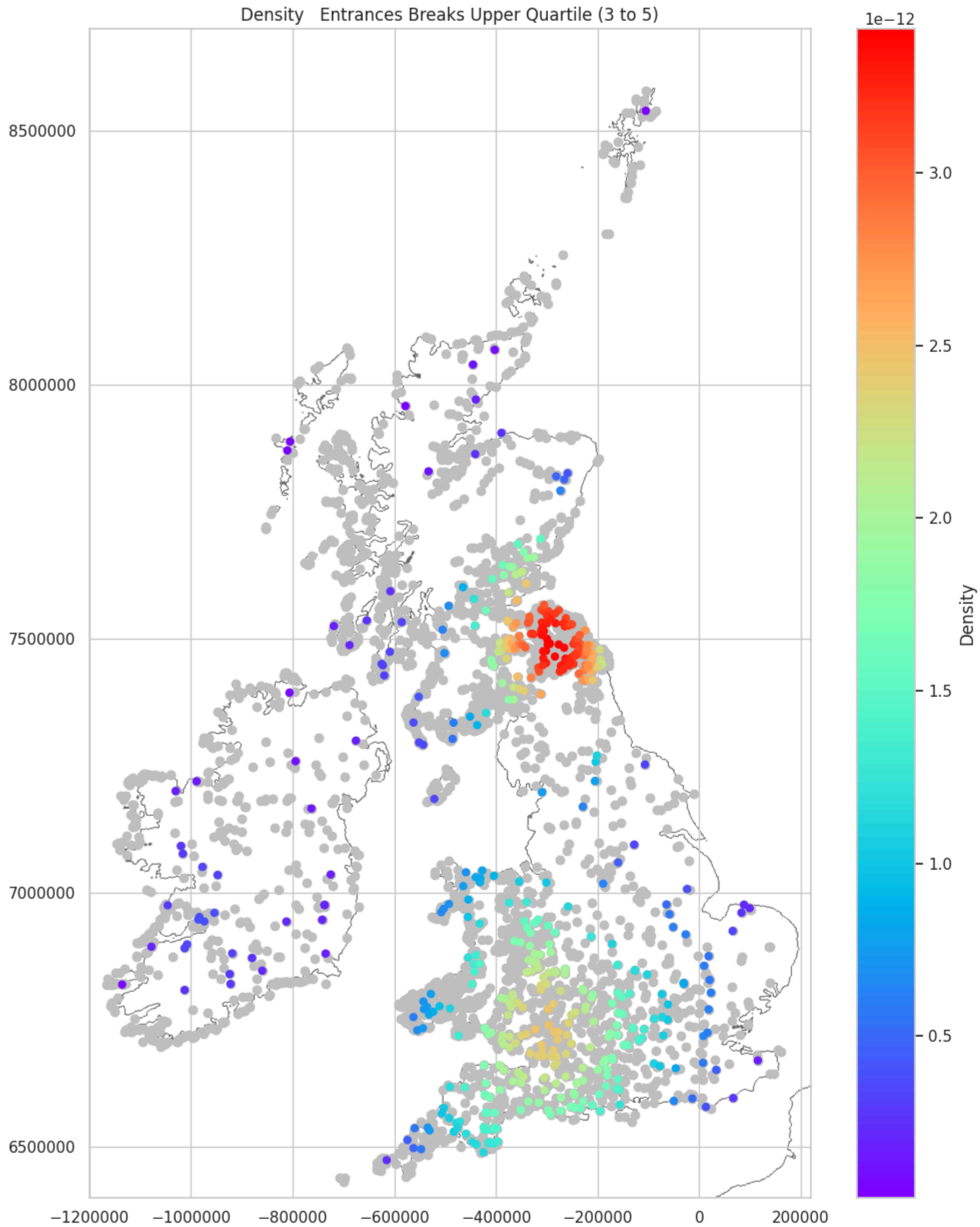
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
11.91%
```

Entrance Breaks Upper Quartile Density Mapped

Hillforts with three to five entrances are cluster in the Northeast and south, central England. There are very few in all other regions. In Ireland, most of this group are in the south.

In [ ]:
```
plot_density_over_grey(eb_upper_q, 'Entrances_Breaks', 'Upper Quartile (3 to 5)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Entrance Breaks Outlier Distribution Mapped

There is a small concentration of hillforts with six or more entrances in the south of England, near the ridge way, and a similar small concentration in the Northeast. Most are peppered over western England, Wales and eastern Scotland. There is a notable survey bias visible in the Northeastern data, as can be seen by the increased density of these hillforts to the south of the Scottish border, in Northumberland. There is a similar recording cluster around Oxford.

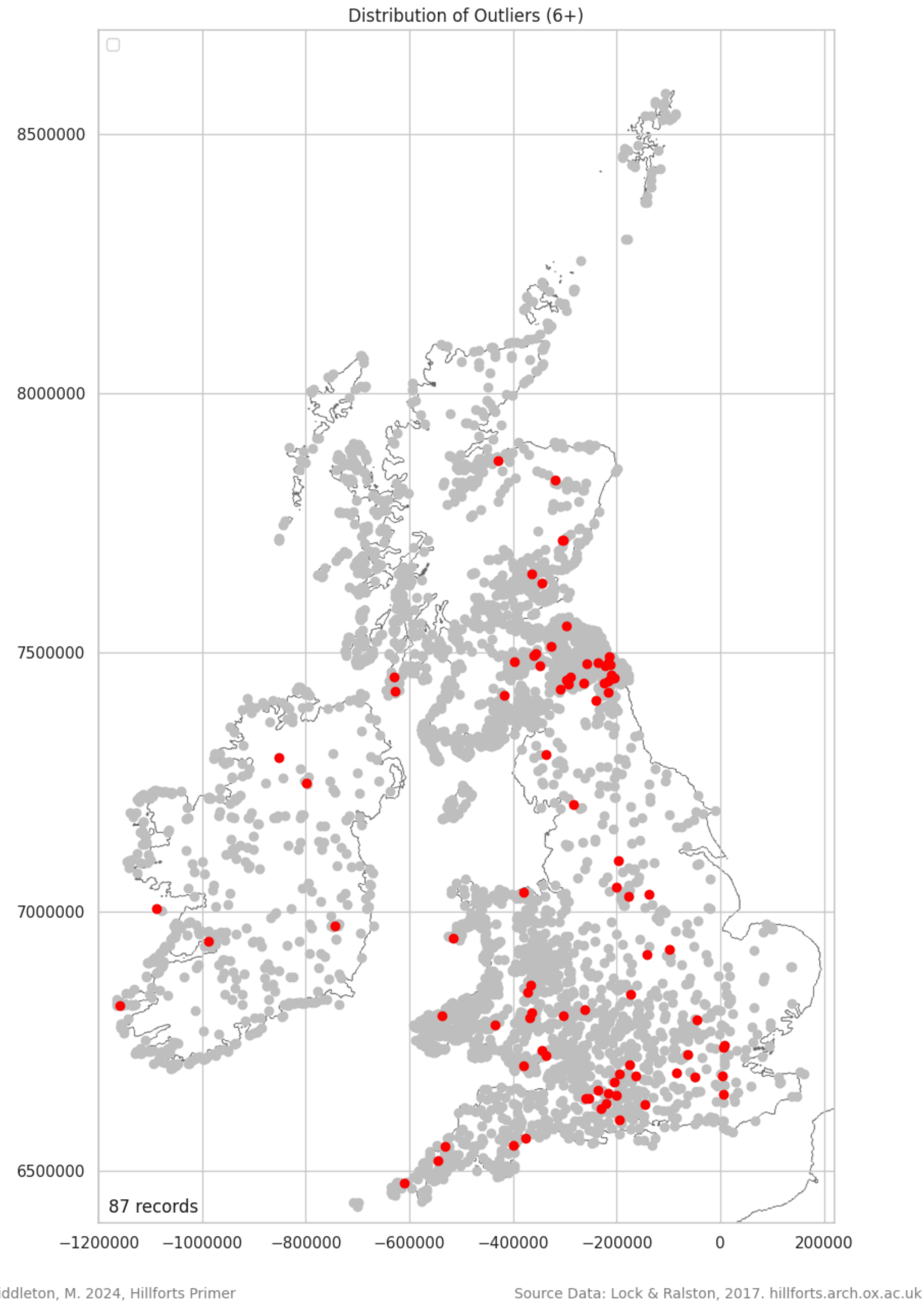
In [ ]:

```
print(f'{round(len(eb_out)/len(location_entrance_data)*100,2)}% of hillforts have six or more entrances (Outliers).'
```

```
2.1% of hillforts have six or more entrances (Outliers).
```

In [ ]:

```
plot_over_grey_numeric(eb_out, 'Entrances_Breaks', 'Distribution of Outliers (6+)')
```




```
2.1%
```

No Entrance Breaks Mapped

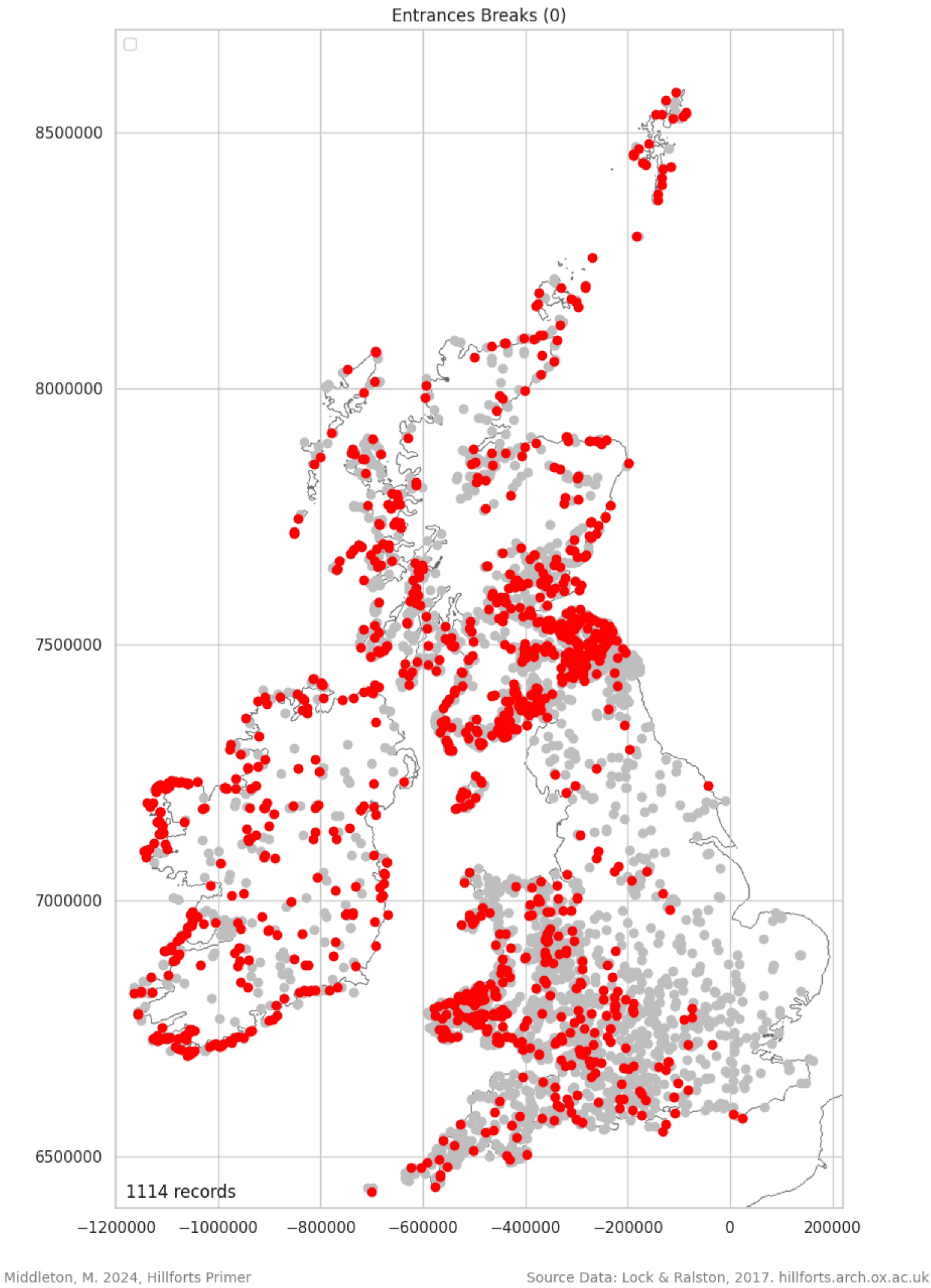
Just over a quarter of hillforts (26.86%) have no recorded entrance. These forts are most common at the northern end of the Northeastern cluster, in Pembrokeshire, up the southern end of the west coast of Scotland, over most of Ireland and peppered across the south west of England. Caution should be taken with regards the data in the Northeastern cluster in that, the southern boundary, between the intense concentration and no hillforts, is close to the England/Scotland border and it is likely that this reflects a recording bias in the data. If this is a recording bias, it does not replicate the bias, seen in other subsets of the data such as, Part 1: Main Boundary Mapped, where the modern border is clearly distinguishable. The fact that this line does not highlight the modern border and it does not mirror the distribution seen in Part1: Northeast Data Mapped, may indicate that this is a meaningful distribution yet, it is still more likely to be the result of a recording bias. Hillforts with no recorded entrance may indicate that this information has not been recorded or there is no evidence of an entrance.

In [ ]:
```
zero_enteances = \
location_entrance_data[location_entrance_data['Entrances_Breaks'] == 0].copy()
zero_enteances['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
print(f'{round(len(zero_enteances)/len(location_entrance_data)*100,2)}% of hillforts have no recorded entrance.')
```

```
26.86% of hillforts have no recorded entrance.
```

In [ ]:
```
zero_enteances_stats = plot_over_grey(zero_enteances, 'Entrances_Breaks', \
                                      'Yes', '(0)')
```




```
26.86%
```

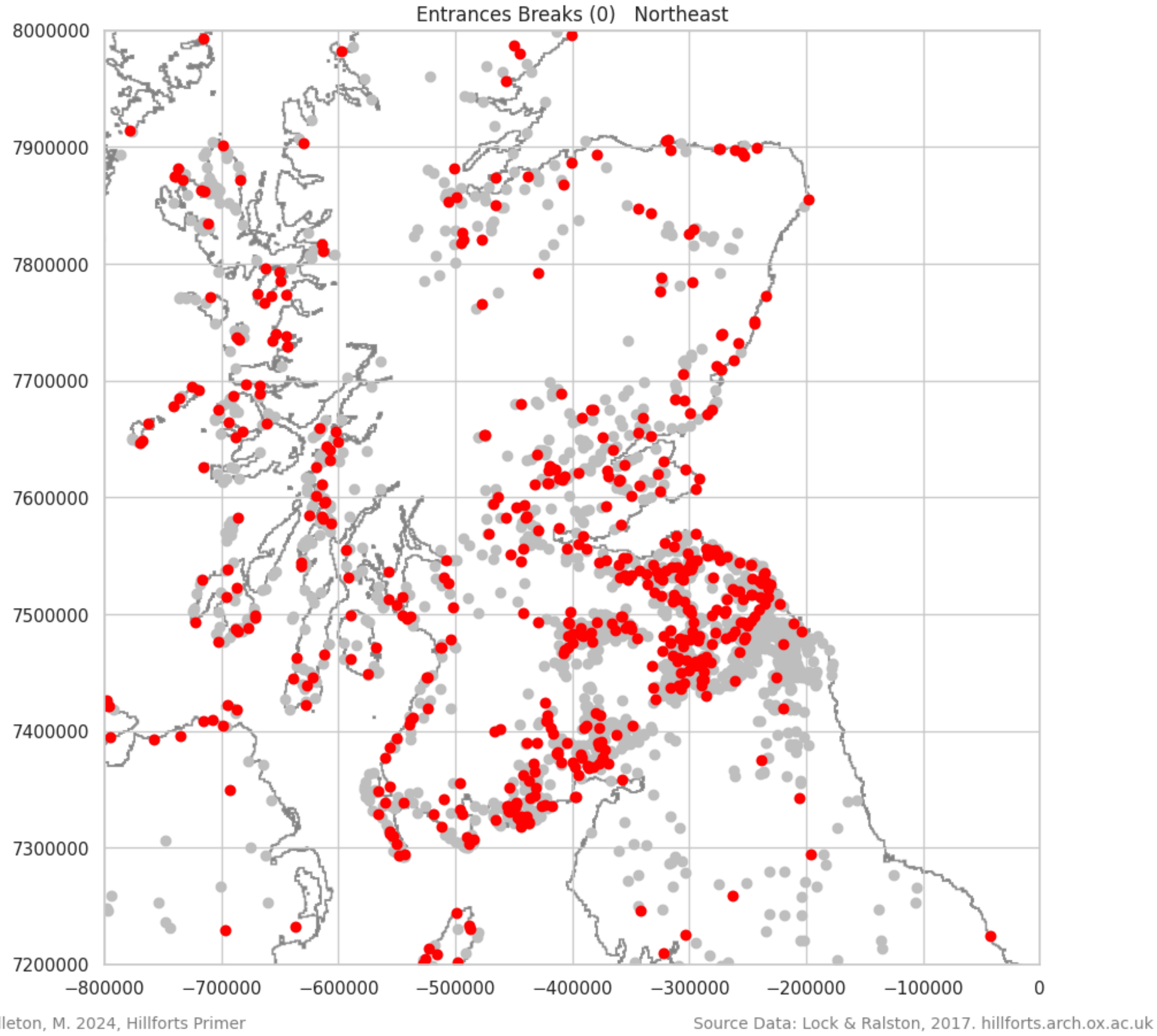
No Entrance Breaks Mapped (Northeast)

This figure shows the boundary between the high concentration of forts with no entrance breaks over the Southern Uplands and the abrupt line where this concentration stops, along the south side of this cluster.

In [ ]:
```
location_entrance_data_ne = \
location_entrance_data[location_entrance_data['Location_Y'] > 7070000].copy()
location_entrance_data_ne = \
location_entrance_data_ne[location_entrance_data_ne['Location_X'] > -800000].copy()
```

```
no_entrances_ne = \
location_entrance_data_ne[location_entrance_data_ne['Entrances_Breaks'] == 0].copy()
no_entrances_ne['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
no_entrances_stats_ne = plot_over_grey_north(no_entrances_ne, \
                                             'Entrances_Breaks', 'Yes', \
                                             '(0) - Northeast')
```

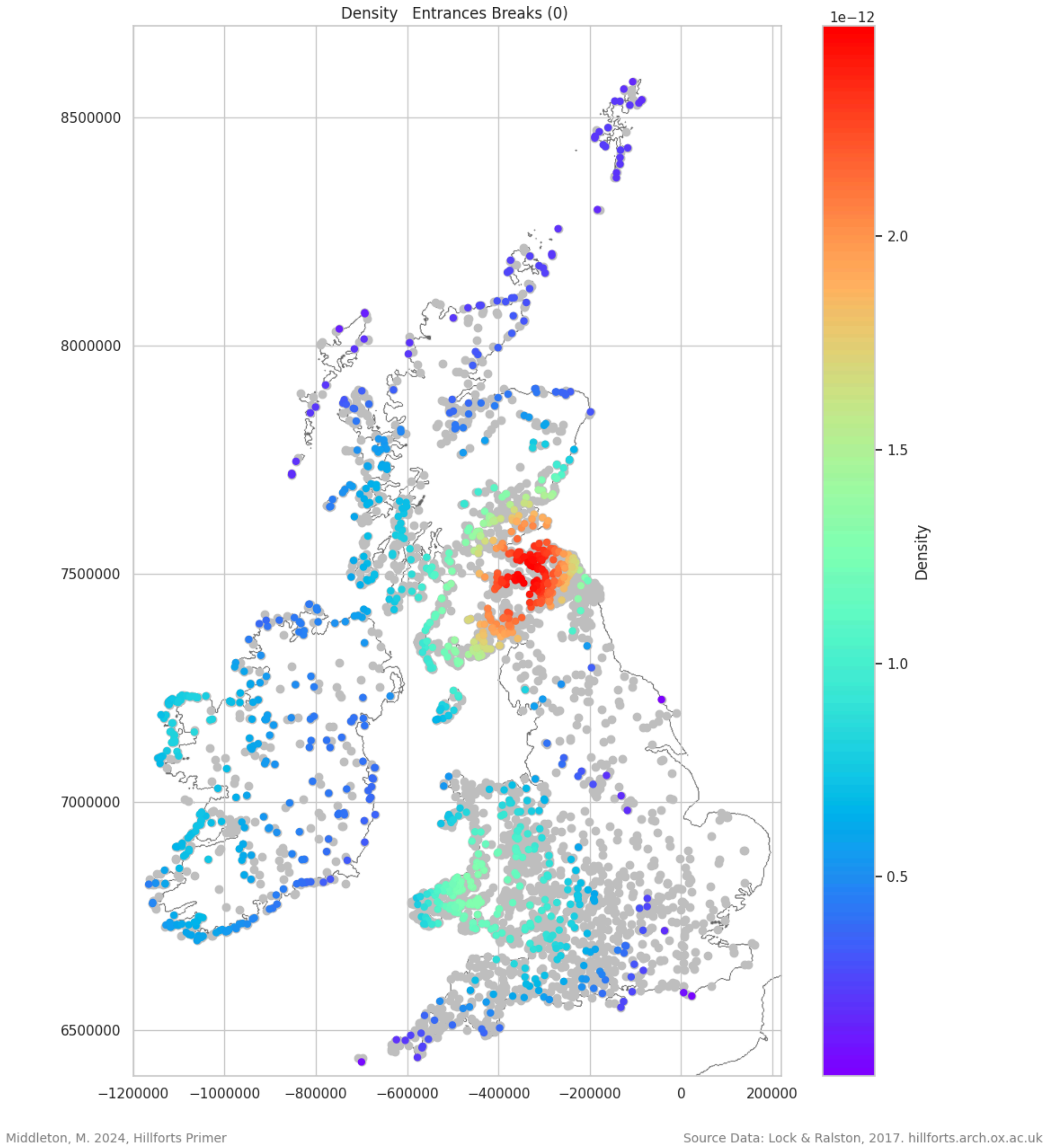


In [ ]:
```
# This code can be used to get details of hillforts within certain x and y coordinate ranges
# To use this code, first run the document using Runtime > Run all, then remove the '#' from the lines
# starting temp below. Once removed press the Run cell button, on this cell, to the left.
# Update the 'Location_X' & 'Location_Y' values as required.
# temp = pd.merge(name_and_number, no_entrances_ne, left_index=True, right_index=True)
# temp = temp[temp['Location_X'].between(-250000, -200000)]
# temp = temp[temp['Location_Y'].between(7500000, 7510000)]
# temp.sort_values(by=['Location_X'], ascending=False)
```

No Entrance Breaks Density Mapped

All five clusters identified in, Part 1: Density Map showing Extent of Boxplots identified in the Atlas Data, can be seen in this subset of the data.

In [ ]:
```
plot_density_over_grey(zero_enteances_stats, 'Entrances_Breaks (0)')
```



One Entrance Break Distribution Mapped

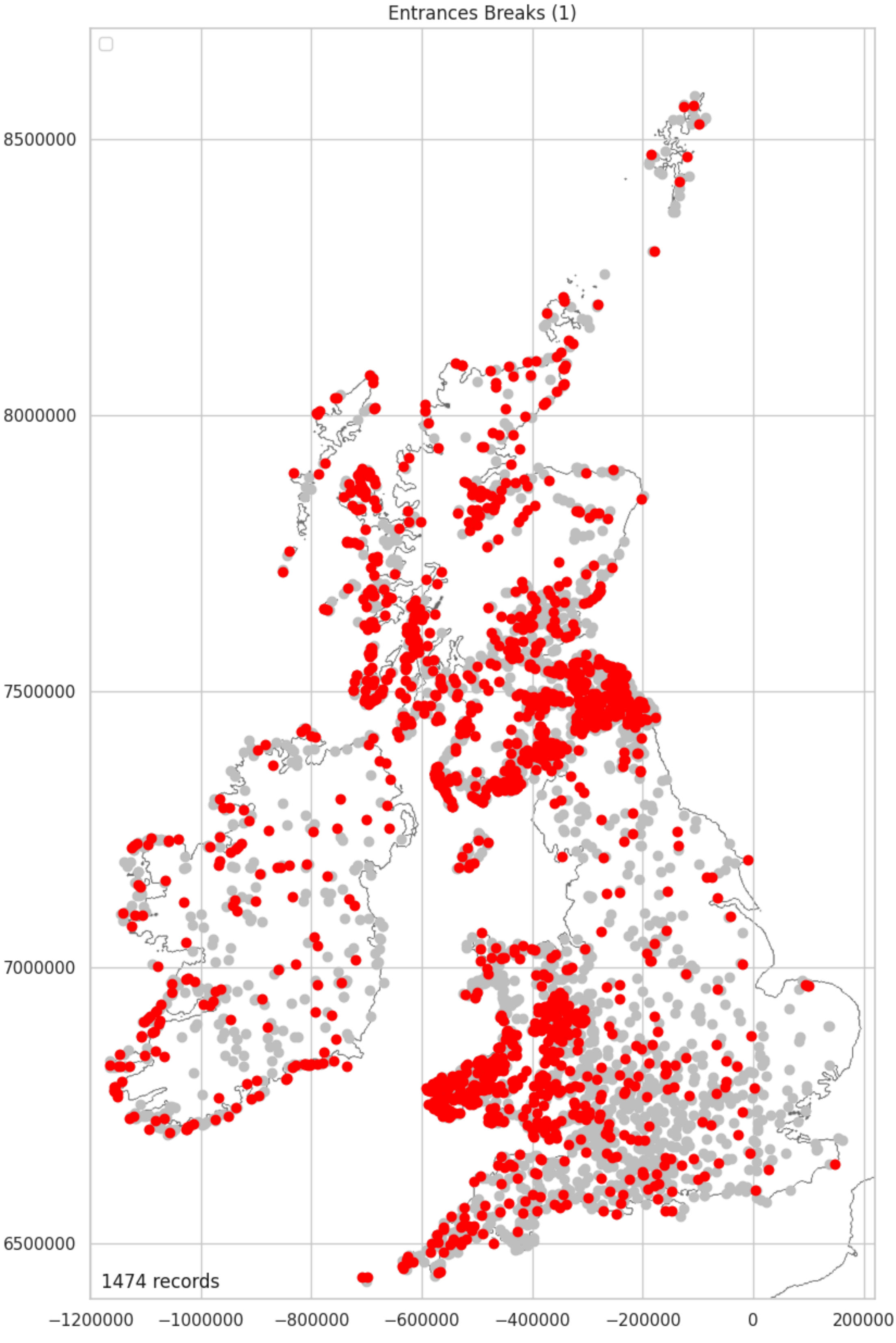
35.54% of hillforts have one recorded entrance. It is noticable how few there are in south central England and northern Wales compared to how many there are over the Shropshire hills and southern Wales. The distinct difference between these areas may indicate a survey bias.

In [ ]:
```
one_entrance = \
location_entrance_data[location_entrance_data['Entrances_Breaks'] == 1].copy()
one_entrance['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
print(f'{round(len(one_entrance)/len(location_entrance_data)*100,2)}% of hillforts have one recorded entrance.')
```

```
35.54% of hillforts have one recorded entrance.
```

In [ ]:
```
one_entrance_stats = \
plot_over_grey(one_entrance, 'Entrances_Breaks', 'Yes', '(1)')
```



Middleton, M. 2024, Hillforts Primer

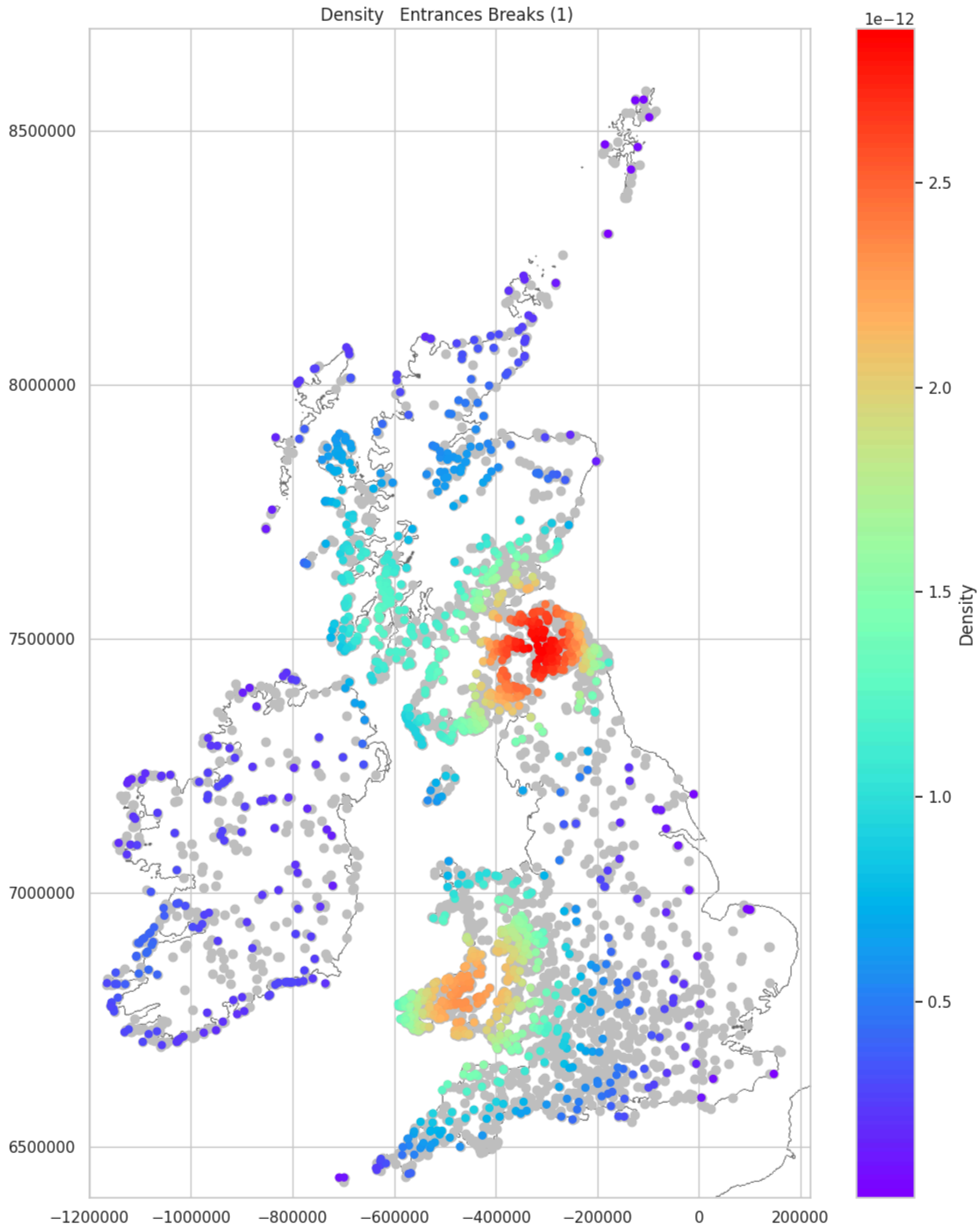
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
35.54%
```

One Entrance Break Density Mapped

Single entrance hillforts are concentrated over the Southern Uplands, the southern Welsh uplands and along the south-western seaboard of Scotland. There is also a notable spread of these forts along the south-west of England and across central and western Ireland as well as clustering along the coasts of northern Scotland and south-western Ireland.

```
In [ ]: plot_density_over_grey(one_entrance_stats, 'Entrances_Breaks (1)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Two Enrance Breaks Distribution Mapped

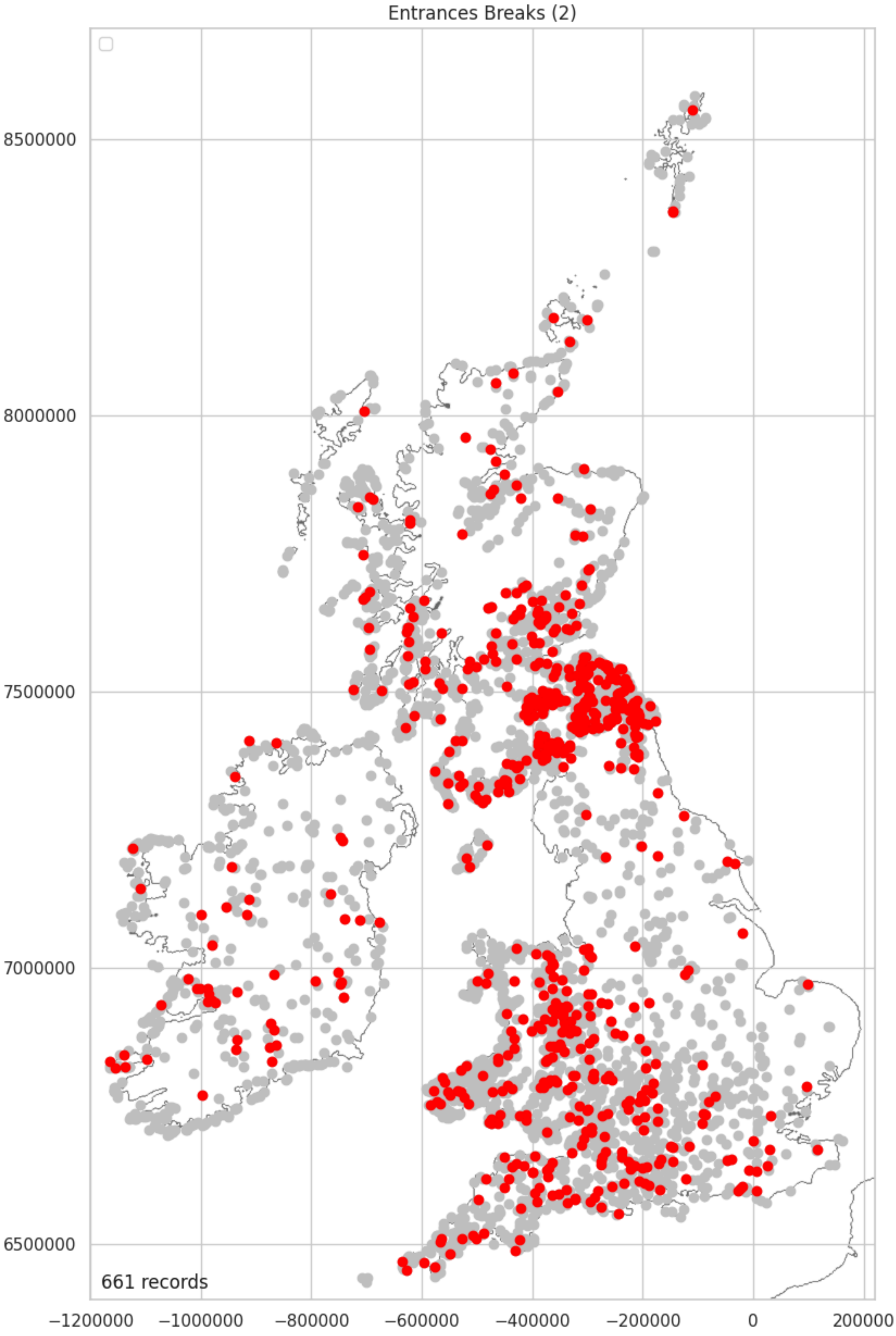
The distribution of two-entrance hillforts is more discreetly concentrated over the eastern Southern Uplands and to the east of the Cambrian Mountains. Interestingly, possible linear alignments of hillforts can be seen in the south of England, with the Ridgeway being the most prominent, running from the Chiltern Hills to Lyme Bay.

In [ ]:
```
two_entrances = \
location_entrance_data[location_entrance_data['Entrances_Breaks'] == 2].copy()
two_entrances['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
print(f'{round(len(two_entrances)/len(location_entrance_data)*100,2)}% of hillforts have two entrances.')
```

```
15.94% of hillforts have two entrances.
```

In [ ]:
```
two_entrances_stats = plot_over_grey(two_entrances, 'Entrances_Breaks', \
                                     'Yes', '(2)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
15.94%
```

Two Entrance Breaks (South) & Possible Corolation to Linear Routes Mapped

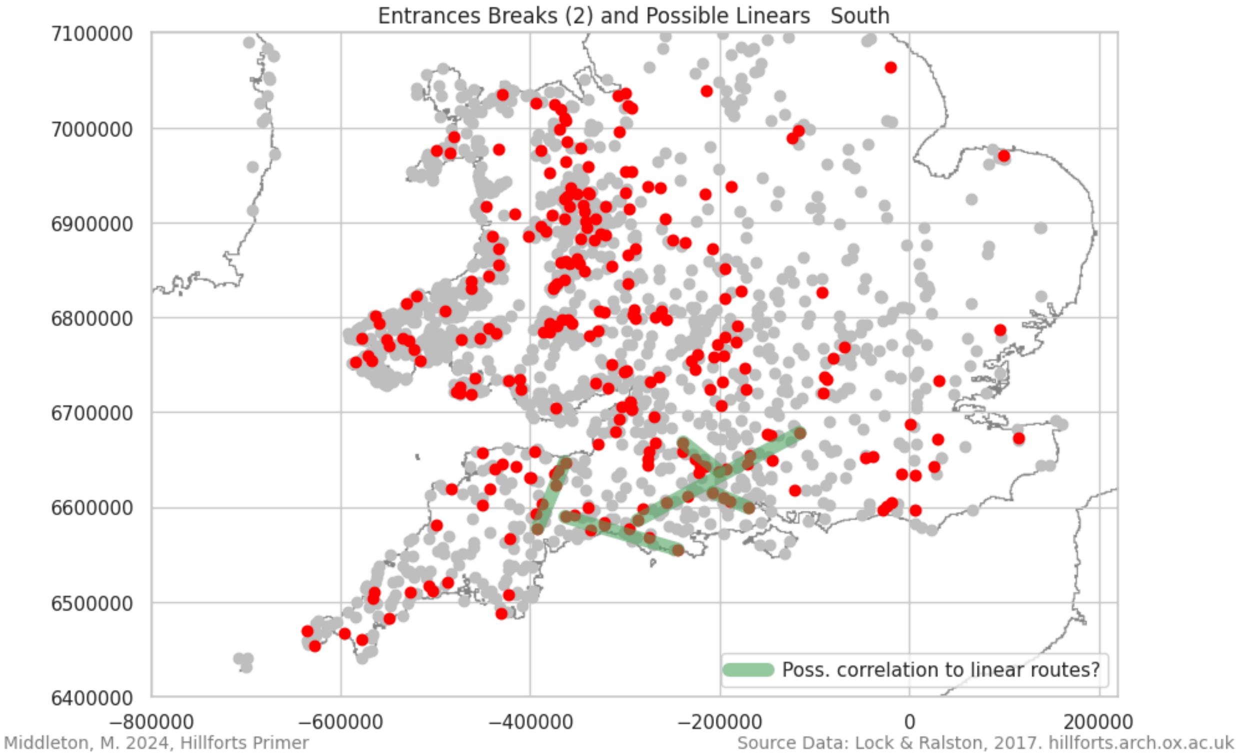
An enlarged extract over the south showing some of the possible linear alignments of hillforts which may be highlighting routes and paths in this area.

In [ ]:

```
location_entrance_data_s = \
location_entrance_data[location_entrance_data['Location_Y'] < 7070000].copy()
location_entrance_data_s = \
location_entrance_data_s[location_entrance_data_s['Location_X'] > -700000].copy()
```

```
two_entrances_south = \
location_entrance_data_s[location_entrance_data_s['Entrances_Breaks'] == 2].copy()
two_entrances_south['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
two_entrances_stats_s = \
plot_over_grey_south(two_entrances_south, 'Entrances_Breaks', 'Yes', \
                     '(2) and Possible Linears - South')
```

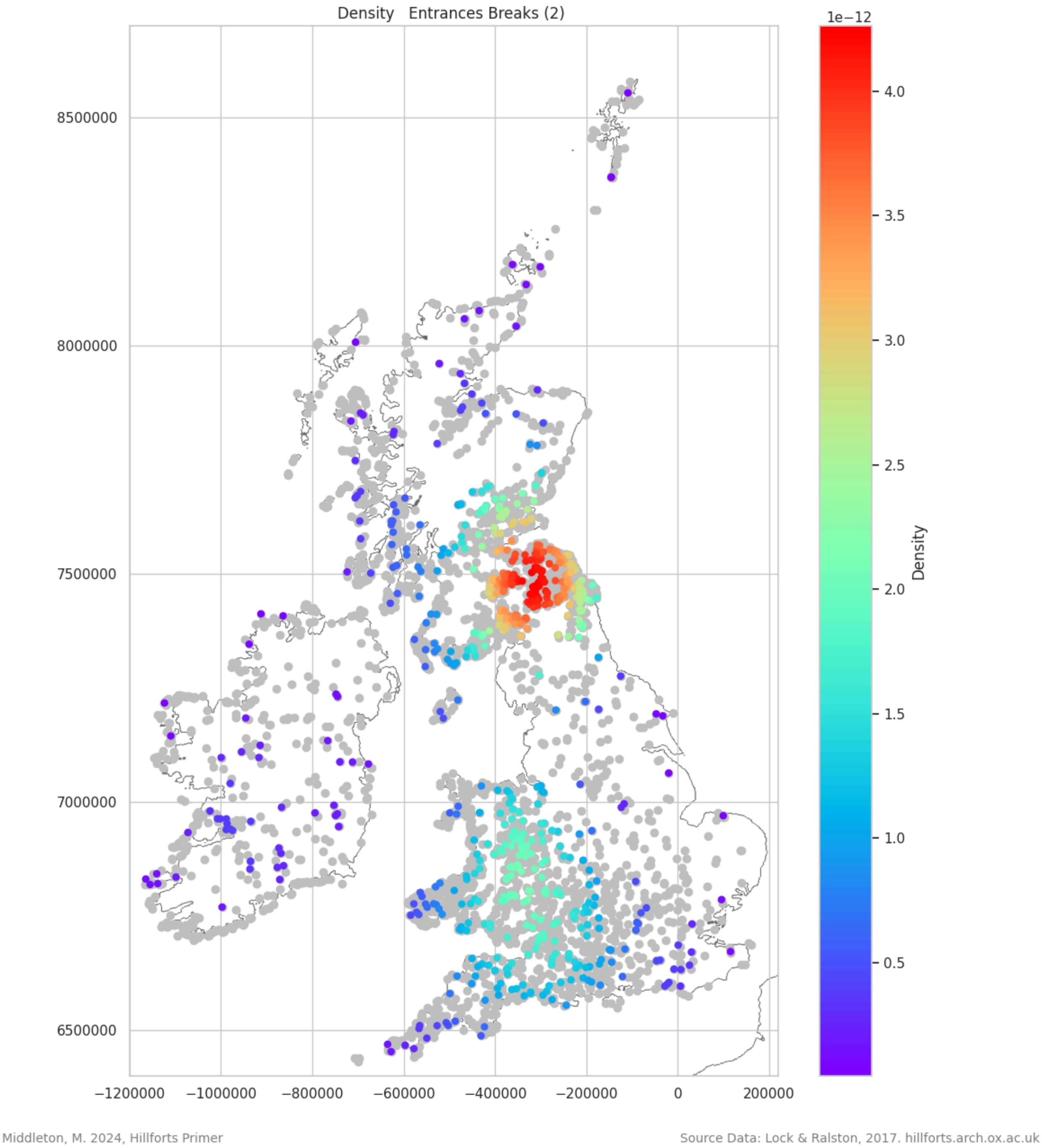


In [ ]:
```
# This code can be used to get details of hillforts within certain x and y coordinate ranges
# To use this code, first run the document using Runtime > Run all, then remove the '#' from the lines
# starting temp below. Once removed press the Run cell button, on this cell, to the left.
# Update the 'Location_X' & 'Location_Y' values as required.
# temp = pd.merge(name_and_number, two_entrances_south, left_index=True, right_index=True)
# temp = temp[temp['Location_X'].between(-210000, -200000)]
# temp = temp[temp['Location_Y'].between(6620000, 6640000)]
# temp
```

Two Entrance Breaks Density Mapped

The focus of two entrance forts is in the Northeast and from the north end of the Cambrian Mountains, then along the eastern fringes of the Cambrian Mountains, down to the western end of south, central England. It is notable that there are almost none of this type around the Irish coast.

In [ ]:
```
plot_density_over_grey(two_entrances_stats, 'Entrances_Breaks (2)')
```



Three Entrance Breaks Distribution Mapped

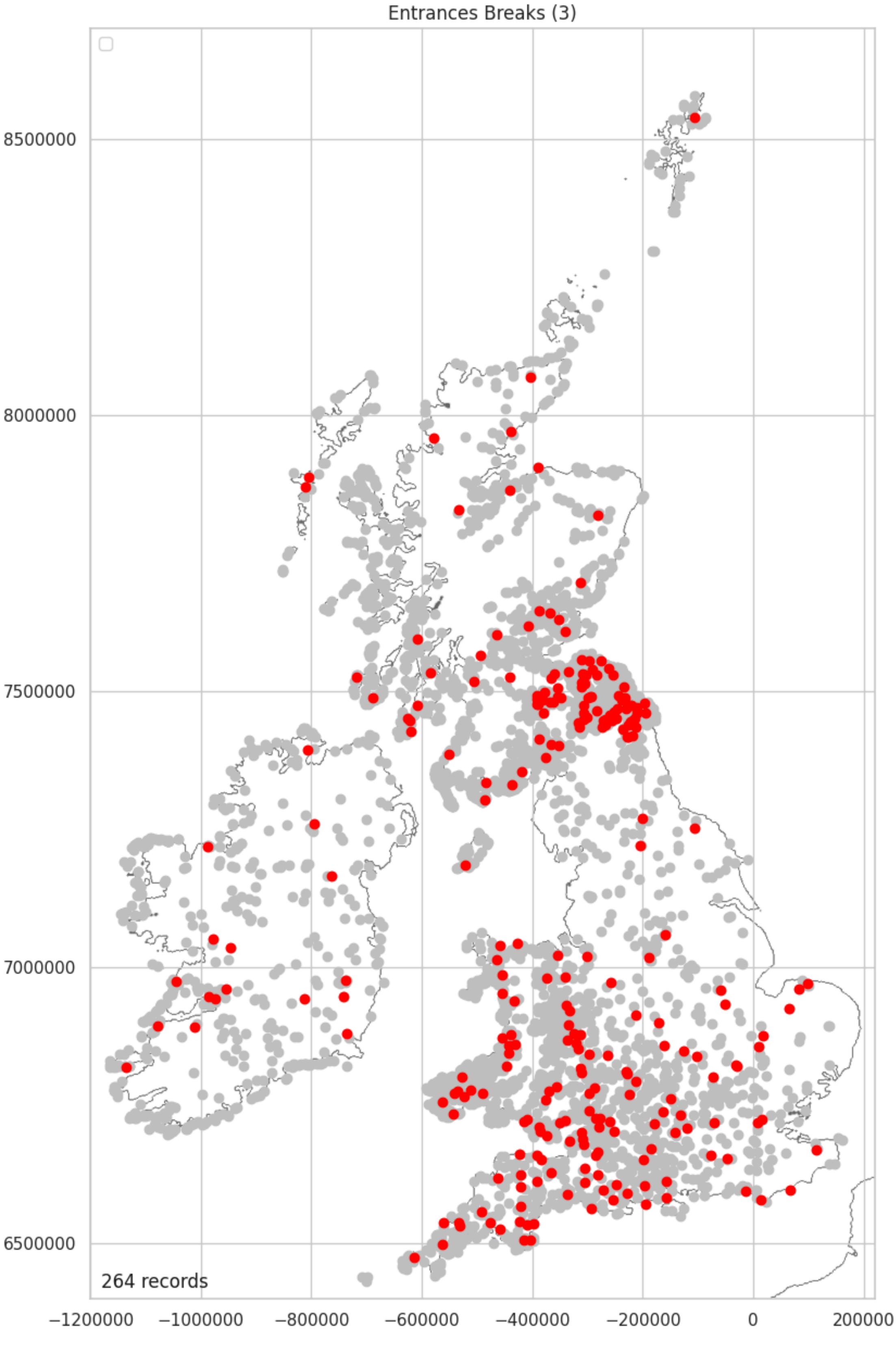
Three entrance forts show a similar distribution to two entrance forts except the focus, in Wales, is now toward the eastern side of the Brecon Beacons. What appears to be a hole at the centre of the Northeast data cluster reflects the local topography with the highlighted forts sitting on the higher ground and the void being the lowland of the Tweed Basin.

In [ ]:
```
three_entrances = \
location_entrance_data[location_entrance_data['Entrances_Breaks'] == 3].copy()
three_entrances['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
print(f'{round(len(three_entrances)/len(location_entrance_data)*100,2)}% of hillforts have three entrances.')
```

```
6.37% of hillforts have three entrances.
```

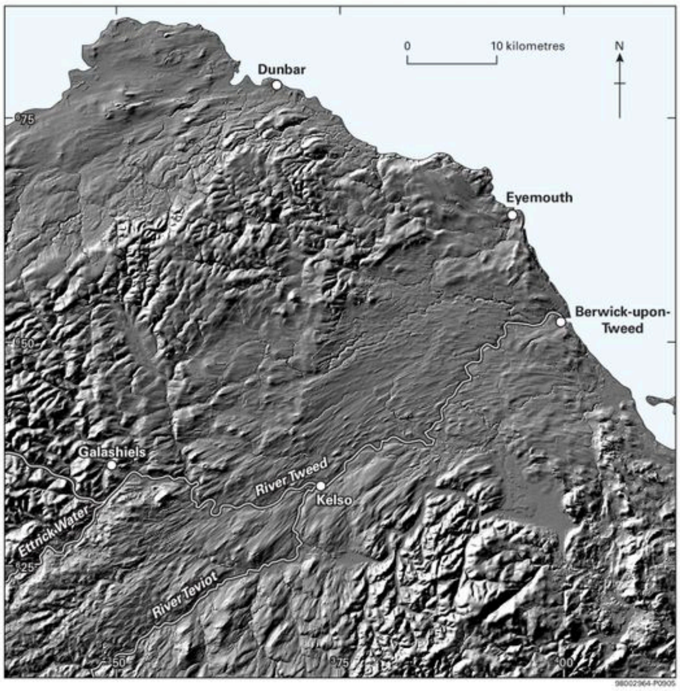
In [ ]:
```
three_entrances_stats = \
plot_over_grey(three_entrances, 'Entrances_Breaks', 'Yes', '(3)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

6.37%

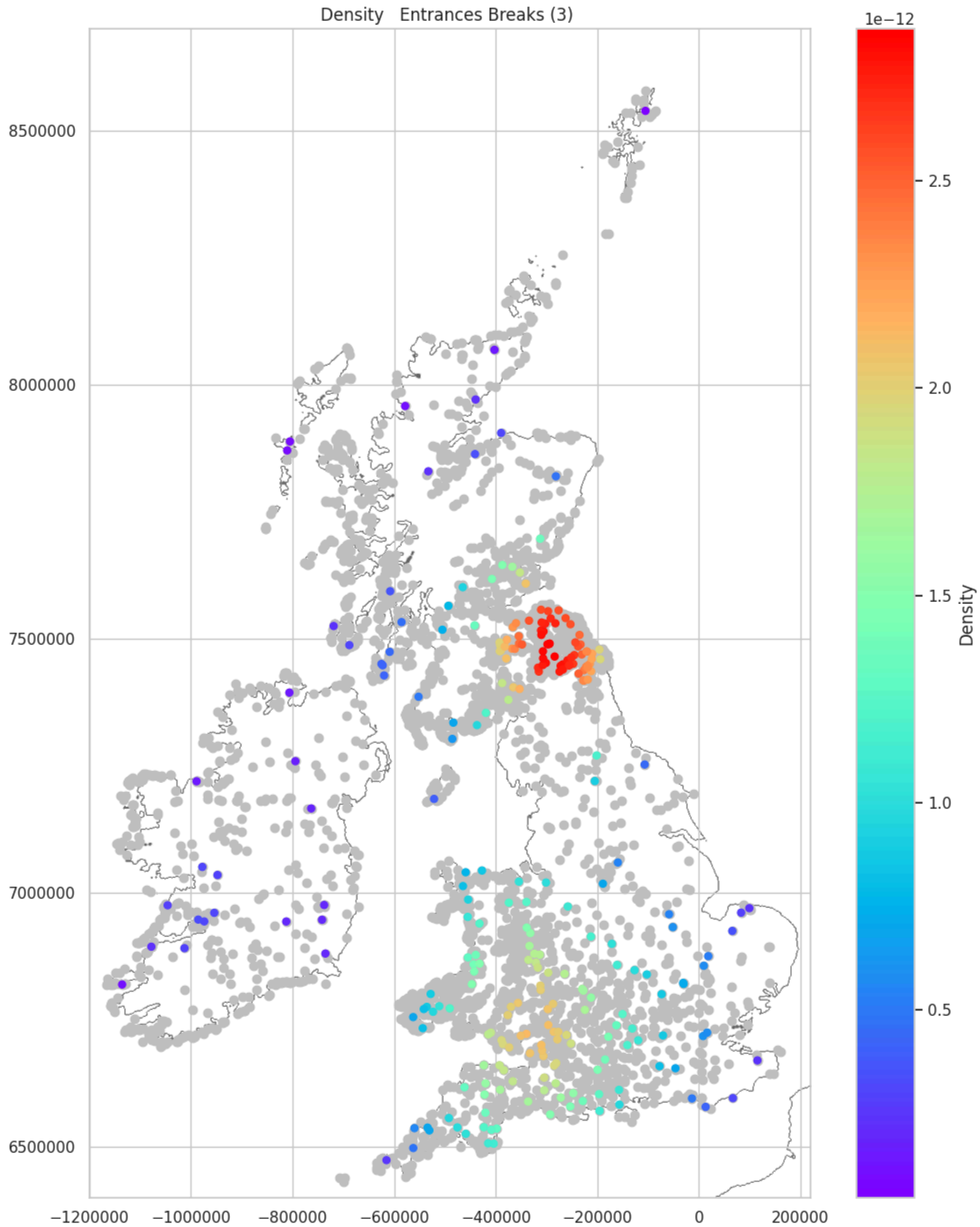


The Tweed Basin
Copyright: British Geological Survey (P912371)
(For use in private study or research for a non-commercial purpose)

Three Entrance Breaks Density Mapped

The density of three entrance hillforts shows a focus over the Northeast. In the south, the distribution is sparse and here the focus of the cluster is toward the River Severn. There are very few of this type out with these two clusters.

In [ ]:
```
plot_density_over_grey(three_entrances_stats, 'Entrances_Breaks (3)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Four Entrance Breaks Distribution Mapped

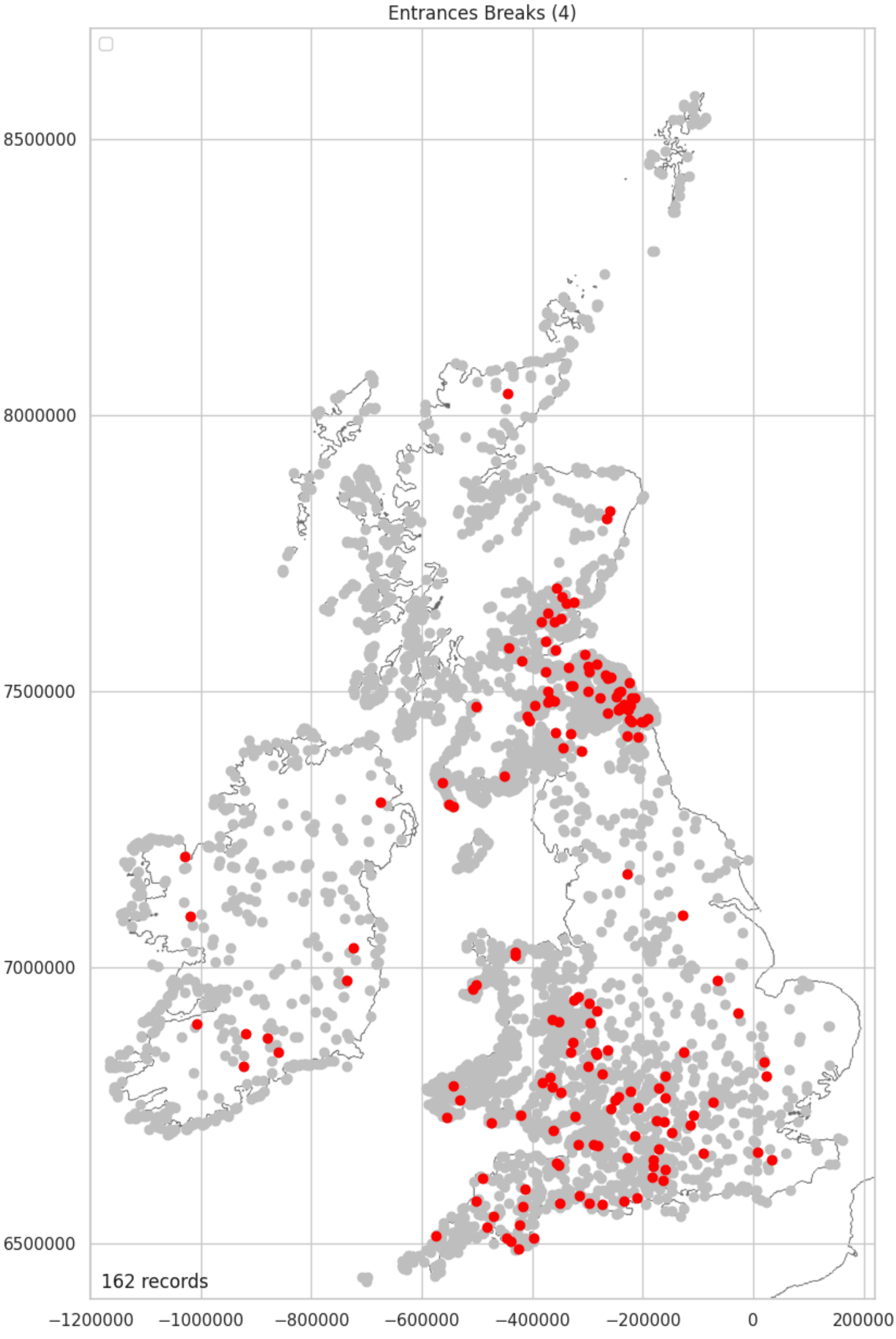
Four entrance forts are almost exclusively located in the Northeast and south central England.

In [ ]:
```
four_entrances = \
location_entrance_data[location_entrance_data['Entrances_Breaks'] == 4].copy()
four_entrances['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
print(f'{round(len(four_entrances)/len(location_entrance_data)*100,2)}% of hillforts have four entrances.')
```

```
3.91% of hillforts have four entrances.
```

In [ ]:
```
four_entrances_stats = \
plot_over_grey(four_entrances, 'Entrances_Breaks', 'Yes', '(4)')
```



Middleton, M. 2024, Hillforts Primer

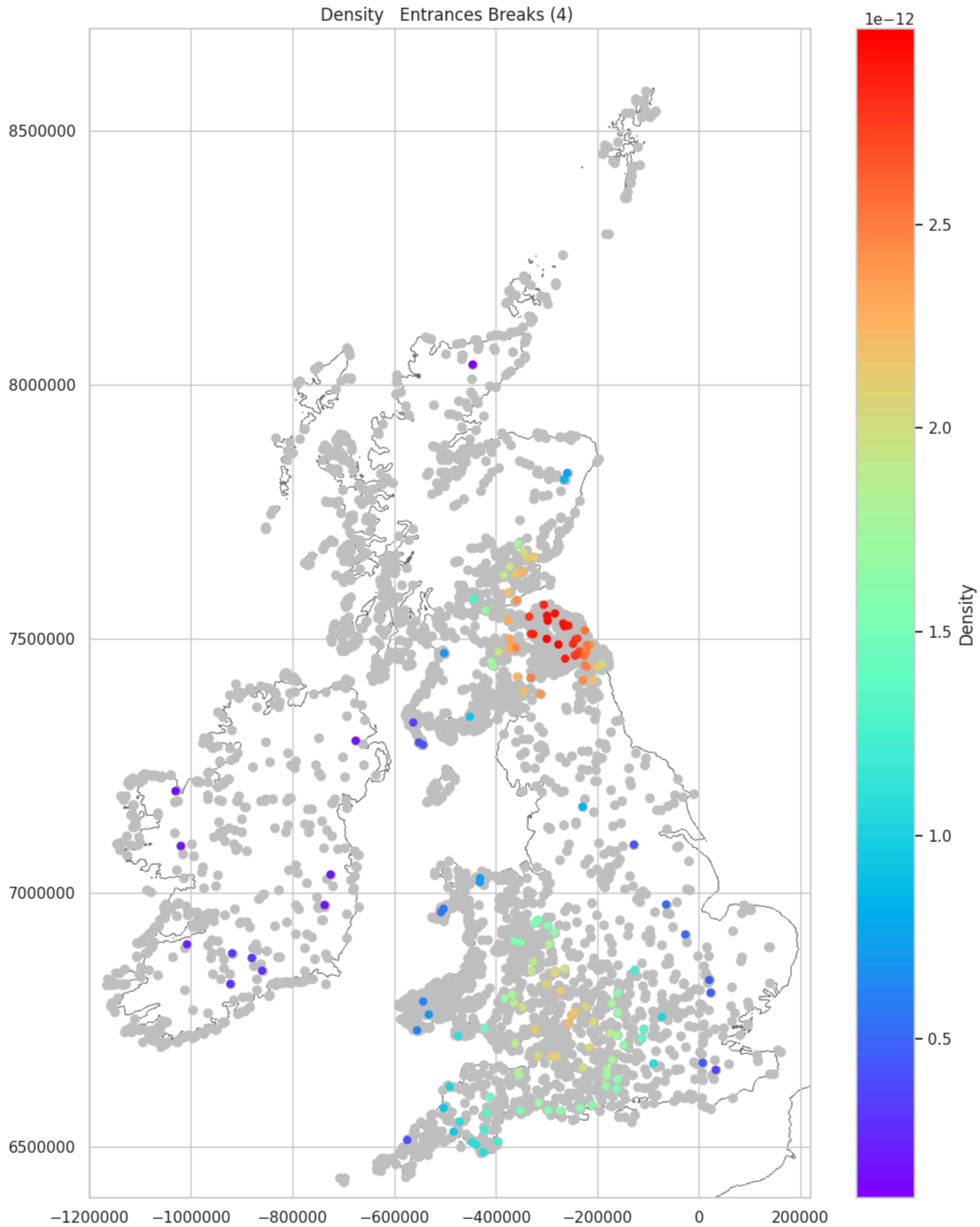
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
3.91%
```

Four Entrance Breaks Density Mapped

The Northeast is the primary focus for four entrance forts. In the South, there is a slight cluster around the River Severn. See Three Entrance Breaks Density Mapped.

```
In [ ]: plot_density_over_grey(four_entrances_stats, 'Entrances_Breaks (4)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Four Entrance Breaks Distribution Mapped (Northeast)

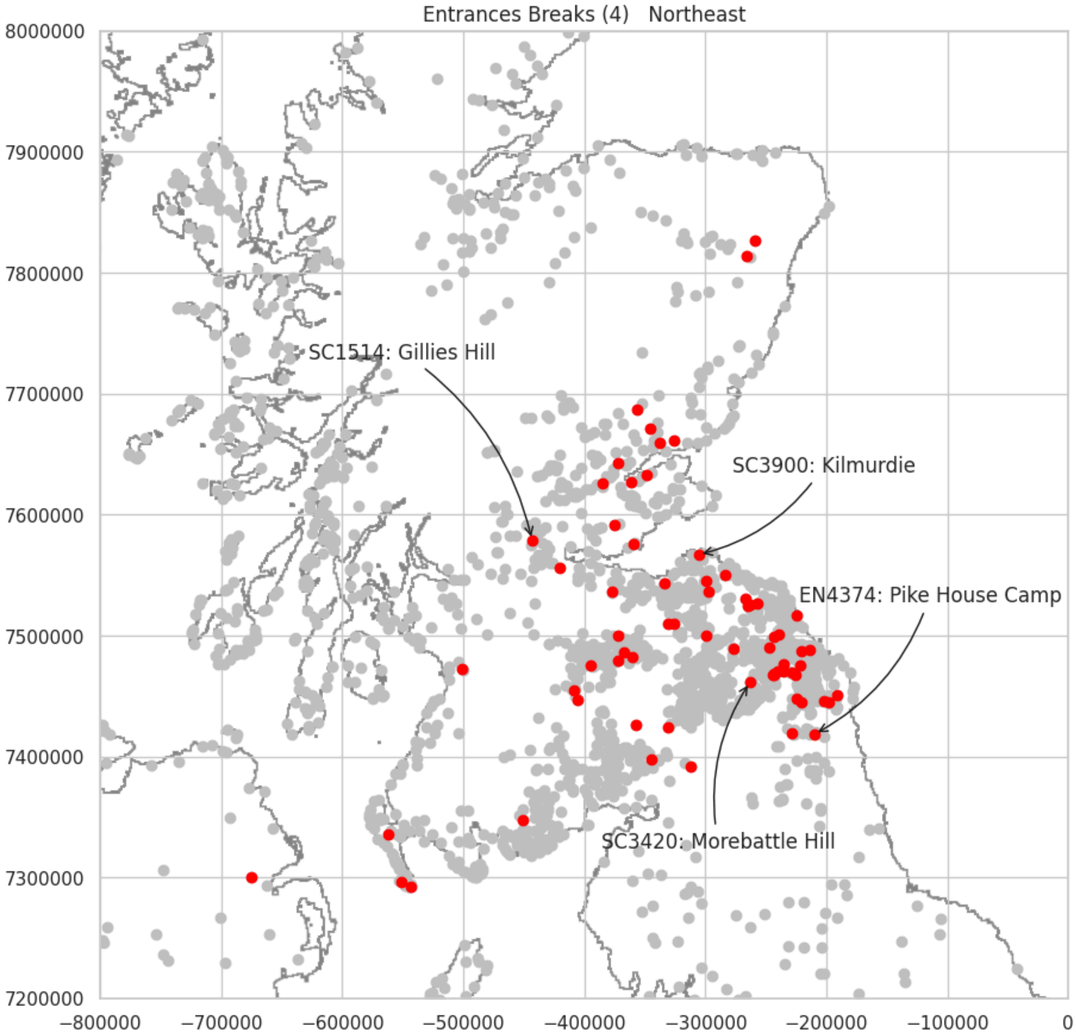
In the Northeast, four entrance forts show hints of alignment. Once such alignment seems to run from Gillies Hill to Morebattle Hill at roughly 30 to 40 km intervals. Another, less defined alignment, looks to run from Pike House Camp, up toward Kilmurdie. There is also a notable cluster around the mouth of the Tay, just north and south of Perth.

In [ ]:

```
four_entrances_ne = \
location_entrance_data_ne[location_entrance_data_ne['Entrances_Breaks'] == 4].copy()
four_entrances_ne['Entrances_Breaks'] = "Yes"
```

In [ ]:

```
four_entrances_stats_ne = \
plot_over_grey_north(four_entrances_ne, 'Entrances_Breaks', 'Yes', \
                     '(4) - Northeast', 'Stirling')
```



Middleton, M. 2024, Hillforts Primer Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

In [ ]:
```
# This code can be used to get details of hillforts within certain x and y coordinate ranges
# To use this code, first run the document using Runtime > Run all, then remove the '#' from the lines
# starting temp below. Once removed press the Run cell button, on this cell, to the left.
# Update the 'Location_X' & 'Location_Y' values as required.
# temp = pd.merge(name_and_number, four_entrances_stats_ne, left_index=True, right_index=True)
# temp = temp[temp['Location_X'].between(-400000, -300000)]
# temp = temp[temp['Location_Y'].between(7600000, 7700000)]
# temp
```

In [ ]:
```
dist = int(np.sqrt( (-443187 - -419817)**2 + (7578896 - 7556141)**2))
dist
```

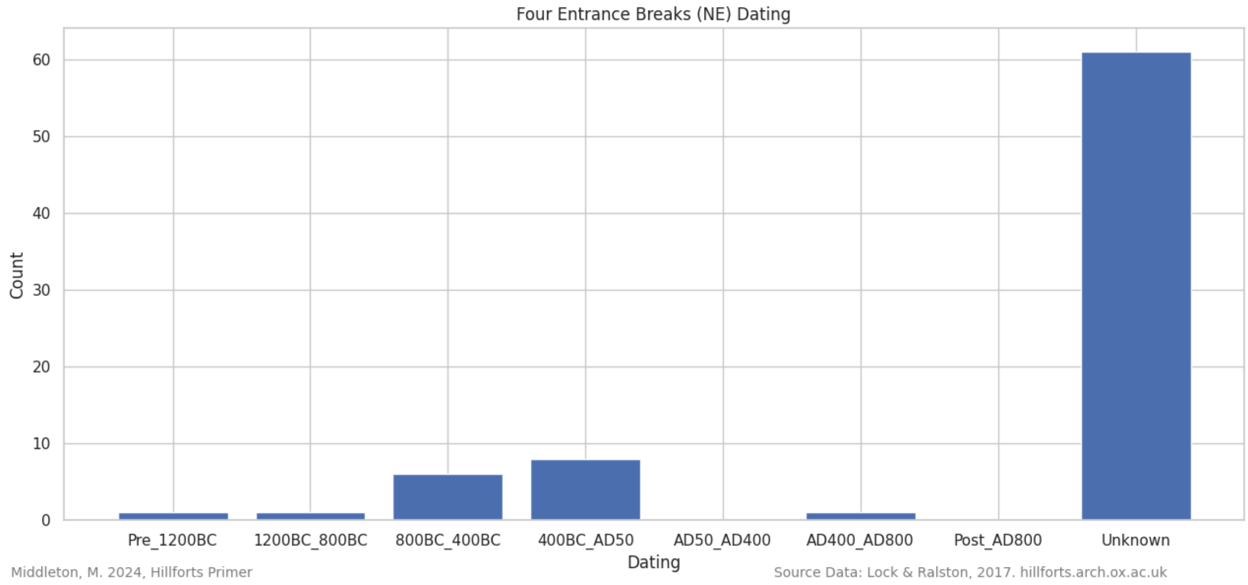
Out[ ]: 32618

Four Entrance Breaks Dating (Northeast)

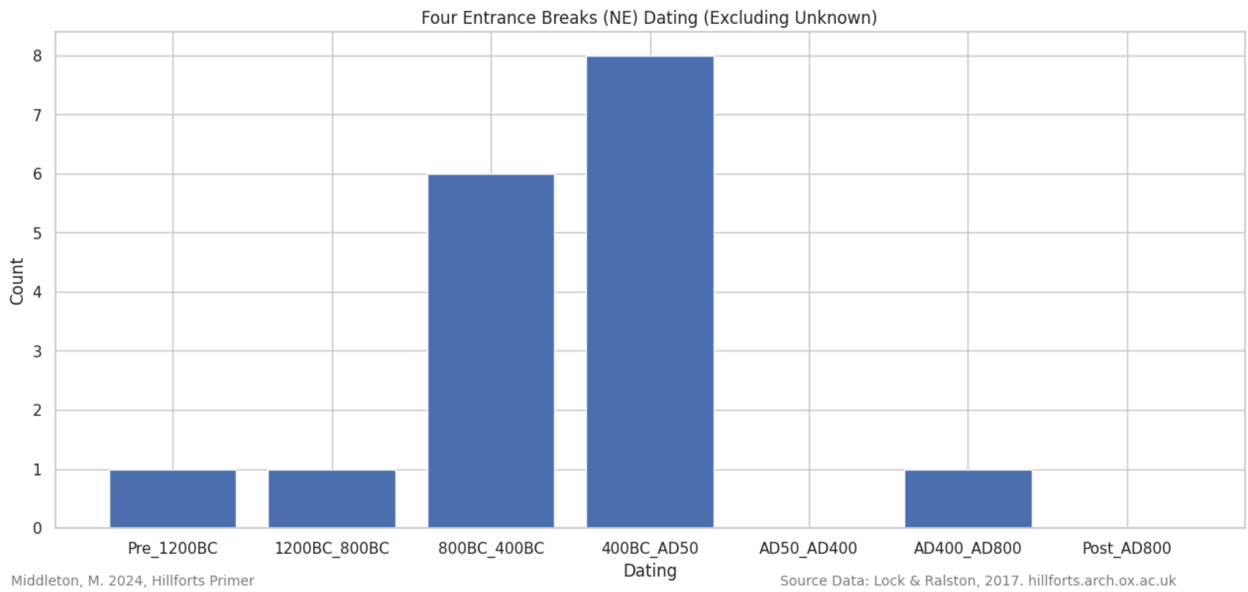
It was considered that the alignment of four entrance forts, and there being a possible relationship between them, might hint at these forts having a different period of construction. In terms of dating the majority of the four entrance breaks hillforts in the Northeast are undated. Of those that are, almost all have dates ranging between 800BC to AD50. There is an interesting lack of dates in the range AD50 to AD400 although it is important to note that the total count of dates is very low and the general distribution of dates is in line with those seen for all hillforts. There is no dating evidence to suggest these forts are related to a different period of construction or reuse.

In [ ]:
```
four_entrances_ne_dates = \
pd.merge(four_entrances_ne, date_data, left_index=True, right_index=True)
```

In [ ]:
```
plot_bar_chart(four_entrances_ne_dates[date_features], 2, 'Dating', 'Count', \
               'Four Entrance Breaks (NE) Dating')
```




```
In [ ]: plot_bar_chart(four_entrances_ne_dates[date_features], 2, 'Dating', 'Count', \
               'Four Entrance Breaks (NE) Dating (Excluding Unknown)', True)
```



Five Entrance Breaks Distribution Mapped

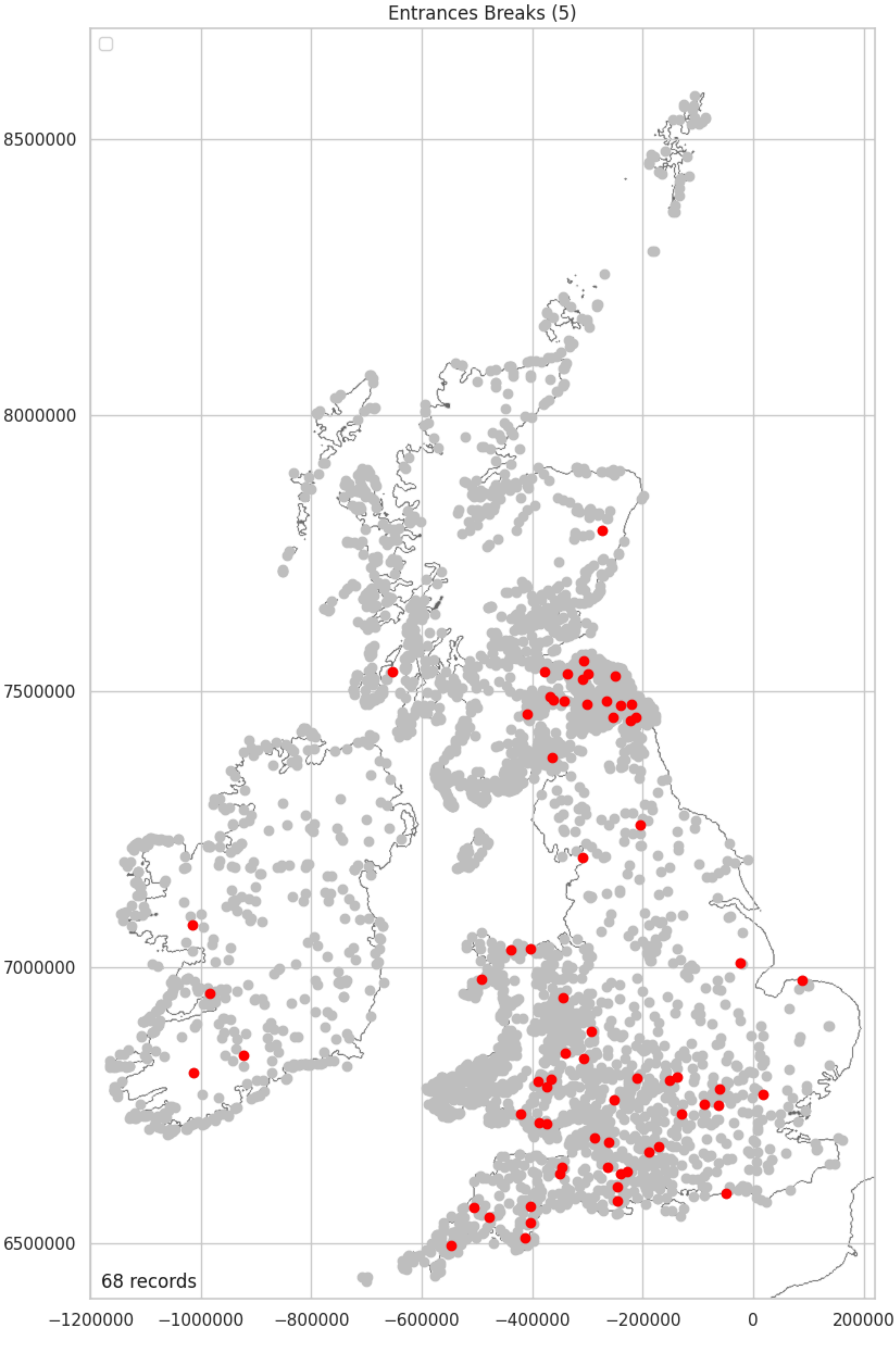
As with three and four entrances above, the Northeast and south central to south west England are the main areas where hillforts with five entrances cluster.

```
In [ ]: five_entrances = \
        location_entrance_data[location_entrance_data['Entrances_Breaks'] == 5].copy()
        five_entrances['Entrances_Breaks'] = "Yes"
```

```
In [ ]: print(f'{round(len(five_entrances)/len(location_entrance_data)*100,2)}% of hillforts have five entrances.')
```

```
1.64% of hillforts have five entrances.
```

```
In [ ]: five_entrances_stats = \
        plot_over_grey(five_entrances, 'Entrances_Breaks', 'Yes', '(5)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
1.64%
```

Entrance Breaks Not Recorded Distribution Mapped

All but one fort in the Northwest and a couple in the Isle of Man, are in England and Wales.

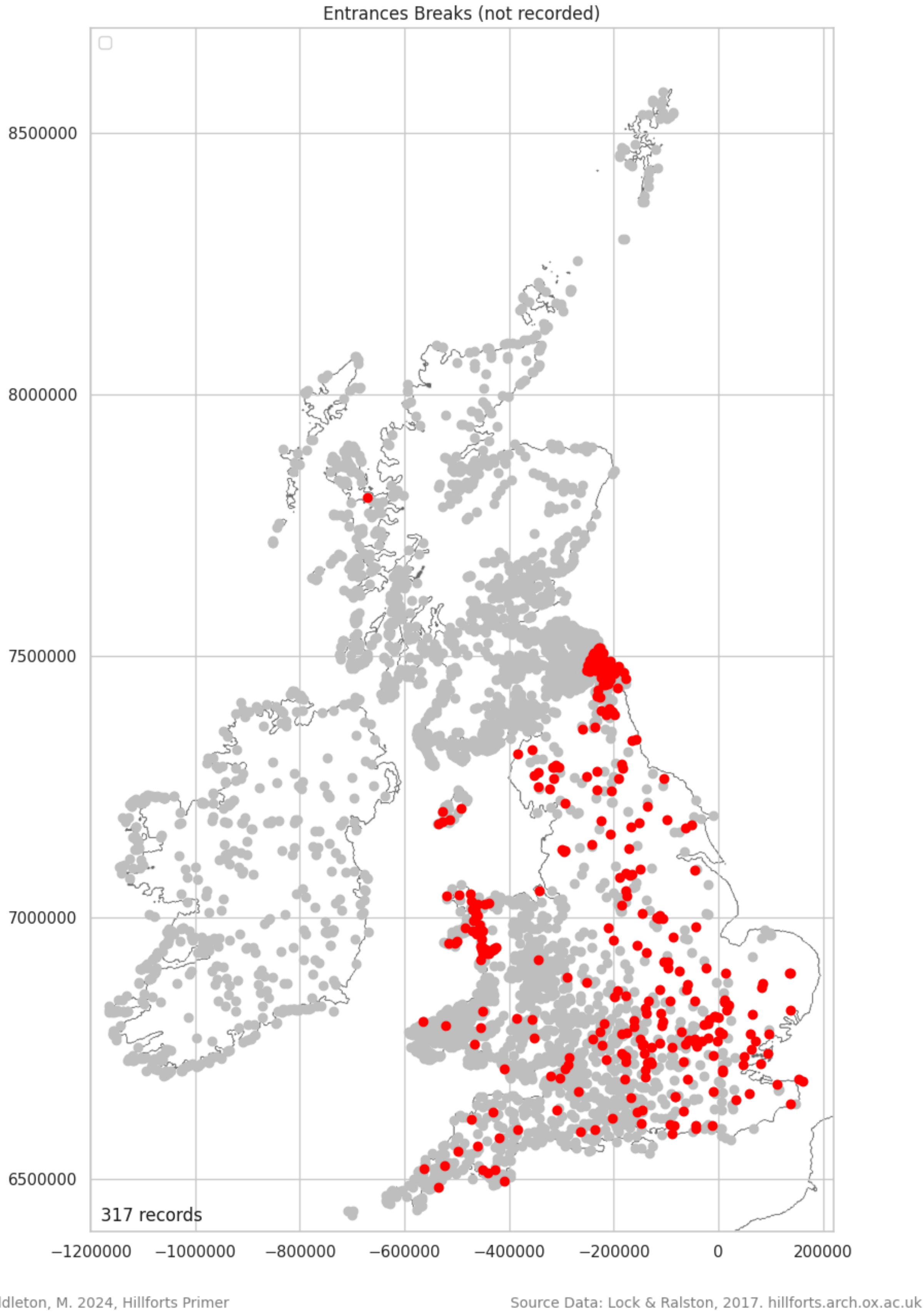
In [ ]:
```
minus_one_entrances = \
location_entrance_data[location_entrance_data['Entrances_Breaks'] == -1].copy()
minus_one_entrances['Entrances_Breaks'] = "Yes"
```

In [ ]:
```
print(f'{round(len(minus_one_entrances)/len(location_entrance_data)*100,2)}% of hillforts have no information record
```

```
7.64% of hillforts have no information recorded regarding entrances.
```

In [ ]:

```
minus_one_entrances_stats = \
plot_over_grey(minus_one_entrances, 'Entrances_Breaks', 'Yes', '(not recorded)')
```




```
7.64%
```

Entrances Original Data Plotted

Entrance Original has a long tail of outliers. 95.44% of hillforts have two original entrances or less. 80.23% have one or less. Only 1.69% of hillforts have four enreances or more.

In [ ]:
```
entrance_numeric_data['Entrances_Original'].value_counts().sort_index()
```

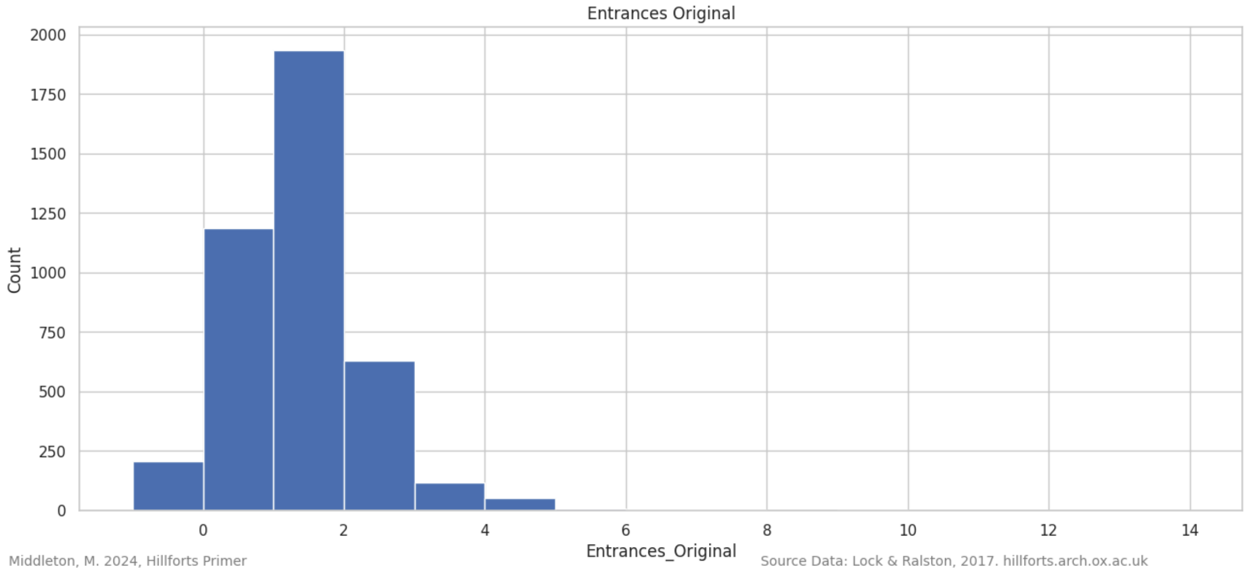
Out[ ]:
```
-1.0      206
 0.0     1186
 1.0     1935
 2.0      631
 3.0      119
 4.0       53
 5.0        8
 6.0        2
 7.0        2
 8.0        1
 9.0        2
 12.0       1
 14.0       1
Name: Entrances_Original, dtype: int64
```

In [ ]:
```
one_orig_ent = entrance_numeric_data[entrance_numeric_data['Entrances_Original']==1]
two_orig_ent_or_less = entrance_numeric_data[entrance_numeric_data['Entrances_Original']<=2] - 206
three_orig_ent_or_less = entrance_numeric_data[entrance_numeric_data['Entrances_Original']<=3] - 206
outlier_orig_ent = entrance_numeric_data[entrance_numeric_data['Entrances_Original']>3] - 206
print(f'{round(len(one_orig_ent)/len(location_entrance_data)*100,2)}% of hillforts have one original entrance.')
print(f'{round(len(two_orig_ent_or_less)/len(location_entrance_data)*100,2)}% of hillforts have two original entranc
print(f'{round(len(three_orig_ent_or_less)/len(location_entrance_data)*100,2)}% of hillforts have three original ent
print(f'{round(len(outlier_orig_ent)/len(location_entrance_data)*100,2)}% of hillforts have four or more original en
```

```
46.66% of hillforts have one original entrance.
95.44% of hillforts have two original entrances or less.
98.31% of hillforts have three original entrances or less.
1.69% of hillforts have four or more original entrances (Outliers).
```

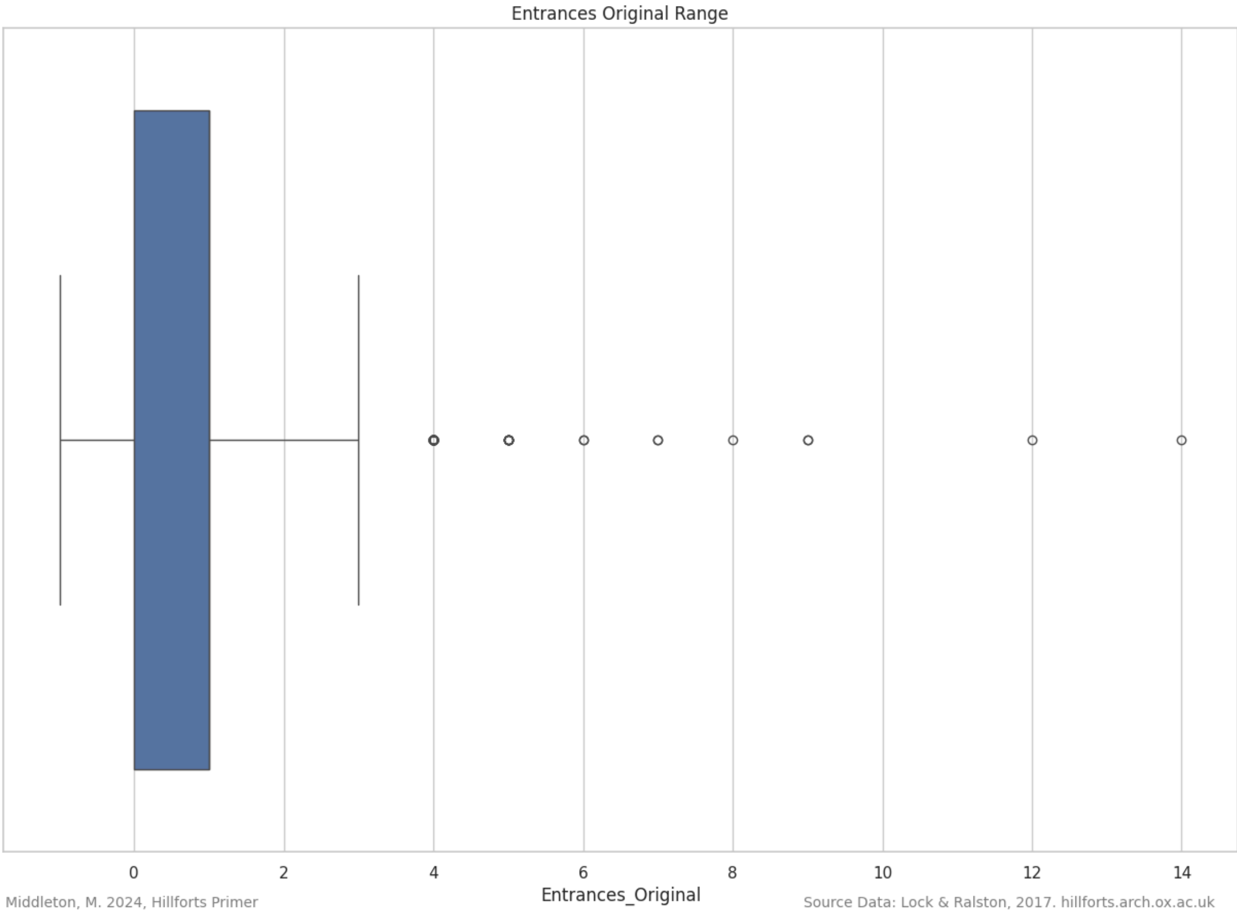
In [ ]:
```
plot_histogram(entrance_numeric_data['Entrances_Original'], \
               'Entrances_Original', 'Entrances_Original')
```

None



Outliers range from four to 14 original entrances. The interquartile range is between zero and one original entrances.

In [ ]:
```
entrances_orignial_data = \
plot_data_range(entrance_numeric_data['Entrances_Original'], \
                'Entrances_Original', "h")
```



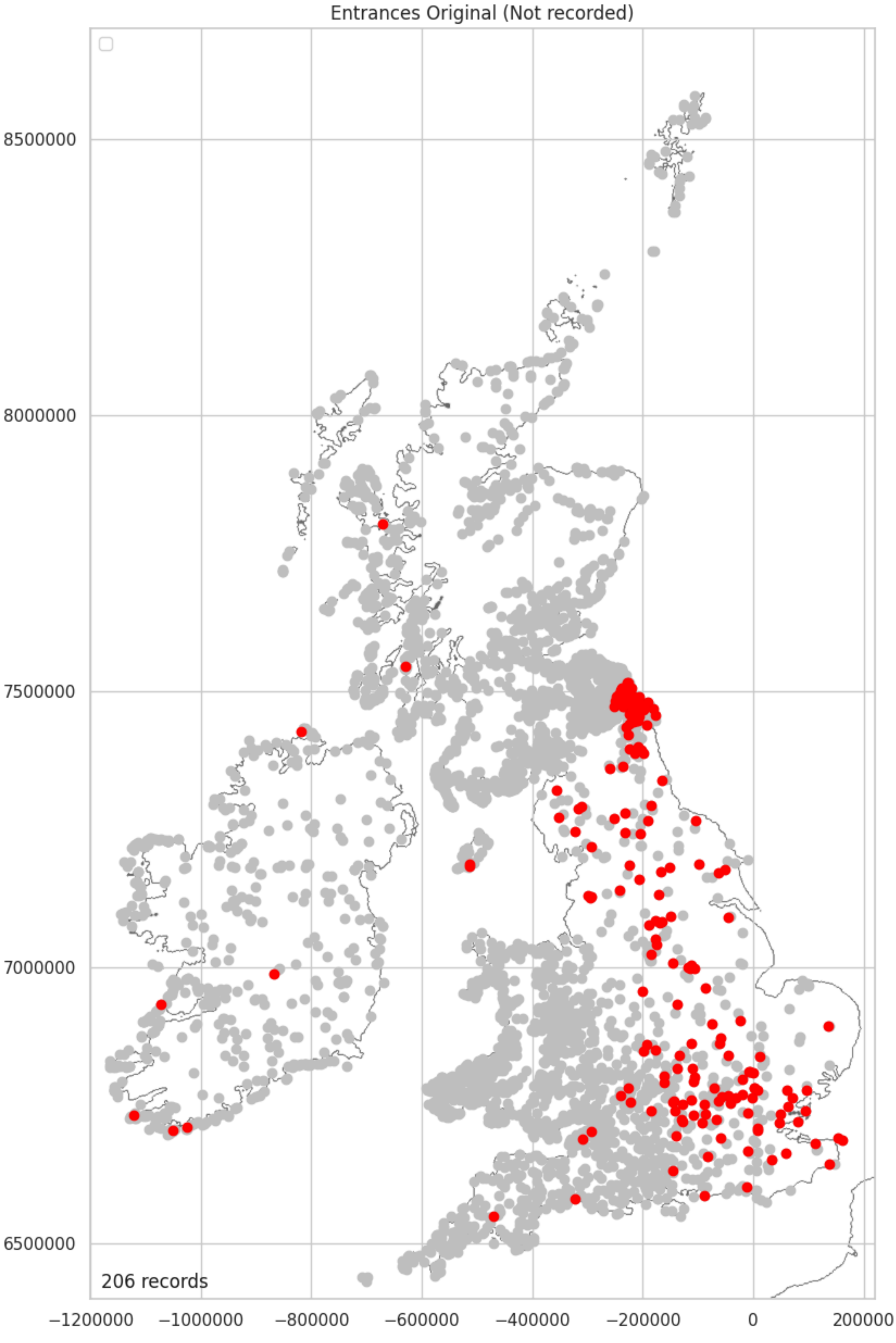
In [ ]:
```
entrances_orignial_data
```

Out[ ]:
```
[-1.0, 0.0, 1.0, 1.0, 3.0]
```

Entrance Original Not Recorded Distribution Mapped

There is a recording bias, in the original entrances data, across England. In England, the focus for recording this data has been in the west and south-west. Only 206 records have no information regarding original entrances and almost all are in the east. All hillforts in Wales and most in Scotland and Ireland have a recorded number of original entrances.

In [ ]:
```
nan_orig_entrance = \
location_entrance_data[location_entrance_data['Entrances_Original']==-1].copy()
nan_orig_entrance['Entrances_Original'] = "Yes"
nan_orig_entrances_stats = plot_over_grey(nan_orig_entrance, \
                                          'Entrances_Original', 'Yes', \
                                          '(Not recorded)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
4.97%
```

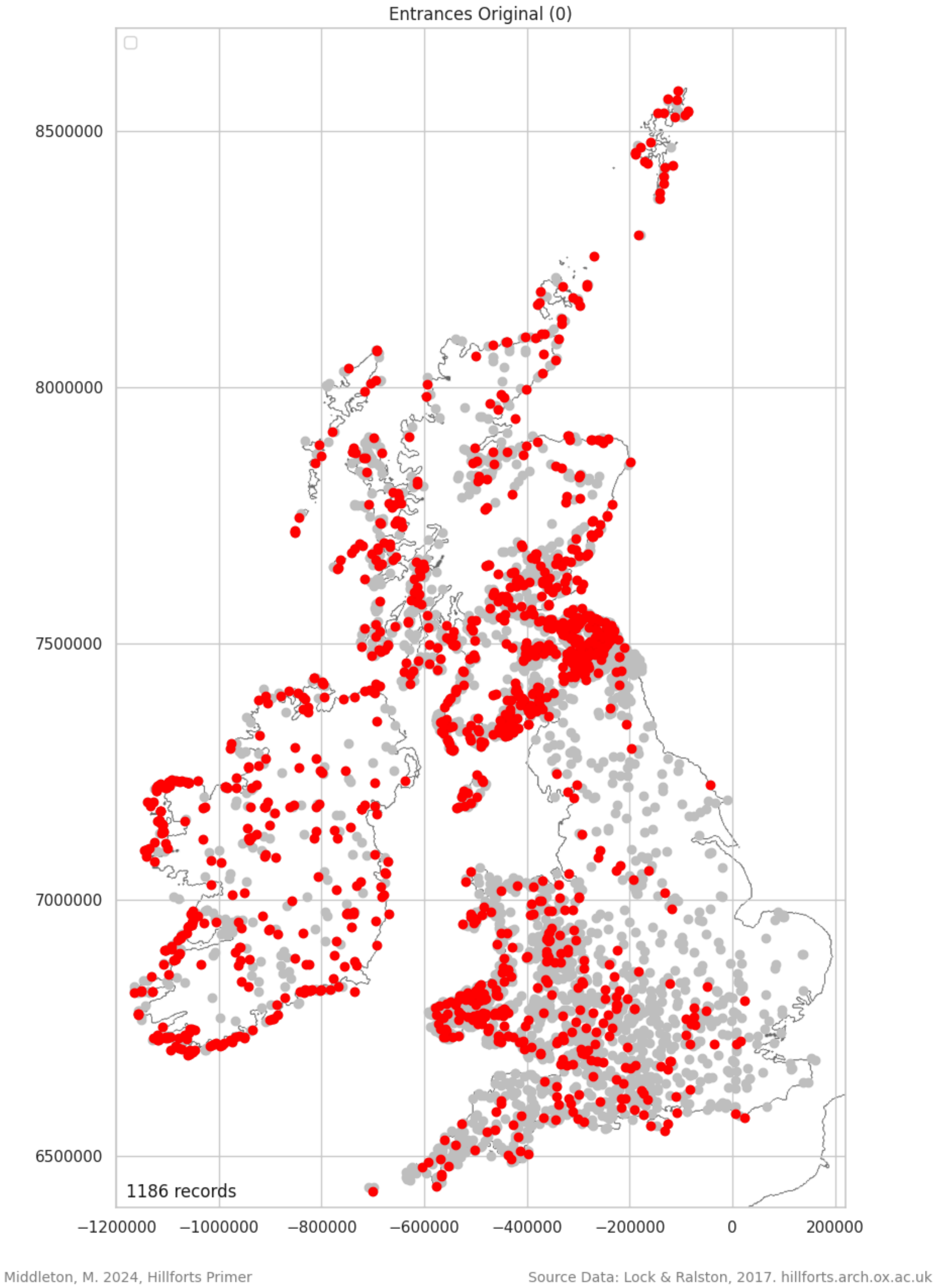
Zero Entrances Original Distribution Mapped

28.6% of hillforts are recorded as not having an original entrance. There is a noticeable lack of records in the east of England which is most likely the result of original entrances not being recorded. See: Entrance Original Not Recorded Distribution Mapped.

In [ ]:

```
zero_orig_entrance = \
location_entrance_data[location_entrance_data['Entrances_Original']==0].copy()
zero_orig_entrance['Entrances_Original'] = "Yes"
```

```
zero_orig_entrances_stats = \
plot_over_grey(zero_orig_entrance, 'Entrances_Original', 'Yes', '(0)')
```

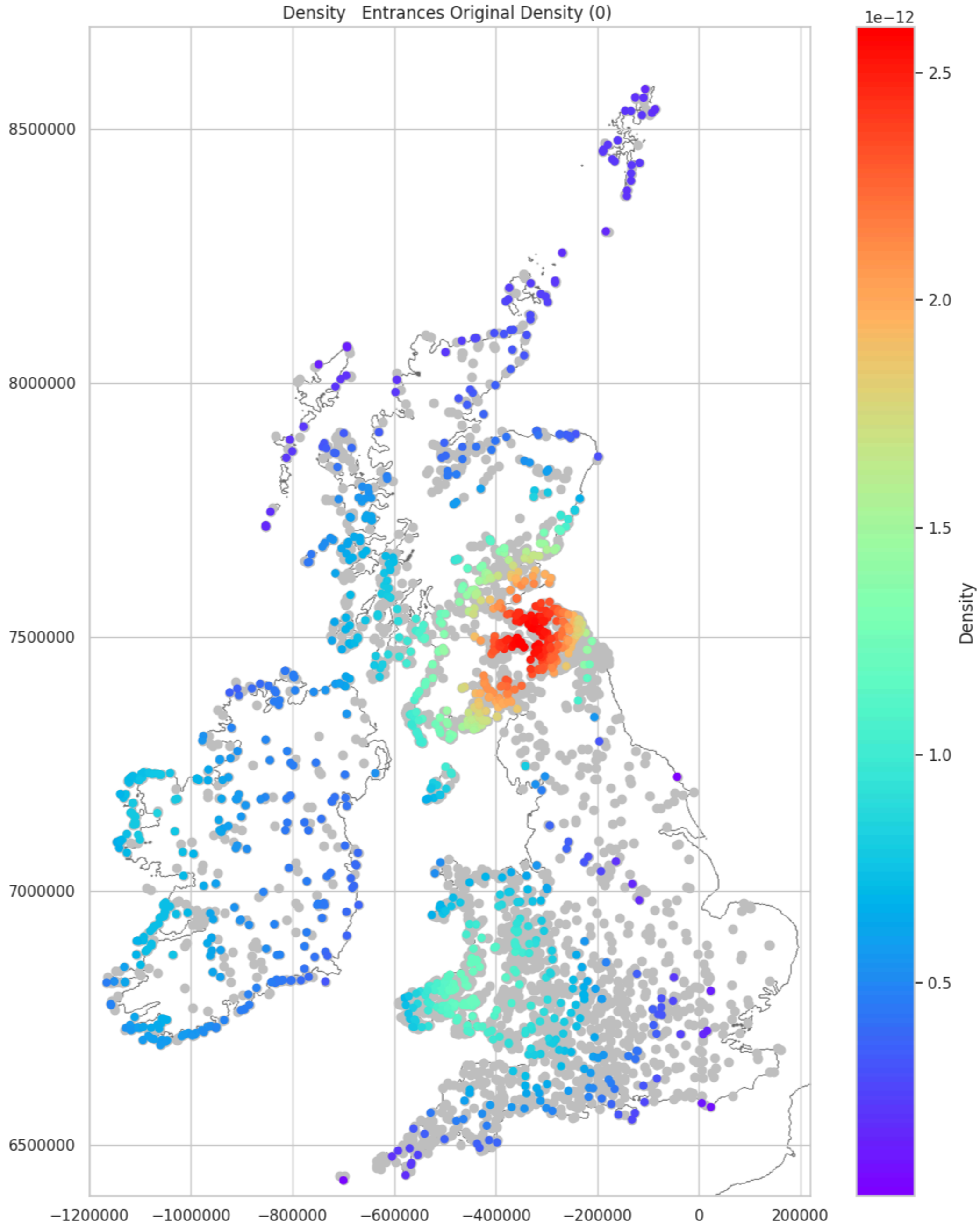



```
28.6%
```

Zero Entrance Original Density Mapped

There is a high degree of similarity between the distribution of hillforts with zero entrances and the distribution clusters seen when plotting the location data in, Part 1: Density Data Transformed Mapped. This may suggest that recording a hillfort as having no original entrances has been used as a shorthand to indicate that this information has not been recorded. If not, it suggests that there is a uniform pattern, across the entire atlas, where original entrances leave no evidence of their existence - perhaps modified by later reuse or an entrance style that leaves no discernible trace.

In [ ]:
```
plot_density_over_grey(zero_orig_entrances_stats, 'Entrances_Original Density (0)')
```



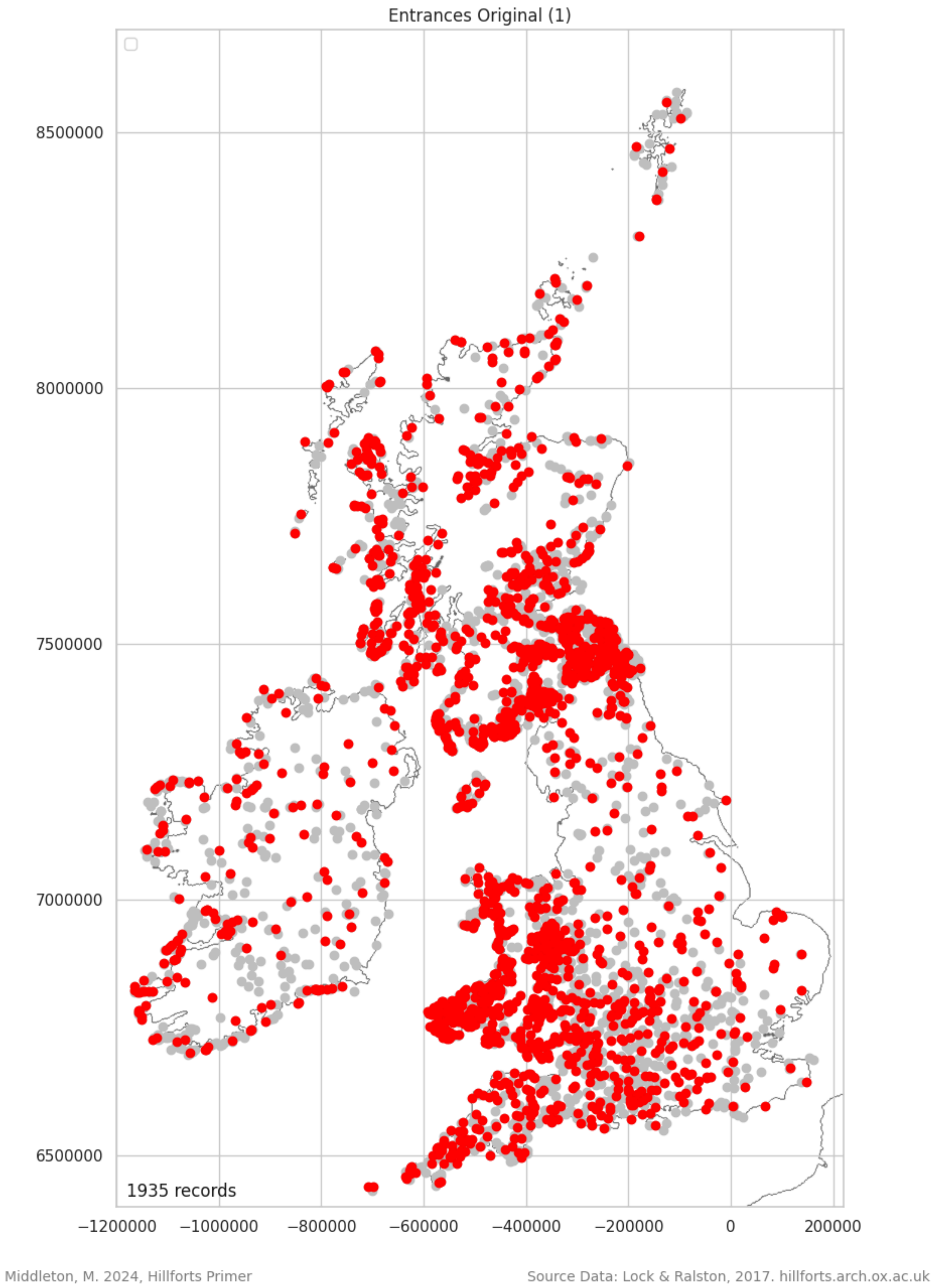
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

One Entrance Original Distribution Mapped

Just under half (46.66%) of all hillforts have a single original entrance.

In [ ]:
```
one_orig_entrance = \
location_entrance_data[location_entrance_data['Entrances_Original']==1].copy()
one_orig_entrance['Entrances_Original'] = "Yes"
one_orig_entrances_stats = \
plot_over_grey(one_orig_entrance, 'Entrances_Original', 'Yes', '(1)')
```

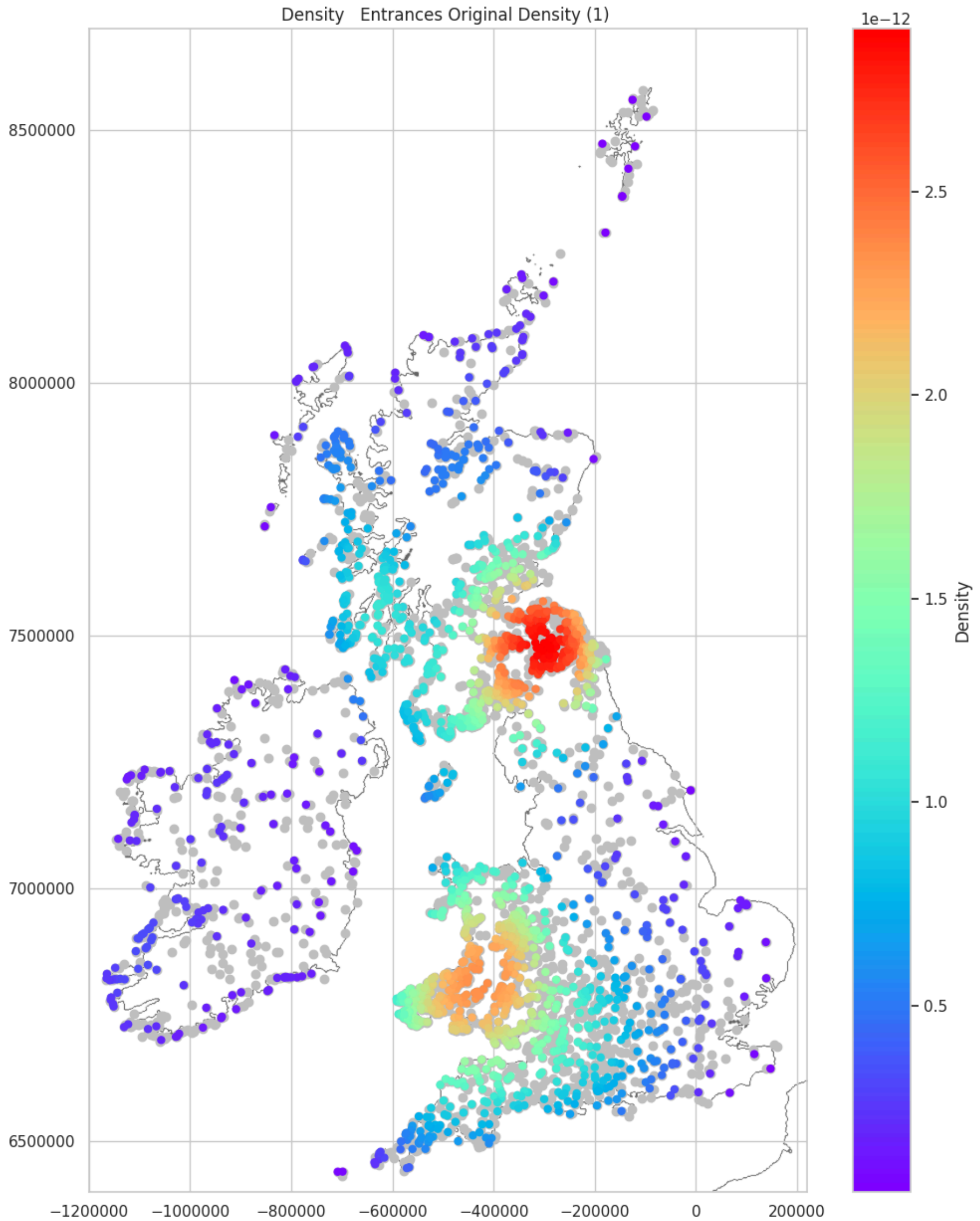



```
46.66%
```

One Entrance Original Density Mapped

Here the distributions in the Northeast, Northwest, and South match the main distributions seen in, Part 1: Density Data Transformed Mapped. In Ireland there is a sparce spread across the entire country but there is no obvious correlation with the two main clusters of forts, seen on the Density Data Transformed plot.

In [ ]:
```
plot_density_over_grey(one_orig_entrances_stats, 'Entrances_Original Density (1)')
```



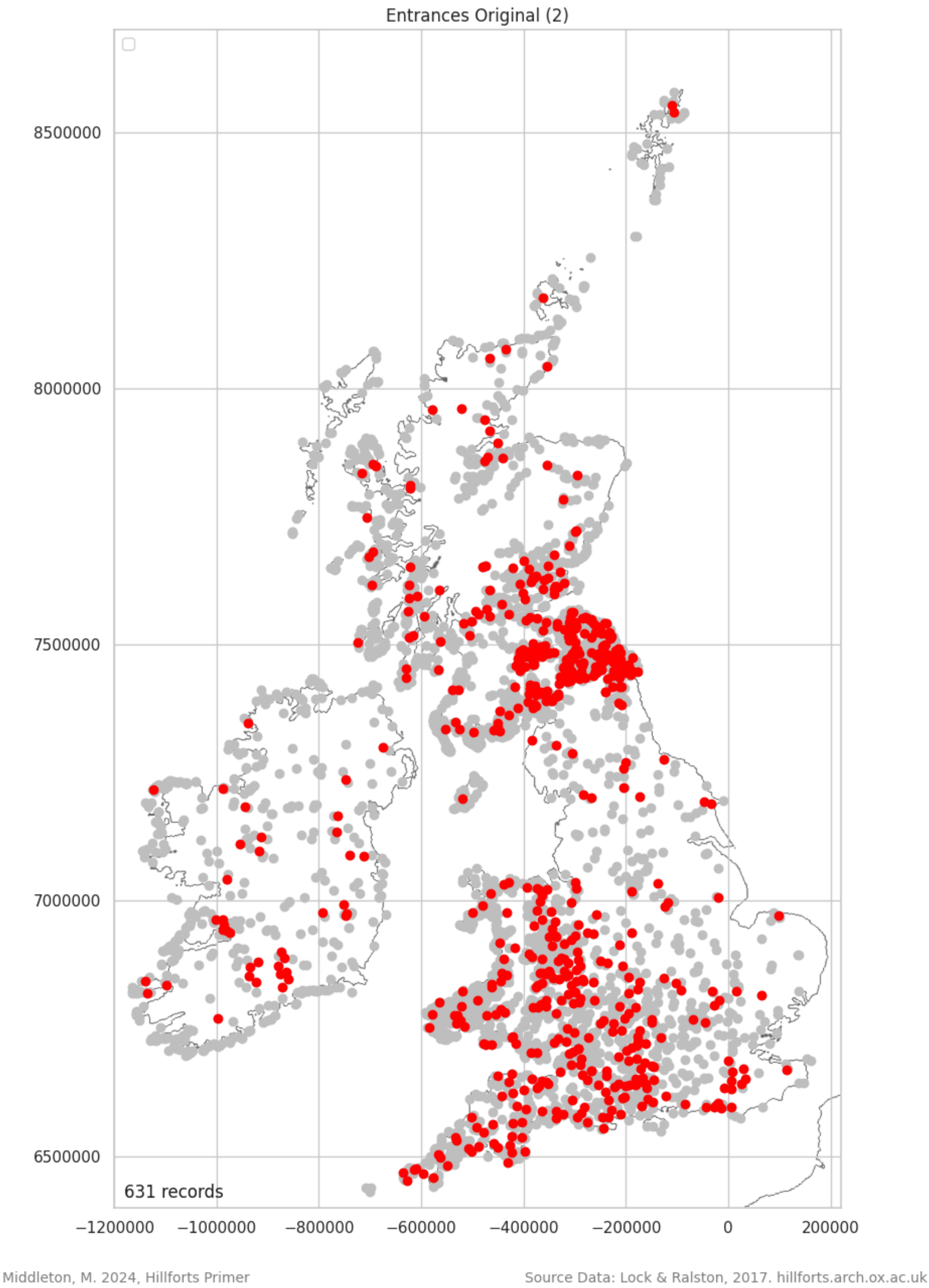
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Two Entrances Original Distribution Mapped

Just 15.22% of hillforts have two original entrances.

In [ ]:
```
two_orig_entrance = \
location_entrance_data[location_entrance_data['Entrances_Original']==2].copy()
two_orig_entrance['Entrances_Original'] = "Yes"
two_orig_entrances_stats = \
plot_over_grey(two_orig_entrance, 'Entrances_Original', 'Yes', '(2)')
```

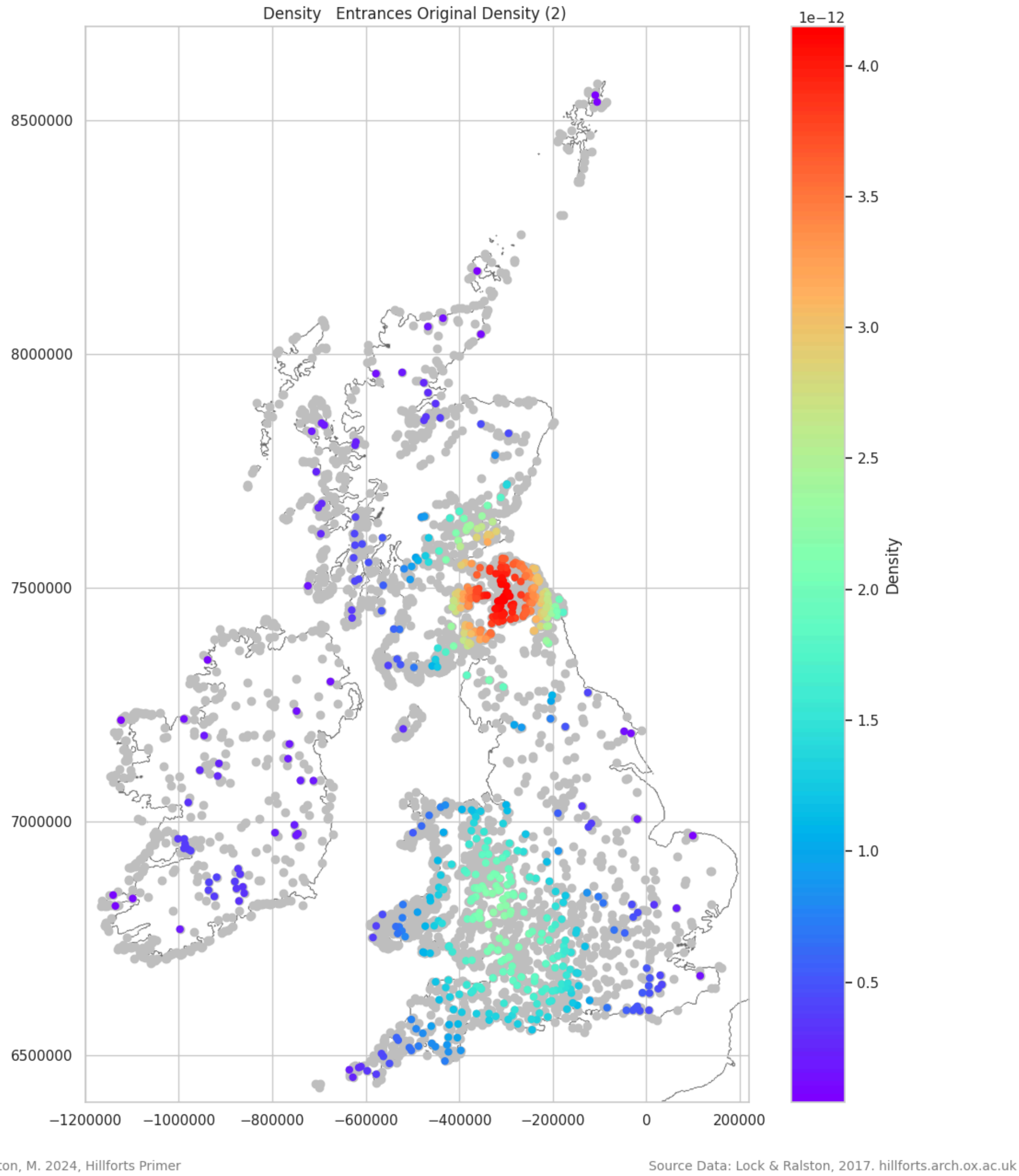


15.22%

Two Entrances Original Density Mapped

There are two main clusters. One in the Northeast and the second to the east of the Cambrian Mountains. The contrast in the intensity and focus of the southern cluster is striking when compared with One Entrance Original Density Mapped, with the two entrance cluster being more diffuse and focussed over the eastern slopes of the Cambrian mountains, into south, central England and down into the South-west.

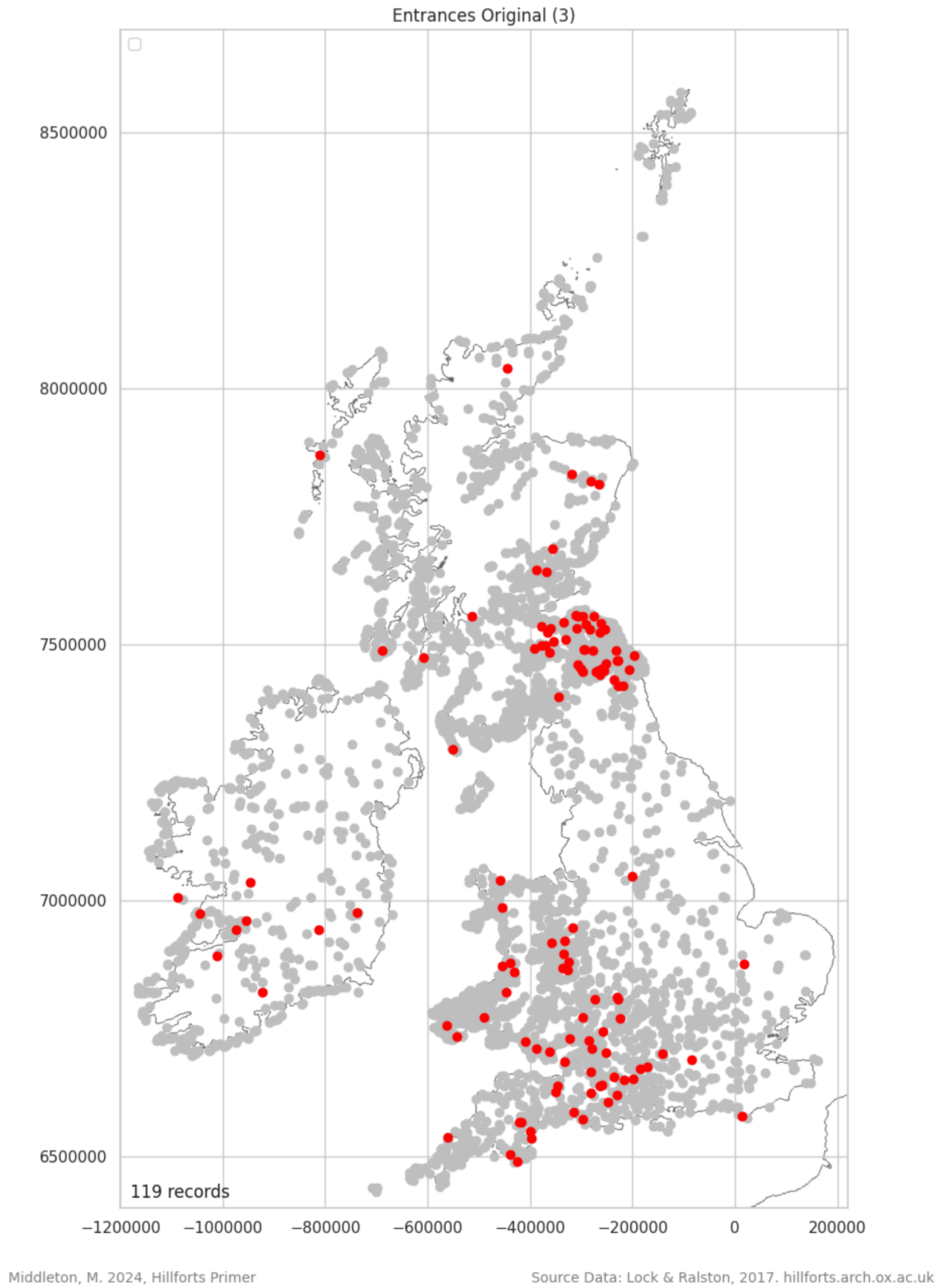
```
In [ ]: plot_density_over_grey(two_orig_entrances_stats, 'Entrances_Original Density (2)')
```



Three Entrances Original Distribution Mapped

Just 2.87% of hillforts have three original entrances.

In [ ]:
```
three_orig_entrance = \
location_entrance_data[location_entrance_data['Entrances_Original']==3].copy()
three_orig_entrance['Entrances_Original'] = "Yes"
three_orig_entrances_stats = \
plot_over_grey(three_orig_entrance, 'Entrances_Original', 'Yes', '(3)')
```

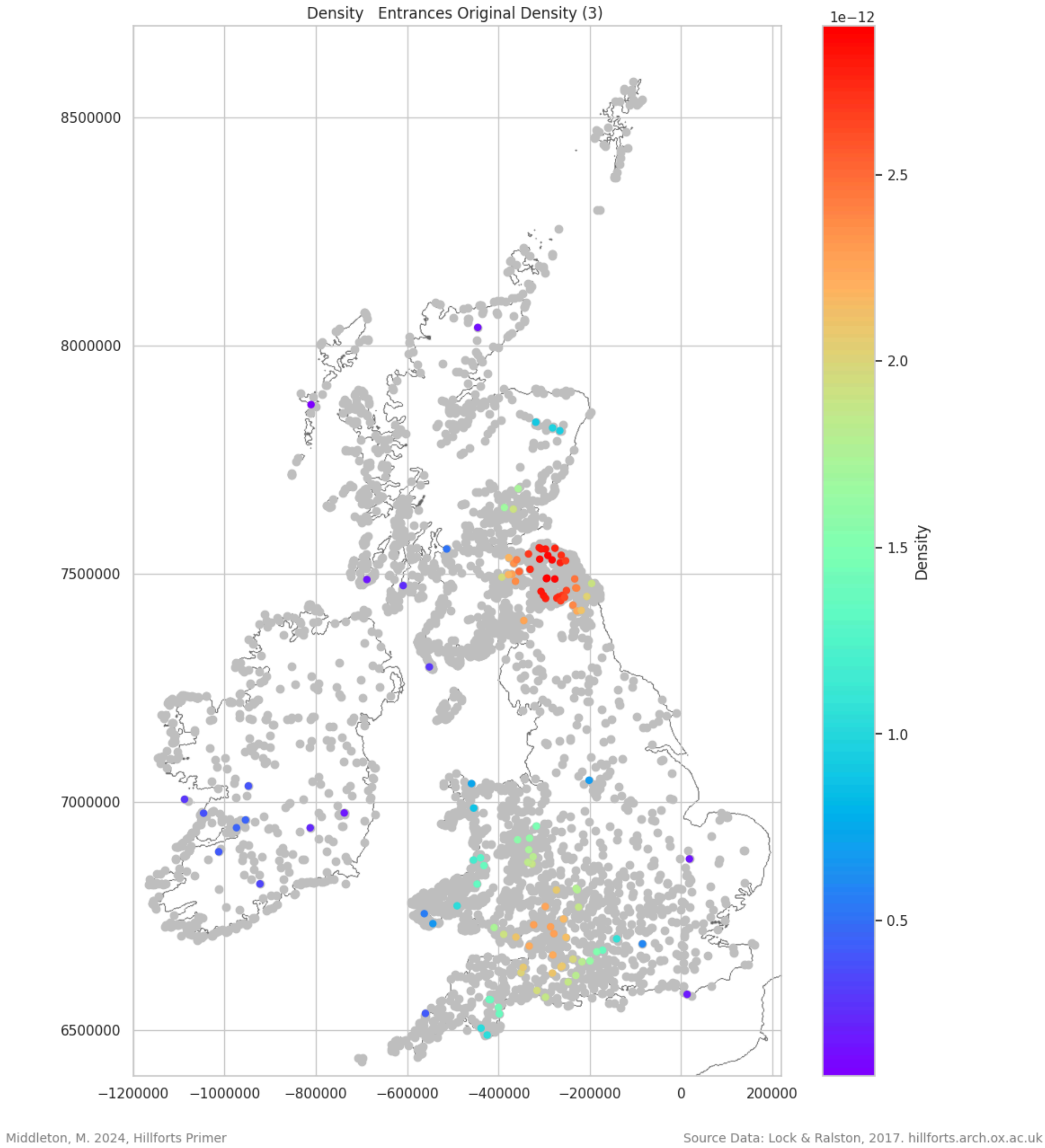


2.87%

Three Entrances Original Density Mapped

Most of these forts are in the Northeast.

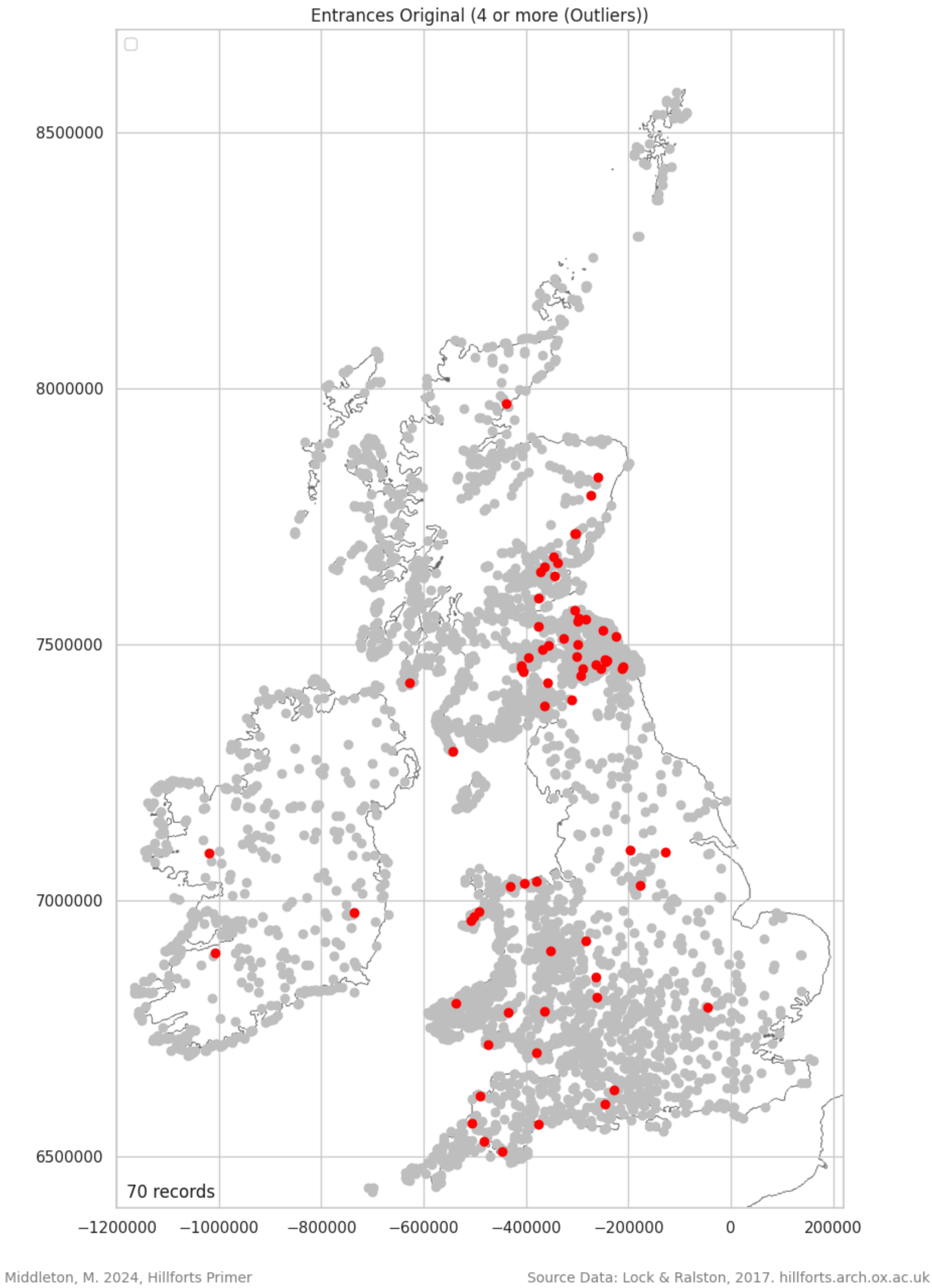
```
In [ ]: plot_density_over_grey(three_orig_entrances_stats, 'Entrances_Original Density (3)')
```



Four Entrances Original Distribution Mapped

Just 70 hillforts (1.69%) have four original entrances.

In [ ]:
```
four_plus_orig_entrance = \
location_entrance_data[location_entrance_data['Entrances_Original']>3].copy()
four_plus_orig_entrance['Entrances_Original'] = "Yes"
four_plus_orig_entrances_stats = \
plot_over_grey(four_plus_orig_entrance, 'Entrances_Original', \
               'Yes', '(4 or more (Outliers))')
```




```
1.69%
```

Entrance Text Data

There are five text features relating to entrances. All contain null values.

```
In [ ]: entrance_text_features = [
         'Entrances_Breaks_Comments',
         'Entrances_Original_Comments',
         'Entrances_Chevaux_Comments',
         'Entrances_Summary',
```

```
 'Related_Entrances']

entrance_text_data = entrance_data[entrance_text_features].copy()
entrance_text_data.head()
```

Out[ ]:

	Entrances_Breaks_Comments	Entrances_Original_Comments	Entrances_Chevaux_Comments	Entrances_Summary	Related_Entrances
0	Two original and four modern gaps.	Two original inturned entrances at SE and SW c...	NaN	Two original entrances; the SE inturned. The S...	In-turned (South east); In-turned (South west)...
1	N entrance damaged by wagon access and possibl...	S entrance original, that on the NW possibly ...	NaN	Entrances difficult to unravel. The S entrance...	In-turned (South); Simple Gap (North west); Ho...
2	Entrances intact	Interesting inturn to N entrance	NaN	The curving N entrance is complex with inturn ...	In-turned (North); Oblique (North); Simple Gap...
3	Modern gap to the S.	Off-set entrance on the E. Possibly another to...	NaN	The gaps to the NE and E have been widened and...	Simple Gap (East); Simple Gap (North east)
4	Probable modern breaks not recorded.	Two entrances are from Phase I and four from P...	NaN	Both Phase I entrances, NE and SW, are out-tur...	Out-turned (North east); Out-turned (South wes...

In [ ]:

```
entrance_text_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 5 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Entrances_Breaks_Comments    1193 non-null   object
 1   Entrances_Original_Comments  1126 non-null   object
 2   Entrances_Chevaux_Comments   77 non-null     object
 3   Entrances_Summary            4132 non-null   object
 4   Related_Entrances            2749 non-null   object
dtypes: object(5)
memory usage: 162.1+ KB
```

Entrance Text Data - Resolve Null Values

Test for 'NA'.

In [ ]:

```
test_cat_list_for_NA(entrance_text_data, entrance_text_features)
```

```
Entrances_Breaks_Comments 0
Entrances_Original_Comments 0
Entrances_Chevaux_Comments 0
Entrances_Summary 0
Related_Entrances 0
```

Fill null values with 'NA'.

In [ ]:

```
entrance_text_data = \
update_cat_list_for_NA(entrance_text_data, entrance_text_features)
entrance_text_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 5 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   Entrances_Breaks_Comments    4147 non-null   object
 1   Entrances_Original_Comments  4147 non-null   object
 2   Entrances_Chevaux_Comments   4147 non-null   object
 3   Entrances_Summary            4147 non-null   object
 4   Related_Entrances            4147 non-null   object
dtypes: object(5)
memory usage: 162.1+ KB
```

Entrance Encodable Data

There are just two encodeable features. Neither contains null values.

In [ ]:

```
entrance_encodeable_features = [
 'Entrances_Guard_Chambers',
 'Entrances_Chevaux']

entrance_encodeable_data = entrance_data[entrance_encodeable_features].copy()
entrance_encodeable_data.head()
```

Out[ ]:

	Entrances_Guard_Chambers	Entrances_Chevaux
0	No	No
1	No	No
2	No	No
3	No	No
4	No	No

In [ ]:
```
entrance_encodeable_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 2 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Entrances_Guard_Chambers  4147 non-null   object
 1   Entrances_Chevaux         4147 non-null   object
dtypes: object(2)
memory usage: 64.9+ KB
```

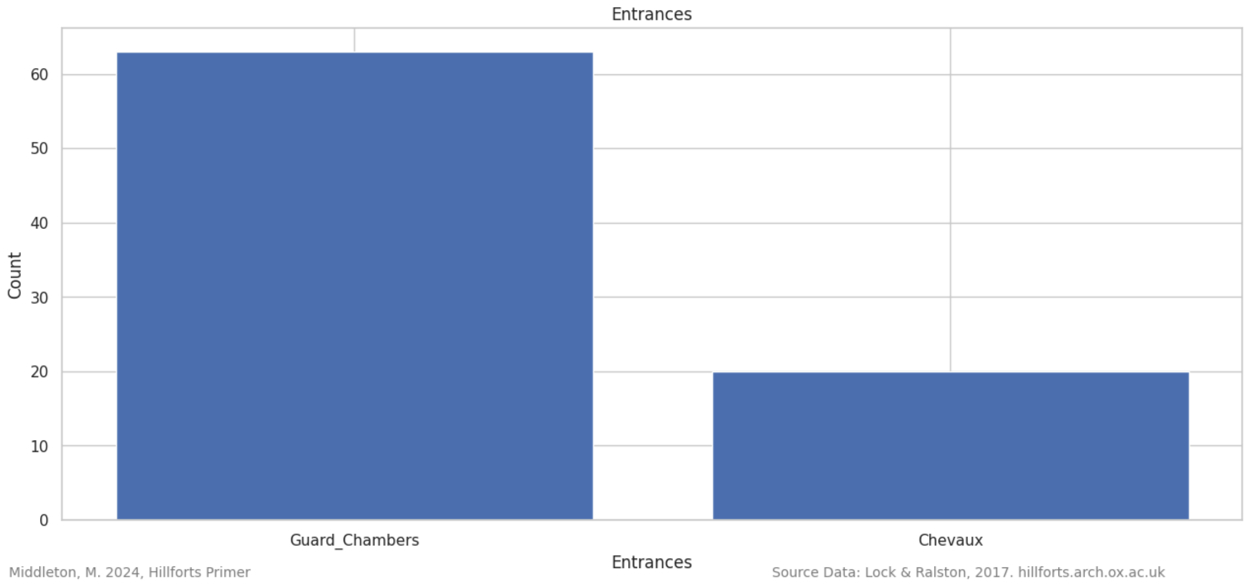
Entrance Encodable Data Plotted

Guard chambers are recorded at 63 hillforts. All but two are in England. Twenty hillforts have a Cheveaux de frise. It is likely that both have a significant survey bias.

In [ ]:
```
for feature in entrance_encodeable_features:
    print(feature + ": " + str(sum(entrance_encodeable_data[feature]=="Yes")))
```

```
Entrances_Guard_Chambers: 63
Entrances_Chevaux: 20
```

In [ ]:
```
plot_bar_chart(entrance_encodeable_data[['Entrances_Guard_Chambers',\
                                        'Entrances_Chevaux'] ], 1, 'Entrances', \
                                        'Count', 'Entrances')
```

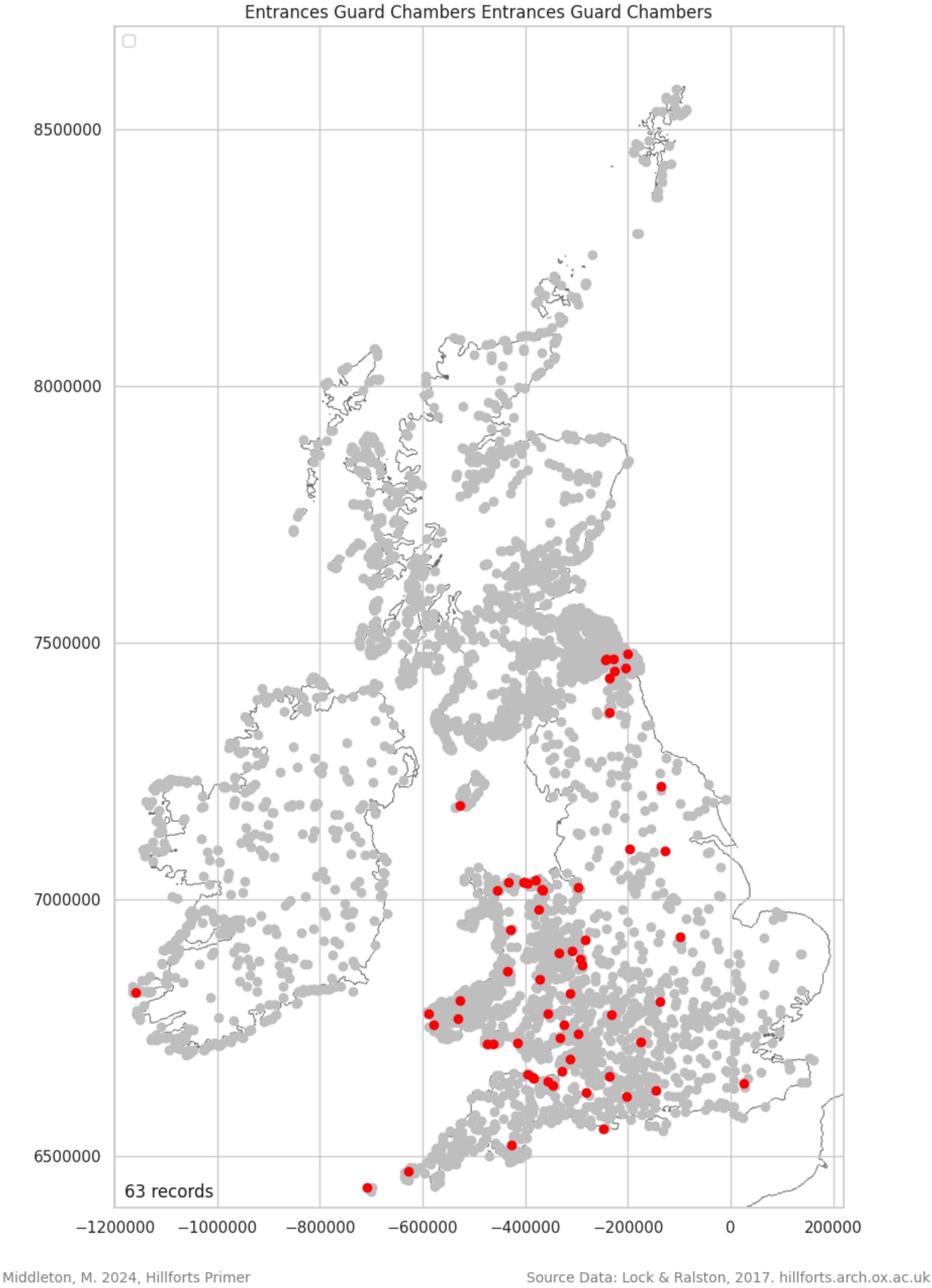


Guard Chambers Mapped

There is a recording bias with all but two of the hillforts recorded being in England and Wales.

In [ ]:
```
location_entrance_encodeable_data = \
pd.merge(location_numeric_data_short, entrance_encodeable_data, \
         left_index=True, right_index=True)
```

In [ ]:
```
enteances_guard_chambers_stats = \
plot_over_grey(location_entrance_encodeable_data, 'Entrances_Guard_Chambers', \
               'Yes', 'Entrances_Guard_Chambers')
```

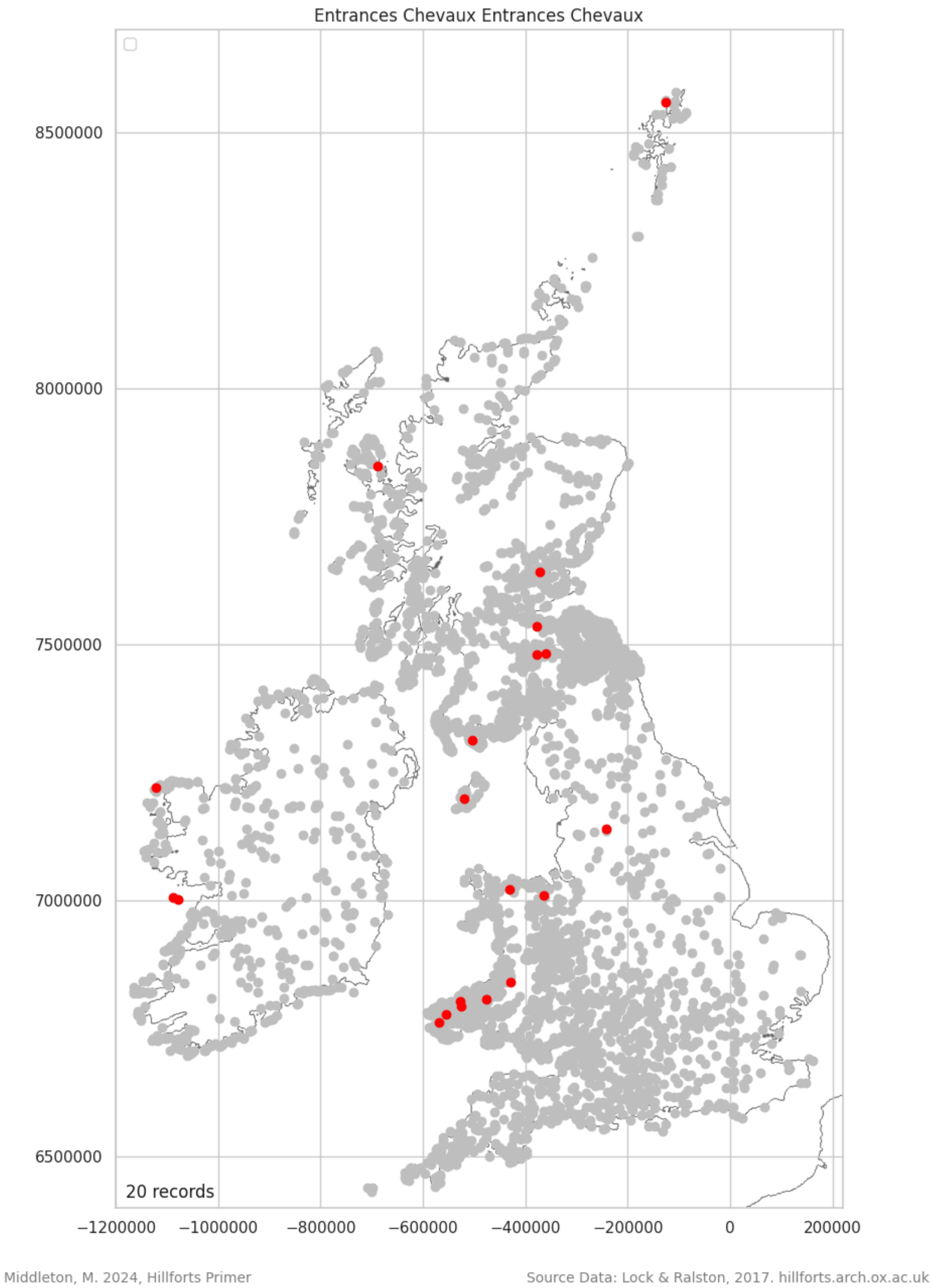



```
1.52%
```

Chevaux-de-frise Mapped

At just 20 examples it is not possible to say anything meaningful about the distribution of Cheveaux de frise other than that they are rare and that most have been recorded in Wales and Scotland.

```
In [ ]: enteances_chevaux_stats = \
        plot_over_grey(location_entrance_encodeable_data, 'Entrances_Chevaux', 'Yes', \
                       'Entrances_Chevaux')
```




```
0.48%
```

Review Entrance Data Split

In [ ]:
```
review_data_split(entrance_data, entrance_numeric_data, entrance_text_data, entrance_encodeable_data)
```

```
Data split good.
```

Entrance Data Package

```
In [ ]: entrance_data_list = [entrance_numeric_data, entrance_text_data, entrance_encodeable_data]
```

Entrance Data Download Packages

If you do not wish to download the data using this document, all the processed data packages, notebooks and images are available here:
https://github.com/MikeDairsie/Hillforts-Primer.

```
In [ ]: download(entrance_data_list, 'entrance_package')
```

Enclosing Data

There are 64 Enclosing Data features which are subgrouped into:

- Area
- Multiperiod
- Circuit
- Ramparts
- Quadrants
- Current (enclosing form)
- Period (enclosing form)
- Surface (enclosing form)
- Excavation
- Gang Working
- Ditches

```
In [ ]: enclosing_features = [
 'Enclosing_Summary',
 'Enclosing_Area_1',
 'Enclosing_Area_2',
 'Enclosing_Area_3',
 'Enclosing_Area_4',
 'Enclosing_Enclosed_Area',
 'Enclosing_Area',
 'Enclosing_Multiperiod',
 'Enclosing_Multiperiod_Comments',
 'Enclosing_Circuit',
 'Enclosing_Circuit_Comments',
 'Enclosing_Max_Ramparts',
 'Enclosing_NE_Quadrant',
 'Enclosing_SE_Quadrant',
 'Enclosing_SW_Quadrant',
 'Enclosing_NW_Quadrant',
 'Enclosing_Quadrant_Comments',
 'Enclosing_Current_Part_Uni',
 'Enclosing_Current_Uni',
 'Enclosing_Current_Part_Bi',
 'Enclosing_Current_Bi',
 'Enclosing_Current_Part_Multi',
 'Enclosing_Current_Multi',
 'Enclosing_Current_Unknown',
 'Enclosing_Period_Part_Uni',
 'Enclosing_Period_Uni',
 'Enclosing_Period_Part_Bi',
 'Enclosing_Period_Bi',
 'Enclosing_Period_Part_Multi',
 'Enclosing_Period_Multi',
 'Enclosing_Surface_None',
 'Enclosing_Surface_Bank',
 'Enclosing_Surface_Wall',
 'Enclosing_Surface_Rubble',
 'Enclosing_Surface_Walk',
 'Enclosing_Surface_Timber',
 'Enclosing_Surface_Vitrification',
 'Enclosing_Surface_Burning',
 'Enclosing_Surface_Palisade',
 'Enclosing_Surface_Counter_Scarp',
 'Enclosing_Surface_Berm',
 'Enclosing_Surface_Unfinished',
 'Enclosing_Surface_Other',
 'Enclosing_Surface_Comments',
 'Enclosing_Excavation_Nothing',
 'Enclosing_Excavation_Bank',
 'Enclosing_Excavation_Wall',
 'Enclosing_Excavation_Murus',
```

```
 'Enclosing_Excavation_Timber_Framed',
 'Enclosing_Excavation_Timber_Laced',
 'Enclosing_Excavation_Vitrification',
 'Enclosing_Excavation_Burning',
 'Enclosing_Excavation_Palisade',
 'Enclosing_Excavation_Counter_Scarp',
 'Enclosing_Excavation_Berm',
 'Enclosing_Excavation_Unfinished',
 'Enclosing_Excavation_No_Known',
 'Enclosing_Excavation_Other',
 'Enclosing_Excavation_Comments',
 'Enclosing_Gang_Working',
 'Enclosing_Gang_Working_Comments',
 'Enclosing_Ditches',
 'Enclosing_Ditches_Number',
 'Enclosing_Ditches_Comments']

enclosing_data = hillforts_data[enclosing_features].copy()
enclosing_data.head()
```

Out[ ]:

	Enclosing_Summary	Enclosing_Area_1	Enclosing_Area_2	Enclosing_Area_3	Enclosing_Area_4	Enclosing_Enclosed_Area	Enclosing_Area	E
0	Univallate hillfort with complete circuit, but...	7.1	NaN	NaN	NaN	7.1	9.3	
1	Defined differentially by single rampart to 5....	4.1	NaN	NaN	NaN	4.1	NaN	
2	Three ramparts and ditches on the N. Although ...	2.8	NaN	NaN	NaN	2.8	NaN	
3	Steep natural scarp artificially scarped with ...	4.8	NaN	NaN	NaN	4.8	NaN	
4	In Phase I, c. 3ha were enclosed by a slight b...	3.0	14.7	NaN	NaN	14.7	NaN	

Enclosing Numeric Data

There are 12 numeric Enclosing features. All contain null values.

In [ ]:

```
enclosing_numeric_features = [
 'Enclosing_Area_1',
 'Enclosing_Area_2',
 'Enclosing_Area_3',
 'Enclosing_Area_4',
 'Enclosing_Enclosed_Area',
 'Enclosing_Area',
 'Enclosing_Max_Ramparts',
 'Enclosing_NE_Quadrant',
 'Enclosing_SE_Quadrant',
 'Enclosing_SW_Quadrant',
 'Enclosing_NW_Quadrant',
 'Enclosing_Ditches_Number']

enclosing_numeric_data = enclosing_data[enclosing_numeric_features].copy()
enclosing_numeric_data.head()
```

Out[ ]:

	Enclosing_Area_1	Enclosing_Area_2	Enclosing_Area_3	Enclosing_Area_4	Enclosing_Enclosed_Area	Enclosing_Area	Enclosing_Max_Rampa
0	7.1	NaN	NaN	NaN	7.1	9.3	1
1	4.1	NaN	NaN	NaN	4.1	NaN	1
2	2.8	NaN	NaN	NaN	2.8	NaN	3
3	4.8	NaN	NaN	NaN	4.8	NaN	1
4	3.0	14.7	NaN	NaN	14.7	NaN	2

In [ ]:

```
enclosing_numeric_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Enclosing_Area_1         3807 non-null   float64
 1   Enclosing_Area_2         335 non-null    float64
 2   Enclosing_Area_3         68 non-null     float64
 3   Enclosing_Area_4         11 non-null     float64
 4   Enclosing_Enclosed_Area  3807 non-null   float64
 5   Enclosing_Area           1263 non-null   float64
 6   Enclosing_Max_Ramparts   3999 non-null   float64
 7   Enclosing_NE_Quadrant    3927 non-null   float64
 8   Enclosing_SE_Quadrant    3899 non-null   float64
 9   Enclosing_SW_Quadrant    3896 non-null   float64
 10  Enclosing_NW_Quadrant    3899 non-null   float64
 11  Enclosing_Ditches_Number 3279 non-null   float64
dtypes: float64(12)
memory usage: 388.9 KB
```

Enclosing Numeric Data - Resolve Null Values

Test for -1.

In [ ]:
```
test_num_list_for_minus_one(enclosing_numeric_data, enclosing_numeric_features)
```

```
Enclosing_Area_1 0
Enclosing_Area_2 0
Enclosing_Area_3 0
Enclosing_Area_4 0
Enclosing_Enclosed_Area 0
Enclosing_Area 0
Enclosing_Max_Ramparts 0
Enclosing_NE_Quadrant 0
Enclosing_SE_Quadrant 0
Enclosing_SW_Quadrant 0
Enclosing_NW_Quadrant 0
Enclosing_Ditches_Number 0
```

Replace null with -1.

In [ ]:
```
enclosing_numeric_data = \
update_num_list_for_minus_one(enclosing_numeric_data, enclosing_numeric_features)
enclosing_numeric_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Enclosing_Area_1         4147 non-null   float64
 1   Enclosing_Area_2         4147 non-null   float64
 2   Enclosing_Area_3         4147 non-null   float64
 3   Enclosing_Area_4         4147 non-null   float64
 4   Enclosing_Enclosed_Area  4147 non-null   float64
 5   Enclosing_Area           4147 non-null   float64
 6   Enclosing_Max_Ramparts   4147 non-null   float64
 7   Enclosing_NE_Quadrant    4147 non-null   float64
 8   Enclosing_SE_Quadrant    4147 non-null   float64
 9   Enclosing_SW_Quadrant    4147 non-null   float64
 10  Enclosing_NW_Quadrant    4147 non-null   float64
 11  Enclosing_Ditches_Number 4147 non-null   float64
dtypes: float64(12)
memory usage: 388.9 KB
```

Enclosing Area 1 Plotted

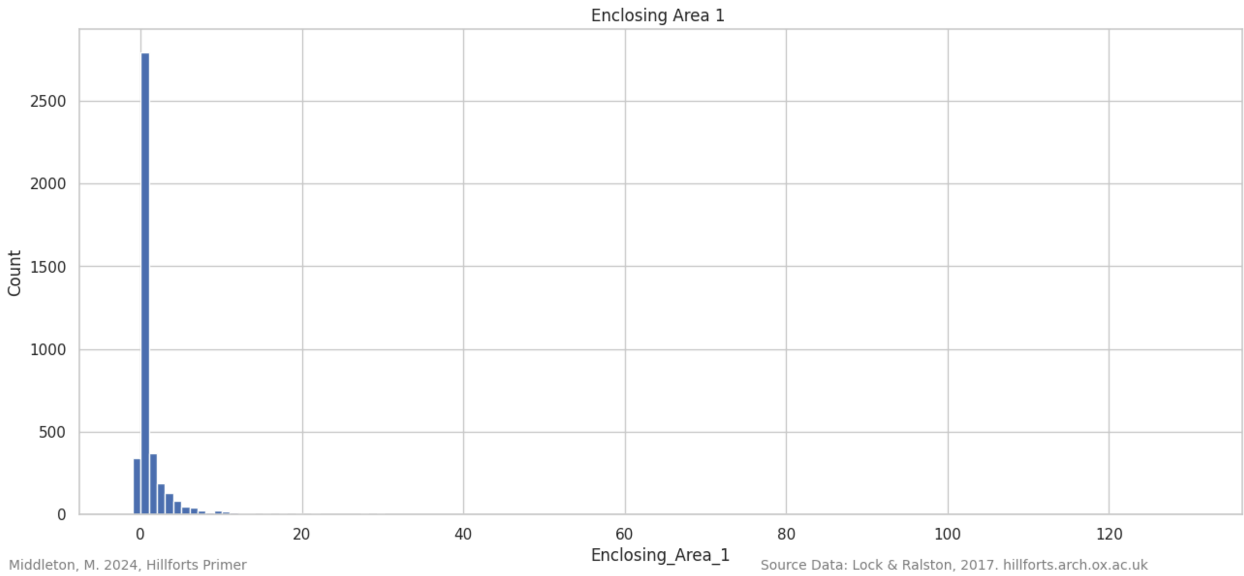
With 3807 entries, Area 1 is the most populated of the Area features and refers to the, "Enclosed area ... within the inner rampart/bank/wall". (Data Structure)

Most forts are less than one hectare in size with outliers up to 130 hectares. The data has a very long tail.

In [ ]:
```
enclosing_numeric_data['Enclosing_Area_1'].describe()
```

```
Out[ ]: count    4147.000000
mean        1.427997
std         5.192075
min        -1.000000
25%         0.130000
50%         0.340000
75%         1.000000
max       130.000000
Name: Enclosing_Area_1, dtype: float64
```

```
In [ ]: plot_bar_chart_numeric(enclosing_numeric_data, 1, \
                               'Enclosing_Area_1', 'Count', 'Enclosing_Area_1', \
                               int(enclosing_numeric_data['Enclosing_Area_1'].max()))
```



Enclosing Area 1 Clipped Plotted

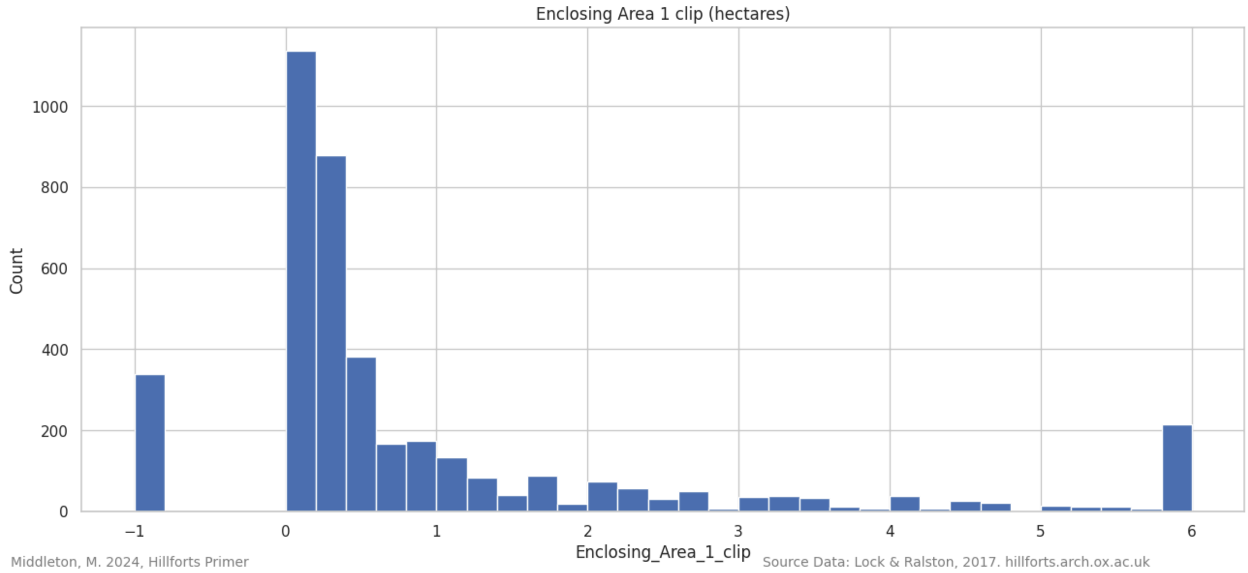
The outliers make it difficult to see the detail in the data at the lower end. To improve the clarity of the plot, the data is capped at 6 Ha. Note that the histogram incudes the null values (-1). All outliers above 6 Ha are collected into the capped value.

Most forts are below 0.5 Ha in size. The majority (95.6%) of forts are below 10.5 Ha in size.

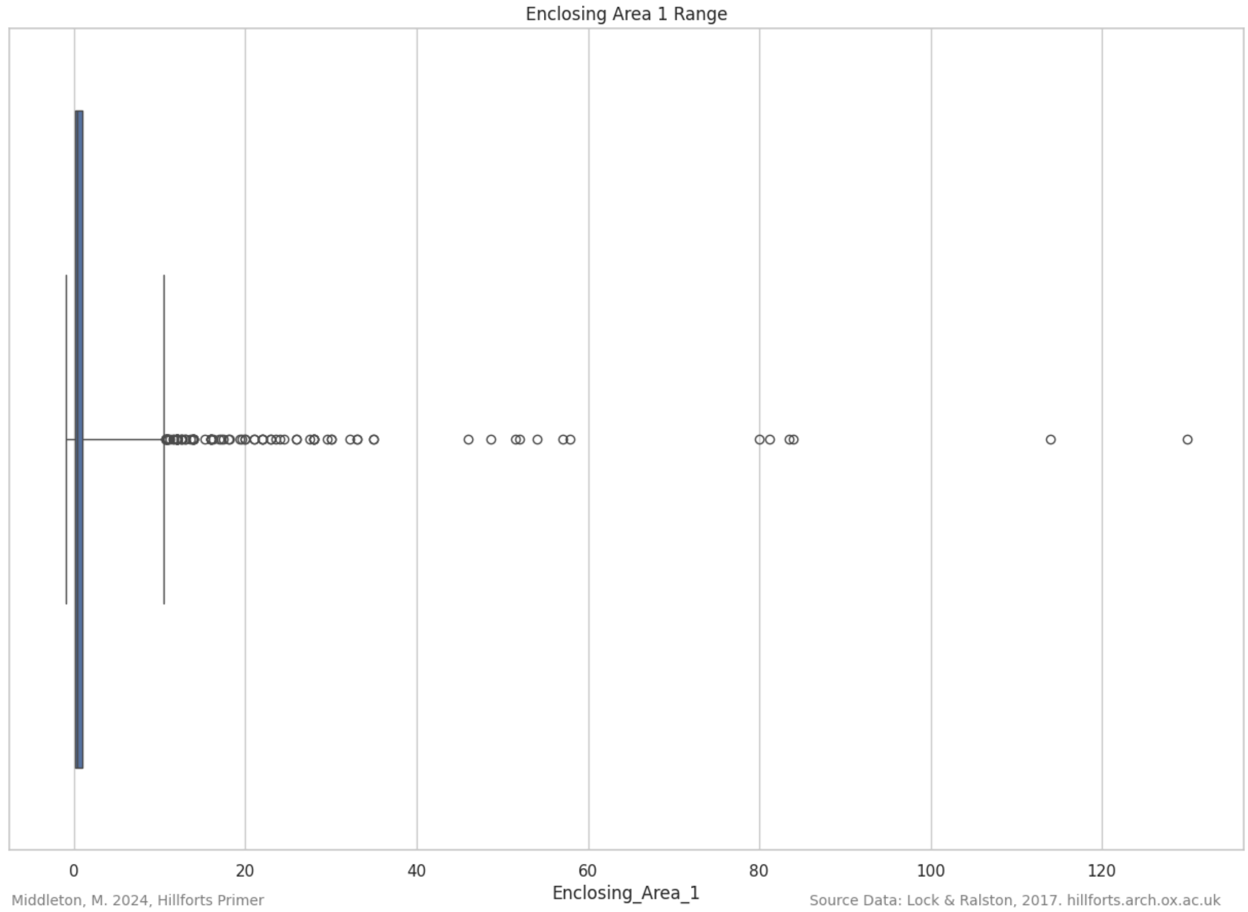
```
In [ ]: enclosing_area_1_data_clip = enclosing_numeric_data.copy()
enclosing_area_1_data_clip['Enclosing_Area_1_clip'] = \
enclosing_area_1_data_clip['Enclosing_Area_1'].\
clip(enclosing_area_1_data_clip['Enclosing_Area_1'], 6, axis=0)
enclosing_area_1_data_clip['Enclosing_Area_1_clip'].describe()
```

```
Out[ ]: count    4147.000000
mean        0.944245
std         1.655712
min        -1.000000
25%         0.130000
50%         0.340000
75%         1.000000
max         6.000000
Name: Enclosing_Area_1_clip, dtype: float64
```

```
In [ ]: plot_bar_chart_numeric(enclosing_area_1_data_clip, 1, 'Enclosing_Area_1_clip', \
                               'Count', 'Enclosing_Area_1_clip', 35, '(hectares)')
```



In [ ]:
```
enclosing_area_1_data = \
plot_data_range(enclosing_numeric_data['Enclosing_Area_1'], \
                'Enclosing_Area_1', "h")
```



In [ ]:
```
enclosing_area_1_data
```

Out[ ]:
```
[-1.0, 0.13, 0.34, 1.0, 10.5]
```

The test below was carried out to review if using -1 for null values might influence the output in terms of the quartile ranges. The question was, does it alter the positive quartile ranges between the minimum value at the start of quarter 1, (-1) to the current maximum value at the top end of quarter 4, (10.5 Ha). The impact of using -1 was tested by changing -1 to -0.01. This was found to make no difference to the positive quartile values. As it had no impact, -1 was retained.

To activate this code, and to confirm the observations above, remove the '#' symbols and re-run the notebook using the menu **Runtime>Run all**.

In [ ]:
```
# """Select area features"""
# area_features = [
#  'Enclosing_Area_1',
```

```
#  'Enclosing_Area_2',
#  'Enclosing_Area_3',
#  'Enclosing_Area_4',
#  'Enclosing_Enclosed_Area',
#  'Enclosing_Area']
```

In [ ]:
```
# """Change -1 to -0.01"""
# for feature in area_features:
#     enclosing_numeric_data[feature] = enclosing_numeric_data[feature].replace(-1,-0.01)
# enclosing_numeric_data.head()
```

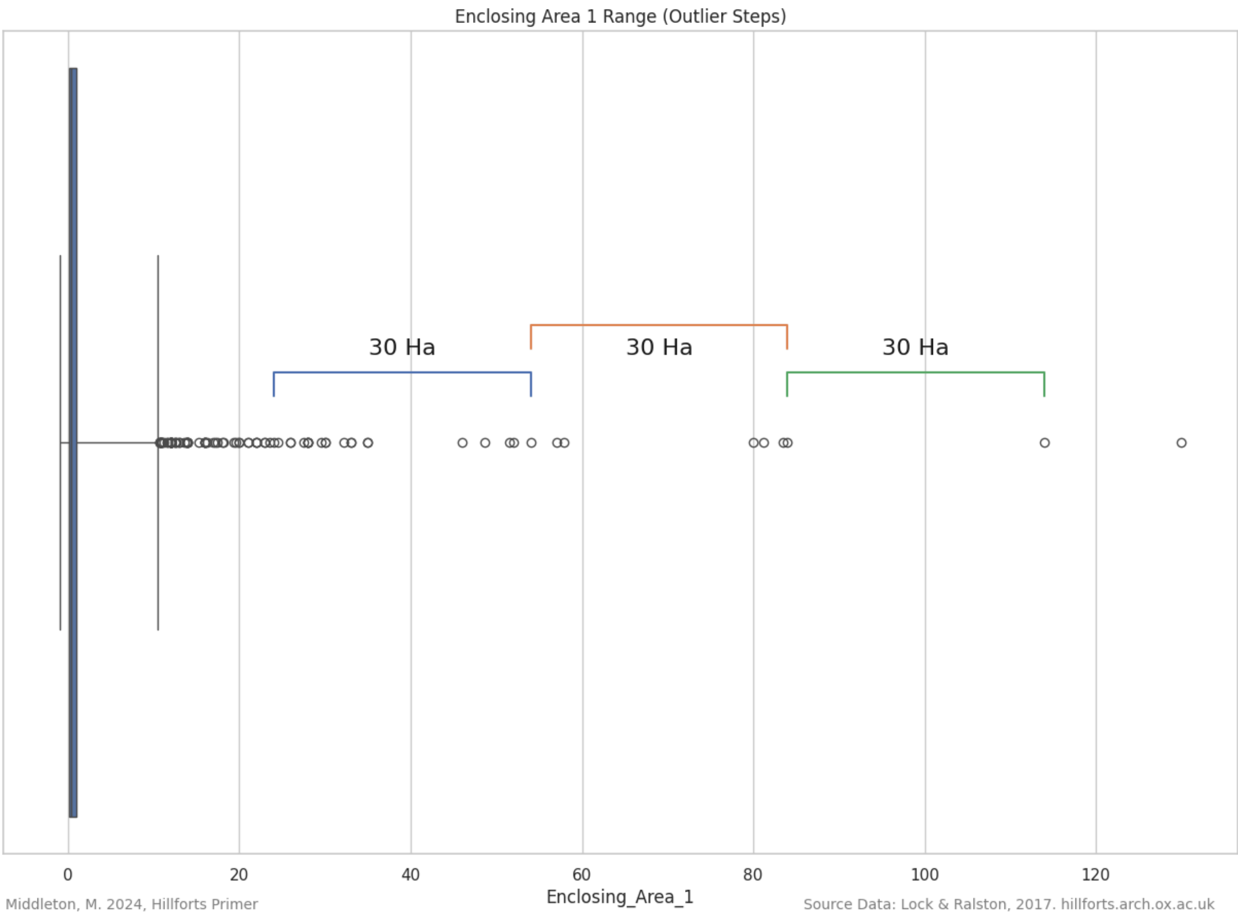
In [ ]:
```
# """Plot new boxplot"""
# Enclosing_Area_1_data_updated = plot_data_range(enclosing_numeric_data['Enclosing_Area_1'], 'Enclosing_Area_1_Upda
```

In [ ]:
```
# """Review new boxplot values"""
# Enclosing_Area_1_data_updated
```

Enclosing Area 1 - Outlier Distribution

The outliers are grouped into four small clusters. The first continues out from the main range; There is then a gap to the next cluster at around 50 Ha; another gap to a small cluster at 80 Ha and then, a final gap, to a pair of sites which are over 110 Ha. One observation is that there is a similarity in the step sizes between these clusters of around 30 Ha. It is important to note that the numbers of sites in these clusters are very small. See: Enclosing Area 1: Regional Boxplots

In [ ]:
```
enclosing_area_1_data = \
plot_data_range_plus(enclosing_numeric_data['Enclosing_Area_1'], \
                     'Enclosing_Area_1', "h")
```

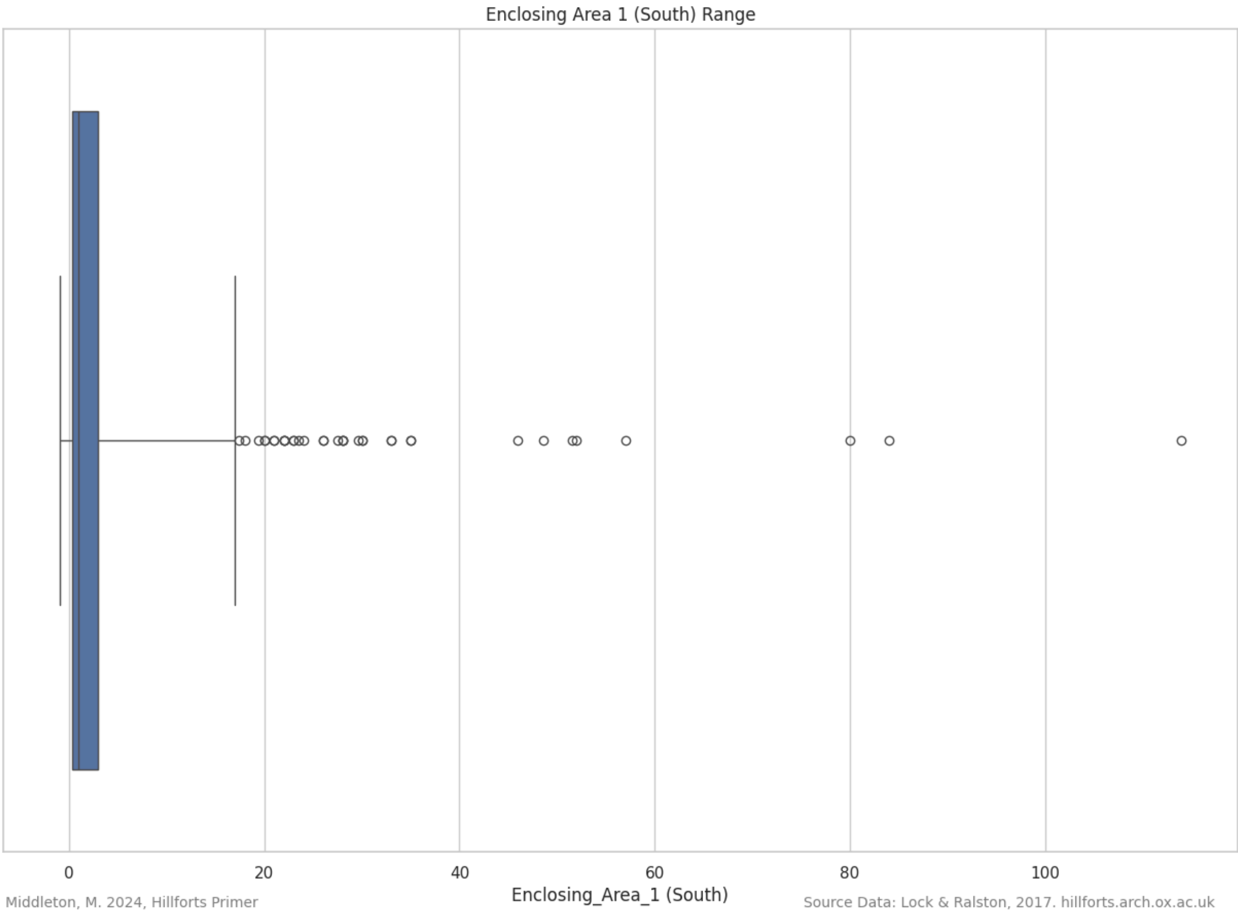


Enclosing Area 1: South Plotted

In the southern data package, 50% of the hillforts sit in a range between 0.3 and 3 hectares and 95.6% of the forts are less than 17 hectares. Most outliers are clustered near the main range, up to the high 30s. There is a small cluster between 40 and 60 hectares, two forts in the 80s and a single fort of 130 Ha. The median is 0.9 hectares and the bar chart shows the majority of forts are at the lower end of the range.

In [ ]:
```
south['uid'] = south.index
location_enclosing_data_south = \
pd.merge(south, enclosing_numeric_data, left_on='uid', right_index=True)
```

In [ ]:
```
enclosing_area_south_data = \
plot_data_range(location_enclosing_data_south['Enclosing_Area_1'], \
                'Enclosing_Area_1 (South)', "h")
```



In [ ]:
```
enclosing_area_south_data
```

Out[ ]:
```
[-1.0, 0.3, 0.9, 3.0, 17.0]
```

In [ ]:
```
location_enclosing_data_south['Enclosing_Area_1'].describe()
```

Out[ ]:
```
count    1555.000000
mean        2.637119
std         6.431279
min        -1.000000
25%         0.300000
50%         0.900000
75%         3.000000
max       114.000000
Name: Enclosing_Area_1, dtype: float64
```

Note how the mean and the median are quite different. The median, 0.9 Ha (the central value in a sorted list of values). Here the mean (2.63 Ha) is larger because of the huge variation in enclosing area. The small number of very large hillforts have an unduly large influence over the mean because the majority of hillforts are very small. A more realistic mean can be achieved by trimming the data to exclude a percentage of the data from the extremes.

In [ ]:
```
trim_pcnt = 0.1 # 10%
location_enclosing_data_trim_mean = \
stats.trim_mean(location_enclosing_data_south['Enclosing_Area_1'], trim_pcnt)
location_enclosing_data_trim_mean
```

Out[ ]:
```
1.5289799196787148
```

In [ ]:
```
test_cat_list_for_NA
```

Out[ ]:

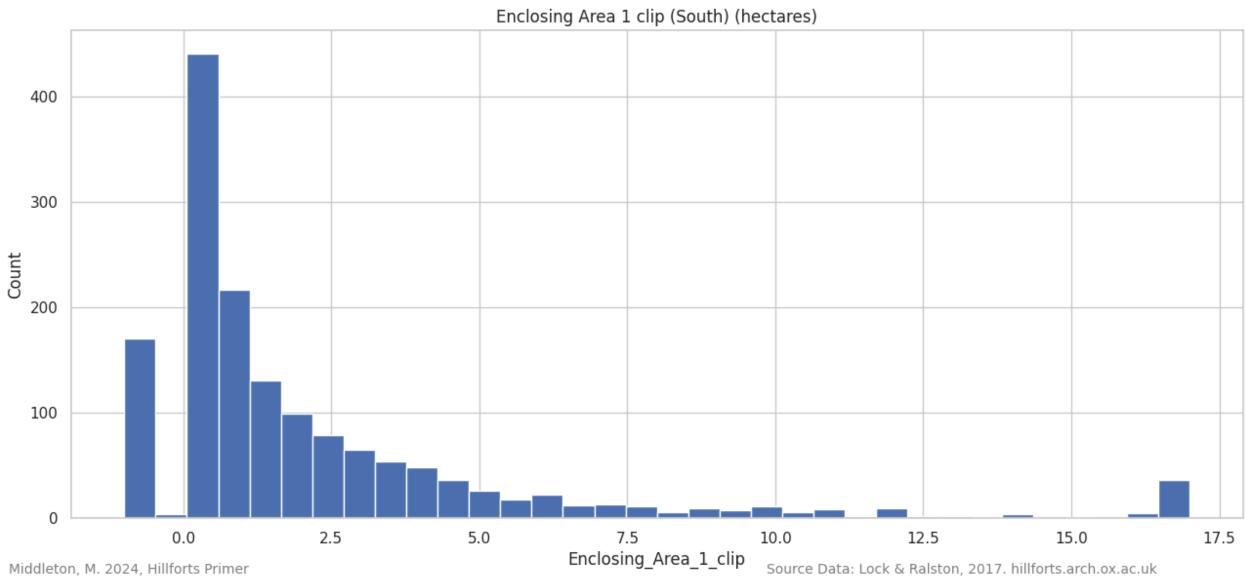
test_cat_list_for_NA
```
def test_cat_list_for_NA(dataframe, cat_list)

<no docstring>
```

To facilitate the reading of the plot, the data is clipped at the 75th percentile (17 Ha). Any data above 17 Ha is grouped at this value.

```
In [ ]: south_enclosing_area_1_data_clip = location_enclosing_data_south.copy()
        south_enclosing_area_1_data_clip['Enclosing_Area_1_clip'] = \
        south_enclosing_area_1_data_clip['Enclosing_Area_1'].\
        clip(south_enclosing_area_1_data_clip['Enclosing_Area_1'], \
             enclosing_area_south_data[4], axis=0)
        south_enclosing_area_1_data_clip['Enclosing_Area_1_clip'].describe()
```

```
Out[ ]: count    1555.000000
        mean        2.236154
        std         3.570042
        min        -1.000000
        25%         0.300000
        50%         0.900000
        75%         3.000000
        max        17.000000
        Name: Enclosing_Area_1_clip, dtype: float64
```

```
In [ ]: plot_bar_chart_numeric(south_enclosing_area_1_data_clip, 1, \
                               'Enclosing_Area_1_clip', 'Count', \
                               'Enclosing_Area_1_clip (South)', \
                               int(enclosing_area_south_data[4]*2), '(hectares)')
```



Enclosing Area 1: Northeast Plotted

In the Northeast, 50% of sites sit within a narrow band between 0.15 and 0.48 Ha and 95.6% of sites are less than 2.6 Ha. Most outliers string out from the top end of the main range up to 10 Ha. There is a single outlier at 24 Ha (1504: Roulston Scar, North Yorkshire) which is located right at the southern edge of the NE data package and may indicate that this fort has characteristics more in line with the southern data. See: Enclosing Area 1: Regional Boxplots.

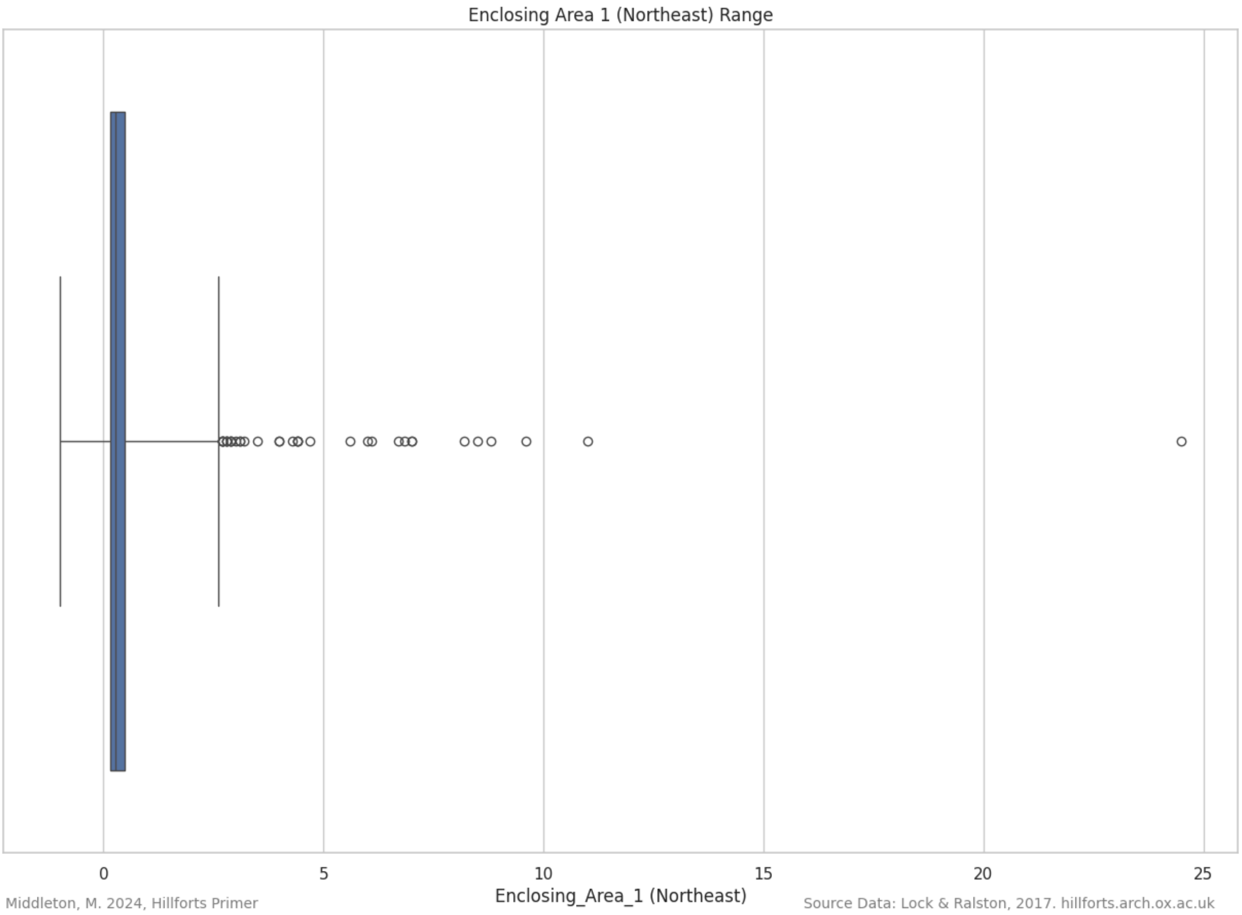
```
In [ ]: north_east['uid'] = north_east.index
        location_enclosing_data_ne = \
        pd.merge(north_east.reset_index(), enclosing_numeric_data, left_on='uid', \
                 right_index=True)
        location_enclosing_data_ne = pd.merge(name_and_number, \
                                              location_enclosing_data_ne, \
                                              left_index=True, right_on='uid')
```

```
In [ ]: location_enclosing_data_ne[location_enclosing_data_ne['Enclosing_Area_1'] > 20]
```

Out[ ]:

	Main_Atlas_Number	Main_Display_Name	index	Location_X	Location_Y	Cluster	uid	Enclosing_Area_1	Enclosing_Area_2	Enclosi
404	1504	Roulston Scar, North Yorkshire (Sutton Bank; C...	1437	-134884	7213231	Northeast	1437	24.5	-1.0	

```
In [ ]: enclosing_area_ne_data = plot_data_range(location_enclosing_data_ne['Enclosing_Area_1'], 'Enclosing_Area_1 (Northeas
```



In [ ]:
```
enclosing_area_ne_data
```

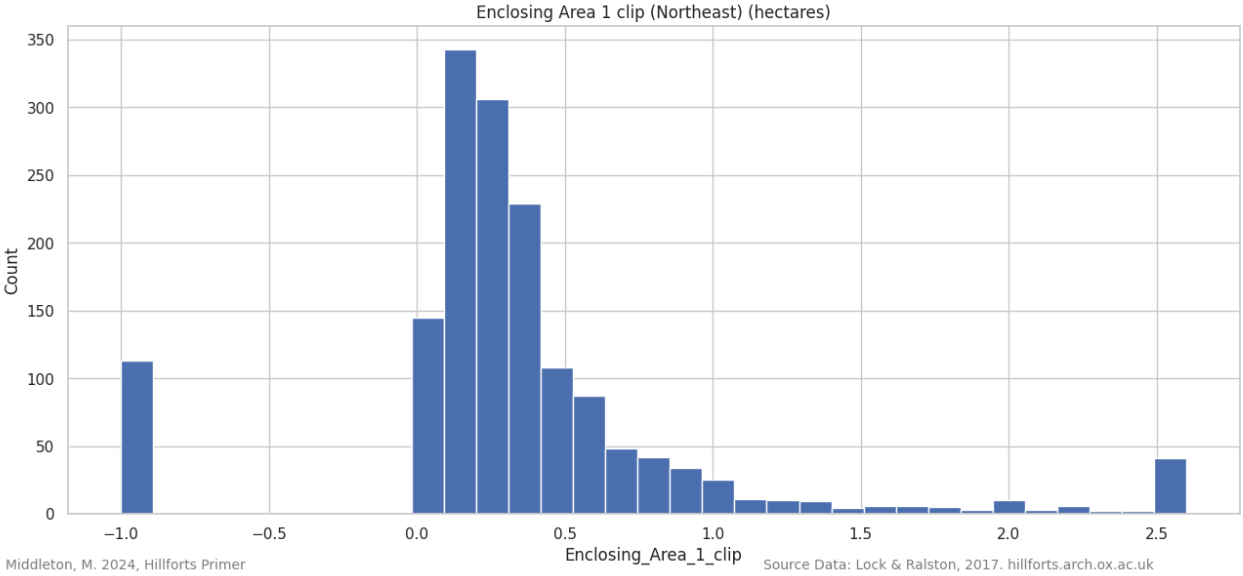
Out[ ]:
```
[-1.0, 0.15, 0.27, 0.48, 2.6]
```

In [ ]:
```
ne_enclosing_area_1_data_clip = location_enclosing_data_ne.copy()
ne_enclosing_area_1_data_clip['Enclosing_Area_1_clip'] = \
ne_enclosing_area_1_data_clip['Enclosing_Area_1'].\
clip(ne_enclosing_area_1_data_clip['Enclosing_Area_1'], \
     enclosing_area_ne_data[4], axis=0)
ne_enclosing_area_1_data_clip['Enclosing_Area_1_clip'].describe()
```

Out[ ]:
```
count    1598.000000
mean        0.354887
std         0.624009
min        -1.000000
25%         0.150000
50%         0.270000
75%         0.480000
max         2.600000
Name: Enclosing_Area_1_clip, dtype: float64
```

The data is clipped at the 75th percentile (2.6 Ha). Any data above 2.6 Ha is grouped at this value.

In [ ]:
```
plot_bar_chart_numeric(ne_enclosing_area_1_data_clip, 1, \
                       'Enclosing_Area_1_clip', 'Count', \
                       'Enclosing_Area_1_clip (Northeast)', \
                       int(enclosing_area_ne_data[4]*13), '(hectares)')
```



Enclosing Area 1: Northwest Plotted

This area is notable for the small size of most forts. The central 50% of sites are contained in a very narrow band between 0.05 and 0.22 Ha and 95.6% being less than 3.3 Ha. There are a small number of outliers up to 11.7 Ha and one single, exceptionally large, fort at 54 Ha (201: Mull of Galloway). See: Enclosing Area 1: Regional Boxplots.

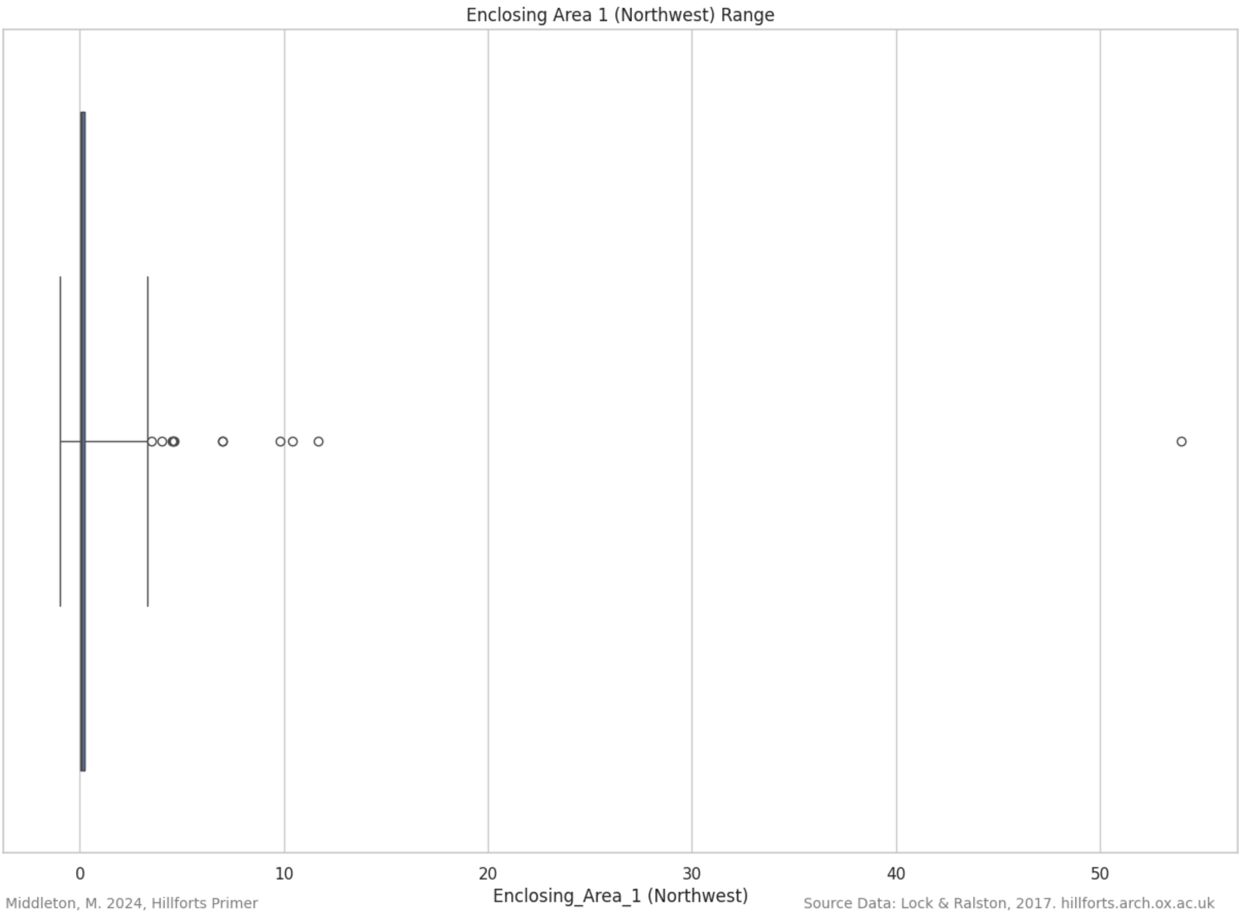
In [ ]:
```
north_west['uid'] = north_west.index
location_enclosing_data_nw = pd.merge(north_west.reset_index(), enclosing_numeric_data, left_on='uid', right_index=T
location_enclosing_data_nw = pd.merge(name_and_number, location_enclosing_data_nw, left_index=True, right_on='uid')
```

In [ ]:
```
location_enclosing_data_nw[location_enclosing_data_nw['Enclosing_Area_1']> 50]
```

Out[ ]:

	Main_Atlas_Number	Main_Display_Name	index	Location_X	Location_Y	Cluster	uid	Enclosing_Area_1	Enclosing_Area_2	Enclosing
36	201	Mull of Galloway, Dumfries & Galloway	194	-542988	7291829	Northwest	194	54.0	-1.0	

In [ ]:
```
enclosing_area_nw_data = \
plot_data_range(location_enclosing_data_nw['Enclosing_Area_1'], \
                'Enclosing_Area_1 (Northwest)', "h")
```



In [ ]:
```
enclosing_area_nw_data
```

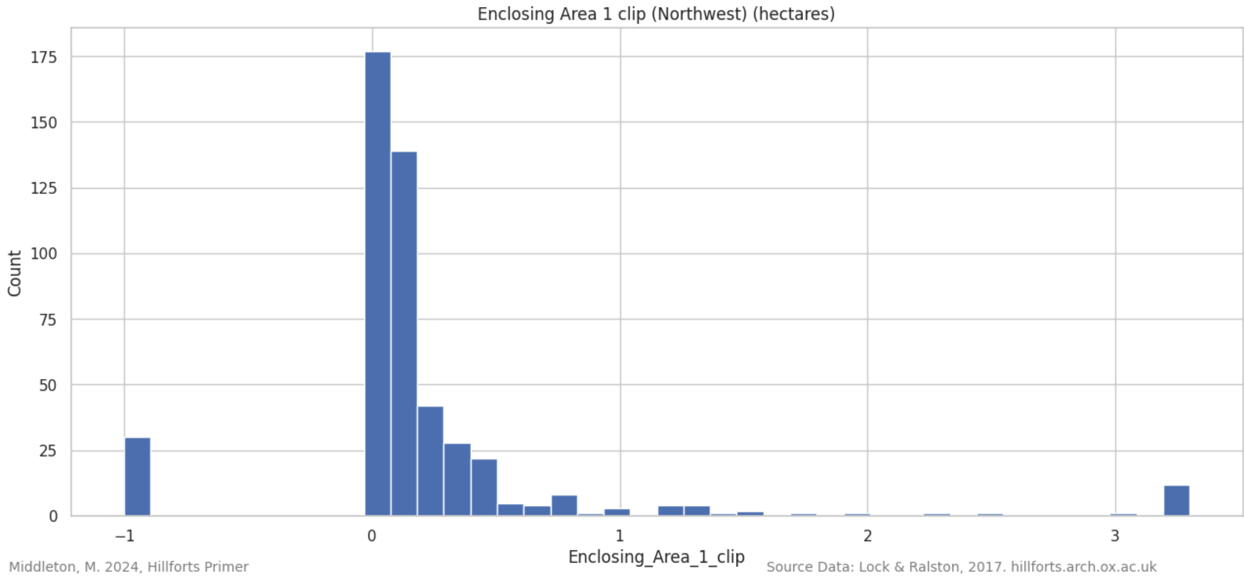
Out[ ]:
```
[-1.0, 0.05, 0.09, 0.22, 3.3]
```

In [ ]:
```
nw_enclosing_area_1_data_clip = location_enclosing_data_nw.copy()
nw_enclosing_area_1_data_clip['Enclosing_Area_1_clip'] = \
nw_enclosing_area_1_data_clip['Enclosing_Area_1'].\
clip(nw_enclosing_area_1_data_clip['Enclosing_Area_1'], \
     enclosing_area_nw_data[4], axis=0)
nw_enclosing_area_1_data_clip['Enclosing_Area_1_clip'].describe()
```

Out[ ]:
```
count    487.000000
mean       0.213747
std        0.656978
min       -1.000000
25%        0.050000
50%        0.090000
75%        0.220000
max        3.300000
Name: Enclosing_Area_1_clip, dtype: float64
```

The data is clipped at the 75th percentile (3.3 Ha). Any data above 3.3 Ha is gouped at this value.

In [ ]:
```
plot_bar_chart_numeric(nw_enclosing_area_1_data_clip, 1, \
                       'Enclosing_Area_1_clip', 'Count', \
                       'Enclosing_Area_1_clip (Northwest)', \
                       int(enclosing_area_south_data[4]*2.4), '(hectares)')
```



Enclosing Area 1: North Ireland Plotted

The central 50% of sites are between 0.07 and 0.79 Ha and 95.6% are less than 11.97 Ha. It is notable that the upper whisker is very long due to the concentration of forts at the lower end of the range and there being a large variance in size among the larger forts up to 11.7 Ha. See the bar chart below and Enclosing Area 1 Distribution of Data by Region. There is one single, atypically large, fort at 57.94 Ha (1104: Inishark (Inis Airc)). See: Enclosing Area 1: Regional Boxplots.

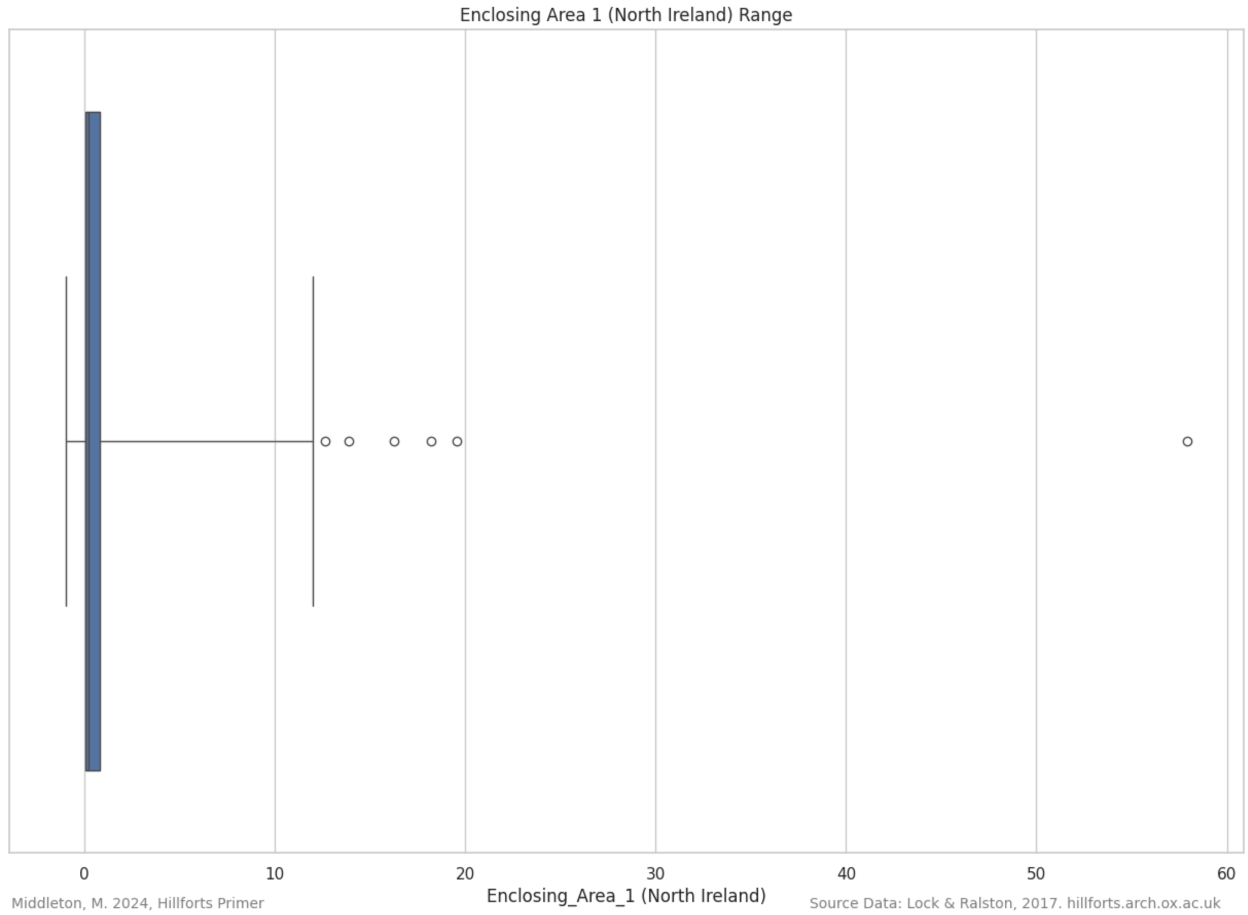
In [ ]:
```
north_ireland['uid'] = north_ireland.index
location_enclosing_data_ireland_n = \
pd.merge(north_ireland.reset_index(), enclosing_numeric_data, \
         left_on='uid', right_index=True)
location_enclosing_data_ireland_n = pd.merge(name_and_number, \
                                             location_enclosing_data_ireland_n, \
                                             left_index=True, right_on='uid')
```

In [ ]:
```
location_enclosing_data_ireland_n\
 [location_enclosing_data_ireland_n['Enclosing_Area_1']> 50]
```

Out[ ]:

	Main_Atlas_Number	Main_Display_Name	index	Location_X	Location_Y	Cluster	uid	Enclosing_Area_1	Enclosing_Area_2	Enclosing_
96	1104	Inishark (Inis Airc), Galway	1076	-1145432	7096391	North Ireland	1076	57.94	-1.0	

In [ ]:
```
enclosing_area_ireland_n_data = \
plot_data_range(location_enclosing_data_ireland_n['Enclosing_Area_1'], \
                'Enclosing_Area_1 (North Ireland)', "h")
```



In [ ]:
```
enclosing_area_ireland_n_data
```

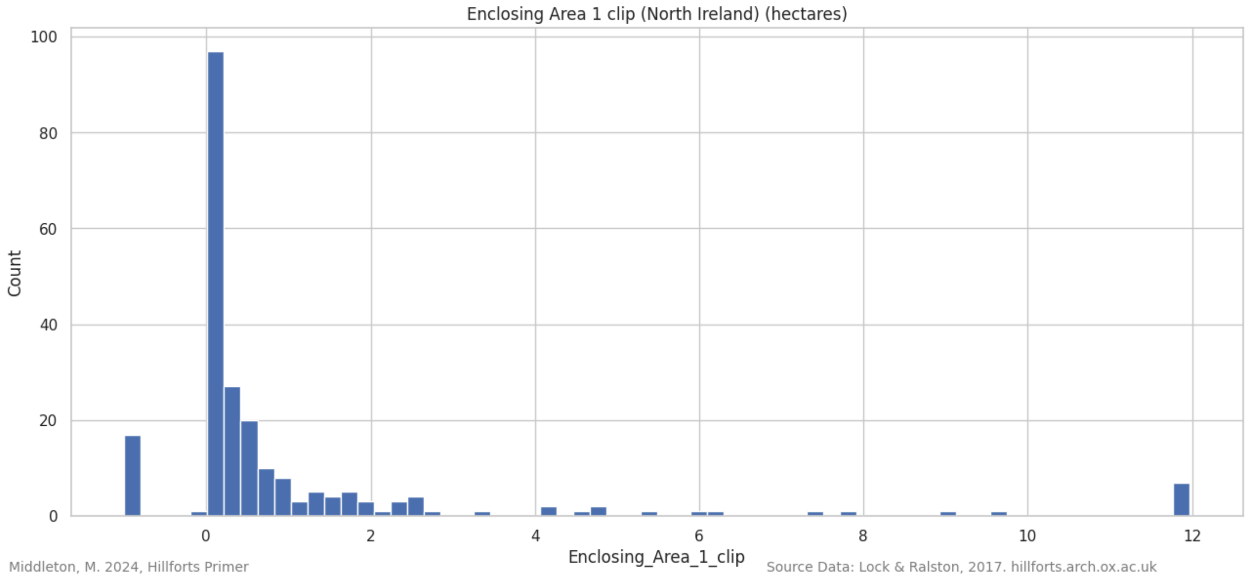
Out[ ]:
```
[-1.0, 0.07, 0.21, 0.79, 11.97]
```

In [ ]:
```
n_ie_enclosing_area_1_data_clip = location_enclosing_data_ireland_n.copy()
n_ie_enclosing_area_1_data_clip['Enclosing_Area_1_clip'] = \
n_ie_enclosing_area_1_data_clip['Enclosing_Area_1'].\
clip(n_ie_enclosing_area_1_data_clip['Enclosing_Area_1'], \
     enclosing_area_ireland_n_data[4], axis=0)
n_ie_enclosing_area_1_data_clip['Enclosing_Area_1_clip'].describe()
```

Out[ ]:
```
count    229.000000
mean       1.046594
std        2.479355
min       -1.000000
25%        0.070000
50%        0.210000
75%        0.790000
max       11.970000
Name: Enclosing_Area_1_clip, dtype: float64
```

The data is clipped at the 75th percentile (11.97 Ha). Any data above 11.97 Ha is gouped at this value.

In [ ]:
```
plot_bar_chart_numeric(n_ie_enclosing_area_1_data_clip, 1, \
                       'Enclosing_Area_1_clip', 'Count', \
                       'Enclosing_Area_1_clip (North Ireland)', \
                       int(enclosing_area_south_data[4]*3.8), '(hectares)')
```



Enclosing Area 1: South Ireland Plotted

The boxplot for South Ireland is compressed due to the huge scale of the outliers in this region. For more clarity see Enclosing Area 1: Regional Boxplots. The central 50% of sites are between 0.11 and 1.3 Ha and 95.6% are less than 12.01 Ha. Like North Ireland, the upper whisker is long due to the concentration of forts below 0.2 Ha and the variance in size among the larger forts up to 12.01 Ha. See the bar chart below. There are three forts over 80 Ha, of which one, at 130 Ha, is enormous (727: Spinans Hill 2). This is the largest fort, by area, recorded in the atlas.

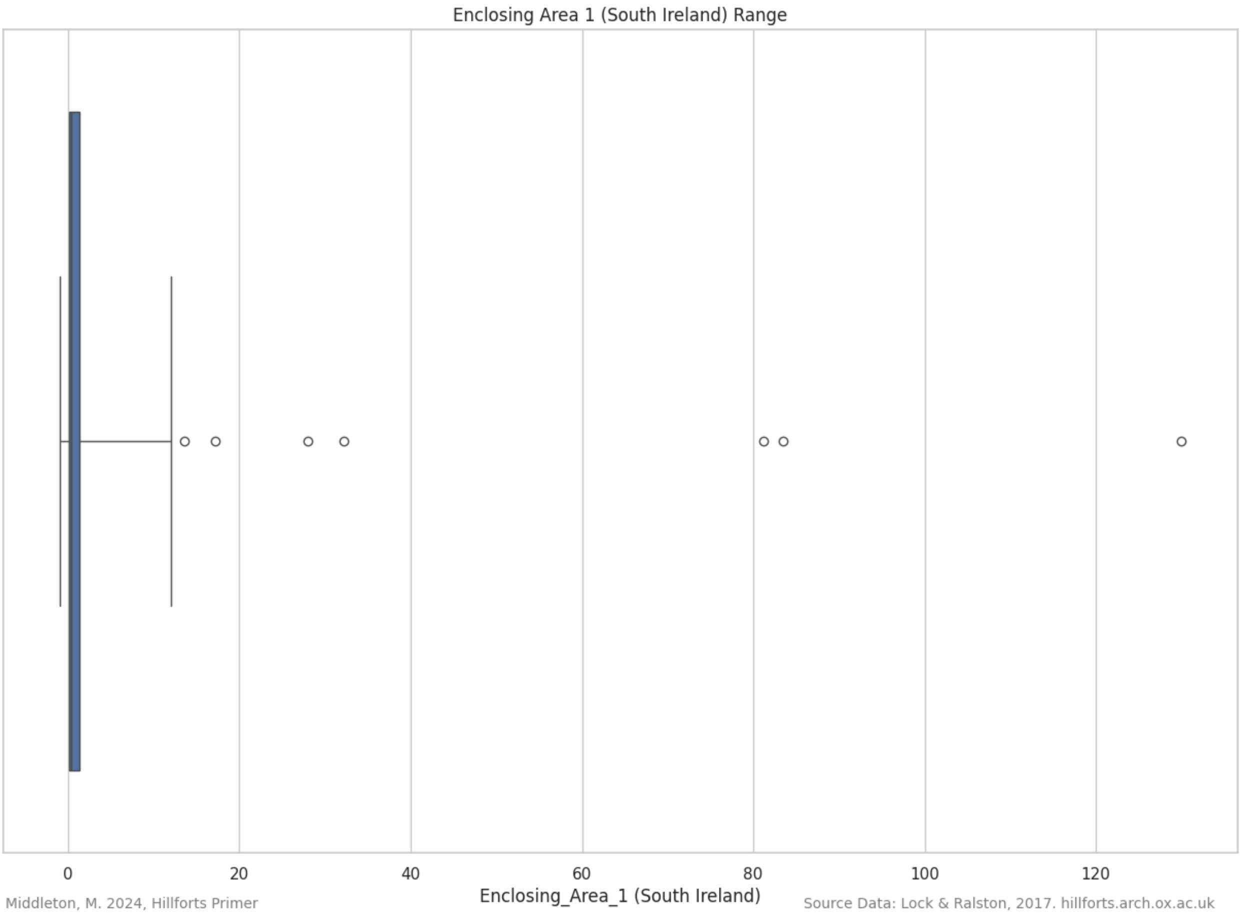
In [ ]:
```
south_ireland['uid'] = south_ireland.index
location_enclosing_data_ireland_s = \
pd.merge(south_ireland.reset_index(), enclosing_numeric_data, left_on='uid', \
         right_index=True)
location_enclosing_data_ireland_s = \
pd.merge(name_and_number, location_enclosing_data_ireland_s, left_index=True, \
         right_on='uid')
```

In [ ]:
```
location_enclosing_data_ireland_s\
 [location_enclosing_data_ireland_s['Enclosing_Area_1']> 80].\
 sort_values(by='Enclosing_Area_1', ascending=False)
```

Out[ ]:

	Main_Atlas_Number	Main_Display_Name	index	Location_X	Location_Y	Cluster	uid	Enclosing_Area_1	Enclosing_Area_2	Enclosing
48	727	Spinans Hill 2, Wicklow (Brusselstown, Spinans...	705	-737471	6976335	South Ireland	705	130.0	-1.0	
264	1970	Ballynacarriga, Cork	1864	-1130188	6726204	South Ireland	1864	83.5	-1.0	
121	901	Downmacpatrick (Old Head), Cork	878	-950310	6730002	South Ireland	878	81.2	-1.0	

In [ ]:
```
enclosing_area_ireland_s_data = \
plot_data_range(location_enclosing_data_ireland_s\
 ['Enclosing_Area_1'], 'Enclosing_Area_1 (South Ireland)', "h")
```



In [ ]:
```
enclosing_area_ireland_s_data
```

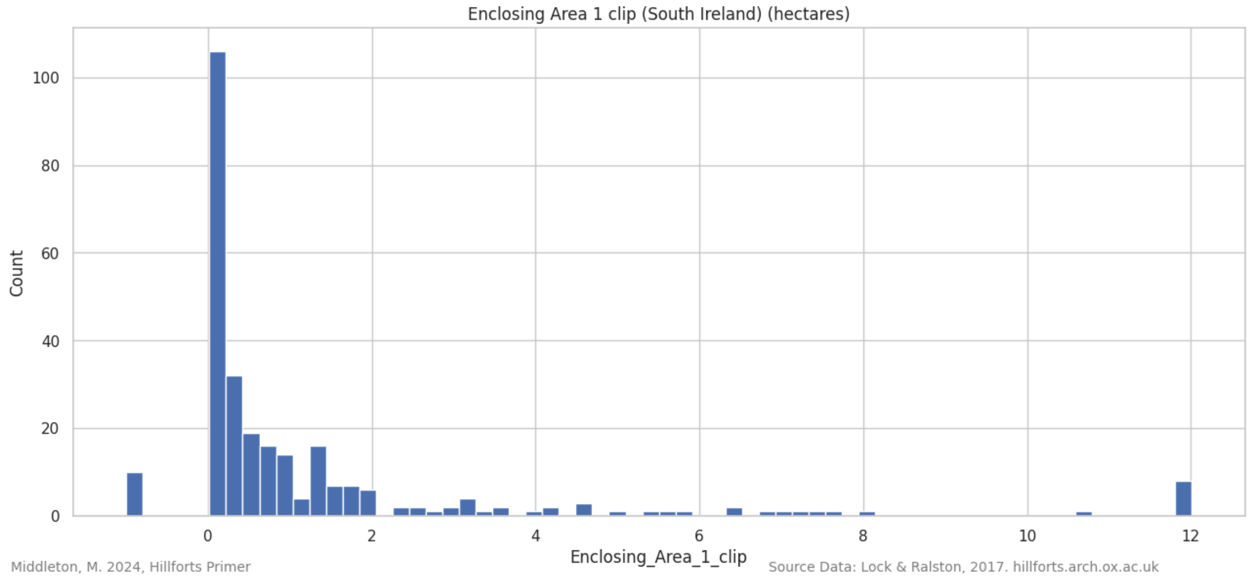
Out[ ]:
```
[-1.0, 0.11, 0.32, 1.2975, 12.01]
```

In [ ]:
```
s_ie_enclosing_area_1_data_clip = location_enclosing_data_ireland_s.copy()
s_ie_enclosing_area_1_data_clip['Enclosing_Area_1_clip'] = \
s_ie_enclosing_area_1_data_clip['Enclosing_Area_1'].\
clip(s_ie_enclosing_area_1_data_clip['Enclosing_Area_1'], \
     enclosing_area_ireland_s_data[4], axis=0)
s_ie_enclosing_area_1_data_clip['Enclosing_Area_1_clip'].describe()
```

Out[ ]:
```
count    278.000000
mean       1.289640
std        2.483341
min       -1.000000
25%        0.110000
50%        0.320000
75%        1.297500
max       12.010000
Name: Enclosing_Area_1_clip, dtype: float64
```

The data is clipped at the 75th percentile (12.01 Ha). Any data above 12.01 Ha is grouped at this value.

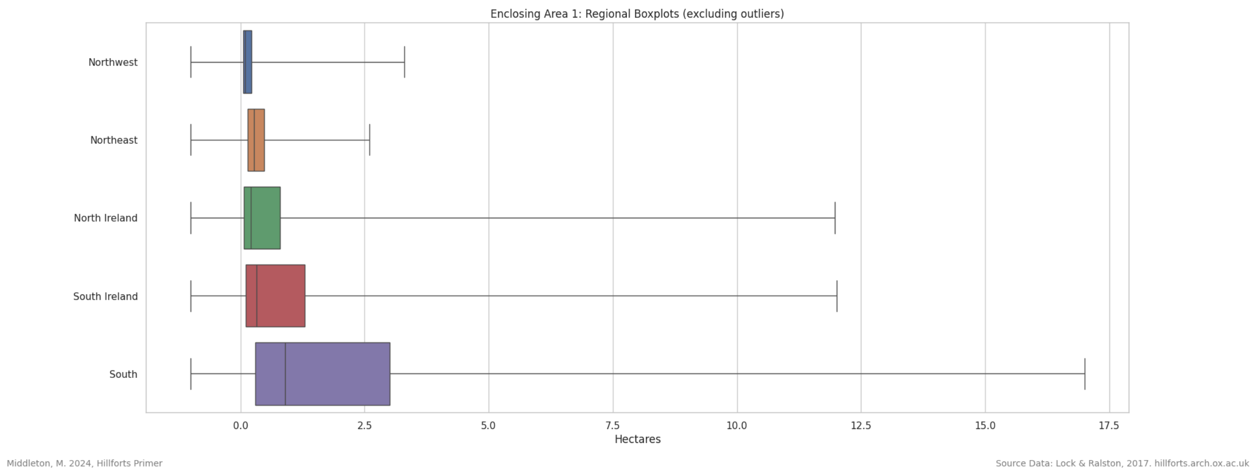
In [ ]:
```
plot_bar_chart_numeric(s_ie_enclosing_area_1_data_clip, 1, \
                       'Enclosing_Area_1_clip', 'Count', \
                       'Enclosing_Area_1_clip (South Ireland)', \
                       int(enclosing_area_south_data[4]*3.8), '(hectares)')
```



Enclosing Area 1: Regional Boxplots

Removing outliers makes it easier to see the detail in the boxplots. They show a clear difference between the North (Scotland and N. England) and South (Wales and S. England). The North is dominated by small forts. The Northwest is notable for its tiny forts, of which the majority sit within a narrow range up to 0.22 Ha. In contrast, the South has a much larger range of fort areas, with an interquartile range between 0.3 and 3 Ha. North Ireland and South Ireland are similar with South Ireland differing in having slightly larger forts overall. The median size of forts in the Northeast and in Ireland are roughly similar ranging from 0.21 to 0.32 Ha. The Northwest are noticeably smaller, with a median of 0.09 Ha and the South are considerably larger, with a median of 0.9 Ha.

In [ ]:
```
regional_dict = \
 {'Northwest': location_enclosing_data_nw['Enclosing_Area_1'], \
  'Northeast': location_enclosing_data_ne['Enclosing_Area_1'], \
  'North Ireland': location_enclosing_data_ireland_n['Enclosing_Area_1'], \
  'South Ireland': location_enclosing_data_ireland_s['Enclosing_Area_1'], \
  'South': location_enclosing_data_south['Enclosing_Area_1']}
plot_data = pd.DataFrame.from_dict(regional_dict)
plt.figure(figsize=(20,8))
ax = sns.boxplot(data=plot_data, orient="h" , whis=[2.2, 97.8], showfliers=False);
add_annotation_plot(ax)
ax.set_xlabel('Hectares')
title = 'Enclosing_Area_1: Regional Boxplots (excluding outliers)'
plt.title(get_print_title(title))
save_fig(title)
plt.show()
```

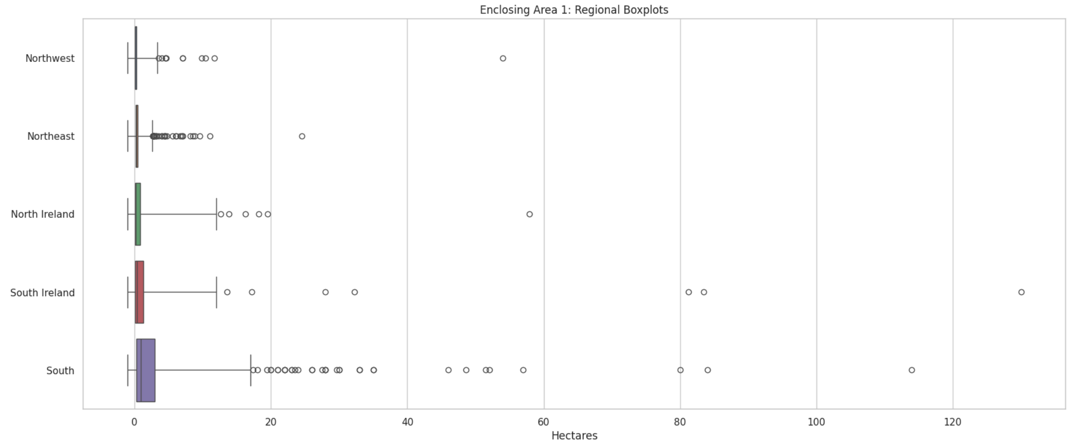


Enclosing Area 1: Outliers

All the regions have outliers, and all have a small number of atypical, large forts. In general, these are very few in number and where outliers are present, they mostly cluster just above the main data range. The steps between outliers, noted in Enclosing Area 1 - Outlier Distribution, are only visible in the south data package.

In [ ]:
```
plt.figure(figsize=(20,8))
ax = sns.boxplot(data=plot_data, orient="h" , whis=[2.2, 97.8], showfliers=True);
add_annotation_plot(ax)
ax.set_xlabel('Hectares')
```

```
title = 'Enclosing_Area_1: Regional Boxplots'
plt.title(get_print_title(title))
save_fig(title)
plt.show()
```



Middleton, M. 2024, Hillforts Primer

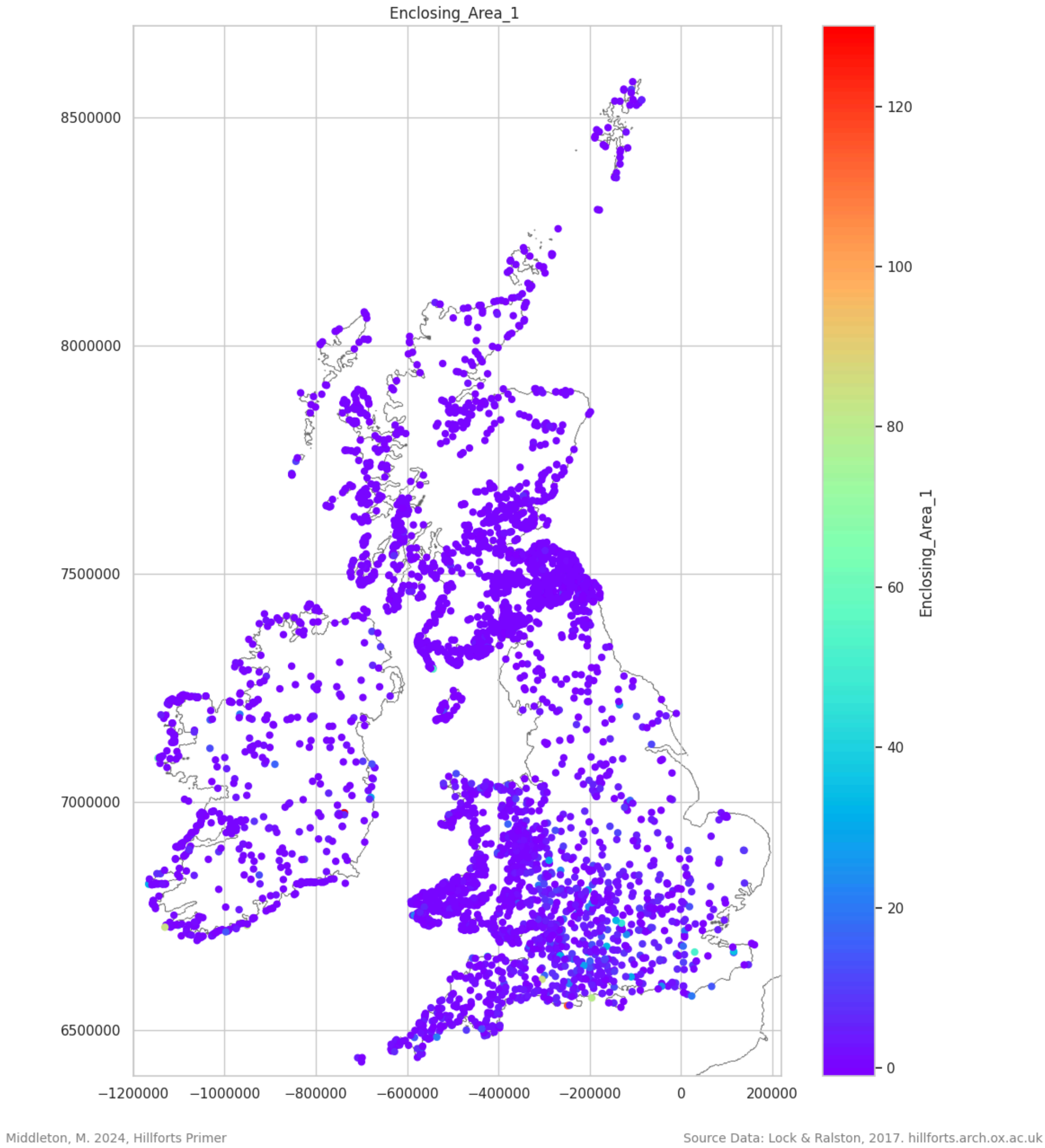
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area 1 Mapped by Size

With most hillforts being less than 1 Ha and 'Enclosing Area 1' having a range up to 130 Ha, the resulting map, based on area, lacks clarity.

In [ ]:
```
location_enclosing_data = \
pd.merge(location_numeric_data_short, enclosing_numeric_data, left_index=True,\
         right_index=True)
```

In [ ]:
```
plot_values(location_enclosing_data, 'Enclosing_Area_1', 'Enclosing_Area_1')
```



Enclosing Area 1 Interquartile Range (mid 50%) Mapped

Plotting the 50% of forts, from the mid-range of the boxplot (the IQR), shows a distribution across the Scottish Borders, the Welsh uplands, the southern end of the west coast of Scotland, the north coast of the South-west Peninsula, coastal sites around the south of Ireland and a peppering of other sites across central Ireland and NE Scotland. What is noticeable is the rarity of forts, in this range, across England. Those, that do fall in England, tend to be at the upper end of the area range. The hillforts in the interquartile range are located predominantly on the eastern Southern Uplands and the Cambrian Mountains.

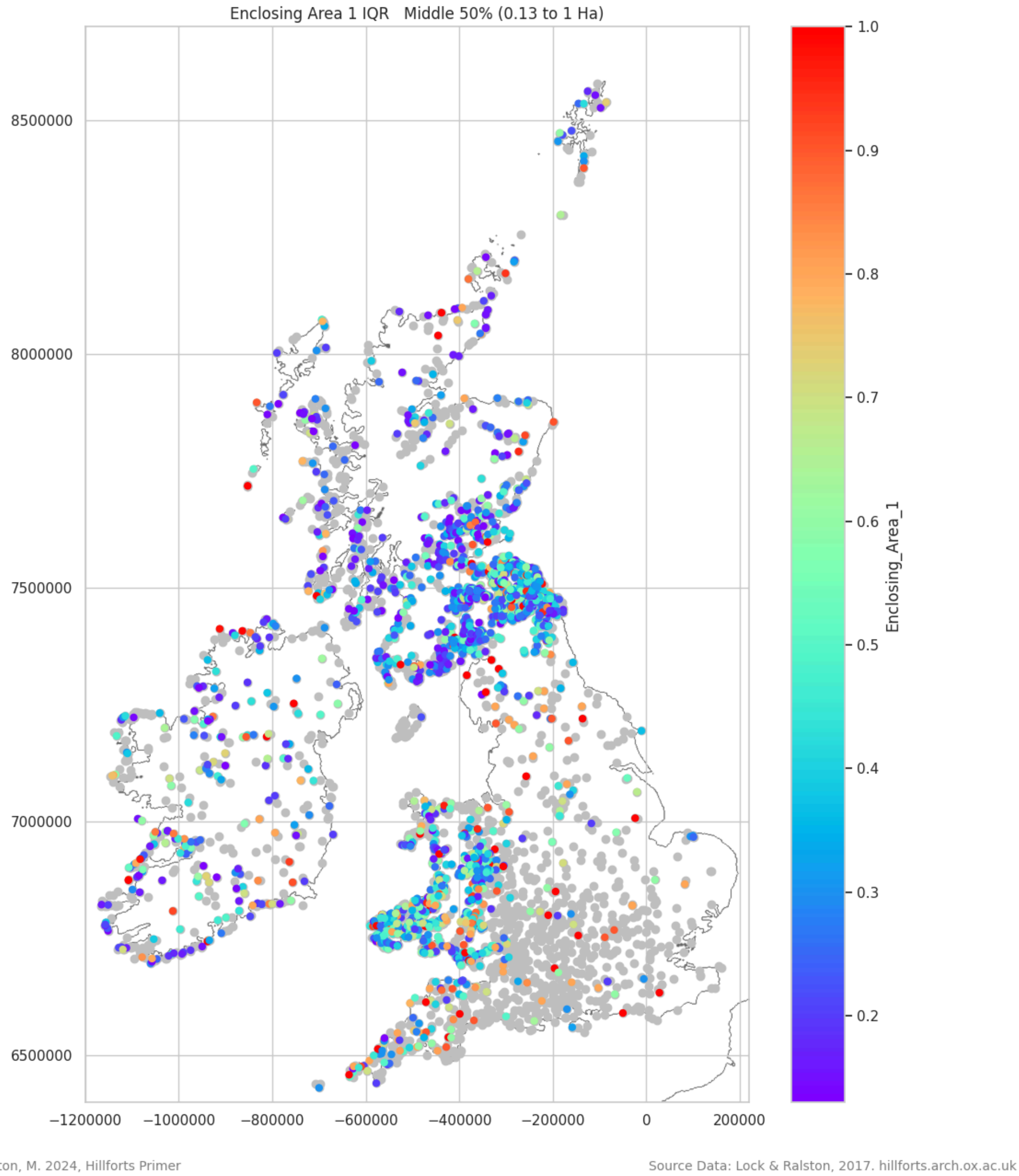
In [ ]:

```
enclosing_area_1_013_1 = location_enclosing_data.copy()
enclosing_area_1_013_1 = \
enclosing_area_1_013_1[enclosing_area_1_013_1['Enclosing_Area_1'].\
                       between(0.13, 1)]
enclosing_area_1_013_1['Enclosing_Area_1'].describe()
```

Out[ ]:

```
count    2100.000000
mean        0.408538
std         0.231068
min         0.130000
25%         0.230000
50%         0.340000
75%         0.550000
max         1.000000
Name: Enclosing_Area_1, dtype: float64
```

```
In [ ]: plot_type_values(enclosing_area_1_013_1, 'Enclosing_Area_1', 'Enclosing_Area_1', \
                         extra='IQR - Middle 50% (0.13 to 1 Ha)')
```

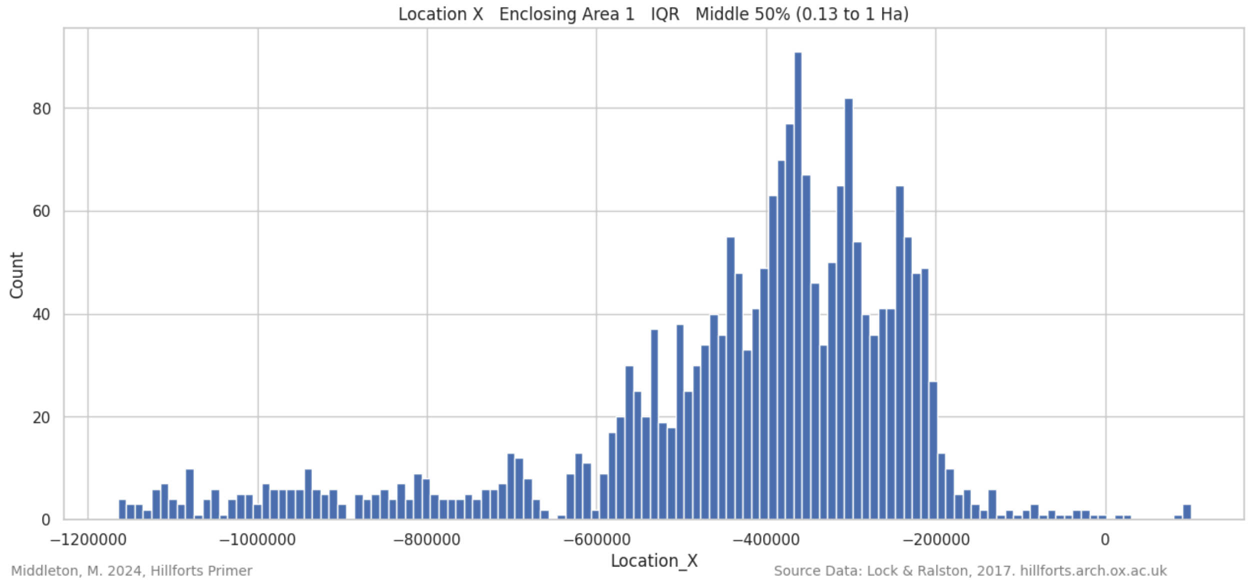


Enclosing Area 1 Interquartile Range (mid 50%) Location_X Plotted

The density peaks towards the east.

```
In [ ]: plot_histogram(enclosing_area_1_013_1['Location_X'], 'Location_X', \
                       'Location_X - Enclosing_Area_1 - IQR - Middle 50% (0.13 to 1 Ha)',\
                       10000)
```

```
10000
```

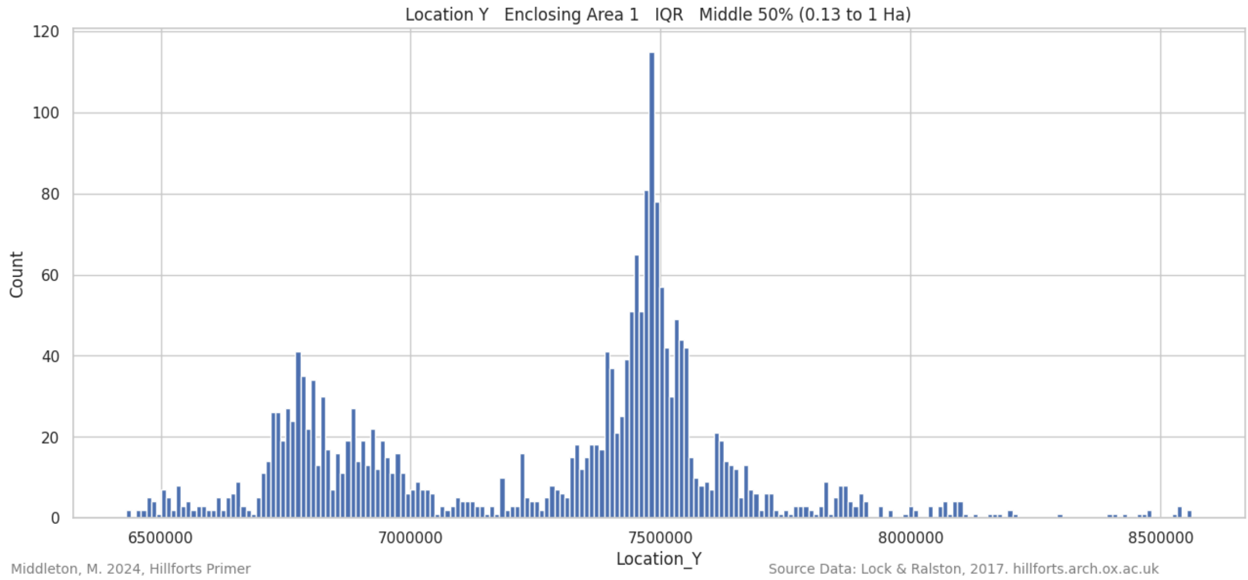


Enclosing Area 1 Interquartile Range (mid 50%) Location_Y Plotted

Plotting the distribution against the Location_Y axis (the northing) shows the peak to the North to be nearly three times that in the South.

```
In [ ]: plot_histogram(enclosing_area_1_013_1['Location_Y'], 'Location_Y', \
                       'Location_Y - Enclosing_Area_1 - IQR - Middle 50% (0.13 to 1 Ha)', 10000)
```

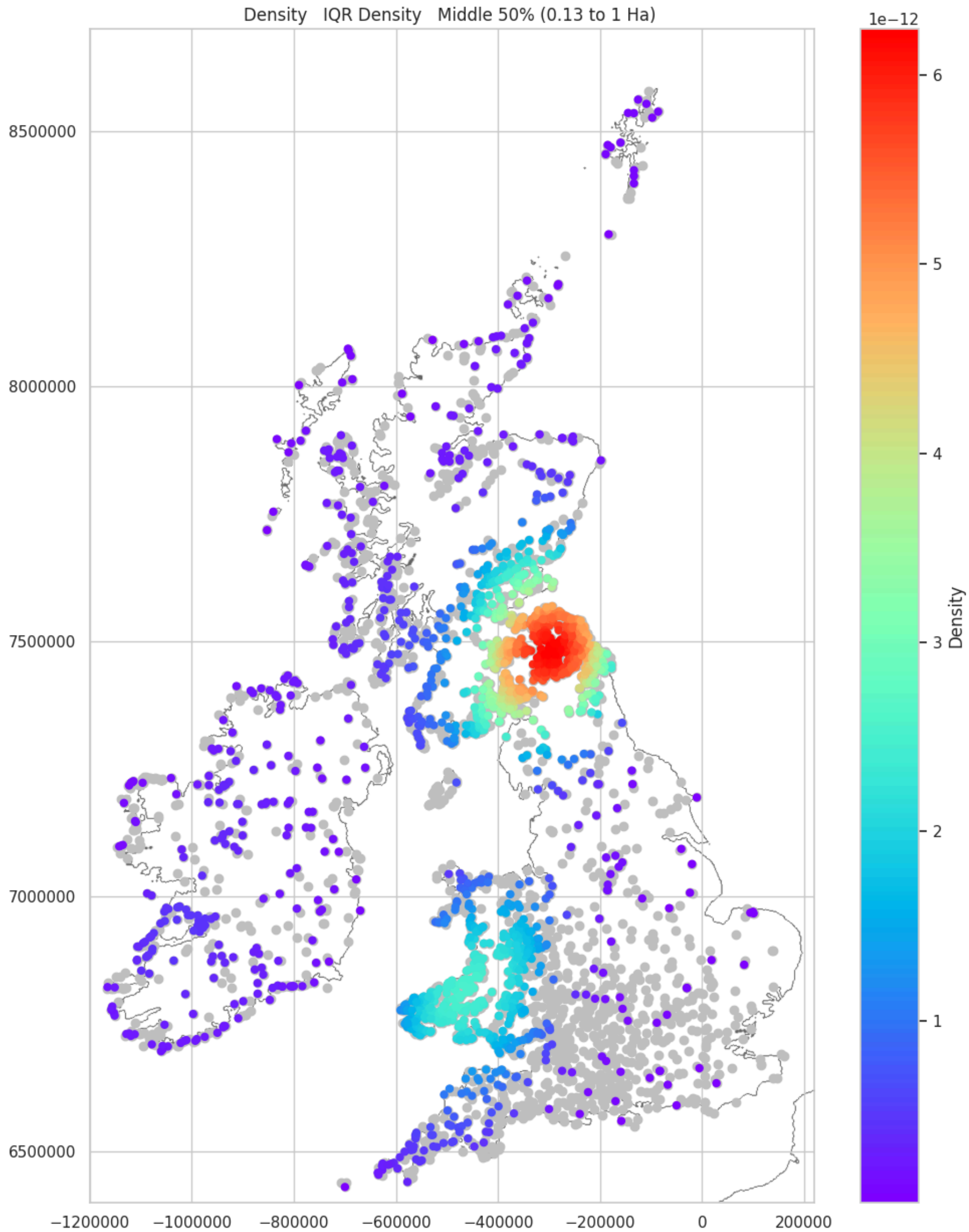
```
10000
```



Enclosing Area 1 Interquartile Range (mid 50%) Density Mapped

The density plot of the interquartile range shows a very intense cluster in the Northeast and a secondary cluster over the southern end of the Cambrian Mountains.

```
In [ ]: plot_density_over_grey(enclosing_area_1_013_1, \
                               'IQR Density - Middle 50% (0.13 to 1 Ha)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area 1 First (lower) and Forth (Upper) Quarters (excluding outliers) Mapped

Mapping the first quarter of sites by area (the blues in the figure below), shows a distribution along the west coast of Scotland and the south, west and north coasts of Ireland. Additionally, there are forts along the Great Glen and forts along the south coast of Fife, and up into Perthshire and Angus. In Wales, there are small clusters of forts at a couple of locations along the west coast and along the Brecon Beacons. There are very few, from this range, within the mainland of Ireland, England or eastern and northern Wales.

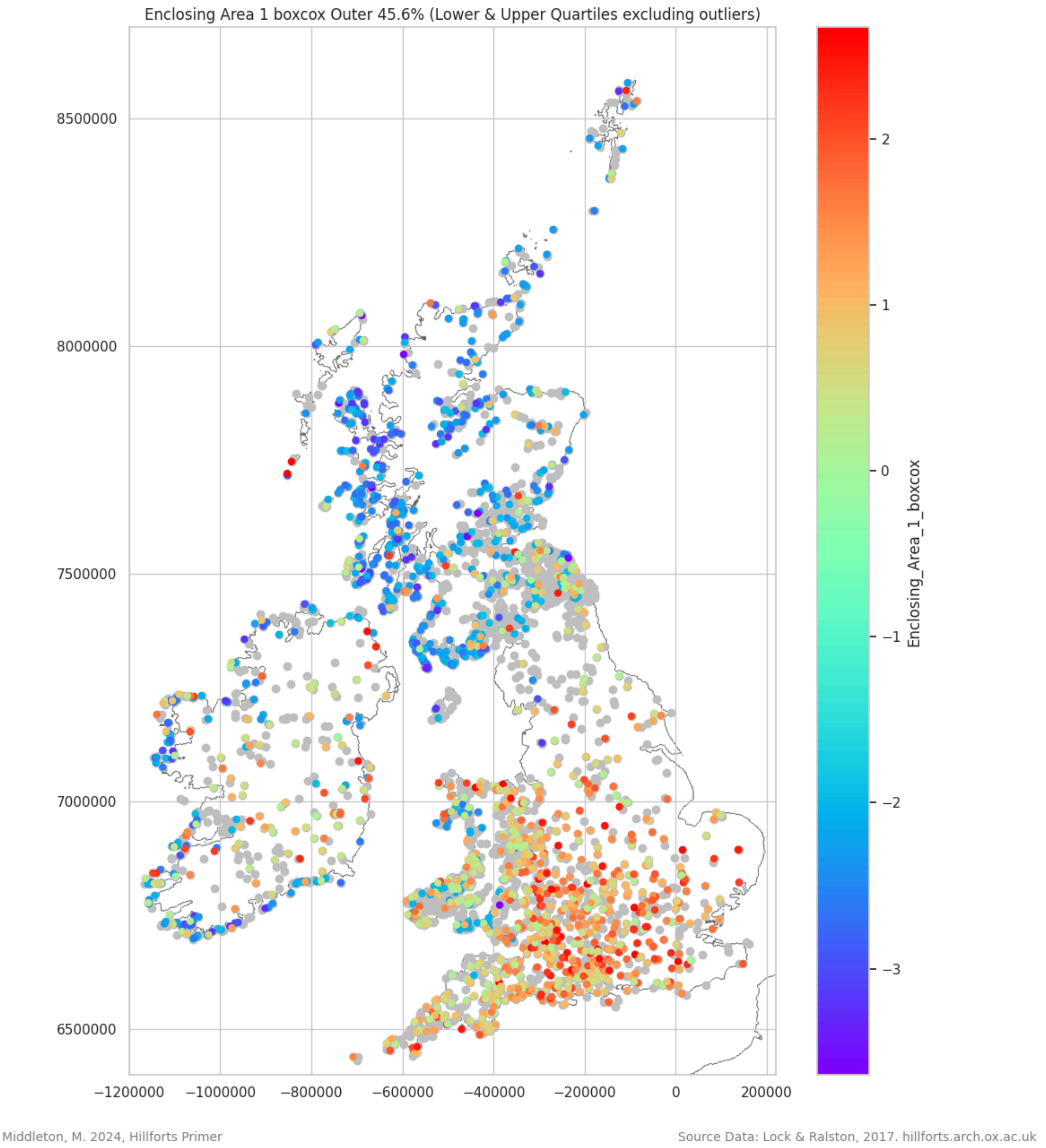
Mapping the fourth quarter, (the greens to red), shows a distribution of sites across eastern Wales, south-central England and into the Southwest Peninsula. There are a sprinkling of sites across the uplands of northern England and eastern Scotland, with a similar concentration across central and southern Ireland. There are noticeably few across western and northern Scotland.

In [ ]:
```
from scipy import stats
```

In [ ]:
```
enclosing_area_1_temp = location_enclosing_data.copy()
enclosing_area_1_low = \
enclosing_area_1_temp[(enclosing_area_1_temp['Enclosing_Area_1']>=0) & \
 (enclosing_area_1_temp['Enclosing_Area_1']<0.13)]
enclosing_area_1_high = \
enclosing_area_1_temp[(enclosing_area_1_temp['Enclosing_Area_1']>1) & \
 (enclosing_area_1_temp['Enclosing_Area_1']<=10.5)]
```

```
enclosing_area_1_high_low = pd.concat([enclosing_area_1_low, \
                                        enclosing_area_1_high])
enclosing_area_1_high_low['Enclosing_Area_1_boxcox'] = \
stats.boxcox(enclosing_area_1_high_low['Enclosing_Area_1'])[0]
```

In [ ]:
```
plot_type_values(enclosing_area_1_high_low, 'Enclosing_Area_1_boxcox', \
                 'Enclosing_Area_1 (Boxcox)', \
                 extra='Outer 45.6% (Lower & Upper Quartiles excluding outliers)')
```

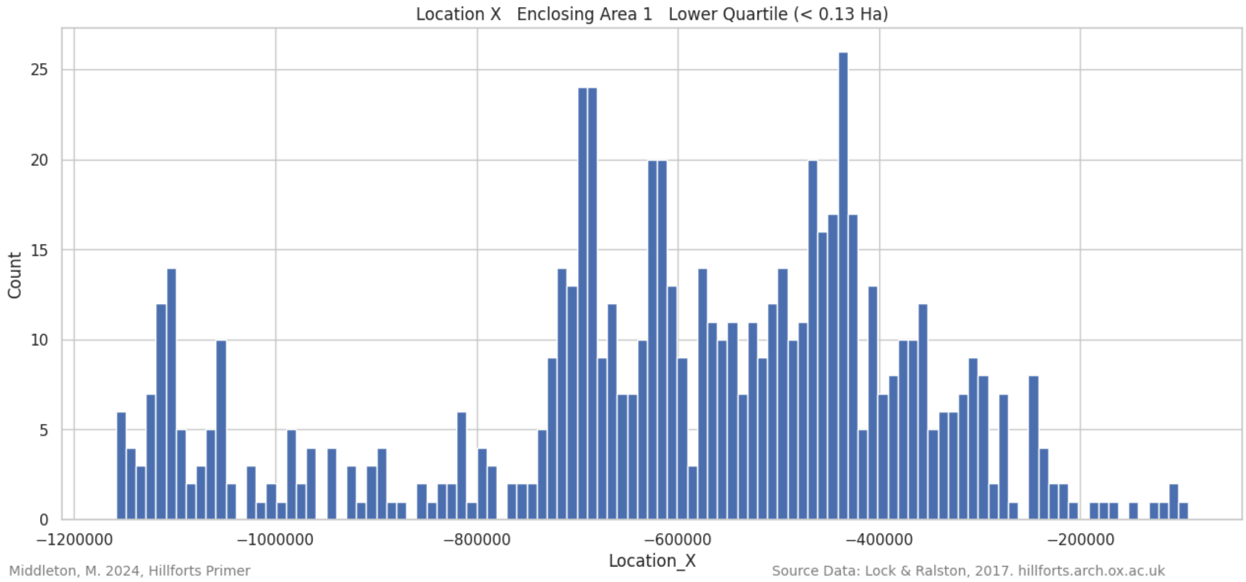


Enclosing Area 1 Lower Quartile (excluding outliers) Location_X Plotted

The Location_X data shows peaks in the data to the west and south of Ireland then three peaks across Scotland. Two over the western seaboard and one around the Great Glen.

In [ ]:
```
plot_histogram(enclosing_area_1_low['Location_X'], 'Location_X', \
               'Location_X - Enclosing_Area_1 - Lower Quartile (< 0.13 Ha)', \
               10000)
```

```
10000
```

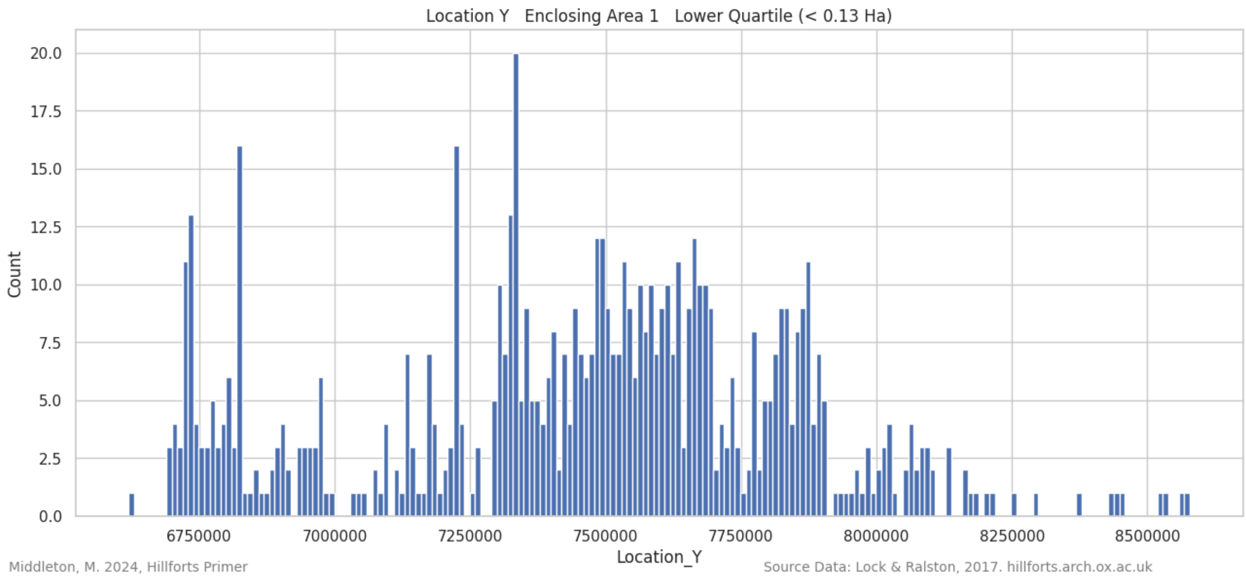


Enclosing Area 1 Lower Quartile (excluding outliers) Location_Y Plotted

In the Location_Y axis there are peaks over the south coast of Ireland and a high peak aligning with the south coast of Galloway. This high peak projects from a broad, lower peak, running the length of the west coast of Scotland, up to Skye.

In [ ]:
```
plot_histogram(enclosing_area_1_low['Location_Y'], 'Location_Y', \
               'Location_Y - Enclosing_Area_1 - Lower Quartile (< 0.13 Ha)', \
               10000)
```

10000

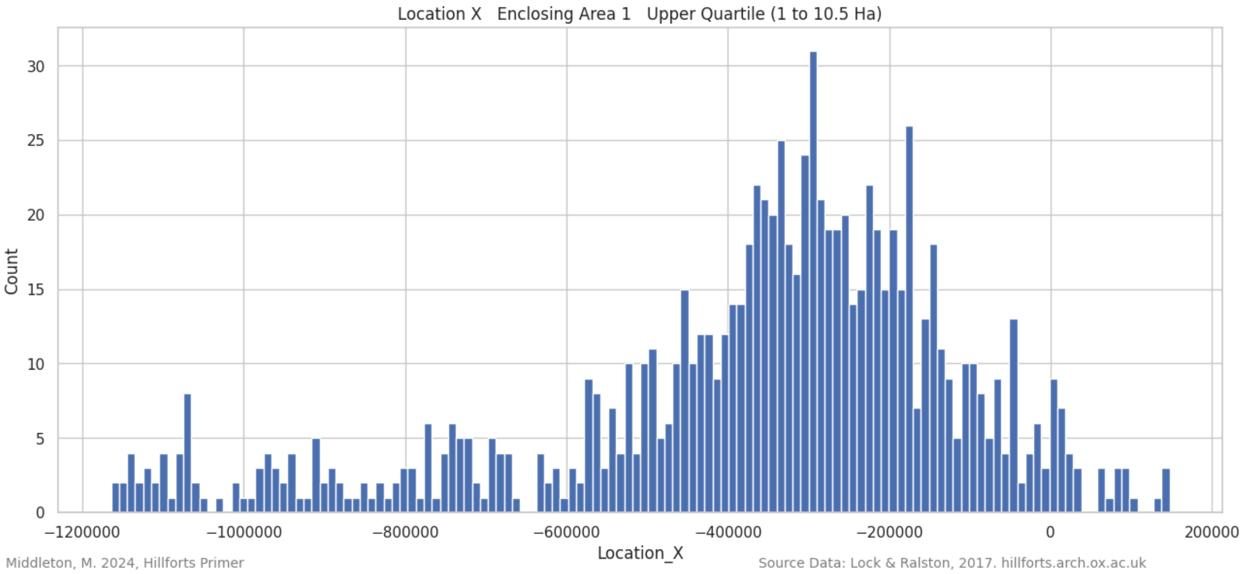


Enclosing Area 1 Upper Quartile (excluding outliers) Location_X Plotted

The fourth quartile Location_X data shows the cluster in southern England to have a broad peak.

In [ ]:
```
plot_histogram(enclosing_area_1_high['Location_X'], \
               'Location_X', \
               'Location_X - Enclosing_Area_1 - Upper Quartile (1 to 10.5 Ha)', \
               10000)
```

10000

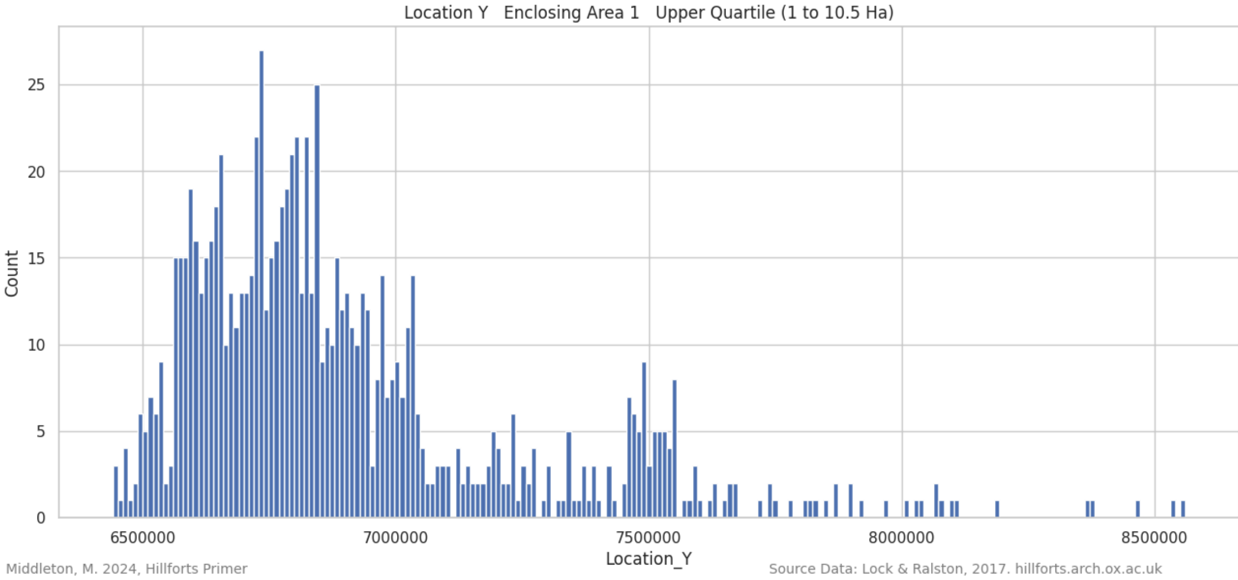


Enclosing Area 1 Upper Quartile (excluding outliers) Location_Y Plotted

In the Location_Y data, although there is a small cluster over the Southern Uplands, the main peak, in the South, is broad and tall.

```
In [ ]: plot_histogram(enclosing_area_1_high['Location_Y'], \
                       'Location_Y', \
                       'Location_Y - Enclosing_Area_1 - Upper Quartile (1 to 10.5 Ha)', \
                       10000)
```

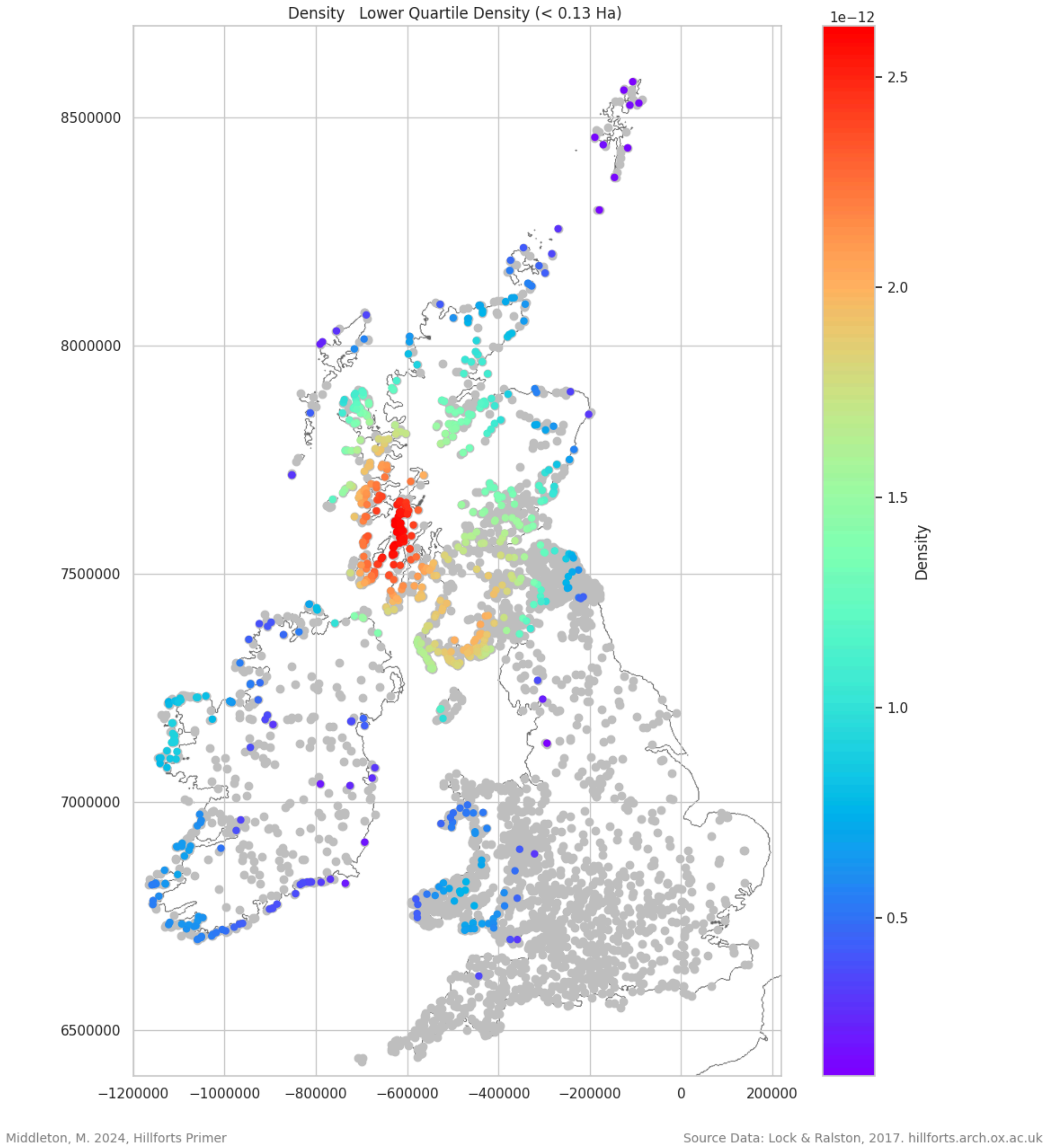
```
10000
```



Enclosing Area 1 Lower Quartile Density Mapped (excluding outliers)

The clusters in the first (lower) quartile are striking. The plot is dominated by the cluster over the western seaboard of Scotland with an unmistakable focus around SC2466: Dunadd. There is a secondary concentration over Galloway and up into the Carsphairn and Lowther hills and a notable cluster toward the eastern end of the Great Glen. In Ireland there is a small cluster on the west coast and there is a small cluster in southern Wales, but it is sparce. Apart from the clusters the other notable feature of this distribution are the areas across England, north Wales and the Southwest where there are almost no forts of this class. Similarly, in Ireland the distribution is very much concentrated around the south and west coast with only a sparce peppering of hillforts inland.

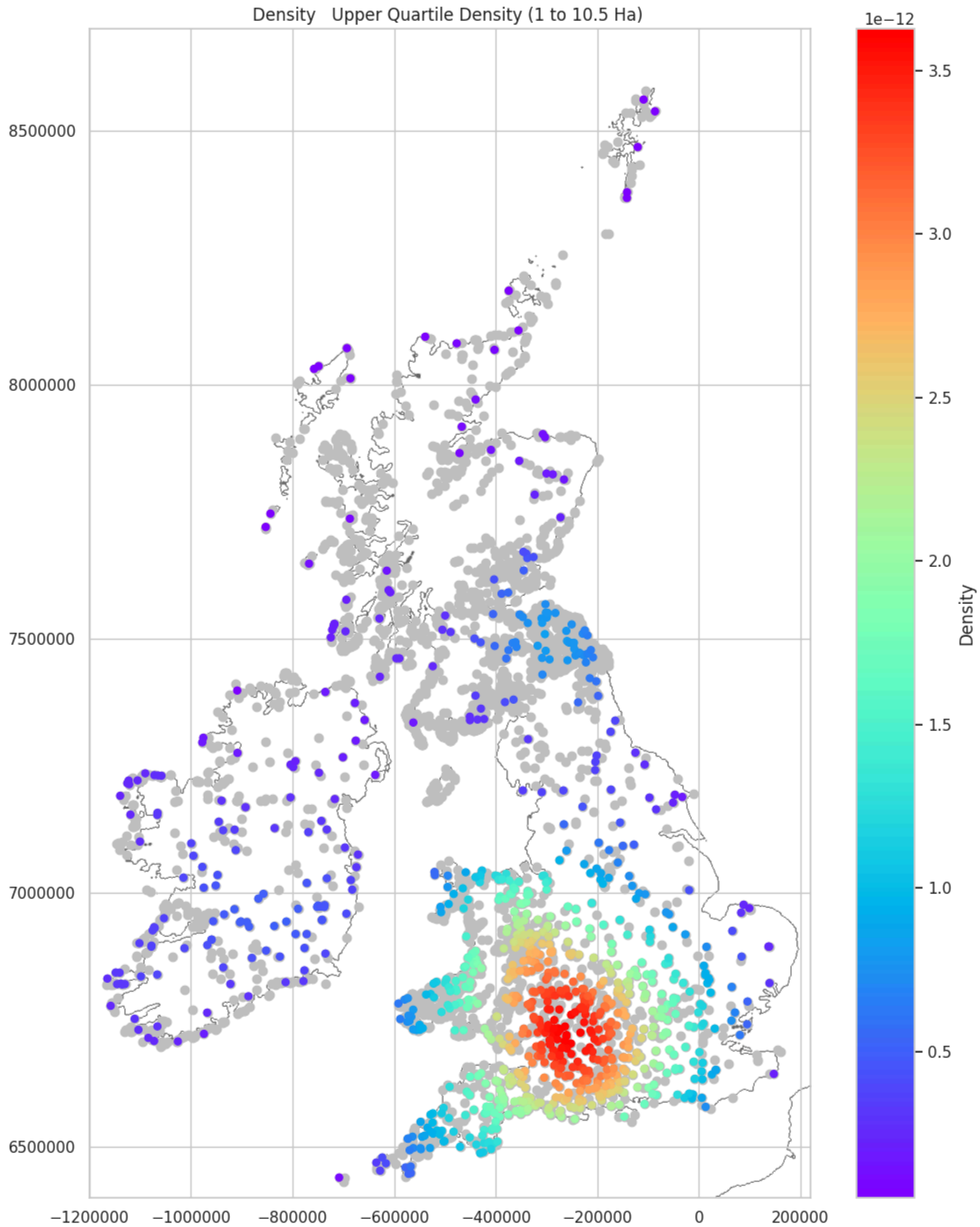
```
In [ ]: plot_density_over_grey(enclosing_area_1_low, \
                               'Lower Quartile Density (< 0.13 Ha)')
```



Enclosing Area 1 Upper Quartile Density Mapped (excluding outliers)

The fourth (upper) quartile is equally striking with a wide cluster focussed over south central England and running into Wales and the Southwest.

In [ ]:
```
plot_density_over_grey(enclosing_area_1_high, \
                       'Upper Quartile Density (1 to 10.5 Ha)')
```



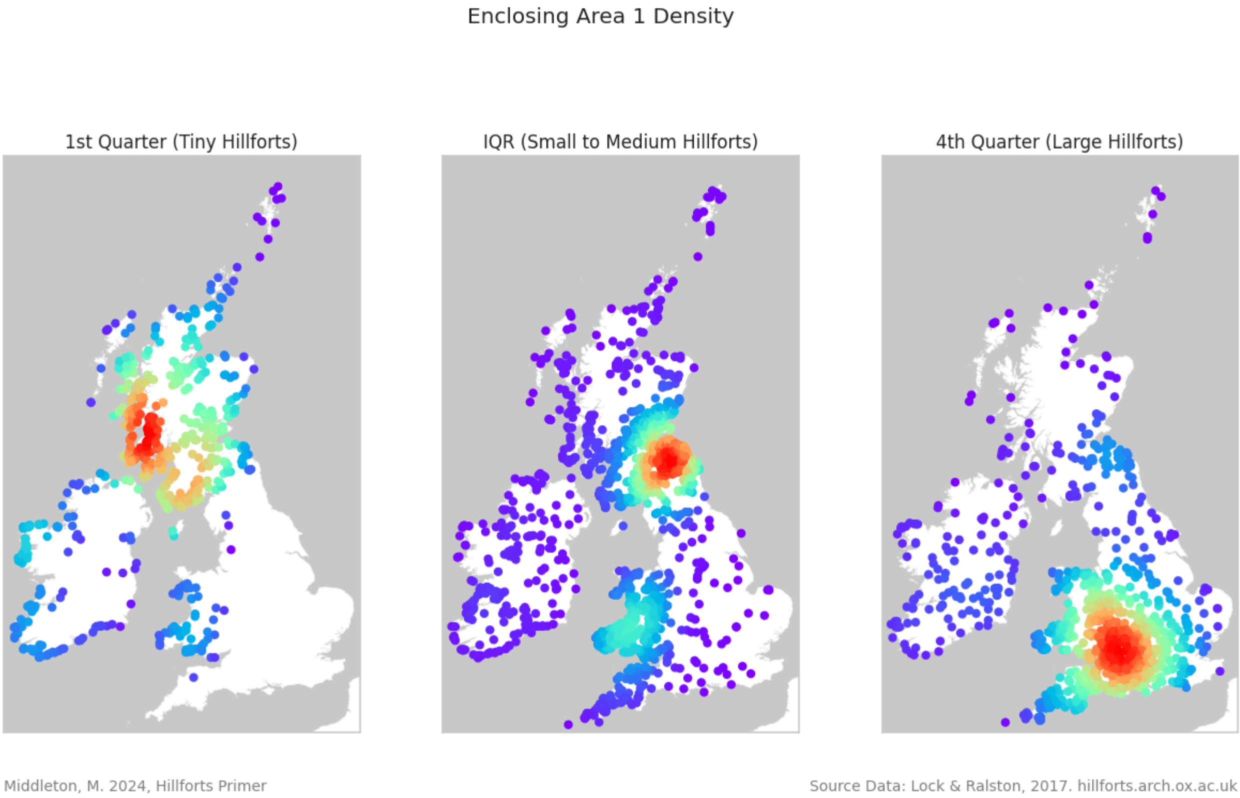
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area 1 Density Summary

The analysis of the Enclosing Area 1 data highlights four, possibly five different clusters. In the 1st quarter, mapping the density of tiny hillforts, there is one intense cluster in the Northwest and a smaller, almost indistinguishable cluster, in the west of Ireland, along the Duvillaun, Achill and Inishkea islands. In the central interquartile range (IQR), of small to medium sized hillforts, there are two more clusters. Here, the most intense cluster is in the Northeast and the smaller, secondary cluster, is in southern Wales. In the 4th quarter, mapping large hillforts, there is one large cluster over south central England. Equally notable are the areas where there are large gaps in the distribution. In the 1st quarter, England, north Wales and the Southwest have almost no recorded tiny hillforts while, less surprisingly, the Highlands and the west coast of Scotland have very few large hillforts.

In [ ]:
```
plot_density_over_grey_three(enclosing_area_1_low, \
                             enclosing_area_1_013_1, enclosing_area_1_high, \
                             'Enclosing Area 1 Density')
```



Enclosing Area 1 Outlier Distribution Mapped (Over 10.5 Ha)

There are 94 outliers that range in size from 10.5 to 130 Ha. Most are located in south central England and 16 in south, central Ireland; There is one in Galloway and one on the Isle of Man.

In [ ]:

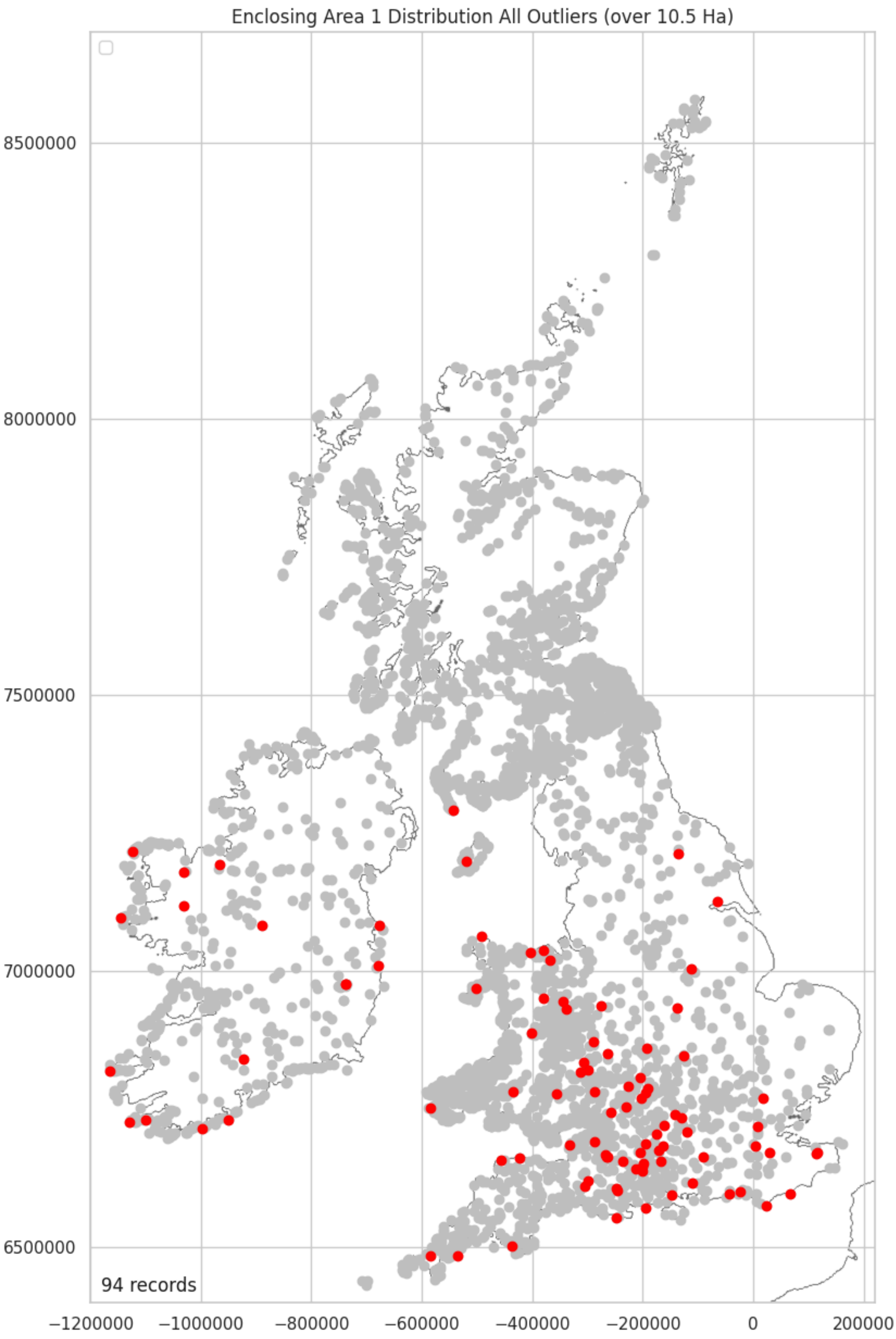
```
enclosing_area_1_105 = location_enclosing_data.copy()
enclosing_area_1_105 = \
enclosing_area_1_105[enclosing_area_1_105['Enclosing_Area_1']>=10.5]
enclosing_area_1_105['Enclosing_Area_1'].describe()
```

Out[ ]:

```
count     94.000000
mean      25.221489
std       22.195230
min       10.500000
25%       12.000000
50%       16.635000
75%       28.000000
max      130.000000
Name: Enclosing_Area_1, dtype: float64
```

In [ ]:

```
plot_over_grey_numeric(enclosing_area_1_105, 'Enclosing_Area_1', \
                       'Enclosing_Area_1 Distribution All Outliers (over 10.5 Ha)')
```



Middleton, M. 2024, Hillforts Primer

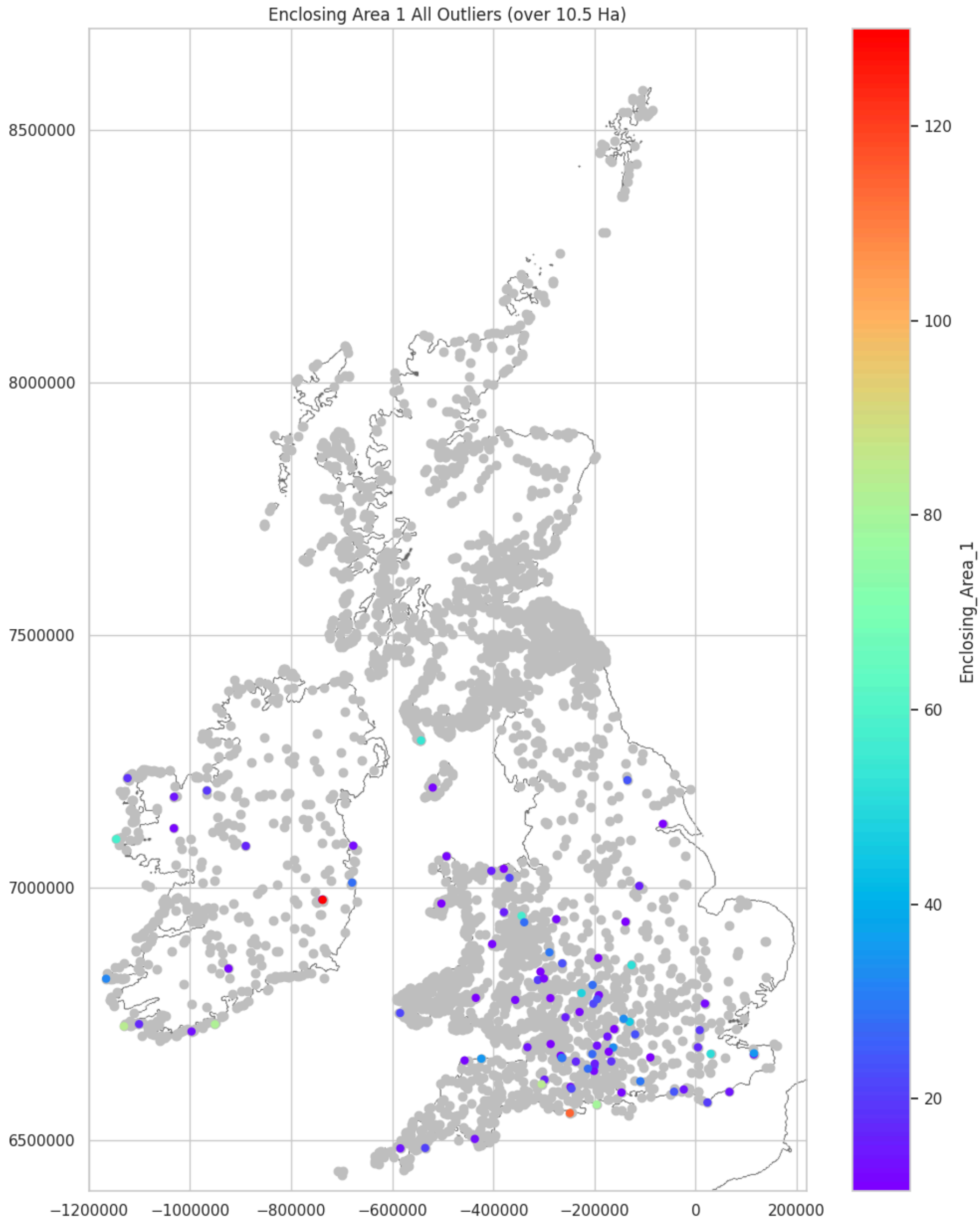
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
2.27%
```

Enclosing Area 1 Outliers Mapped by Size (Over 10.5 Ha)

Within the outliers over 10.5 Ha, there are two very large hillforts over 100 Ha. Otherwise, most are around 20 Ha or less. In the mid-range the plot highlights an alignment of forts, over 40 Ha running from the Thames up toward north Wales (light blue).

In [ ]:
```
plot_type_values(enclosing_area_1_105, 'Enclosing_Area_1', \
                 'Enclosing_Area_1', extra='All Outliers (over 10.5 Ha)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

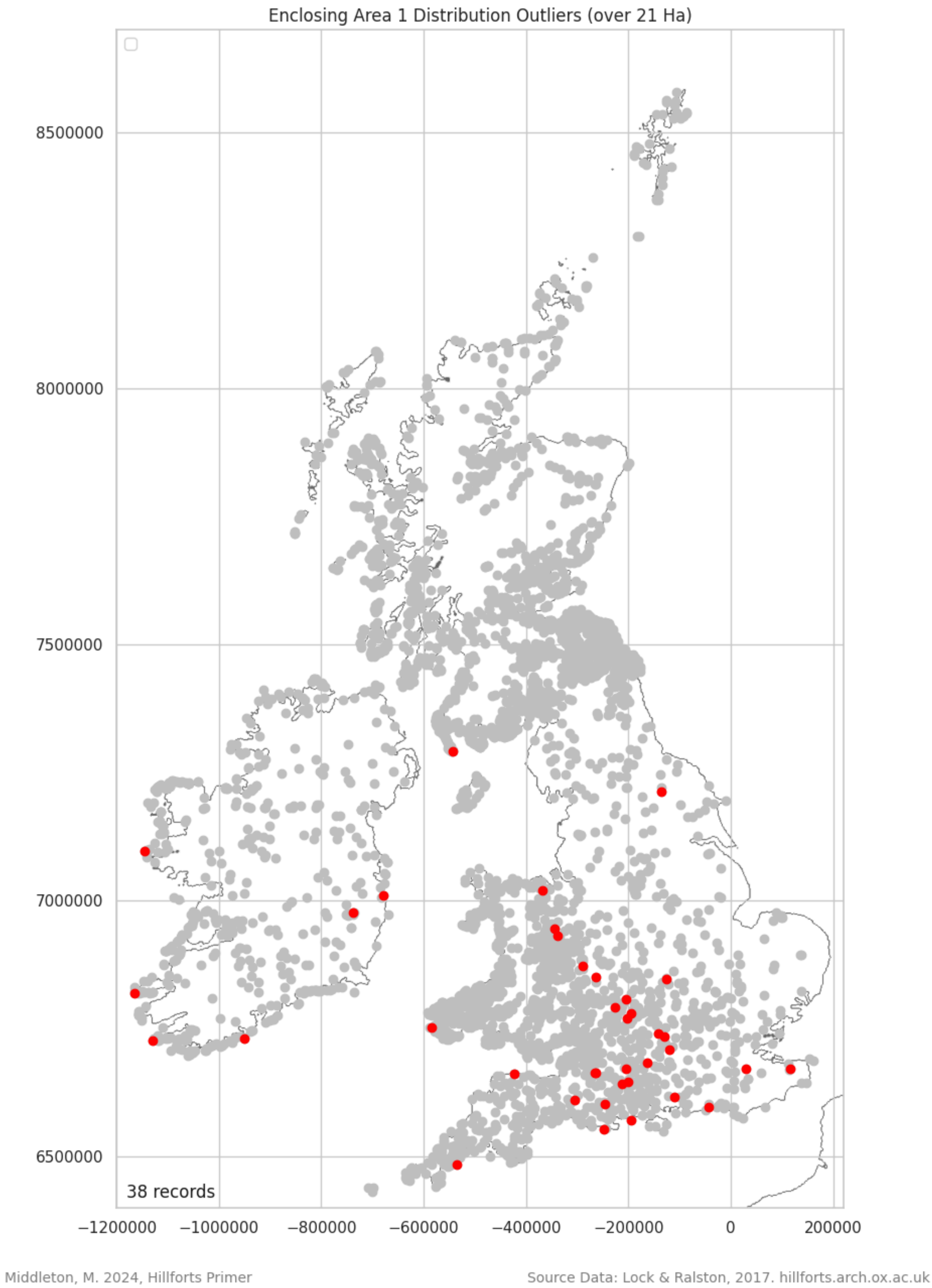
Enclosing Area 1: Distribution of Outliers Over 21 Ha Mapped

After multiple tests to filter the data for forts over various sizes, a possible alignment of hillforts was isolated for forts over 21 Ha. See Appendix 1 where the straight section of the alignment from (1155) Penycloddiau, Denbighshire (Pen y Cloddiau) to (139) Bozedown Camp, Oxfordshire (Binditch) is hypothesis tested and show this alignment as likely to be meaningful.

In [ ]:
```
enclosing_area_1_21 = location_enclosing_data.copy()
enclosing_area_1_21 = \
enclosing_area_1_21[enclosing_area_1_21['Enclosing_Area_1']>=21]
enclosing_area_1_21['Enclosing_Area_1'].describe()
```

Out[ ]:
```
count     38.000000
mean      42.501053
std       26.680691
min       21.000000
25%       24.875000
50%       30.000000
75%       51.875000
max      130.000000
Name: Enclosing_Area_1, dtype: float64
```

In [ ]:
```
plot_over_grey_numeric(enclosing_area_1_21, 'Enclosing_Area_1', \
                       'Enclosing_Area_1 Distribution Outliers (over 21 Ha)', '')
```




```
0.92%
```

Enclosing Area 1 Hillforts Over 21 Ha Mapped by Size

Clipping the maximum area at 80 Ha allows variation in the smaller sites to be visible. What can be seen is that most hillforts in the band from the Thames to north Wales are at the lower end of the area range. These are interspersed with forts in the mid-range (blue green). The largest forts are on or near the south coast or in Ireland.

In [ ]:
```
enclosing_area_1_21_clip = enclosing_area_1_21.copy()
enclosing_area_1_21_clip['Enclosing_Area_1_clip'] = \
enclosing_area_1_21_clip['Enclosing_Area_1'].\
```

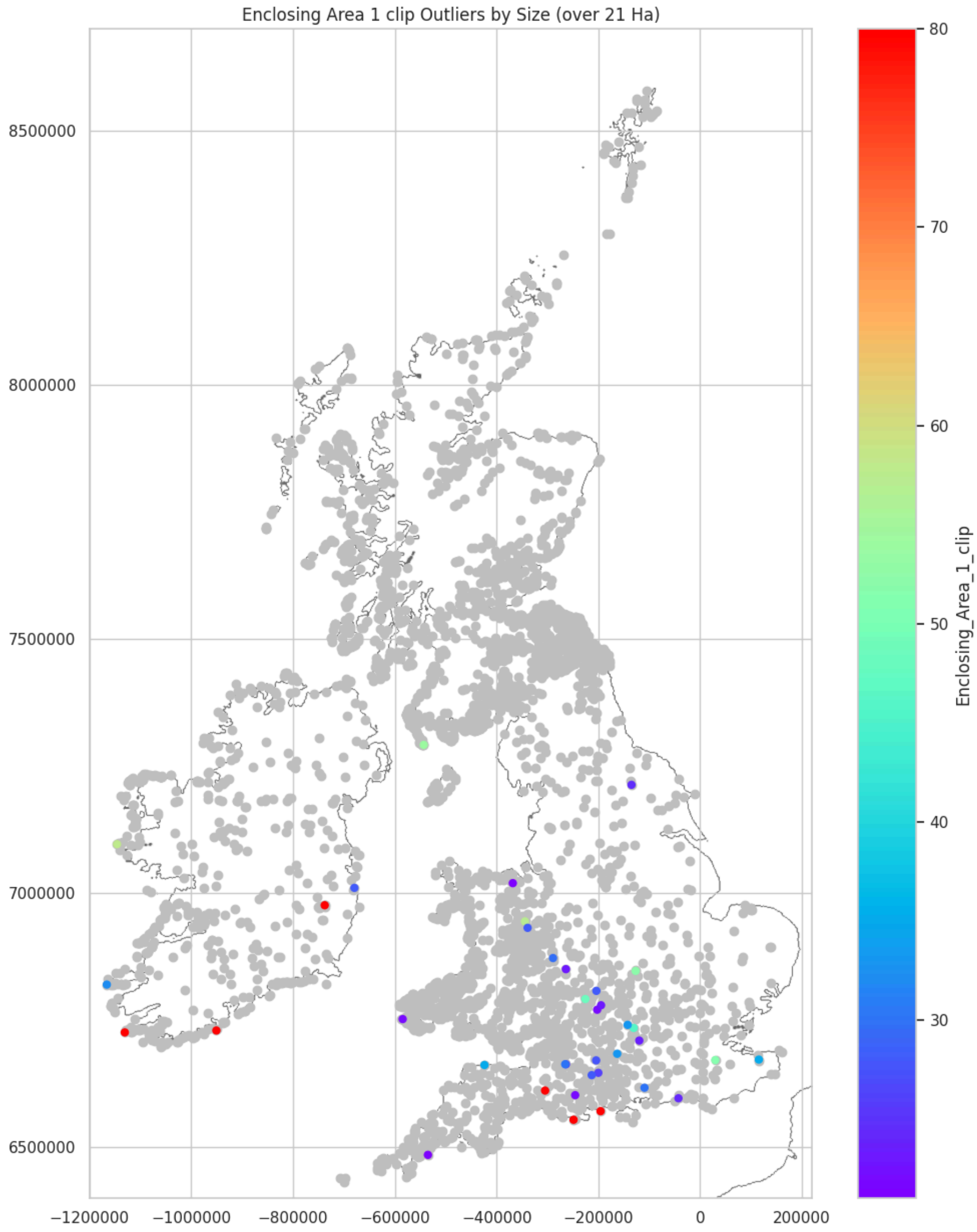
```
clip(enclosing_area_1_21_clip['Enclosing_Area_1'], 80, axis=0)
enclosing_area_1_21_clip['Enclosing_Area_1_clip'].describe()
```

Out[ ]:

```
count     38.000000
mean      40.061579
std       20.450771
min       21.000000
25%       24.875000
50%       30.000000
75%       51.875000
max       80.000000
Name: Enclosing_Area_1_clip, dtype: float64
```

In [ ]:

```
plot_type_values(enclosing_area_1_21_clip, 'Enclosing_Area_1_clip', \
                 'Enclosing_Area_1_clip', extra='Outliers by Size (over 21 Ha)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area 1: Hillfort Density Transformed Overlayed by Hillforts Over 21 Ha

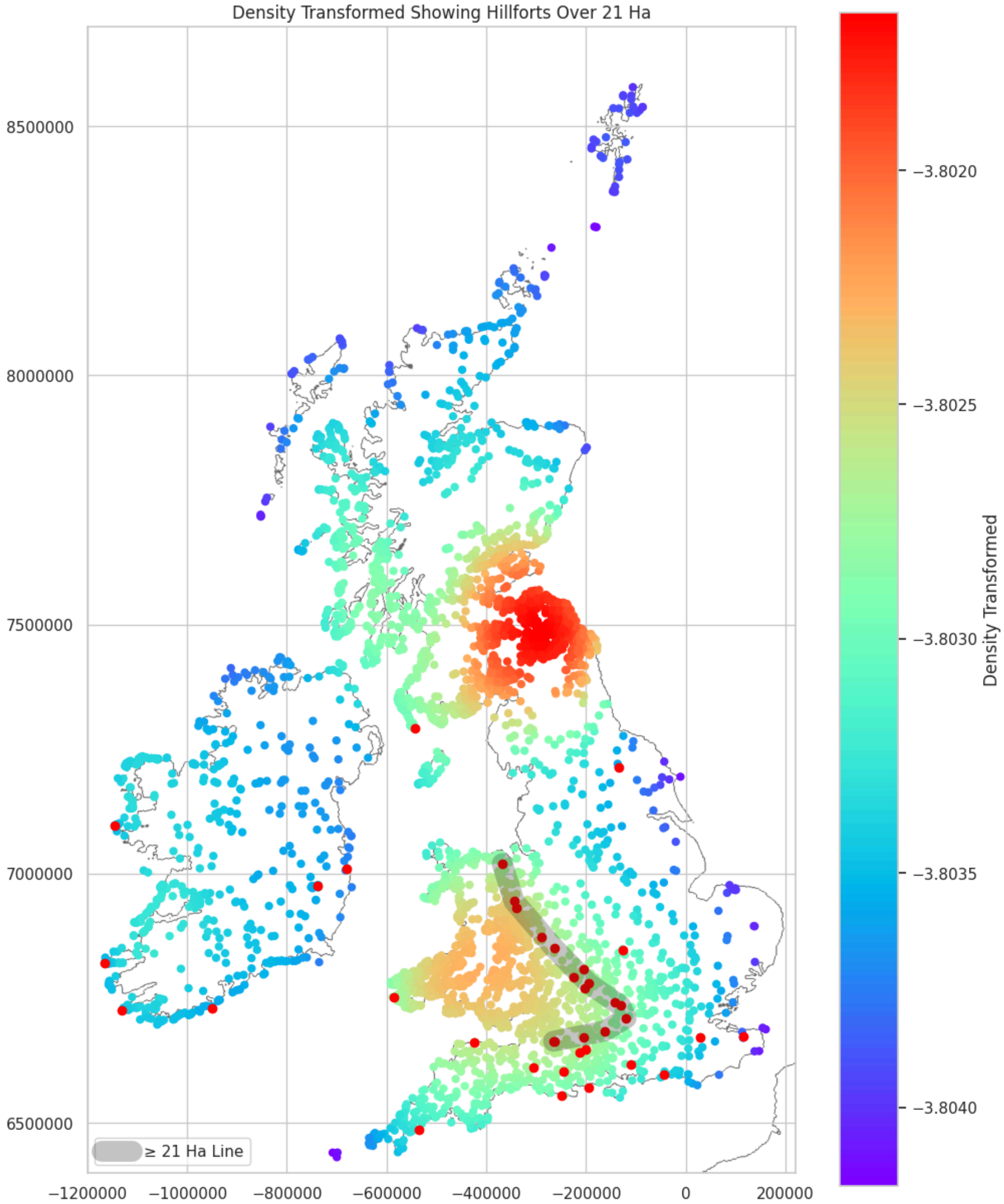
Plotting the hillforts over 21 Ha against the hillfort density shows the alignment of forts sit along the eastern fringe of the southern density cluster. The forts are located roughly along the transition from the orange to green on the density map (-3.8025). This line has been annotated the, '≥ 21 Ha Line'.

In [ ]:

```
transformed_location_numeric_data_short = location_data.copy()
transformed_location_numeric_data_short['Density_trans'], best_lambda = \
```

```
stats.boxcox(transformed_location_numeric_data_short['Density'])
```

In [ ]:
```
density_scatter_lines(transformed_location_numeric_data_short, \
                      enclosing_area_1_21, \
                      'Density Transformed Showing Hillforts Over 21 Ha', True)
```



Middleton, M. 2024, Hillforts Primer

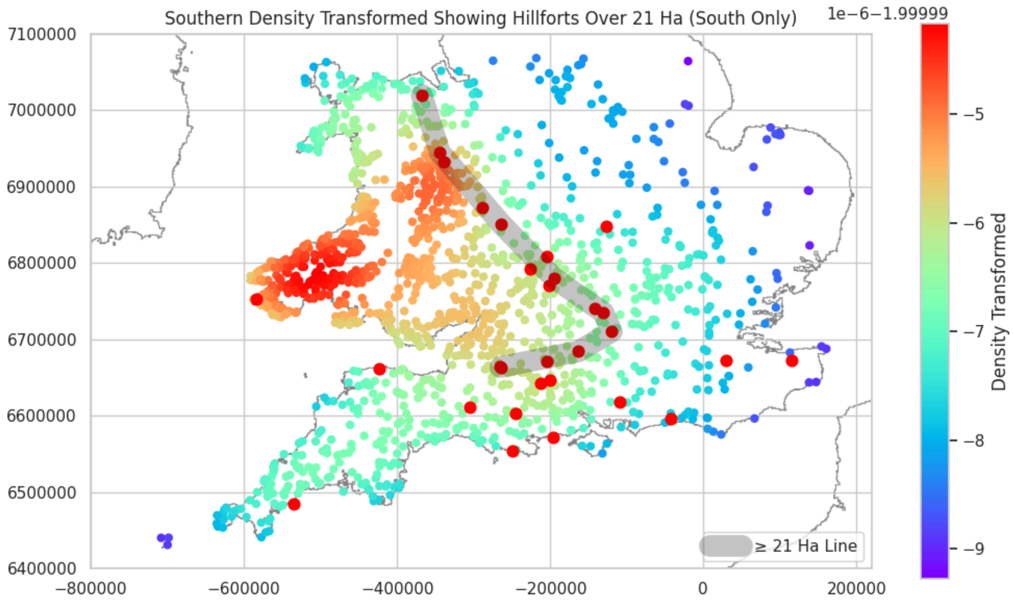
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area 1: Southern Hillfort Density Transformed overlayed by hillforts over 21 Ha

The same map showing only the southern data.

In [ ]:
```
cluster_south = south.copy()
enclosing_area_1_21_s = \
enclosing_area_1_21[enclosing_area_1_21['Location_X']>-600000]
cluster_south = add_density(cluster_south)
cluster_south['Density_trans'] = stats.boxcox(cluster_south['Density'], 0.5)
```

In [ ]:
```
south_density_scatter_lines(cluster_south, enclosing_area_1_21_s, \
                            'Southern Density Transformed Showing Hillforts Over 21 Ha (South Only)', \
                            True, False)
```



Middleton, M. 2024, Hillforts Primer

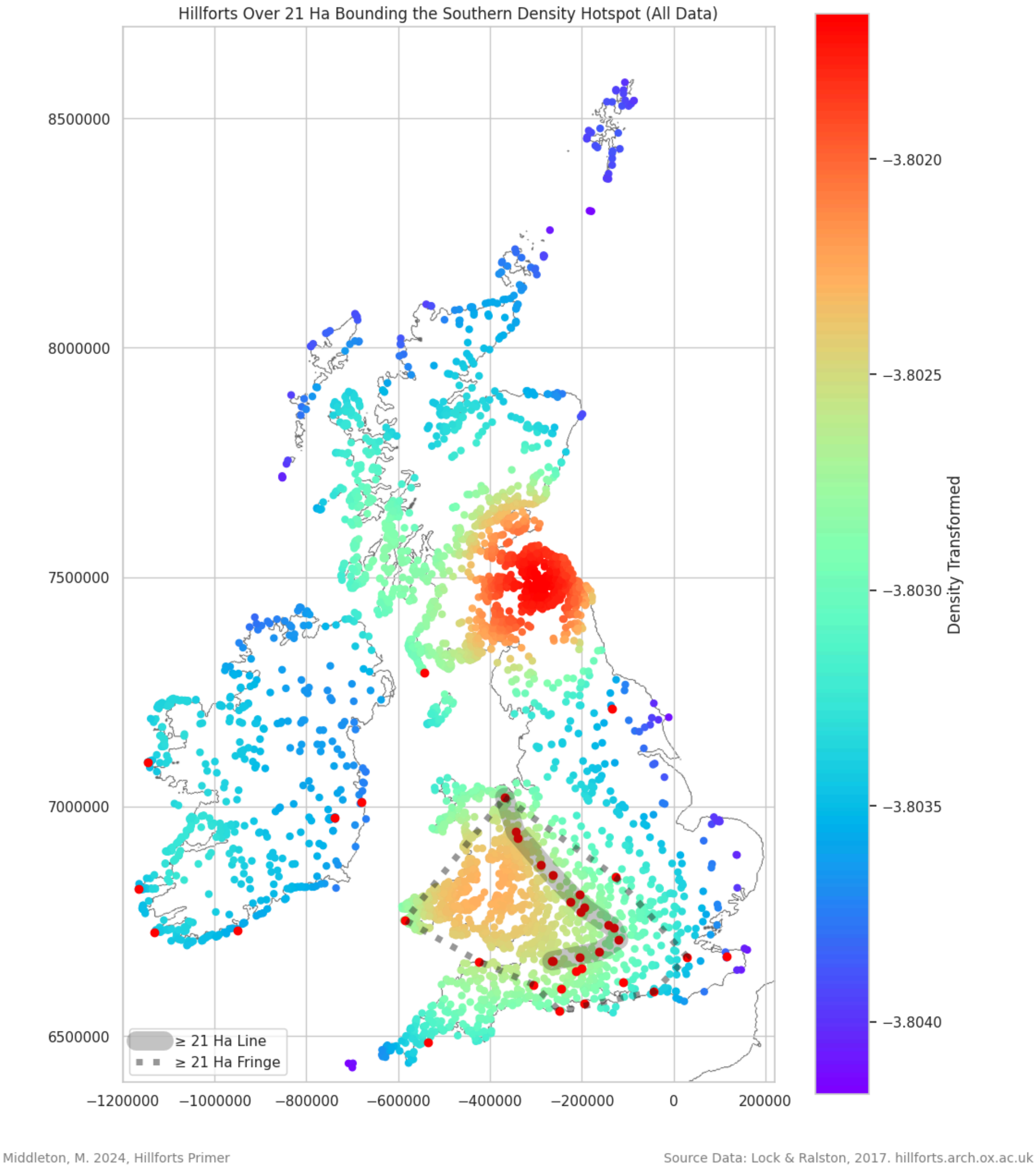
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area 1: Hillforts over 21 Ha Bounding the Southern Density Cluster

Most of the remaining outliers over 21 Ha in the South are located on the fringe of the southern density cluster. These have been annotated as the, '≥ 21 Ha Fringe'.

Both lines are based on a very small number of hillforts and are therefore highly speculative. There are only 38 hillforts greater than or equal to 21 Ha. This equates to 0.92% of all hillforts. These are not just the outliers (which are classified as lying in the outer 4.4% of a distribution), these are the outliers within the outliers. They are the most unusual hillforts in terms of Enclosing Area 1. For these hillforts to be distributed in such a uniform alignment is highly unlikely and can be shown not to be a random distribution in Appendix 1. For this class of forts to align with the edge of the most intense concentration of hillforts, seen in the southern density cluster, supports the idea that this alignment is not a coincidence. These hillforts seem to be positioned for a purpose. Could these be forts on a frontier between two cultural groups or perhaps these are forts focussed on trade, capable of hosting large gatherings of people and animals? It is hoped these observations will encourage a more detailed analysis.

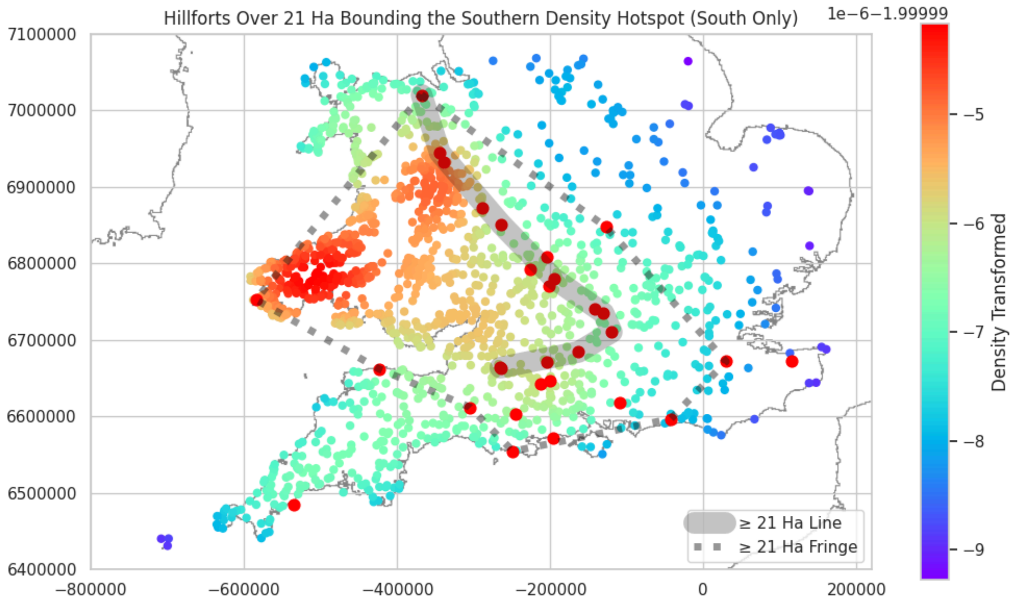
In [ ]:
```
density_scatter_lines(transformed_location_numeric_data_short, \
                      enclosing_area_1_21, \
                      'Hillforts Over 21 Ha Bounding the Southern Density Hotspot (All Data)',\
                      True, True)
```



Enclosing Area 1: Hillforts over 21 Ha Bounding the Southern Density Cluster (South Only)

The same plot showing only the southern data.

In [ ]:
```
south_density_scatter_lines(cluster_south, enclosing_area_1_21_s, \
                            'Hillforts Over 21 Ha Bounding the Southern Density Hotspot (South Only)', \
                            True, True)
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

A full list of hillforts, of 21 hectares and over, in the southern data package.

In [ ]:
```
greater_than_21ha_south = pd.merge(name_and_number, enclosing_area_1_21_s, \
                                   left_index=True, right_index=True)
greater_than_21ha_south\
 [['Main_Atlas_Number', 'Main_Display_Name', 'Enclosing_Area_1', \
   'Location_X', 'Location_Y']].sort_values(by='Enclosing_Area_1').\
   style.hide_index()
```

```
<ipython-input-248-398796f1c553>:6: FutureWarning: this method is deprecated in favour of `Styler.hide(axis="index")
`
  style.hide_index()
```

Out[ ]:

Main_Atlas_Number	Main_Display_Name	Enclosing_Area_1	Location_X	Location_Y
1155	Penycloddiau, Denbighshire (Pen y Cloddiau)	21.000000	-367969	7019842
643	Dodman Castle, Cornwall (Dodman Point; The Dodman)	21.000000	-534770	6485285
753	Norbury Camp, Northleach, Gloucestershire	22.000000	-202278	6770873
1997	Wooltack Point, Pembrokeshire (Deer Park Fort)	22.000000	-584307	6752378
3595	Hod Hill, Dorset	22.000000	-245476	6602754
757	Salmonsbury Camp, Gloucestershire	23.000000	-194654	6779602
367	Woodbury, Great Witley, Worcestershire (Woodbury Hill)	23.000000	-263690	6850593
139	Bozedown Camp, Oxfordshire (Binditch)	23.500000	-119597	6710127
3749	Cissbury Ring, West Sussex (Cissbury Camp)	24.000000	-42657	6596650
1504	Roulston Scar, North Yorkshire (Sutton Bank; Casten Dike)	24.500000	-134884	7213231
404	Ogbury Camp, Wiltshire	26.000000	-200007	6646748
461	Tedbury Camp, Somerset	26.000000	-263616	6663457
389	Casterley Camp, Wiltshire (Catterley Banks)	27.500000	-204306	6671153
756	Willersey Camp, Gloucestershire (Willersey Hill Camp)	28.000000	-203808	6807893
427	Ebsbury Hill, Wiltshire (Grovely Earthworks)	28.000000	-212997	6642126
1276	y Breiddin, Powys (Breddin Hillfort; The Breiddin Hillfort; Breiddin Hill Camp)	28.000000	-338884	6931868
91	Titterstone Clee, Shropshire	29.600000	-289034	6872454
3795	Butser Hill, Hampshire	30.000000	-109375	6617356
464	Wadbury Camp, Somerset (Wadbury Hillfort)	30.000000	-265052	6663609
173	Abingdon, The Vineyard, Oxfordshire	33.000000	-142388	6740992
97	Walbury Camp, West Berkshire	33.000000	-162994	6684152
3459	Countisbury Castle, Devon (Shoulsbury; Wind Hill)	35.000000	-423647	6662041
3823	Homestall Wood, Kent	35.000000	115292	6672762
172	Dyke Hills, Oxfordshire (Dike Hills)	46.000000	-130542	6735058
760	Nottingham Hill Camp, Gloucestershire	48.600000	-225551	6791821
3774	Oldbury Camp, Kent	51.500000	29679	6671658
773	Borough Hill 1, Northamptonshire (Borough Hill)	52.000000	-126771	6847138
201	Mull of Galloway, Dumfries & Galloway	54.000000	-542988	7291829
71	Llanymynech Hill, Powys	57.000000	-344171	6944572
3594	Hengistbury Head, Dorset	80.000000	-195572	6571275
448	Ham Hill, Somerset (Hamdon Hill Camp)	84.000000	-304545	6611780
3582	Bindon Hill, Dorset	114.000000	-248650	6554366

Enclosing Area 1: Southern Hillforts Over 21 Ha Dates

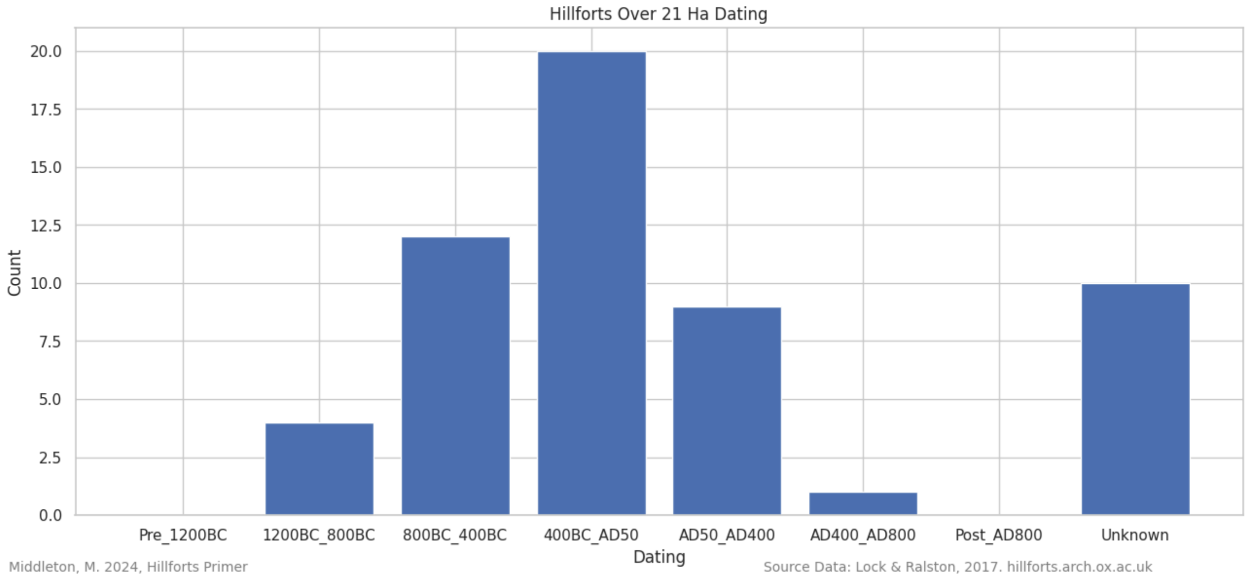
Most of the hillforts over 21 Ha, in the southern data, have dating evidence and the plot is consistent to that seen in Part 3: Date Data Plotted (Excluding No Dates) and Part3: Dating by Region, where the forts have dates from the late Bronze Age through to the Early Medieval with the highest peak being in the late Iron Age.

In [ ]:

```
greater_than_21ha_south_dates = \
pd.merge(greater_than_21ha_south, date_data, left_index=True, right_index=True)
```

In [ ]:

```
plot_bar_chart(greater_than_21ha_south_dates[date_features], 2, \
               'Dating', 'Count', 'Hillforts Over 21 Ha Dating')
```



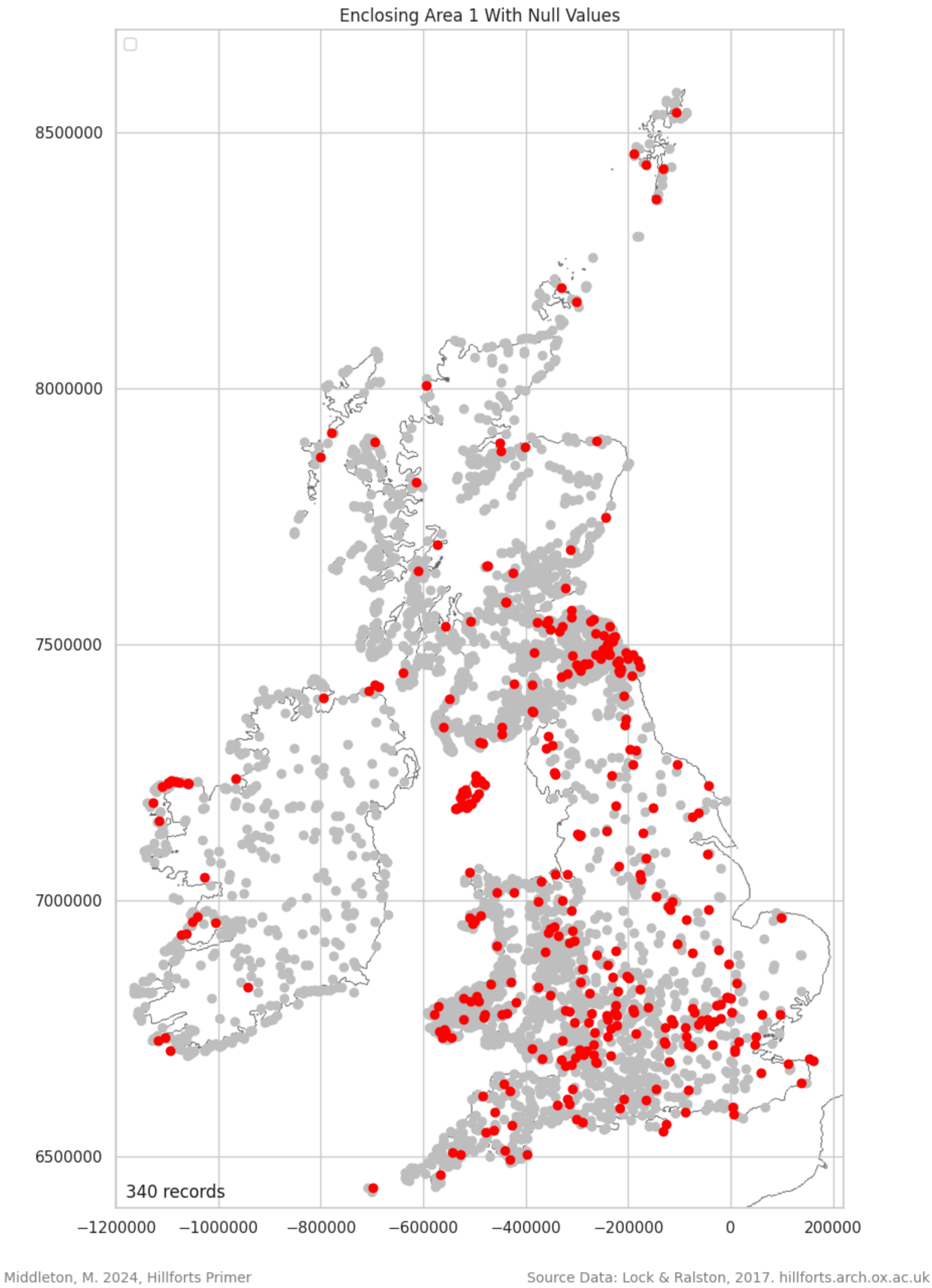
Enclosing Area 1 Null Values Mapped

There are 340 records where no 'Enclosing_Area_1' area is recorded.

In [ ]:
```
enclosing_area_1_minus1 = location_enclosing_data.copy()
enclosing_area_1_minus1 = \
enclosing_area_1_minus1[enclosing_area_1_minus1['Enclosing_Area_1']<0]
enclosing_area_1_minus1['Enclosing_Area_1'].describe()
```

Out[ ]:
```
count    340.0
mean      -1.0
std        0.0
min       -1.0
25%       -1.0
50%       -1.0
75%       -1.0
max       -1.0
Name: Enclosing_Area_1, dtype: float64
```

In [ ]:
```
plot_over_grey_numeric(enclosing_area_1_minus1, 'Enclosing_Area_1', \
                       'Enclosing_Area_1 With Null Values')
```




```
8.2%
```

Enclosing Area 2, 3 & 4

There are only 335 hillforts with an Enclosing_Area_2, 68 with an Enclosing_Area_3 and 11 with an Enclosing_Area_4. These additional area features have been used to capture the increased areas of hillforts when including, "outer enclosing works". (Data Structure)

In [ ]:
```
hillforts_data[['Enclosing_Area_2',
 'Enclosing_Area_3',
 'Enclosing_Area_4']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Enclosing_Area_2  335 non-null    float64
 1   Enclosing_Area_3  68 non-null     float64
 2   Enclosing_Area_4  11 non-null     float64
dtypes: float64(3)
memory usage: 97.3 KB
```

In [ ]:
```
enclosing_area_2_short = \
location_enclosing_data[location_enclosing_data['Enclosing_Area_2']>=0]
enclosing_area_3_short = \
location_enclosing_data[location_enclosing_data['Enclosing_Area_3']>=0]
enclosing_area_4_short = \
location_enclosing_data[location_enclosing_data['Enclosing_Area_4']>=0]
```

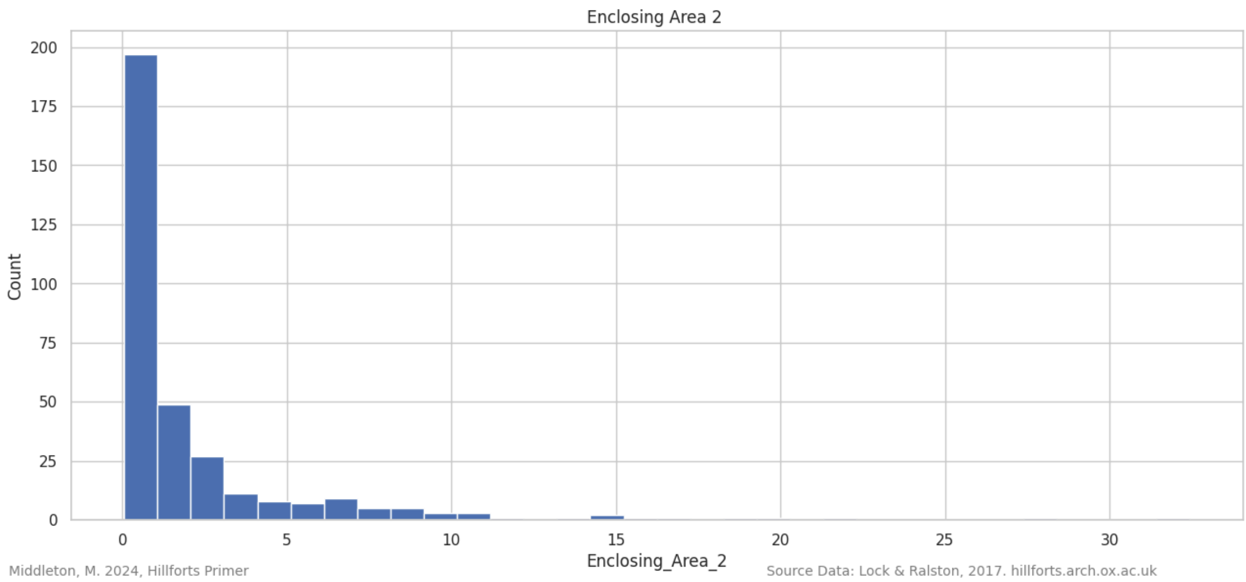
Enclosing Area 2 Plotted

Like Enclosing_Area_1, the area of most hillforts, with an Enclosing_Area_2, are small. The spread of 95.6% of the data is quite wide, running from 0.12 to 14.46 Ha but, the interquartile range (the middle 50% of the data) only ranges from 0.4 to 2.21 Ha.

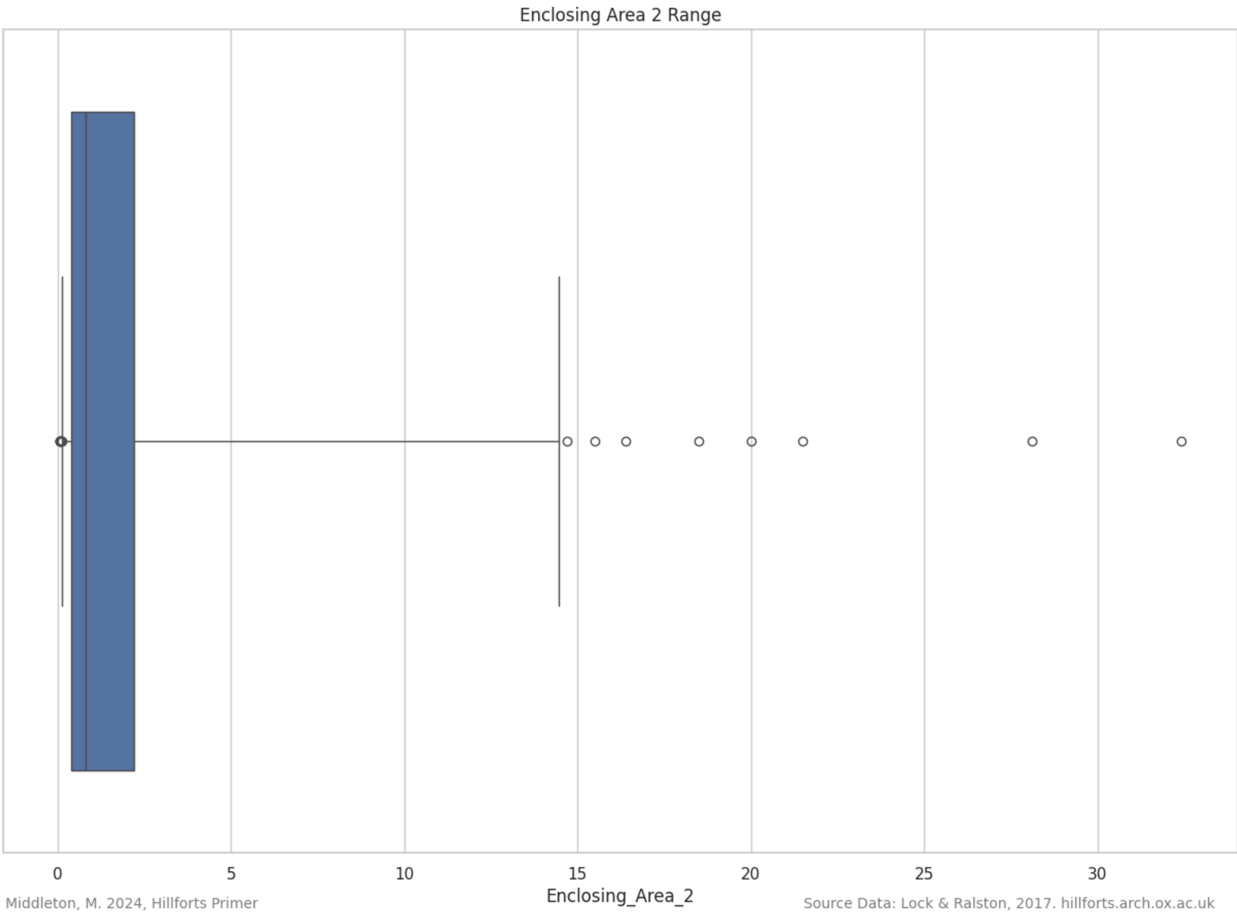
In [ ]:
```
enclosing_area_2_short['Enclosing_Area_2'].describe()
```

Out[ ]:
```
count    335.000000
mean       2.277761
std        3.924060
min        0.050000
25%        0.400000
50%        0.800000
75%        2.210000
max       32.430000
Name: Enclosing_Area_2, dtype: float64
```

In [ ]:
```
plot_bar_chart_numeric(enclosing_area_2_short, 1, 'Enclosing_Area_2', \
                       'Count', 'Enclosing_Area_2', \
                       int(enclosing_numeric_data['Enclosing_Area_2'].max()))
```



In [ ]:
```
enclosing_area_2_data = \
plot_data_range(enclosing_area_2_short['Enclosing_Area_2'].\
                reset_index(drop = True), 'Enclosing_Area_2', "h")
```



In [ ]:
```
enclosing_area_2_data
```

Out[ ]:
```
[0.12, 0.4, 0.8, 2.21, 14.46]
```

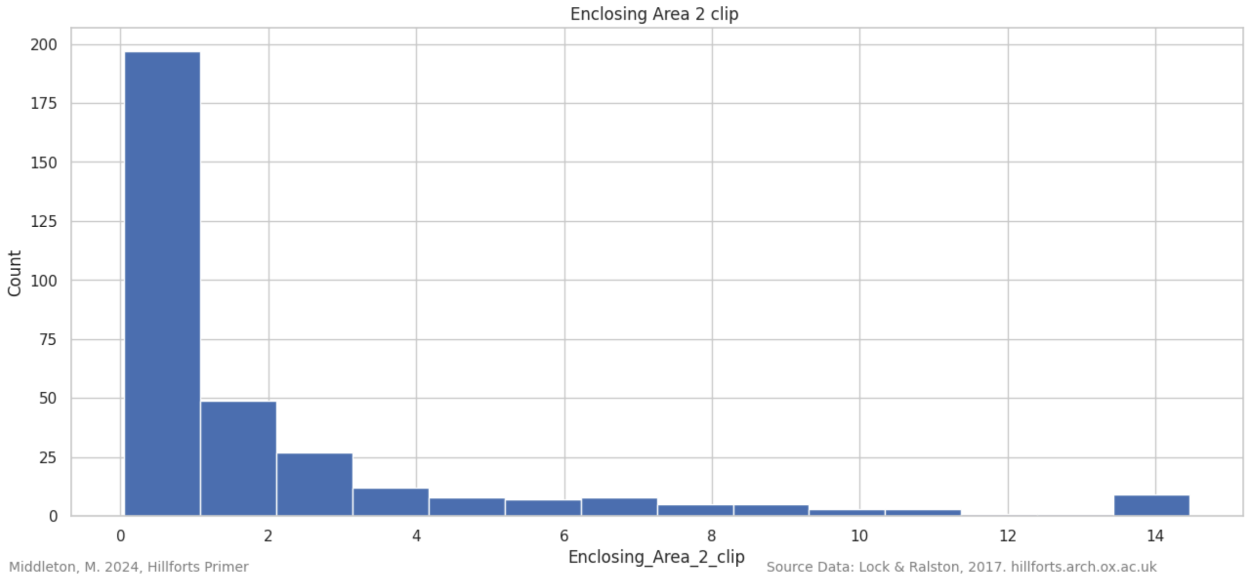
Enclosing Area 2 Clipped Plotted

To help visualise the data, outliers have been clipped. All values beyond 14.46 HA have been pooled into this value.

In [ ]:
```
enclosing_area_2_data_clip = enclosing_area_2_short.copy()
enclosing_area_2_data_clip['Enclosing_Area_2_clip'] = \
enclosing_area_2_data_clip['Enclosing_Area_2'].\
clip(enclosing_area_2_data_clip['Enclosing_Area_2'], \
     enclosing_area_2_data[-1], axis=0)
enclosing_area_2_data_clip['Enclosing_Area_2_clip'].describe()
```

Out[ ]:
```
count    335.000000
mean       2.124119
std        3.128999
min        0.050000
25%        0.400000
50%        0.800000
75%        2.210000
max       14.460000
Name: Enclosing_Area_2_clip, dtype: float64
```

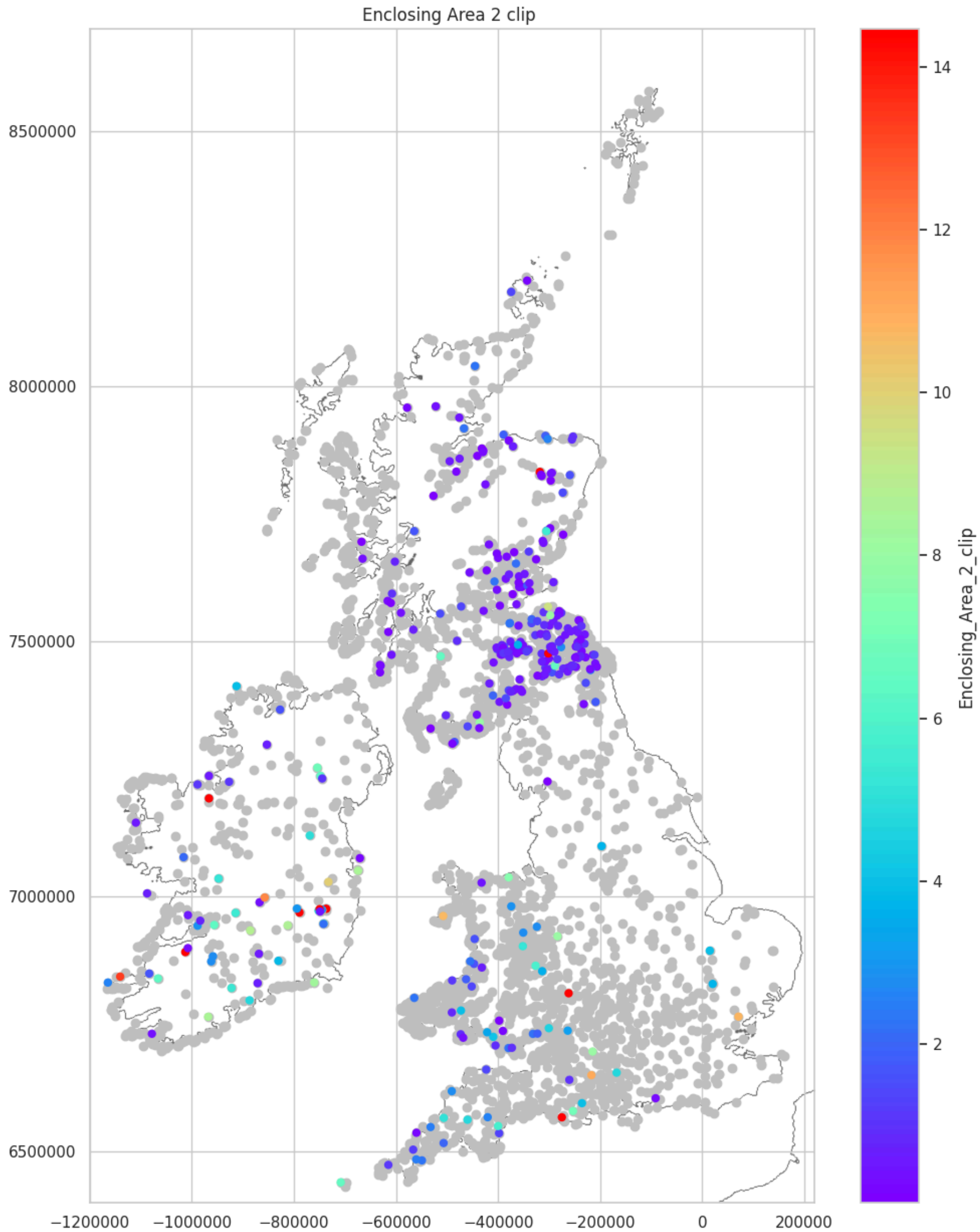
In [ ]:
```
plot_bar_chart_numeric(enclosing_area_2_data_clip, 1, \
                       'Enclosing_Area_2_clip', 'Count', \
                       'Enclosing_Area_2_clip', \
                       int(enclosing_area_2_data_clip['Enclosing_Area_2_clip'].max()))
```



Enclosing Area 2 Clipped Mapped

The distribution of this data suggests there is a survey bias and that many hillforts with outer works have not had an Enclosing_Area_2 recorded. Of those that have, most are in the Northeast.

In [ ]:
```
plot_type_values(enclosing_area_2_data_clip, 'Enclosing_Area_2_clip', \
                 'Enclosing_Area_2_clip')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

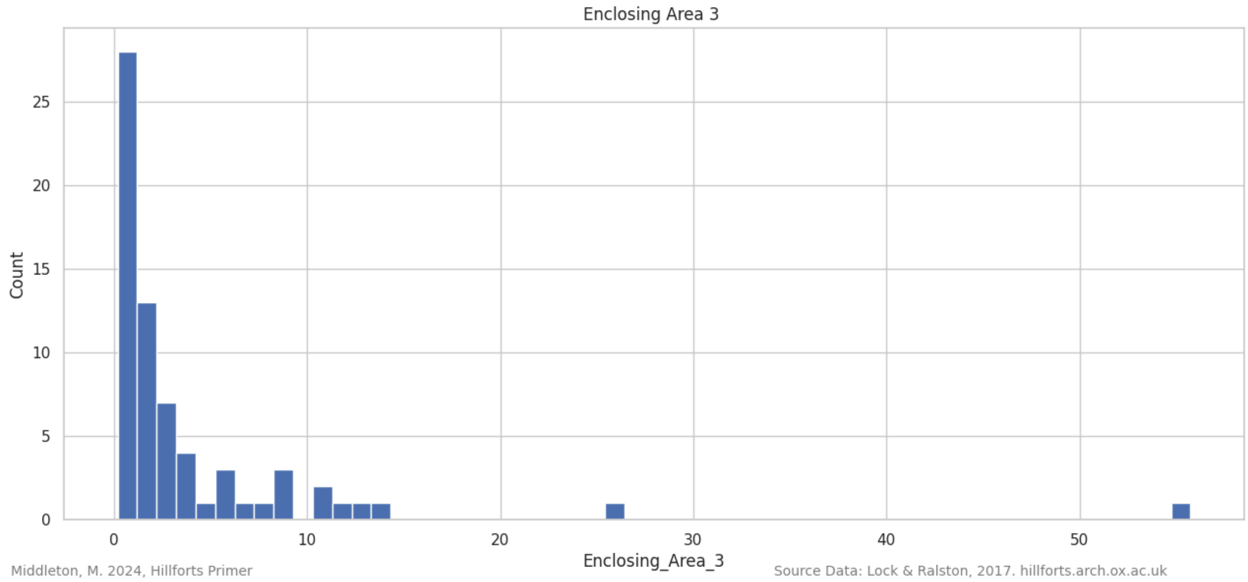
Enclosing Area 3 Plotted

Only 68 hillforts have an Enclosing_Area_3. They follow the same pattern as seen in Enclosing_Area_2, with most being quite small and most located in the North. As with Enclosing_Area_2, it is likely that this data contains a survey bias.

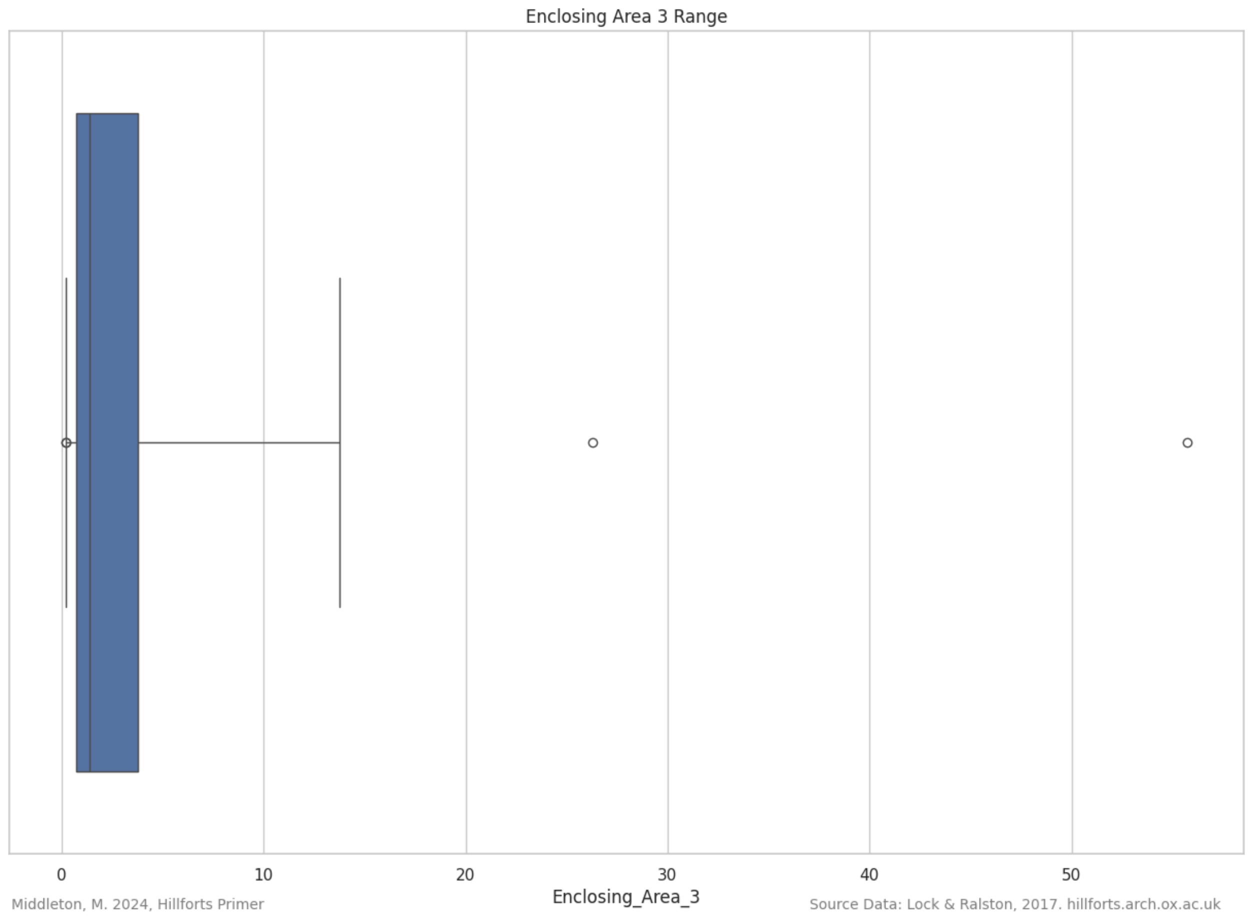
In [ ]:
```
enclosing_area_3_short['Enclosing_Area_3'].describe()
```

Out[ ]:
```
count    68.000000
mean      4.037647
std       7.742350
min       0.190000
25%       0.747500
50%       1.400000
75%       3.775000
max      55.740000
Name: Enclosing_Area_3, dtype: float64
```

In [ ]:
```
plot_bar_chart_numeric(enclosing_area_3_short, 1, 'Enclosing_Area_3', \
                       'Count', 'Enclosing_Area_3', \
                       int(enclosing_numeric_data['Enclosing_Area_3'].max()))
```



In [ ]:
```
enclosing_area_3_data = \
plot_data_range(enclosing_area_3_short['Enclosing_Area_3'].\
                reset_index(drop = True), 'Enclosing_Area_3', "h")
```



In [ ]:
```
enclosing_area_3_data
```

Out[ ]:
```
[0.23, 0.7475, 1.4, 3.7750000000000004, 13.75]
```

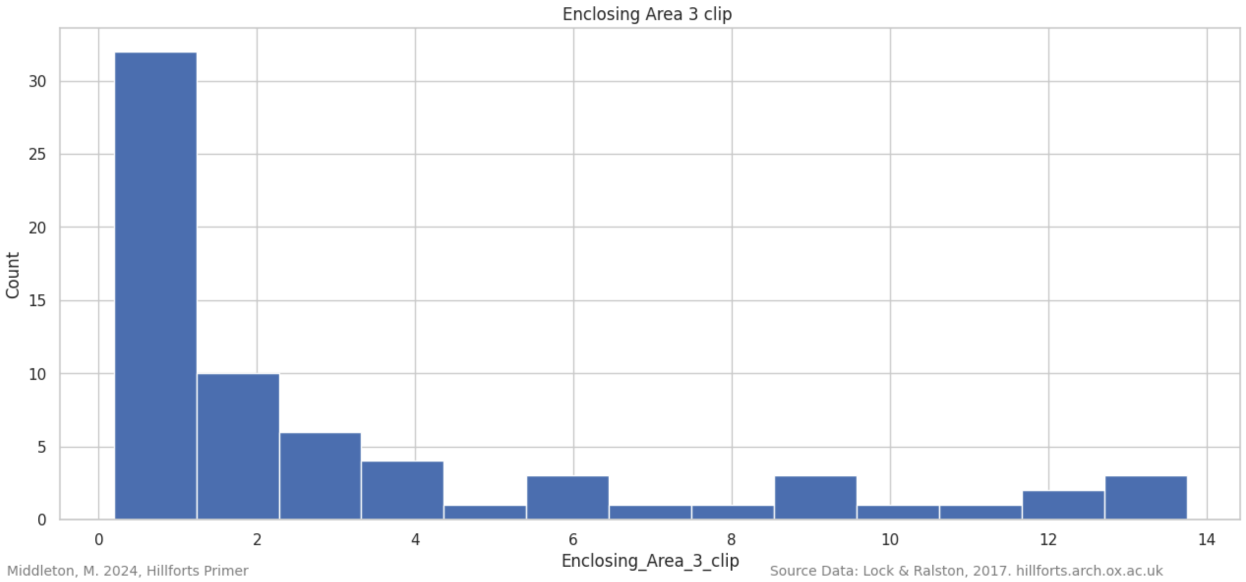
Enclosing Area 3 Clipped Plotted

To help visualise the data, outliers have been clipped. All values beyond 13.75 HA have been pooled into this value.

In [ ]:
```
enclosing_area_3_data_clip = enclosing_area_3_short.copy()
enclosing_area_3_data_clip['Enclosing_Area_3_clip'] = \
enclosing_area_3_data_clip['Enclosing_Area_3'].\
clip(enclosing_area_3_data_clip['Enclosing_Area_3'], \
     enclosing_area_3_data[-1], axis=0)
enclosing_area_3_data_clip['Enclosing_Area_3_clip'].describe()
```

Out[ ]:
```
count    68.000000
mean      3.236176
std       3.852233
min       0.190000
25%       0.747500
50%       1.400000
75%       3.775000
max      13.750000
Name: Enclosing_Area_3_clip, dtype: float64
```

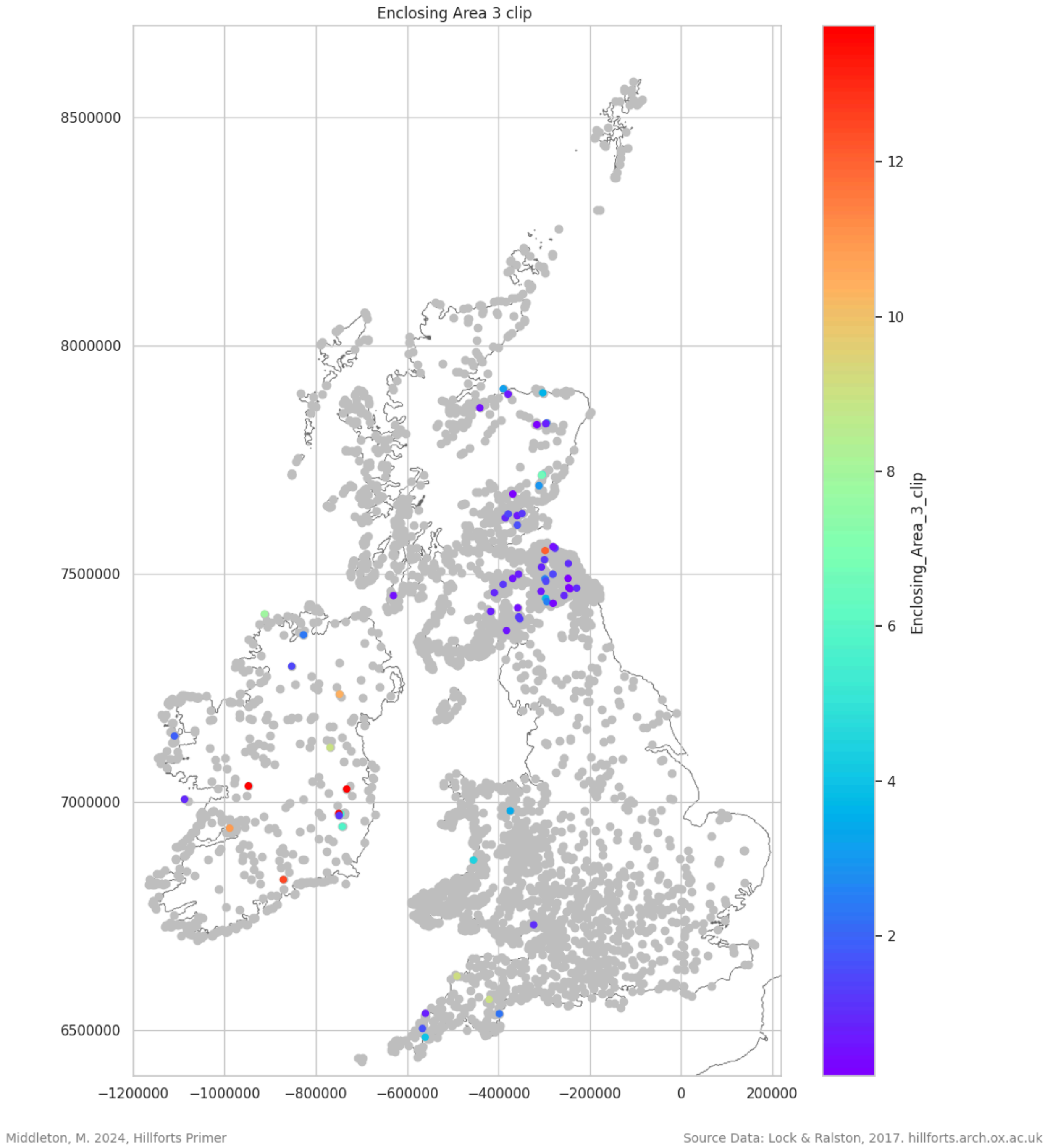
In [ ]:
```
plot_bar_chart_numeric(enclosing_area_3_data_clip, 1, \
                       'Enclosing_Area_3_clip', 'Count', \
                       'Enclosing_Area_3_clip', \
                       int(enclosing_area_3_data_clip['Enclosing_Area_3_clip'].max()))
```



Enclosing Area 3 Clipped Mapped

The forts in this class are mostly located in the Northeast. This, and the low number of records in this class, suggests that this data has a survey bias toward this area.

In [ ]:
```
plot_type_values(enclosing_area_3_data_clip, 'Enclosing_Area_3_clip', \
                 'Enclosing_Area_3_clip')
```



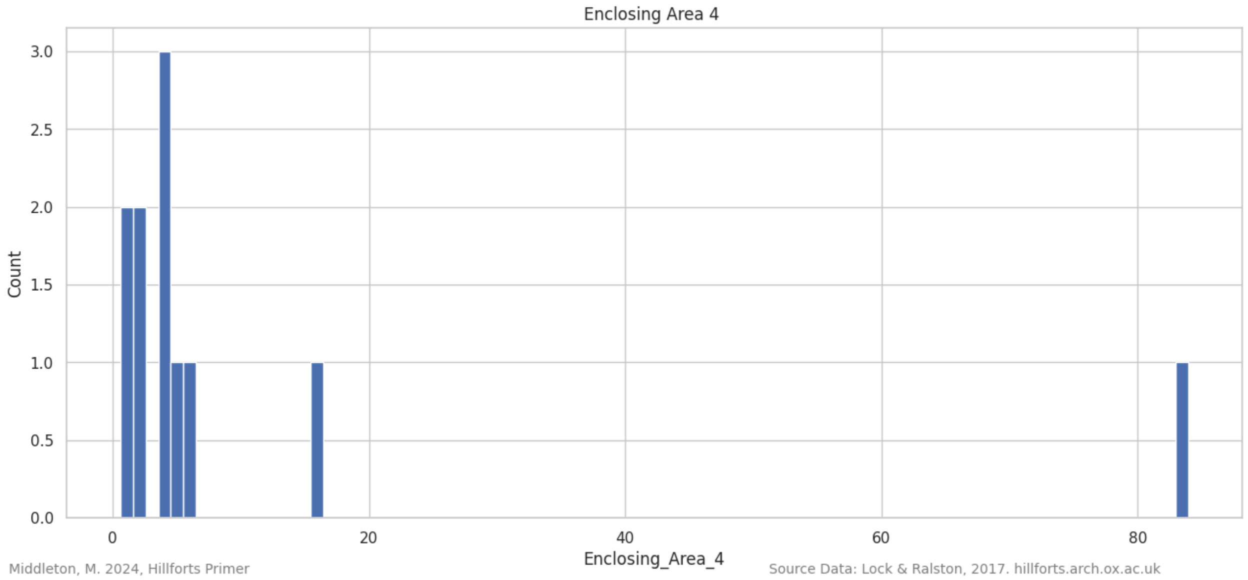
Enclosing Area 4 Plotted

Only 11 hillforts have a fourth outer work recorded.

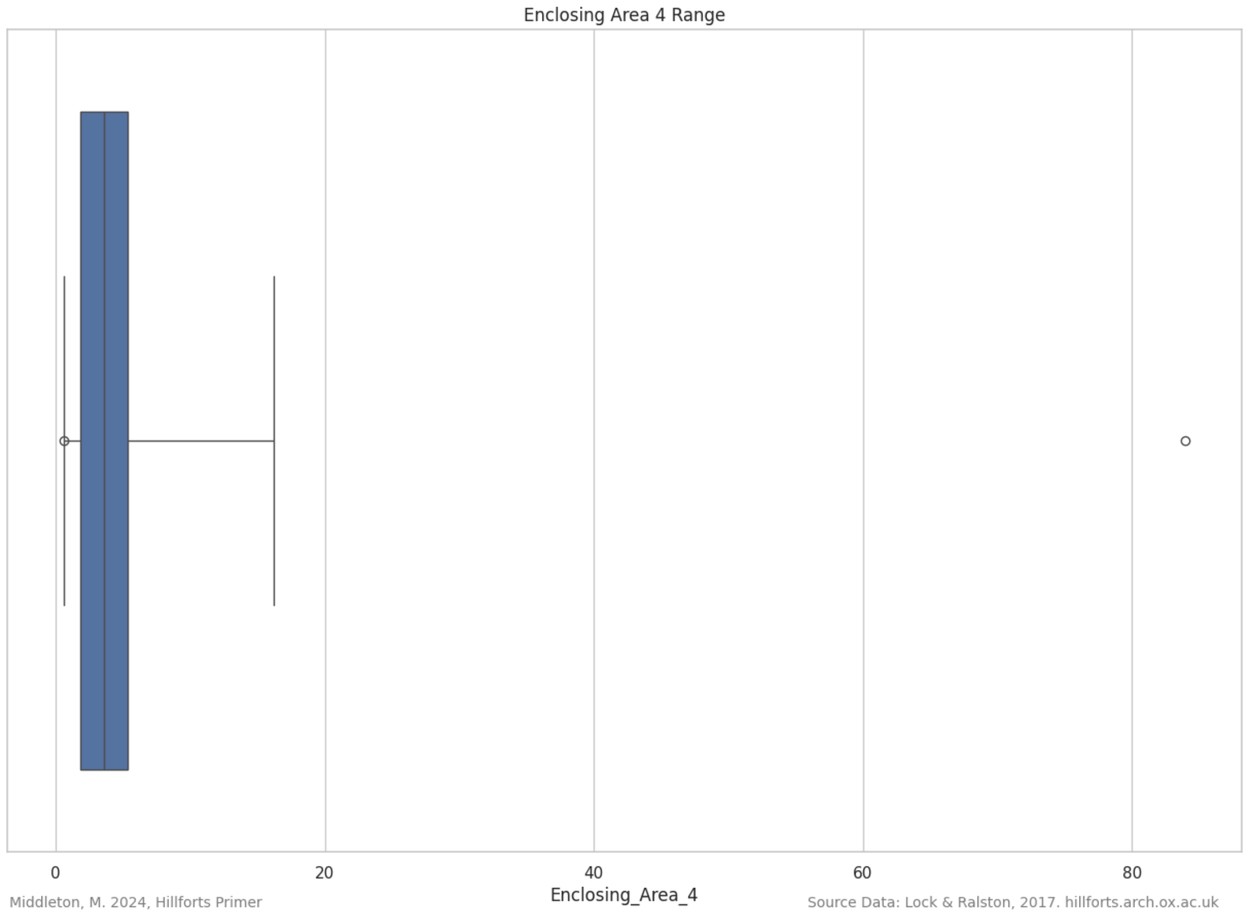
In [ ]:
```
enclosing_area_4_short['Enclosing_Area_4'].describe()
```

Out[ ]:
```
count    11.000000
mean     11.568182
std      24.404416
min       0.560000
25%       1.830000
50%       3.600000
75%       5.350000
max      84.000000
Name: Enclosing_Area_4, dtype: float64
```

In [ ]:
```
plot_bar_chart_numeric(enclosing_area_4_short, 1, \
                        'Enclosing_Area_4', 'Count', 'Enclosing_Area_4', \
                        int(enclosing_numeric_data['Enclosing_Area_4'].max()))
```



In [ ]:
```
enclosing_area_4_data = \
plot_data_range(enclosing_area_4_short['Enclosing_Area_4'].\
                reset_index(drop = True), 'Enclosing_Area_4', "h")
```



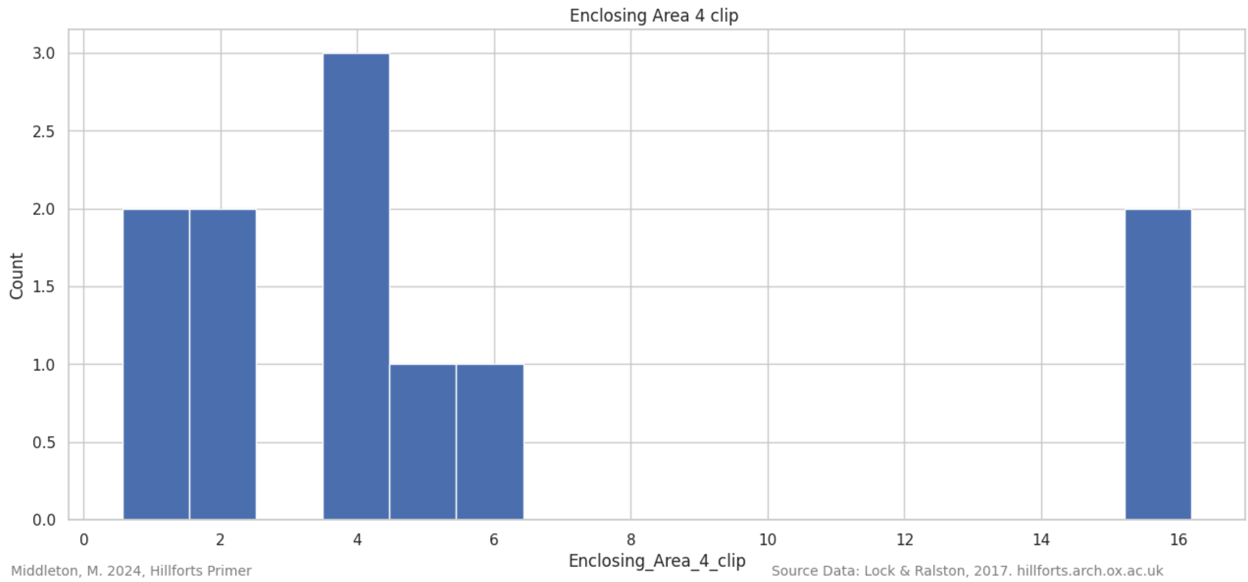
Enclosing Area 4 Clipped Plotted

This group contains forts in a range up to 16.2 Ha and a single huge hillfort at 84 Ha. To aid in visualising this data this outlier has been pooled to 16.2 Ha.

In [ ]:
```
enclosing_area_4_data_clip = enclosing_area_4_short.copy()
enclosing_area_4_data_clip['Enclosing_Area_4_clip'] = \
enclosing_area_4_data_clip['Enclosing_Area_4'].\
clip(enclosing_area_4_data_clip['Enclosing_Area_4'], enclosing_area_4_data[-1], \
     axis=0)
enclosing_area_4_data_clip['Enclosing_Area_4_clip'].describe()
```

```
Out[ ]: count    11.000000
mean      5.404545
std       5.594157
min       0.560000
25%       1.830000
50%       3.600000
75%       5.350000
max      16.200000
Name: Enclosing_Area_4_clip, dtype: float64
```

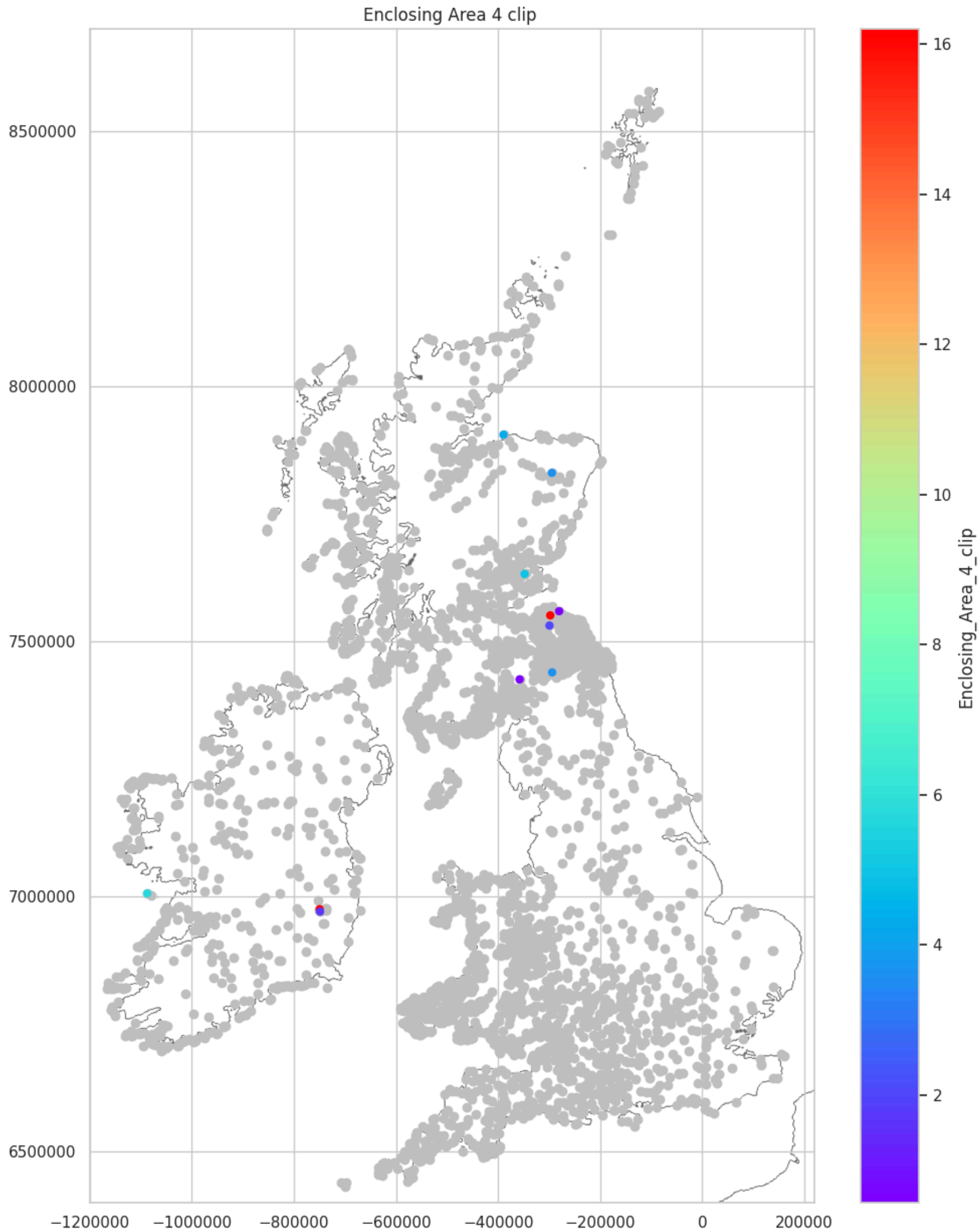
```
In [ ]: plot_bar_chart_numeric(enclosing_area_4_data_clip, 1, \
                        'Enclosing_Area_4_clip', 'Count', 'Enclosing_Area_4_clip', \
                        int(enclosing_area_4_data_clip['Enclosing_Area_4_clip'].max()))
```



Enclosing Area 4 Clipped Mapped

The forts in this class are mostly located in the Northeast. This, and the low number of records, suggest this data has a survey bias.

```
In [ ]: plot_type_values(enclosing_area_4_data_clip, 'Enclosing_Area_4_clip', \
                  'Enclosing_Area_4_clip')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Enclosed Area : Difference between Enclosing Enclosed Area and Enclosing Area 1 Plotted

Note: Enclosing_Enclosed_Area should not be confused with Enclosing_Area.

There are 3807 hillfort records that have both 'Enclosing_Enclosed_Area' and 'Enclosing_Area_1' recorded. 313 of these hillforts have an 'Enclosing_Enclosed_Area' that is larger than 'Enclosing_Area_1'. Of these, the majority are between, 0.27 and 1.96 Ha larger. The largest difference is 79.33 Ha.

In [ ]:
```
#Hillforts with an 'Enclosing Enclosed Area'
enclosing_enclosed_area = location_enclosing_data.copy()
enclosing_enclosed_area = \
enclosing_enclosed_area[enclosing_enclosed_area['Enclosing_Enclosed_Area']>=0]
enclosing_enclosed_area['Enclosing_Enclosed_Area'].describe()
```

Out[ ]:
```
count    3807.000000
mean        1.823261
std         5.652301
min         0.010000
25%         0.200000
50%         0.410000
75%         1.400000
max       130.000000
Name: Enclosing_Enclosed_Area, dtype: float64
```

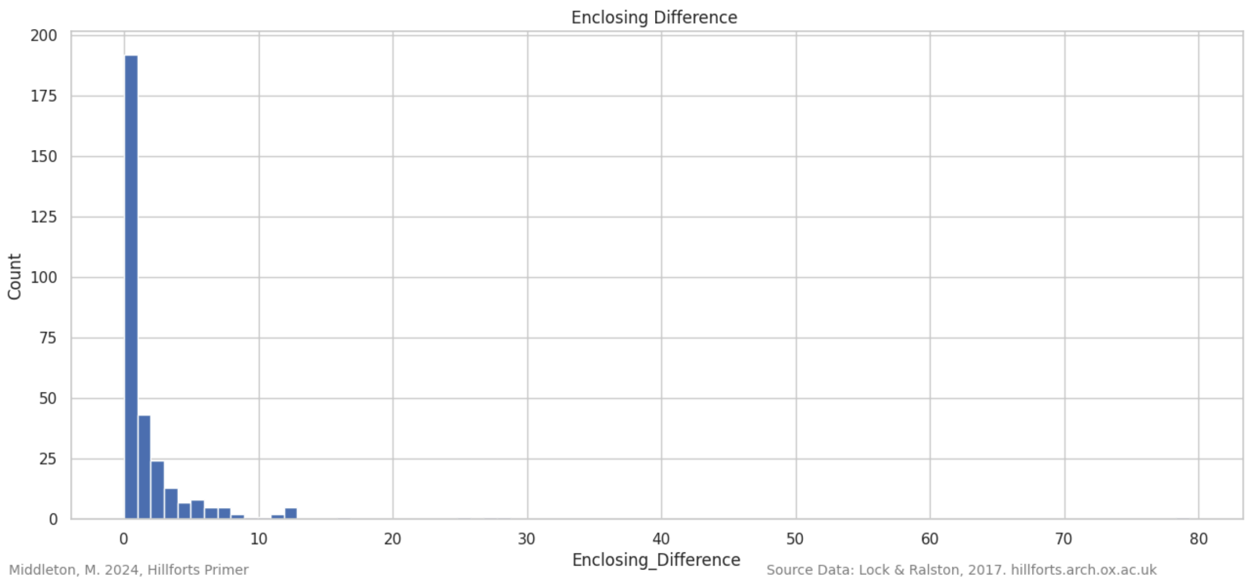
In [ ]:
```
#The difference in area between 'Enclosing Enclosed Area' and 'Enclosing Area 1'
enclosing_enclosed_area['Enclosing_Difference'] = \
enclosing_enclosed_area['Enclosing_Enclosed_Area'] - \
enclosing_enclosed_area['Enclosing_Area_1']
enclosing_enclosed_area['Enclosing_Difference'].describe()
enclosing_difference = \
enclosing_enclosed_area[enclosing_enclosed_area['Enclosing_Difference']>0]
enclosing_difference['Enclosing_Difference'].describe()
```

Out[ ]:
```
count    313.000000
mean       2.170128
std        5.629909
min        0.010000
25%        0.270000
50%        0.600000
75%        1.970000
max       79.330000
Name: Enclosing_Difference, dtype: float64
```

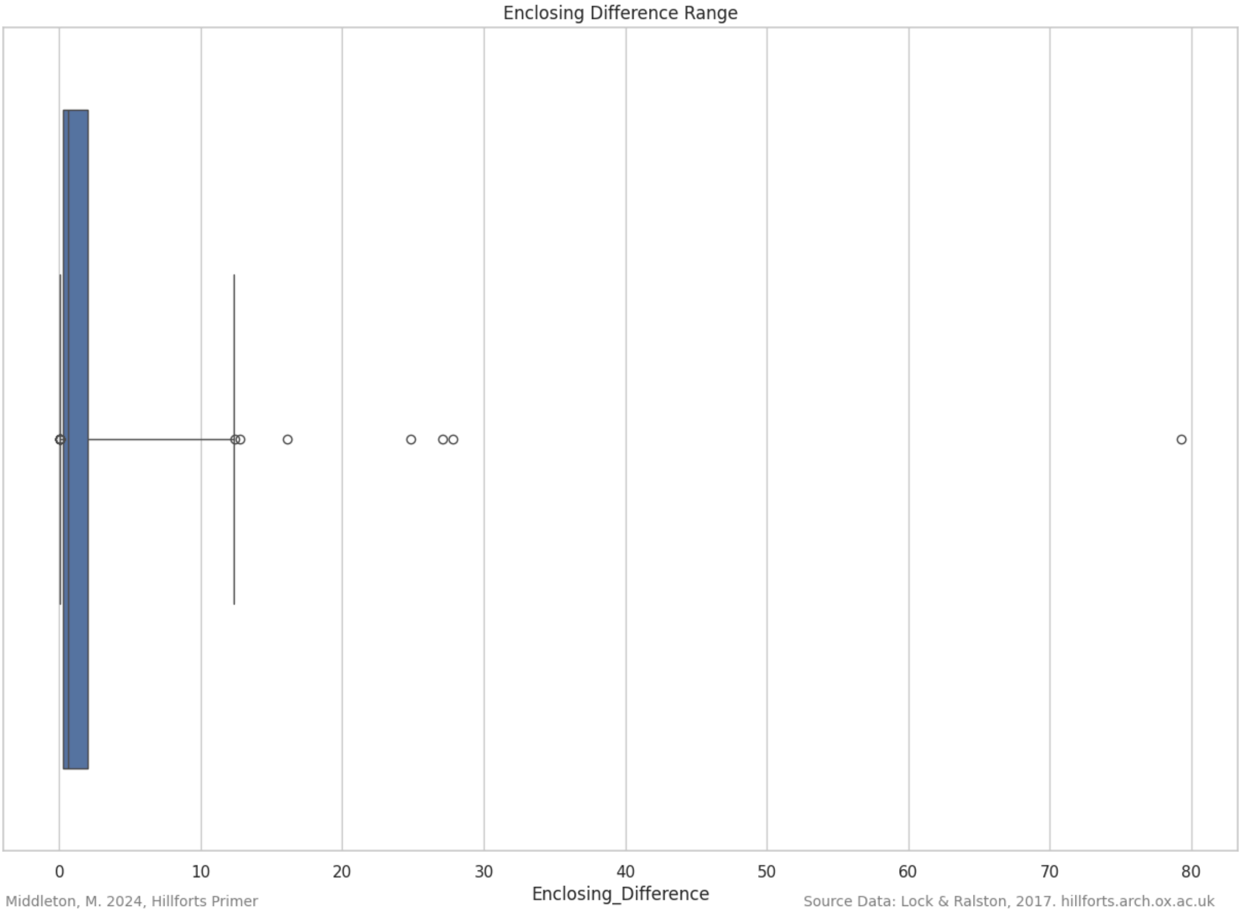
In [ ]:
```
#Number of 'Enclosing Enclosed Area' records where there is no 'Enclosing Area 1'
eea_but_no_ea1 = \
enclosing_enclosed_area[enclosing_enclosed_area['Enclosing_Area_1']==-1]
len(eea_but_no_ea1['Enclosing_Difference'])
```

Out[ ]:
```
0
```

In [ ]:
```
plot_bar_chart_numeric(enclosing_difference, 1, 'Enclosing_Difference', \
                       'Count', 'Enclosing_Difference', 80)
```



In [ ]:
```
enclosing_difference_data = \
plot_data_range(enclosing_difference['Enclosing_Difference'].\
                reset_index(drop = True), 'Enclosing_Difference', "h")
```



In [ ]:
```
enclosing_difference_data
```

Out[ ]:
```
[0.04999999999999999, 0.27, 0.6, 1.9699999999999989, 12.3]
```

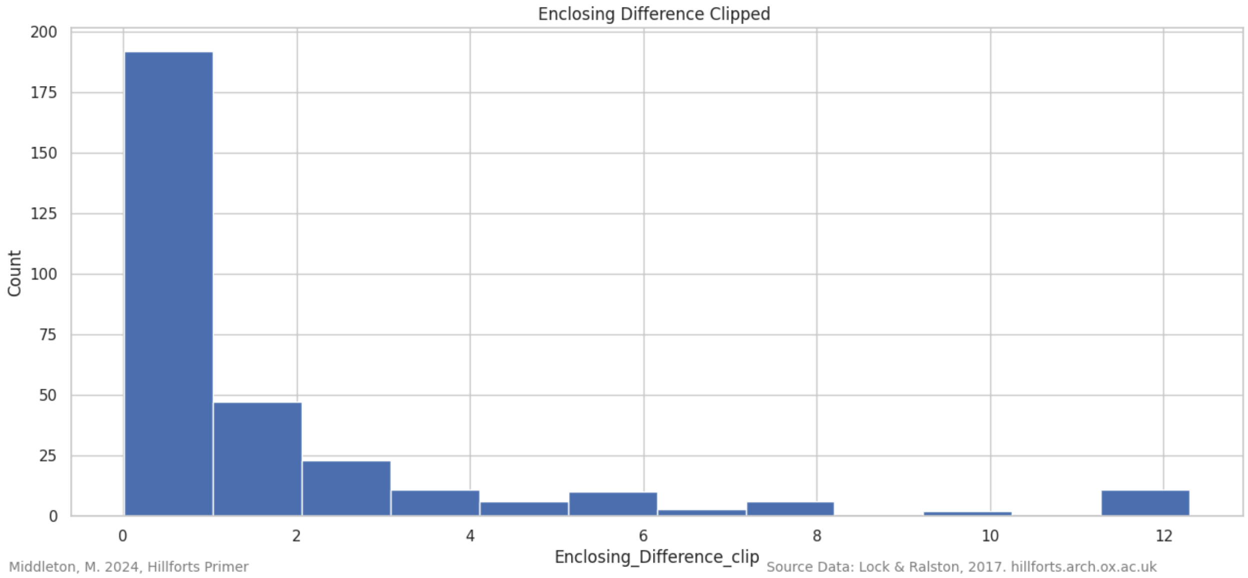
Enclosing Enclosed Area: Difference between Enclosing Enclosed Area and Enclosing Area 1 Clipped Plotted

To facilitate plotting the data has been clipped to 16 Ha. All values beyond this have been pooled into this value.

In [ ]:
```
enclosing_difference_data_clip = enclosing_difference.copy()
enclosing_difference_data_clip['Enclosing_Difference_clip'] = \
enclosing_difference_data_clip['Enclosing_Difference'].\
clip(enclosing_difference_data_clip['Enclosing_Difference'], \
     enclosing_difference_data[-1], axis=0)
enclosing_difference_data_clip['Enclosing_Difference_clip'].describe()
```

Out[ ]:
```
count    313.000000
mean       1.805176
std        2.779261
min        0.010000
25%        0.270000
50%        0.600000
75%        1.970000
max       12.300000
Name: Enclosing_Difference_clip, dtype: float64
```

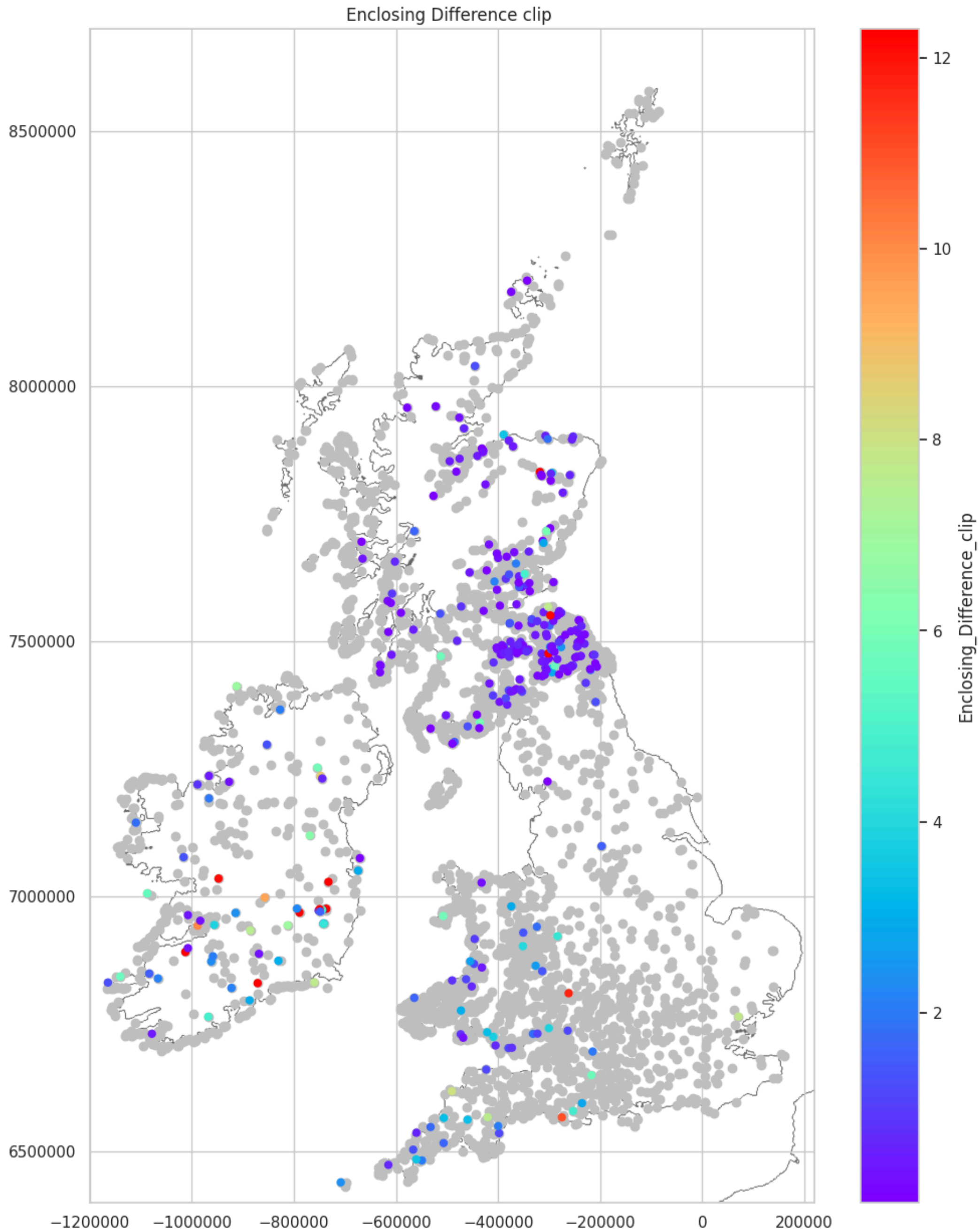
In [ ]:
```
plot_bar_chart_numeric(enclosing_difference_data_clip, 1, \
                       'Enclosing_Difference_clip', 'Count', \
                       'Enclosing Difference Clipped', \
                       int(enclosing_difference_data_clip['Enclosing_Difference_clip'].max()))
```



Enclosing Enclosed Area: Difference between Enclosing Enclosed Area and Enclosing Area 1 Clipped Mapped

Most of the hillforts with an Enclosing_Enclosed_Area are located in the Northeast. This suggests there is a recording bias.

In [ ]:
```
plot_type_values(enclosing_difference_data_clip, \
                 'Enclosing_Difference_clip', 'Enclosing_Difference_Clip')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Area Plotted

Note: Not to be confused with Enclosing_Enclosed_Area.

There are only 1259 hillforts with a recorded Enclosing_Area, the area "within the inner rampart/bank/wall where measurable", compared to 3807 hillforts that have an Enclosing_Area_1 recorded (Data Structure). The areas range from 0.02 Ha to 160 Ha. 95.6% range between 0.06 Ha and 16 Ha. The interquartile range is 0.34, to 1.435 Ha.

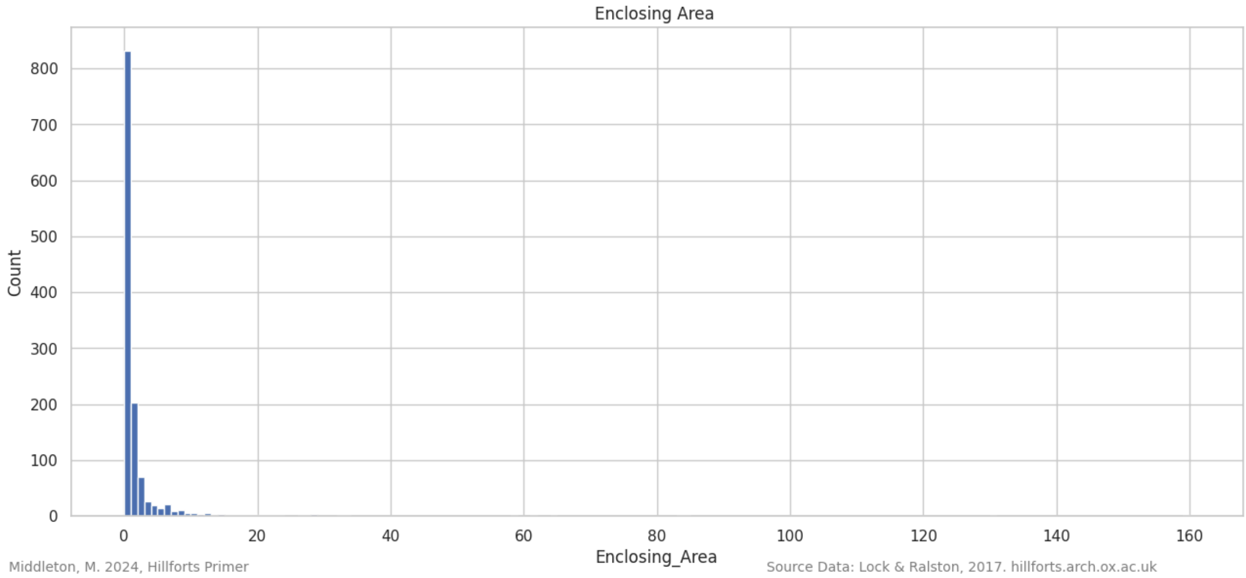
In [ ]:

```
enclosing_area = location_enclosing_data.copy()
enclosing_area = enclosing_area[enclosing_area['Enclosing_Area']>0]
enclosing_area['Enclosing_Area'].describe()
```

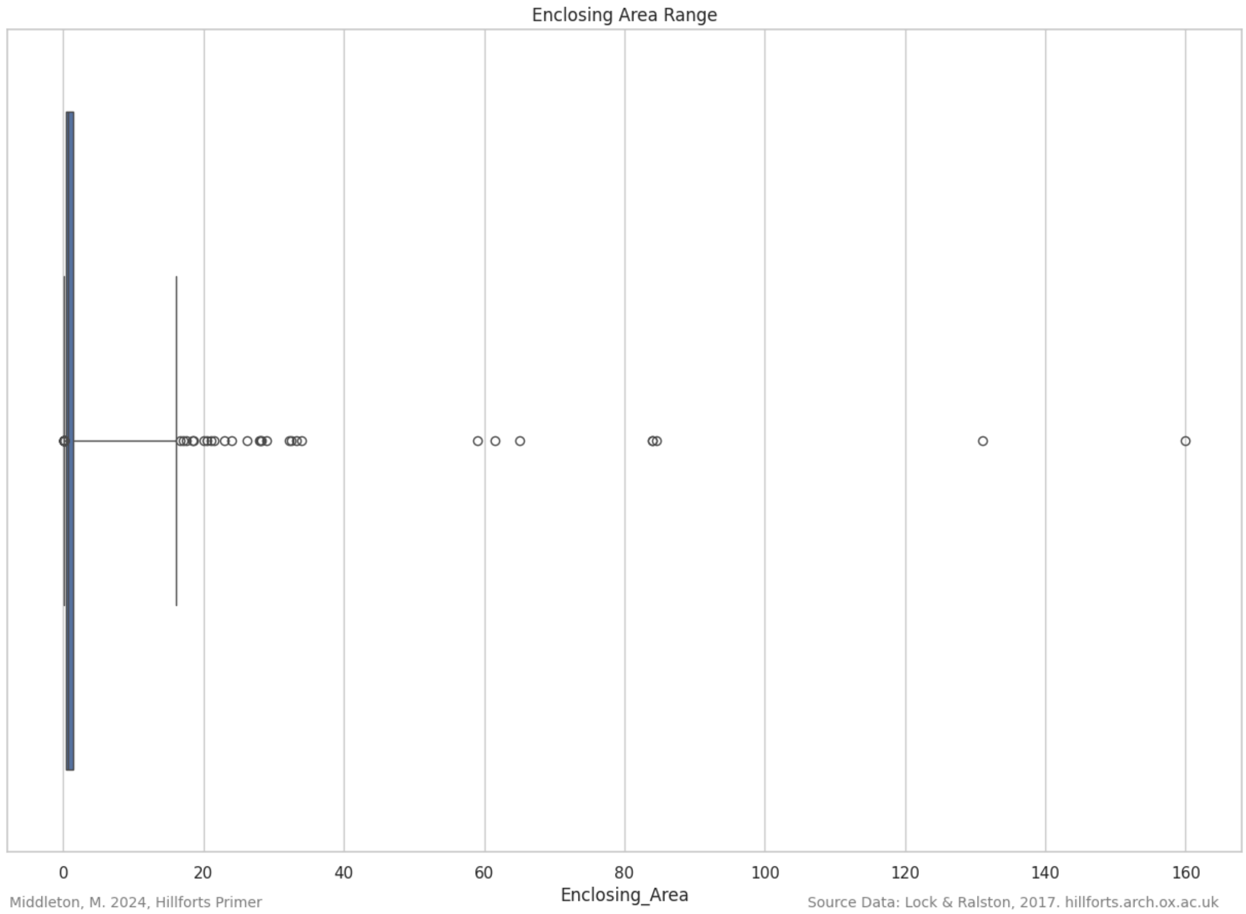
Out[ ]:

```
count    1259.000000
mean        2.325655
std         8.436951
min         0.020000
25%         0.340000
50%         0.700000
75%         1.450000
max       160.000000
Name: Enclosing_Area, dtype: float64
```

In [ ]:
```
plot_bar_chart_numeric(enclosing_area, 1, 'Enclosing_Area', 'Count', \
                       'Enclosing_Area', \
                       int(enclosing_numeric_data['Enclosing_Area'].max()))
```



In [ ]:
```
enclosing_area_data = \
plot_data_range(enclosing_area['Enclosing_Area'].reset_index(drop = True), \
                'Enclosing_Area', "h")
```



In [ ]:
```
enclosing_area_data
```

Out[ ]:
```
[0.06, 0.34, 0.7, 1.45, 16.0]
```

Enclosing Area Clipped Plotted

To facilitate plotting the data is clipped to 16 Ha. All values beyond will be pooled into this value.

In [ ]:
```
enclosing_area_clip = enclosing_area.copy()
enclosing_area_clip['Enclosing_Area_clip'] = \
enclosing_area_clip['Enclosing_Area'].clip(enclosing_area_clip['Enclosing_Area'],\
```

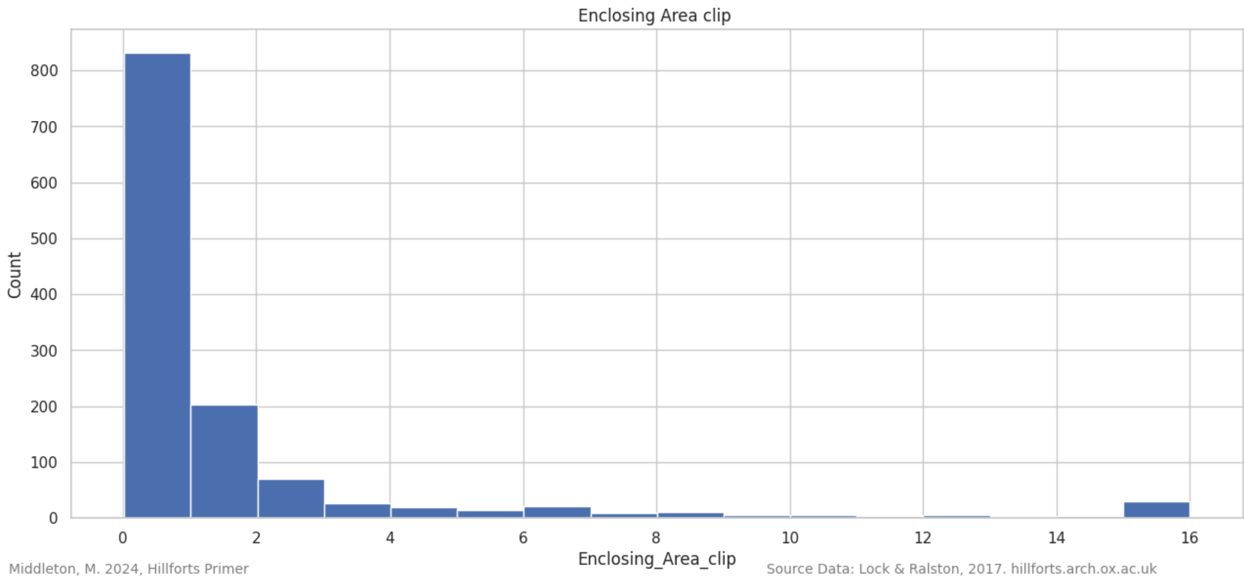
```
                                            enclosing_area_data[-1], axis=0)
enclosing_area_clip['Enclosing_Area_clip'].describe()
```

Out[ ]:

```
count    1259.000000
mean        1.713487
std         3.047517
min         0.020000
25%         0.340000
50%         0.700000
75%         1.450000
max        16.000000
Name: Enclosing_Area_clip, dtype: float64
```

In [ ]:

```
plot_bar_chart_numeric(enclosing_area_clip, 1, 'Enclosing_Area_clip', \
                       'Count', 'Enclosing_Area_clip', \
                       int(enclosing_area_clip['Enclosing_Area_clip'].max()))
```

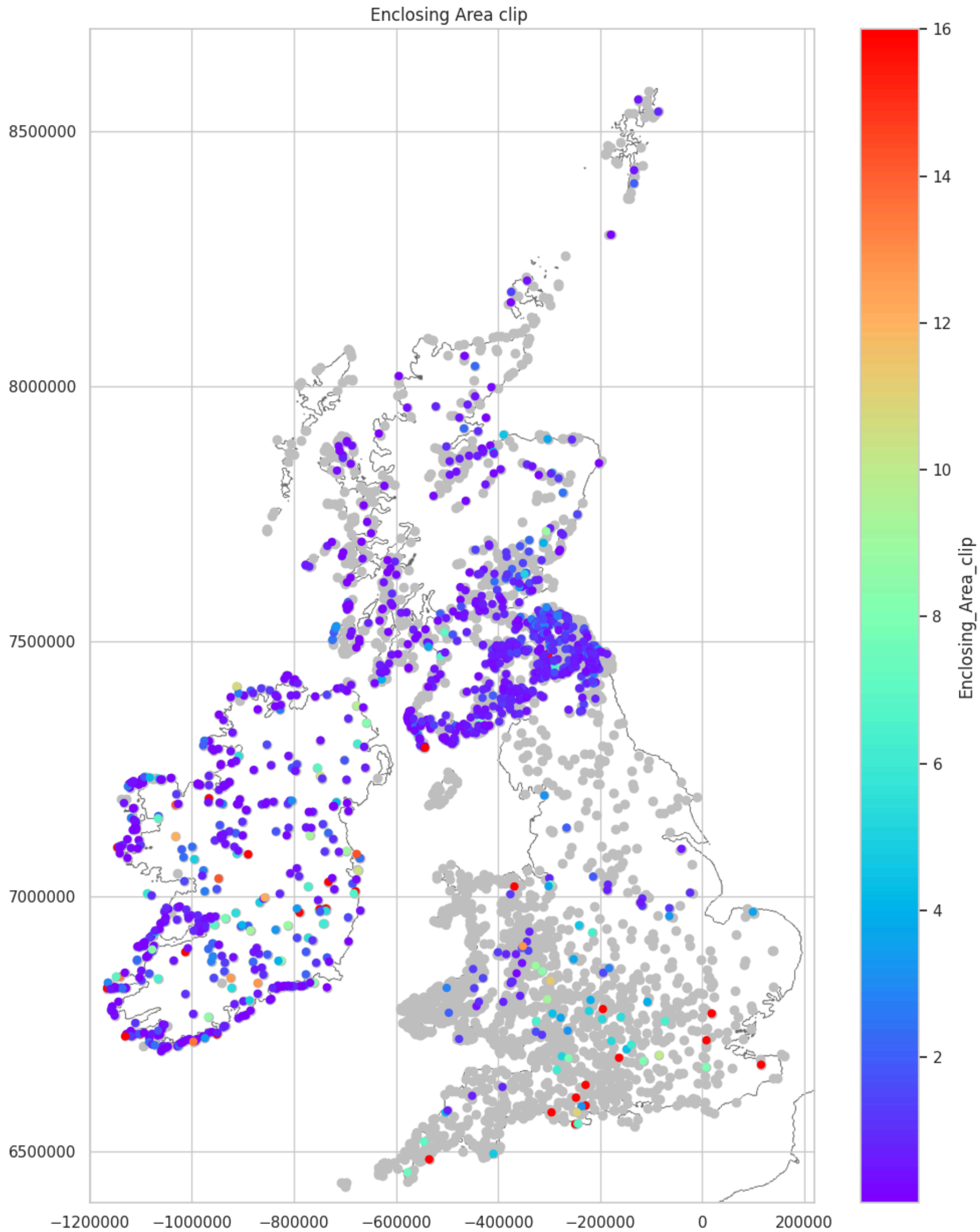


Enclosing Area Clipped Mapped

There is a recording bias toward Ireland and Scotland. What data there is in England and Wales seems to follow the pattern observed in Enclosed_Area_1, where larger hillforts are located in the South. Similarly, in the Irish and Scottish data the larger forts are located in south central Ireland.

In [ ]:

```
plot_type_values(enclosing_area_clip, 'Enclosing_Area_clip', \
                 'Enclosing Area Clipped')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ramparts Plotted

Most hillforts (75.77%) have one or two ramparts. 95.6% have four or less. 113 hillforts have no ramparts while, with 10 ramparts, West-Town, Waterford, in Ireland, is the fort with the most.

In [ ]:

```
ramparts_location_enc_data = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']>=0]
ramparts_location_enc_data['Enclosing_Max_Ramparts'].value_counts().sort_index()
```

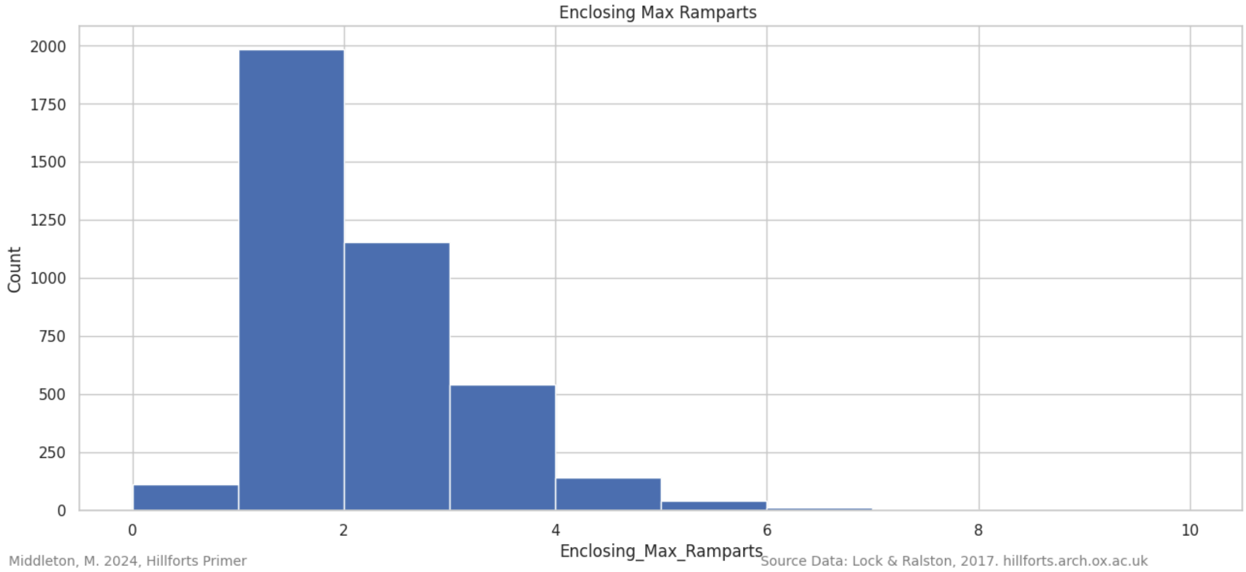
Out[ ]:

```
0.0      113
1.0     1985
2.0     1156
3.0      542
4.0      141
5.0       43
6.0       11
7.0        5
8.0        2
10.0       1
Name: Enclosing_Max_Ramparts, dtype: int64
```

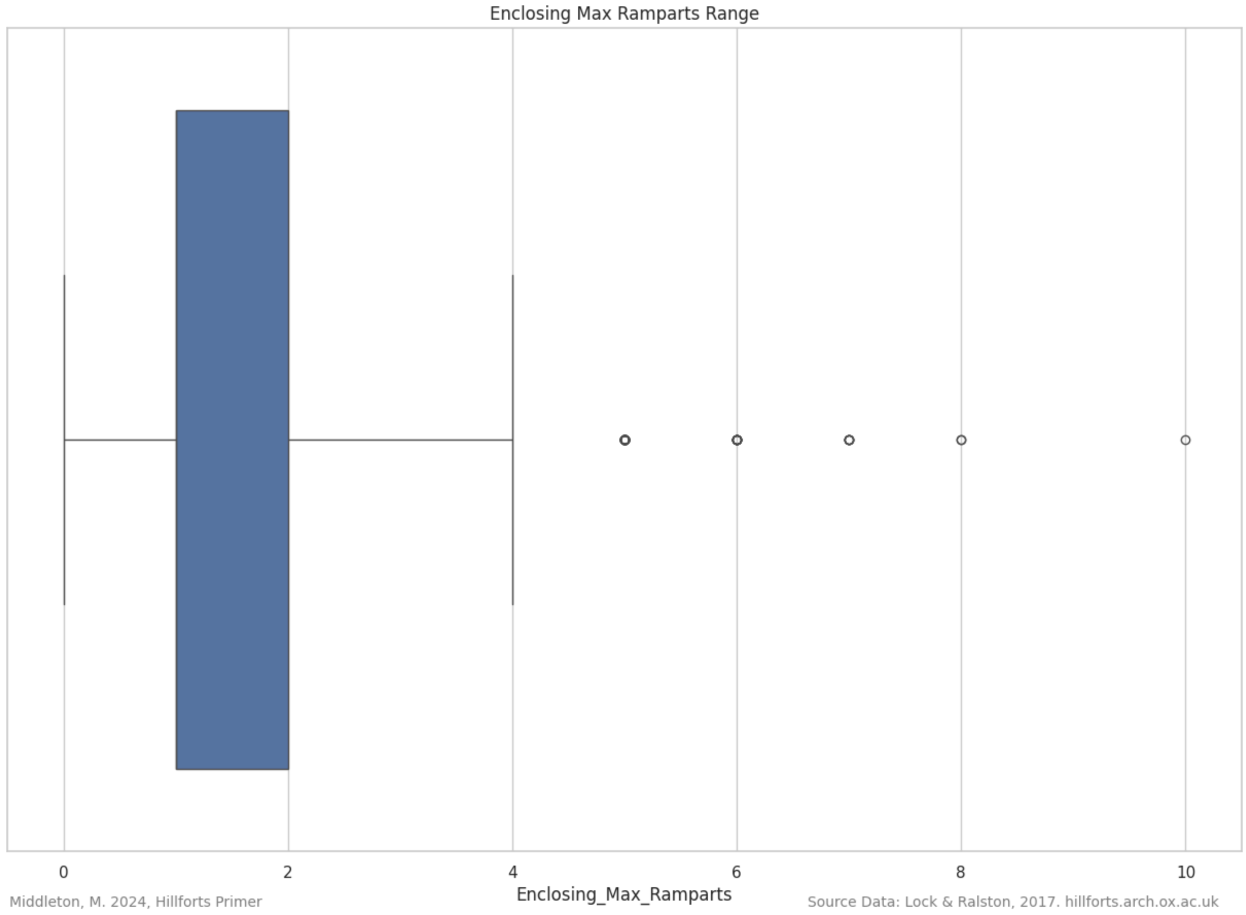
In [ ]:

```
plot_bar_chart_numeric(ramparts_location_enc_data, 1, \
                       'Enclosing_Max_Ramparts', 'Count', \
```

```
                    'Enclosing_Max_Ramparts', \
                    int(ramparts_location_enc_data['Enclosing_Max_Ramparts'].max()))
```



In [ ]:
```
ramparts_data = \
plot_data_range(ramparts_location_enc_data['Enclosing_Max_Ramparts'].\
                reset_index(drop = True), 'Enclosing_Max_Ramparts', "h")
```



In [ ]:
```
ramparts_data
```

Out[ ]:
```
[0.0, 1.0, 1.0, 2.0, 4.0]
```

Ramparts Clipped Mapped

To aid visualising the ramparts data, outliers are clipped to four ramparts. Any fort with more than four ramparts is pooled into this value. The clipped plot is still difficult to interpret so individual values will be reviewed below.

In [ ]:
```
ramparts_clip = ramparts_location_enc_data.copy()
ramparts_clip['Enclosing_Max_Ramparts_clip'] = \
ramparts_clip['Enclosing_Max_Ramparts'].\
```

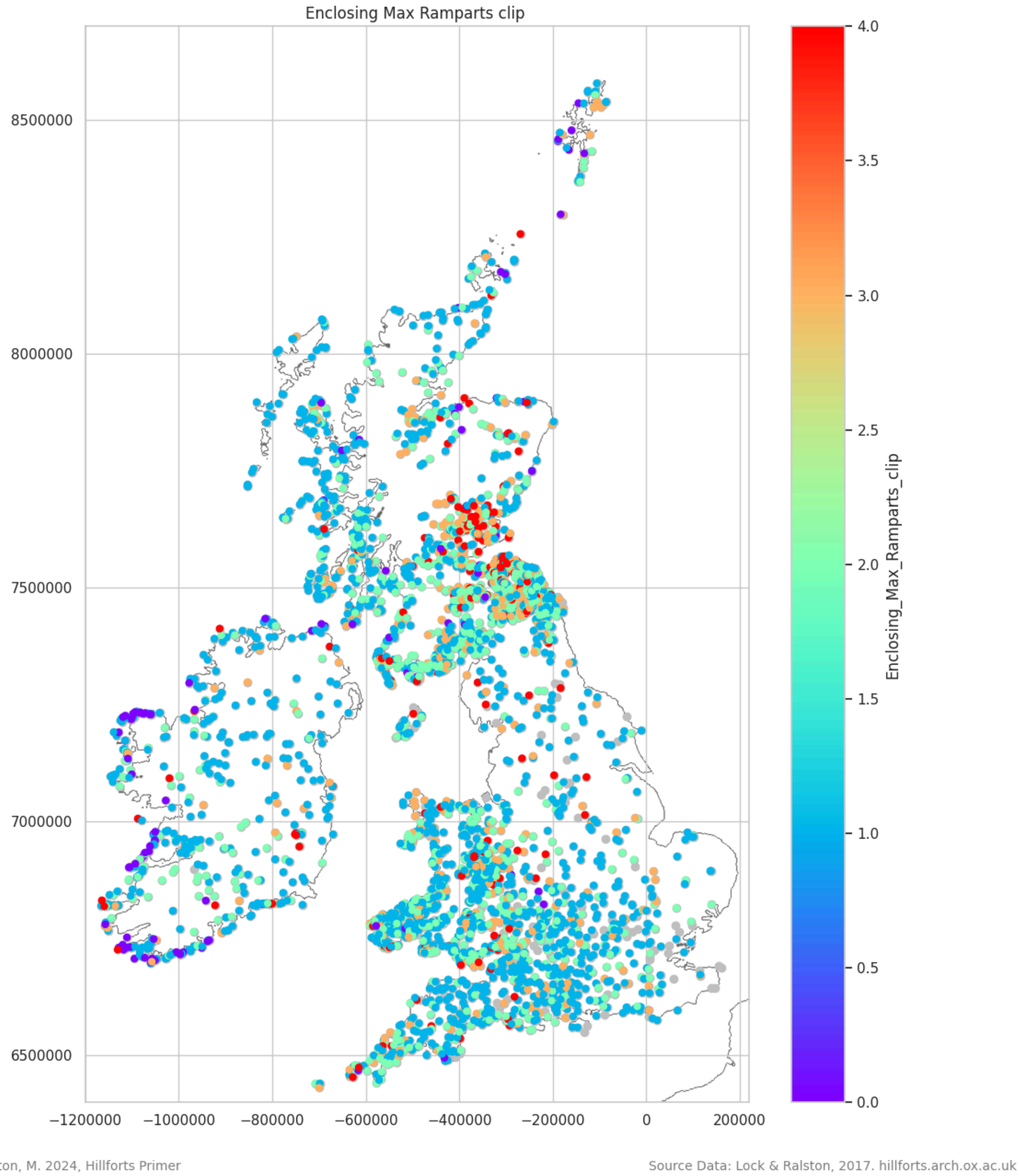
```
clip(ramparts_clip['Enclosing_Max_Ramparts'], ramparts_data[-1], axis=0)
ramparts_clip['Enclosing_Max_Ramparts_clip'].value_counts().sort_index()
```

Out[ ]:

```
0.0     113
1.0    1985
2.0    1156
3.0     542
4.0     203
Name: Enclosing_Max_Ramparts_clip, dtype: int64
```

In [ ]:

```
plot_type_values(ramparts_clip, 'Enclosing_Max_Ramparts_clip', \
                 'Enclosing_Max_Ramparts_clip')
```

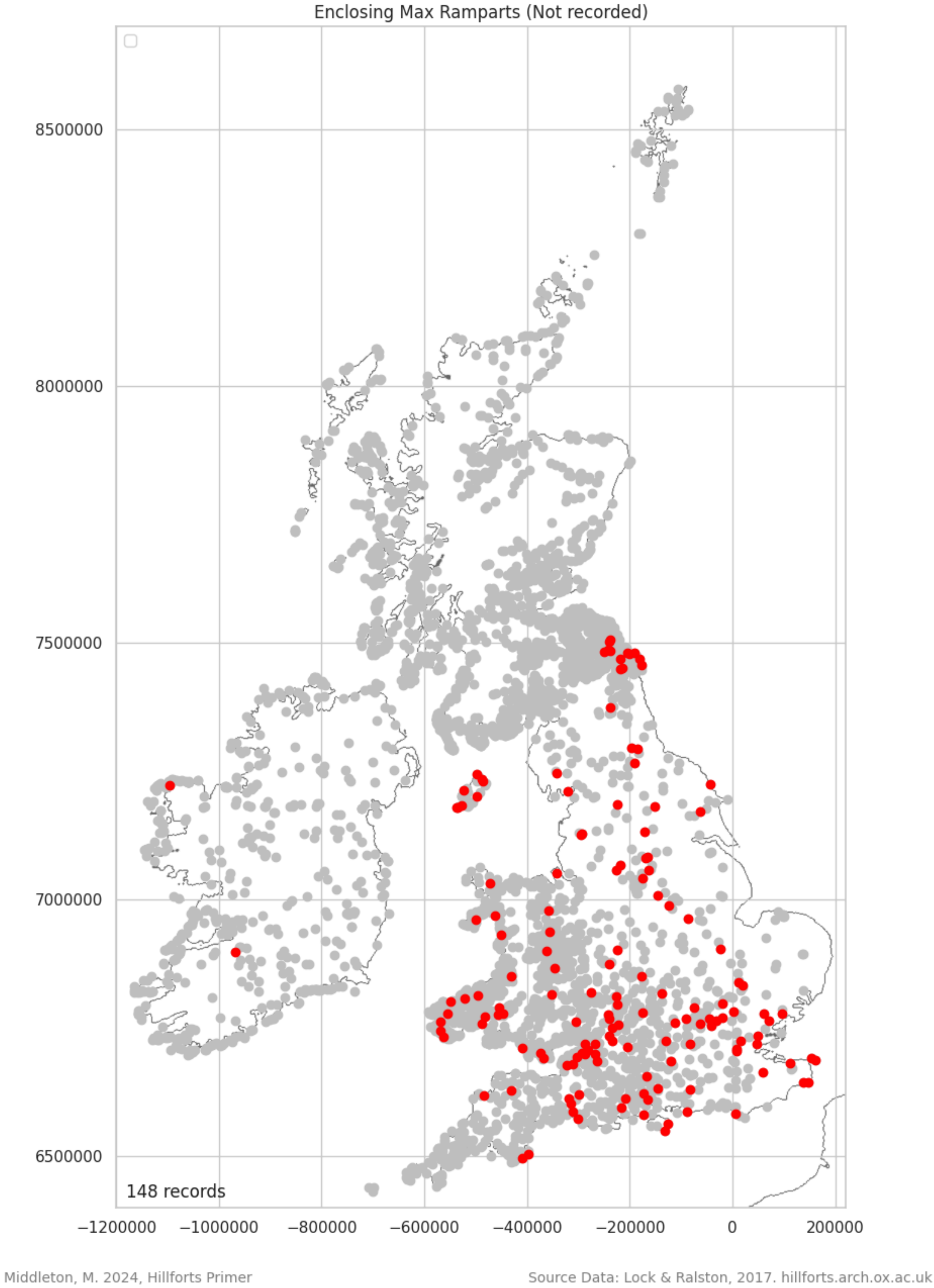


Ramparts Mapped (Not Recorded)

Just 148 (3.57%) of hillforts have not had the presence of ramparts recorded. Almost all are in England, Wales and the Isle of Man.

In [ ]:

```
nan_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==-1].copy()
nan_ramparts['Enclosing_Max_Ramparts'] = "Yes"
nan_ramparts_stats = \
plot_over_grey(nan_ramparts, 'Enclosing_Max_Ramparts', 'Yes', '(Not recorded)')
```



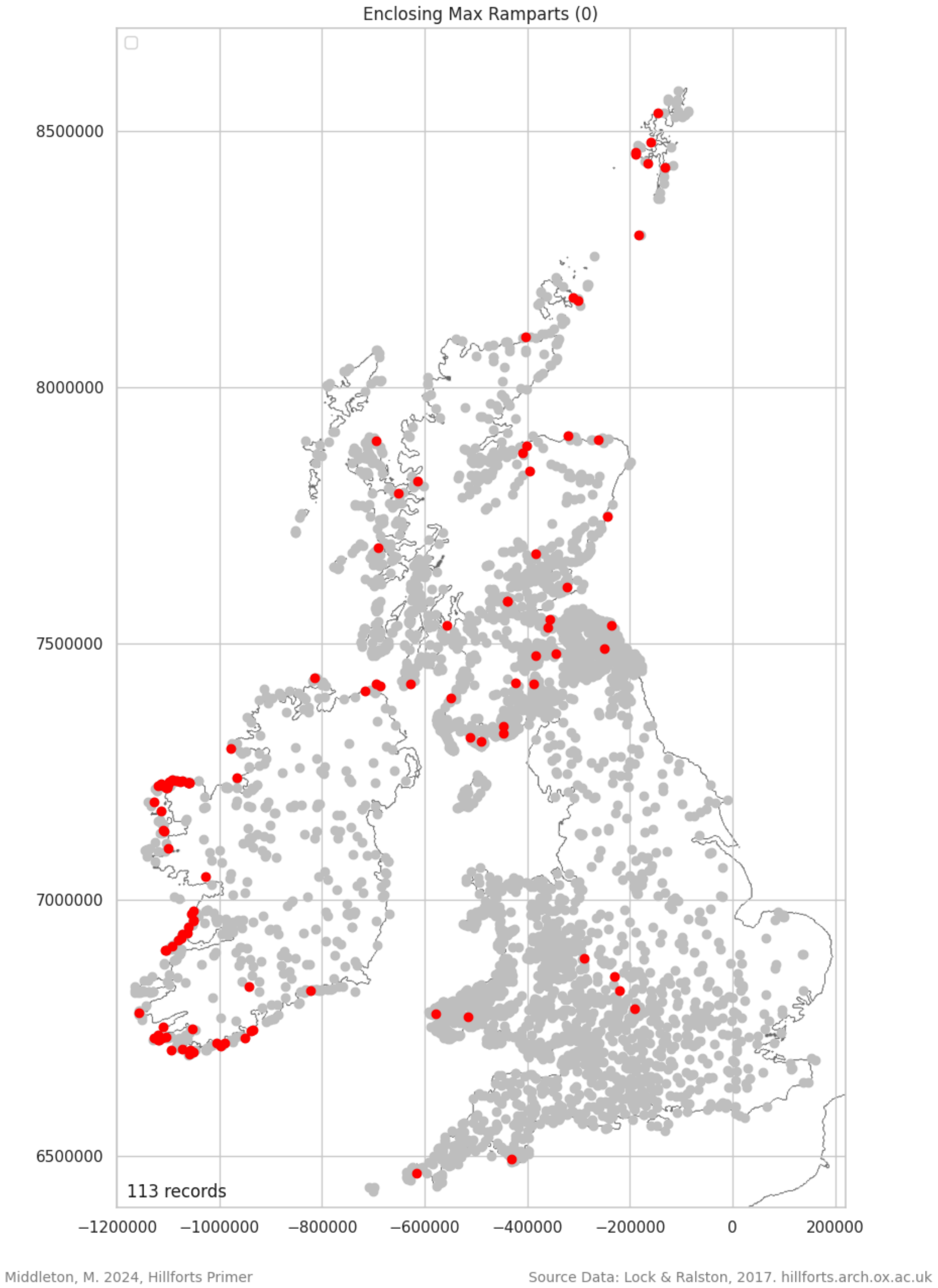

```
3.57%
```

Ramparts Mapped (0)

Hillforts without ramparts are dominated by the coastal forts of Ireland. There is a peppering across Scotland with many again located on the coast. There are very few in England and Wales.

In [ ]:
```
zero_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==0].copy()
zero_ramparts['Enclosing_Max_Ramparts'] = "Yes"
```

```
zero_ramparts_stats = plot_over_grey(zero_ramparts, \
                                     'Enclosing_Max_Ramparts', 'Yes', '(0)')
```

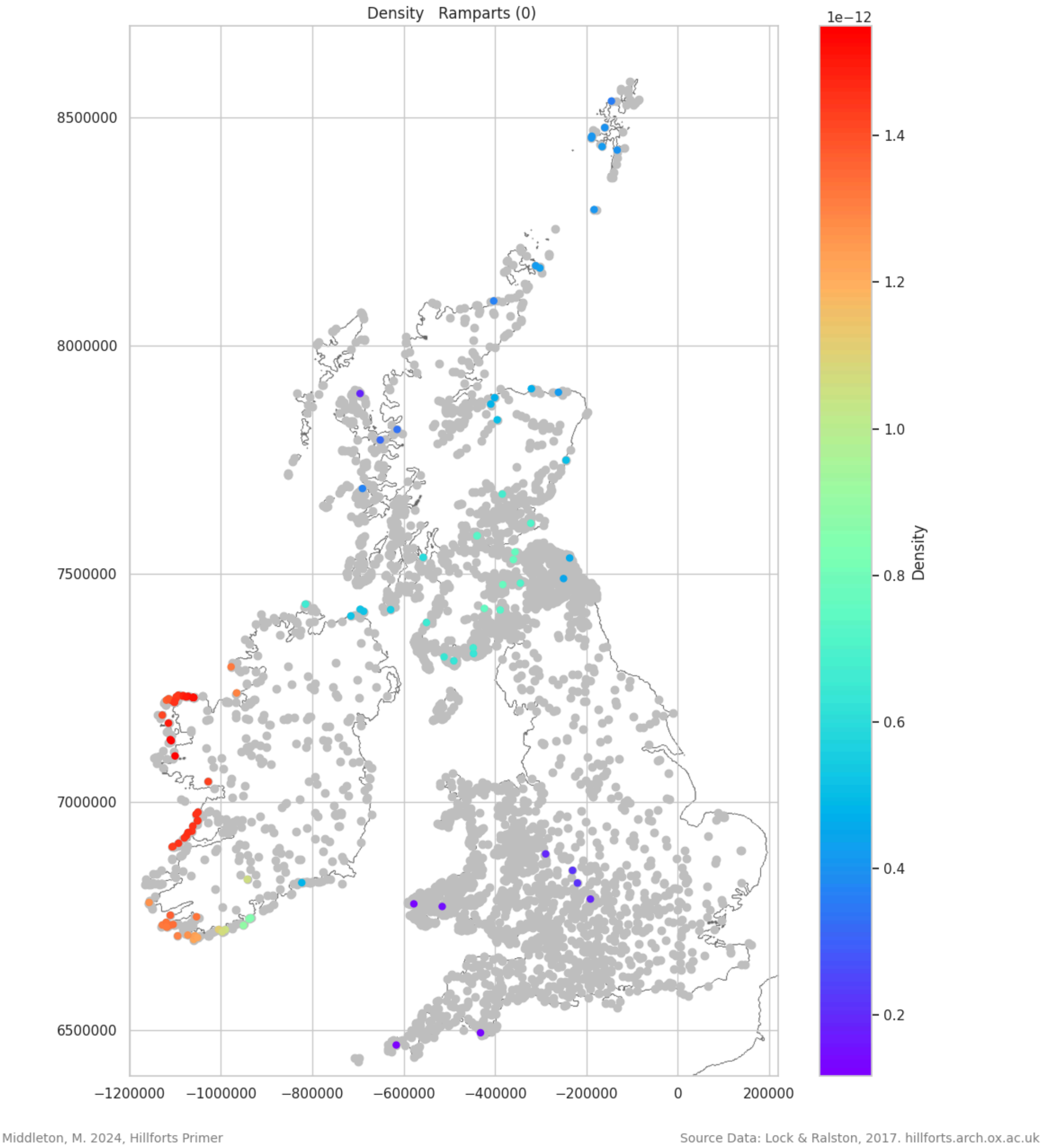



```
2.72%
```

Ramparts Density Mapped (0)

The west and south coast of Ireland is the focus of hillforts with no ramparts.

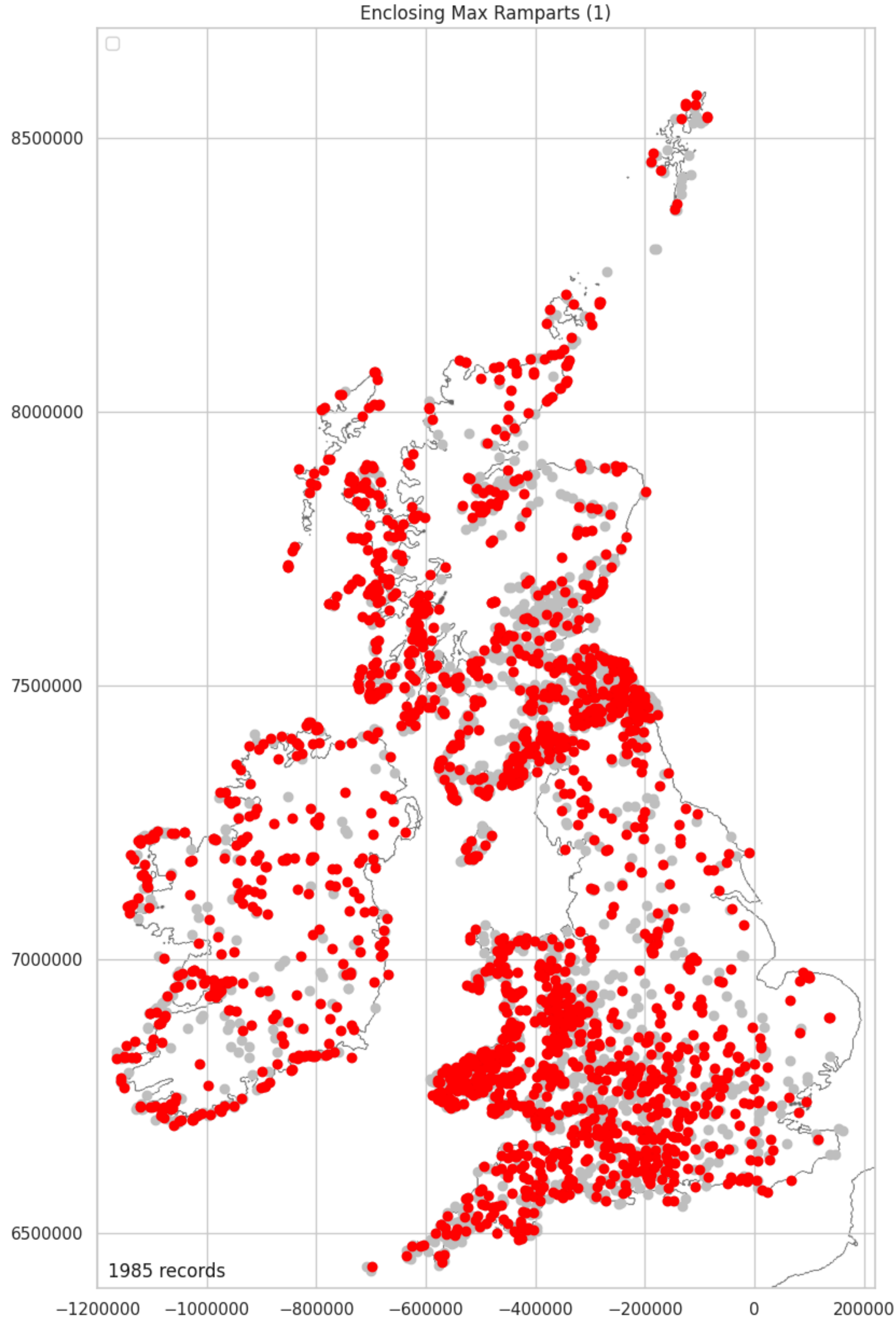
In [ ]:
```
plot_density_over_grey(zero_ramparts_stats, 'Ramparts (0)')
```



Ramparts Mapped (1)

Hillforts with a single rampart occur right across the Atlas. At 1985 examples (47.87%), a single rampart is the most common rampart layout.

In [ ]:
```
one_rampart = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==1].copy()
one_rampart['Enclosing_Max_Ramparts'] = "Yes"
one_rampart_stats = plot_over_grey(one_rampart, \
                                   'Enclosing_Max_Ramparts', 'Yes', '(1)')
```



Middleton, M. 2024, Hillforts Primer

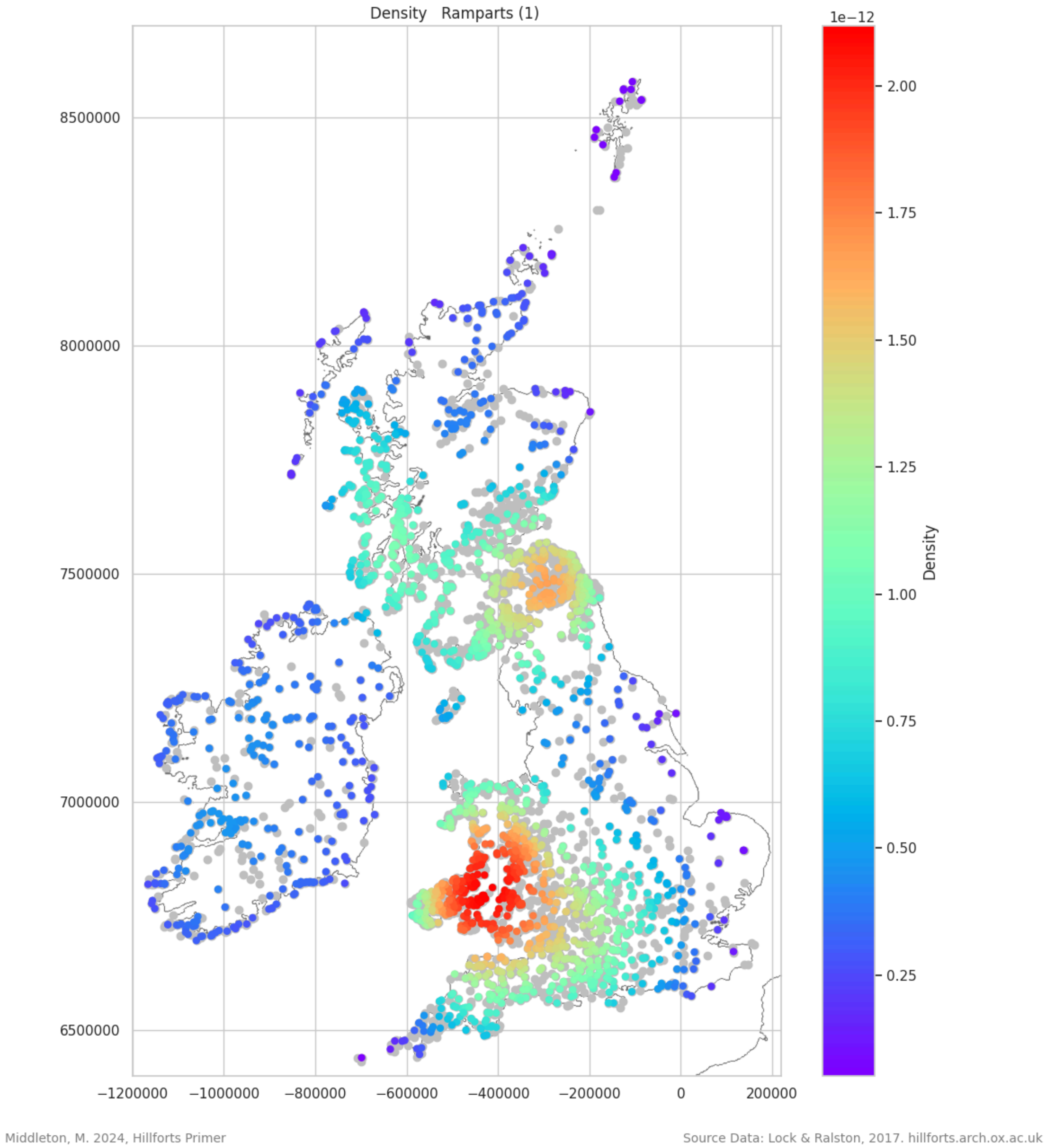
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
47.87%
```

Ramparts Density Mapped (1)

The density of single rampart forts is most intense in the South, over the southern end of the Cambrian Mountains. This contrasts to the general distribution seen in Part 1: Density Data Mapped where the most intense cluster was in the Northeast. The Northeast does show a cluster, but this is far less intense than that seen in the South. A third cluster can be seen in the Northwest. The distribution across Ireland is very uniform, and there are no significant concentrations.

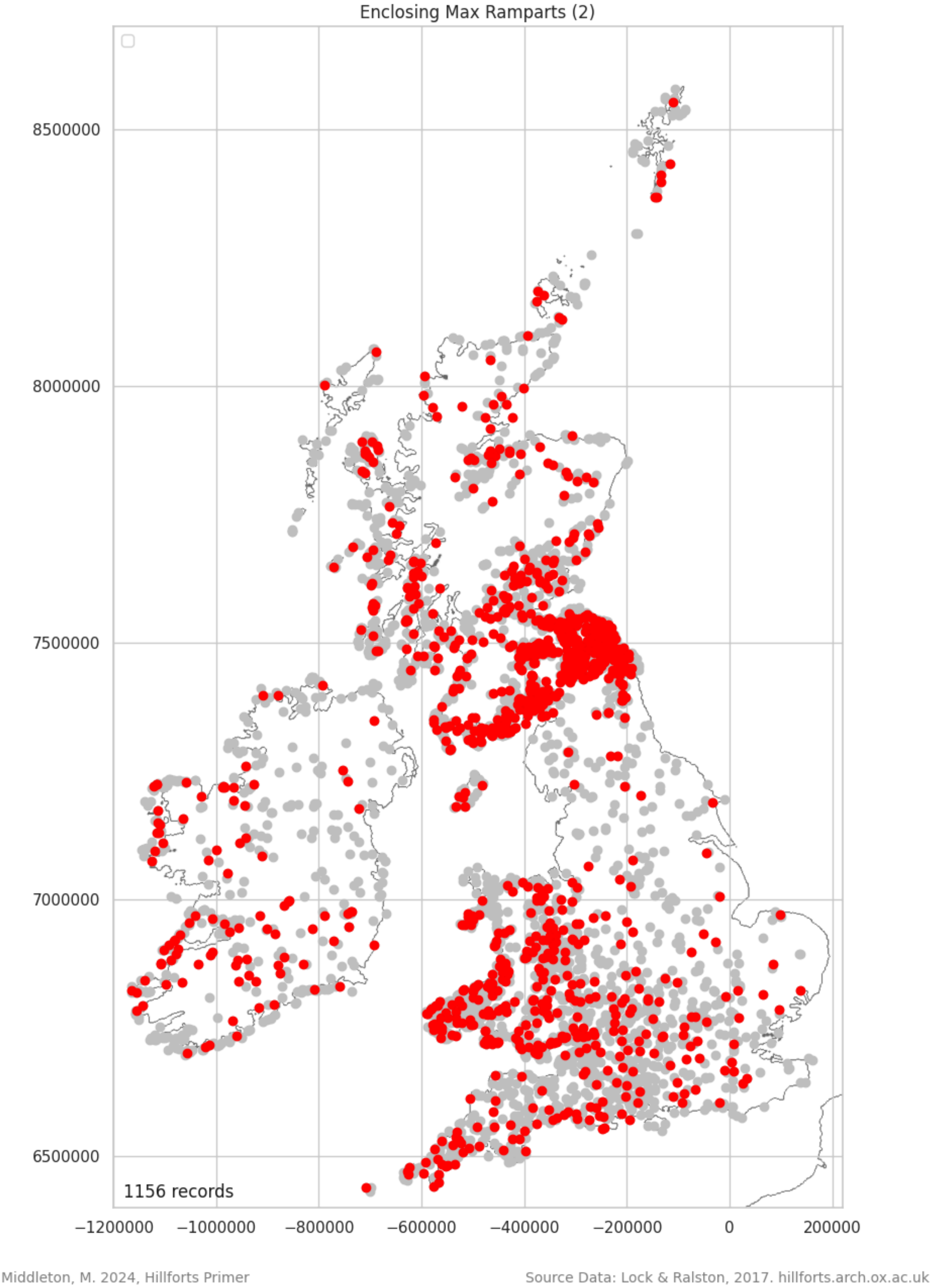
In [ ]:
```
plot_density_over_grey(one_rampart_stats, 'Ramparts (1)')
```



Ramparts Mapped (2)

There are 1356 (27.88%) hillforts with two ramparts. They are distributed mostly in the South, North and across southern Ireland.

In [ ]:
```
two_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==2].copy()
two_ramparts['Enclosing_Max_Ramparts'] = "Yes"
two_ramparts_stats = \
plot_over_grey(two_ramparts, 'Enclosing_Max_Ramparts', 'Yes', '(2)')
```

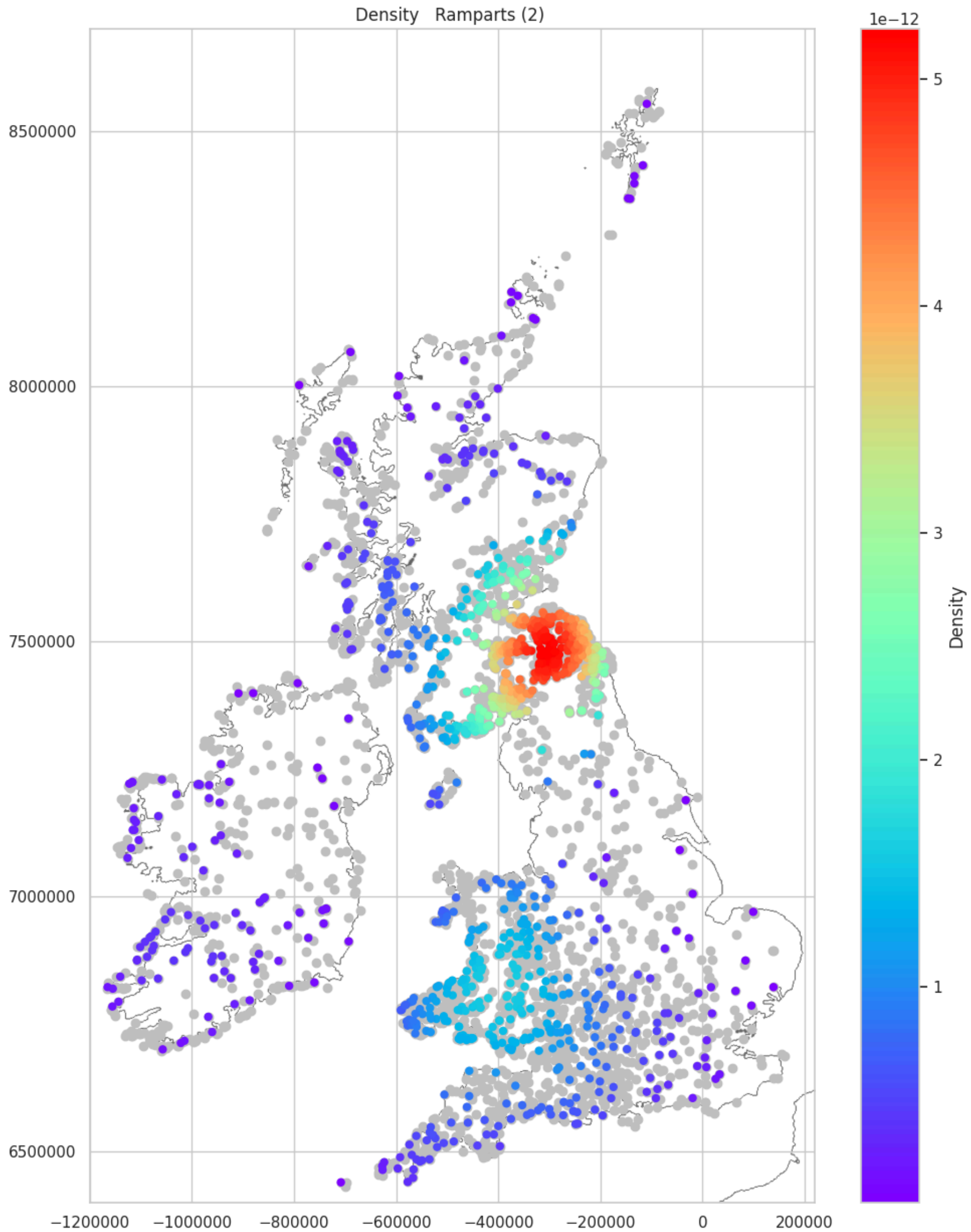


27.88%

Ramparts Density Mapped (2)

Hillforts with two ramparts cluster, most intensely, in the Northeast. There is a secondary, weak cluster, at the southern end of the Cambrian Mountains.

In [ ]:
```
plot_density_over_grey(two_ramparts_stats, 'Ramparts (2)')
```



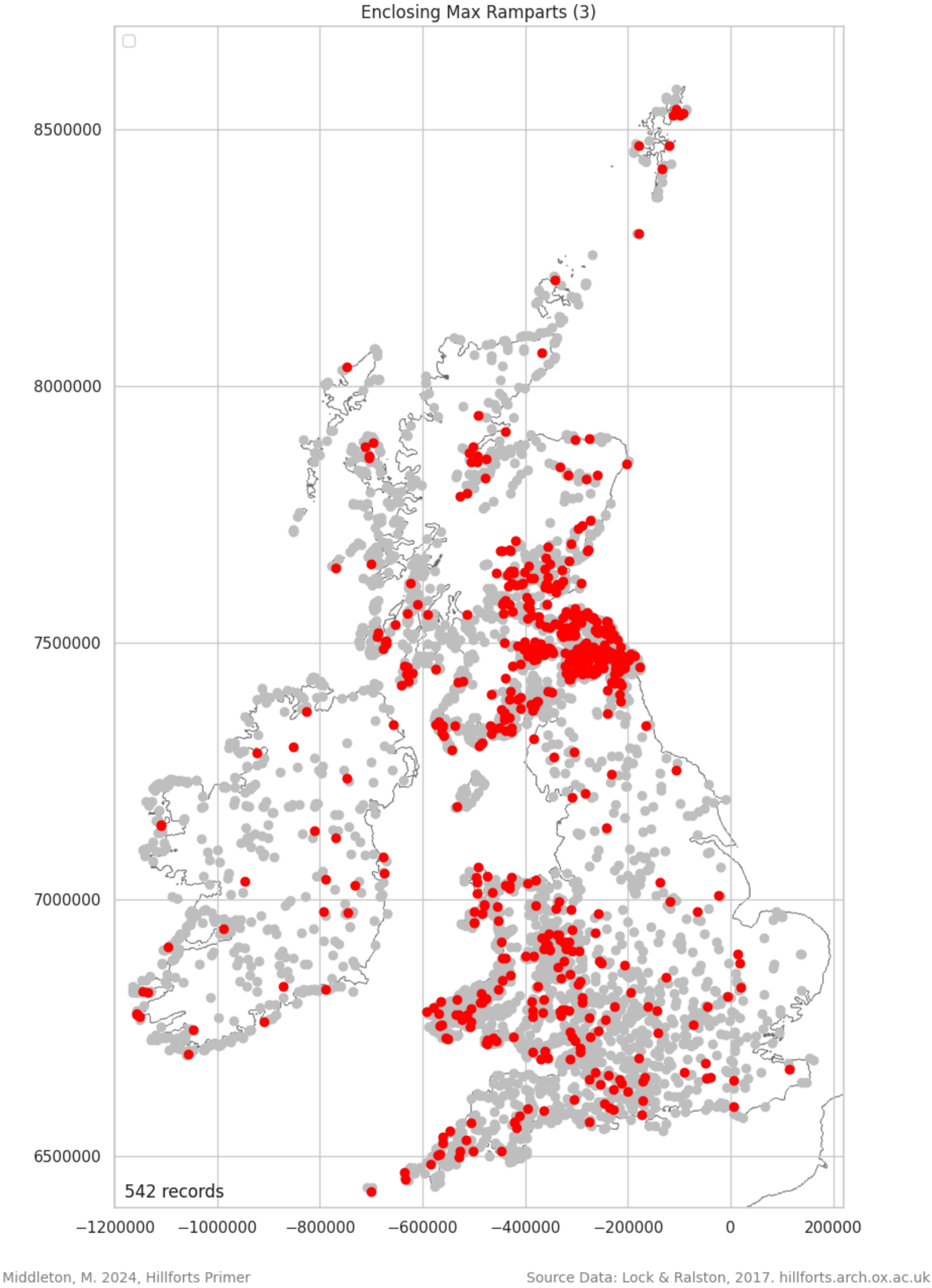
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ramparts Mapped (3)

542 hillforts (13.07%) are recorded as having three ramparts. These cluster in the Northeast and South and they are peppered lightly across Ireland.

In [ ]:
```
three_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==3].copy()
three_ramparts['Enclosing_Max_Ramparts'] = "Yes"
three_ramparts_stats = plot_over_grey(three_ramparts, \
                                      'Enclosing_Max_Ramparts', 'Yes', '(3)')
```

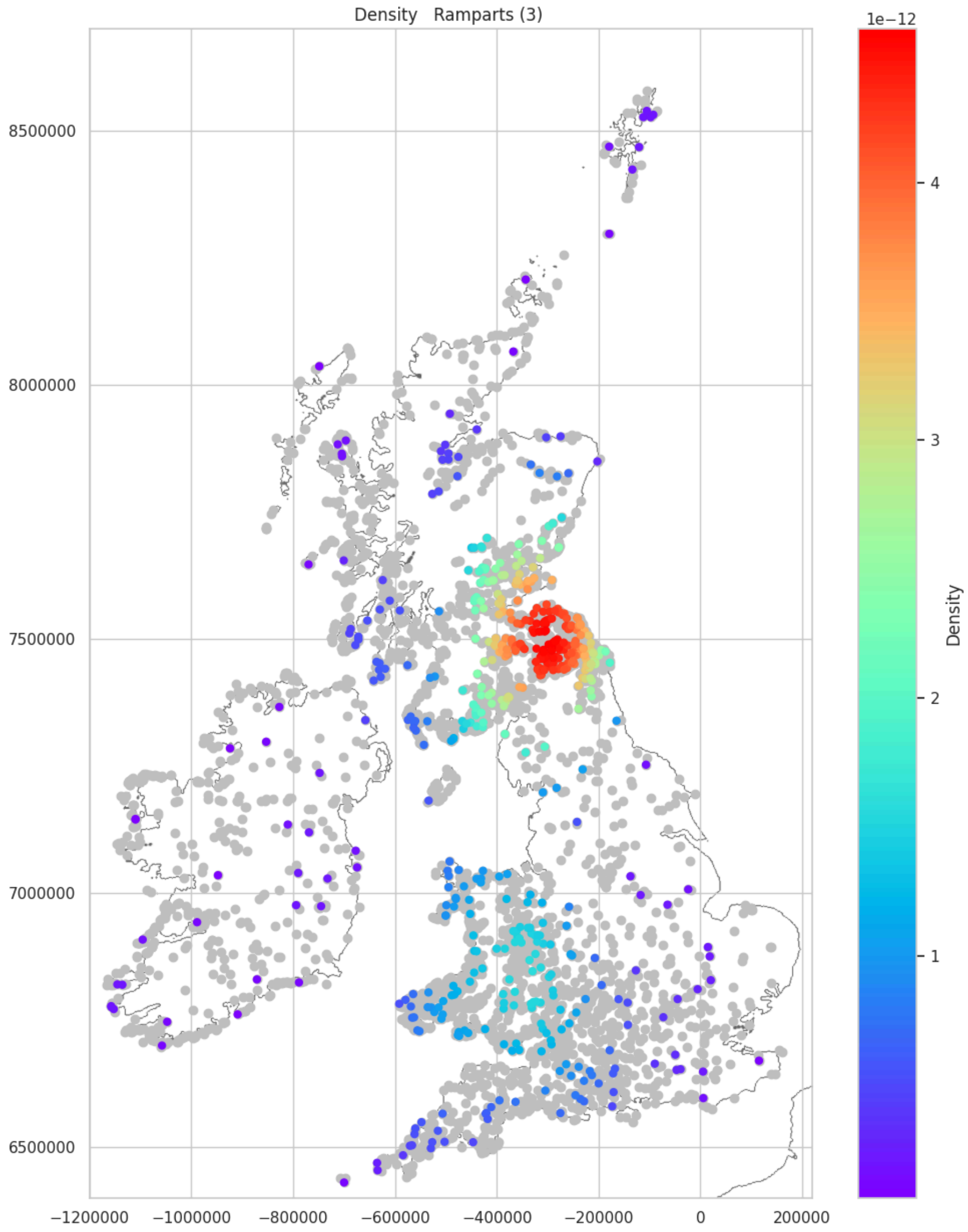



```
13.07%
```

Ramparts Density Mapped (3)

The main focus of hillforts with three ramparts is in the Northeast. There is a weak clustering along the eastern fringe of the Cambrian Mountains.

In [ ]:
```
plot_density_over_grey(three_ramparts_stats, 'Ramparts (3)')
```



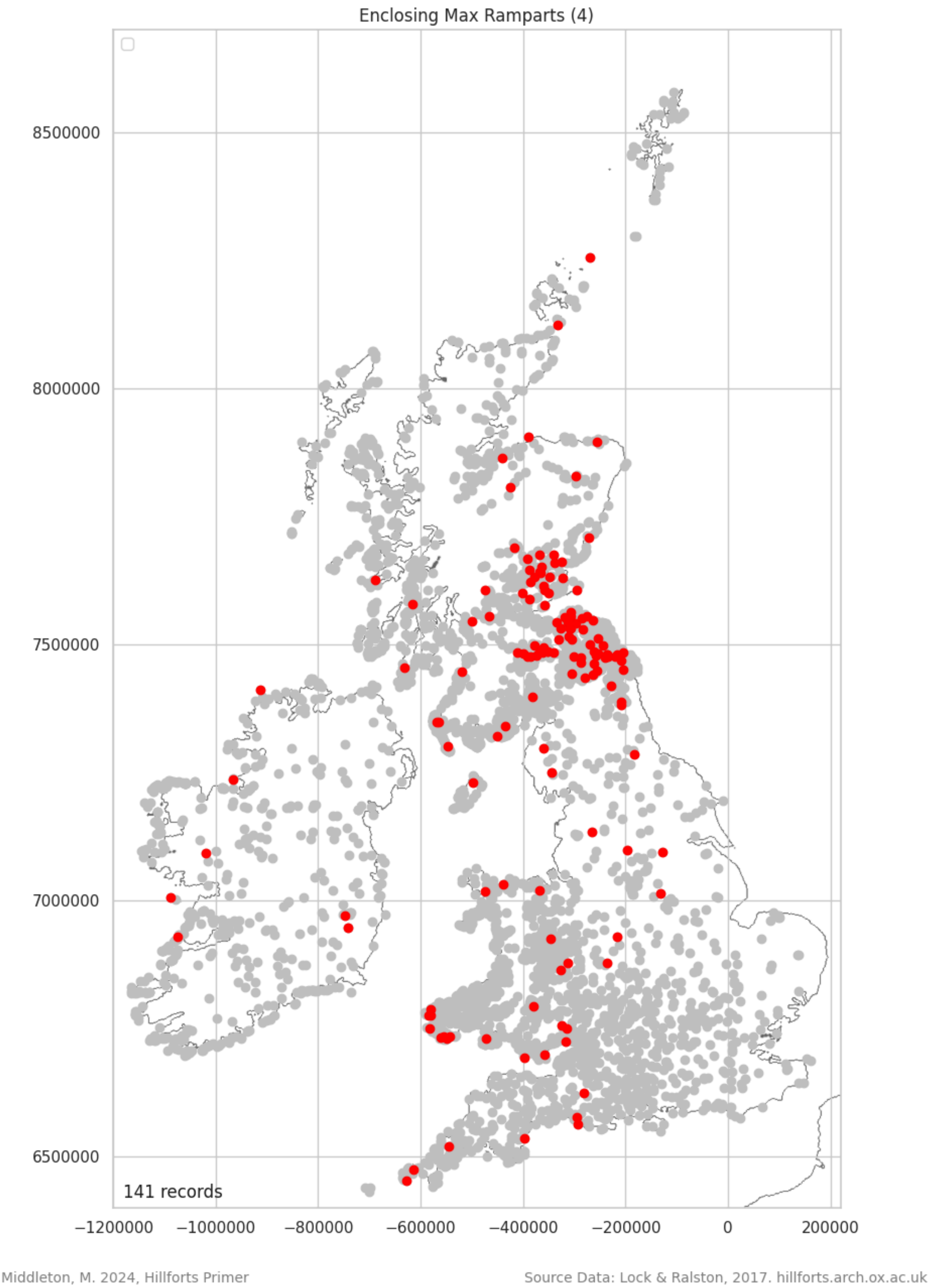
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ramparts Mapped (4)

141 hillforts (3.4%) have four ramparts. The distribution of these is noticeably concentrated in the Northeast and up into Fife, Perthshire and Angus.

In [ ]:
```
four_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==4].copy()
four_ramparts['Enclosing_Max_Ramparts'] = "Yes"
four_ramparts_stats = plot_over_grey(four_ramparts, \
                                     'Enclosing_Max_Ramparts', 'Yes', '(4)')
```

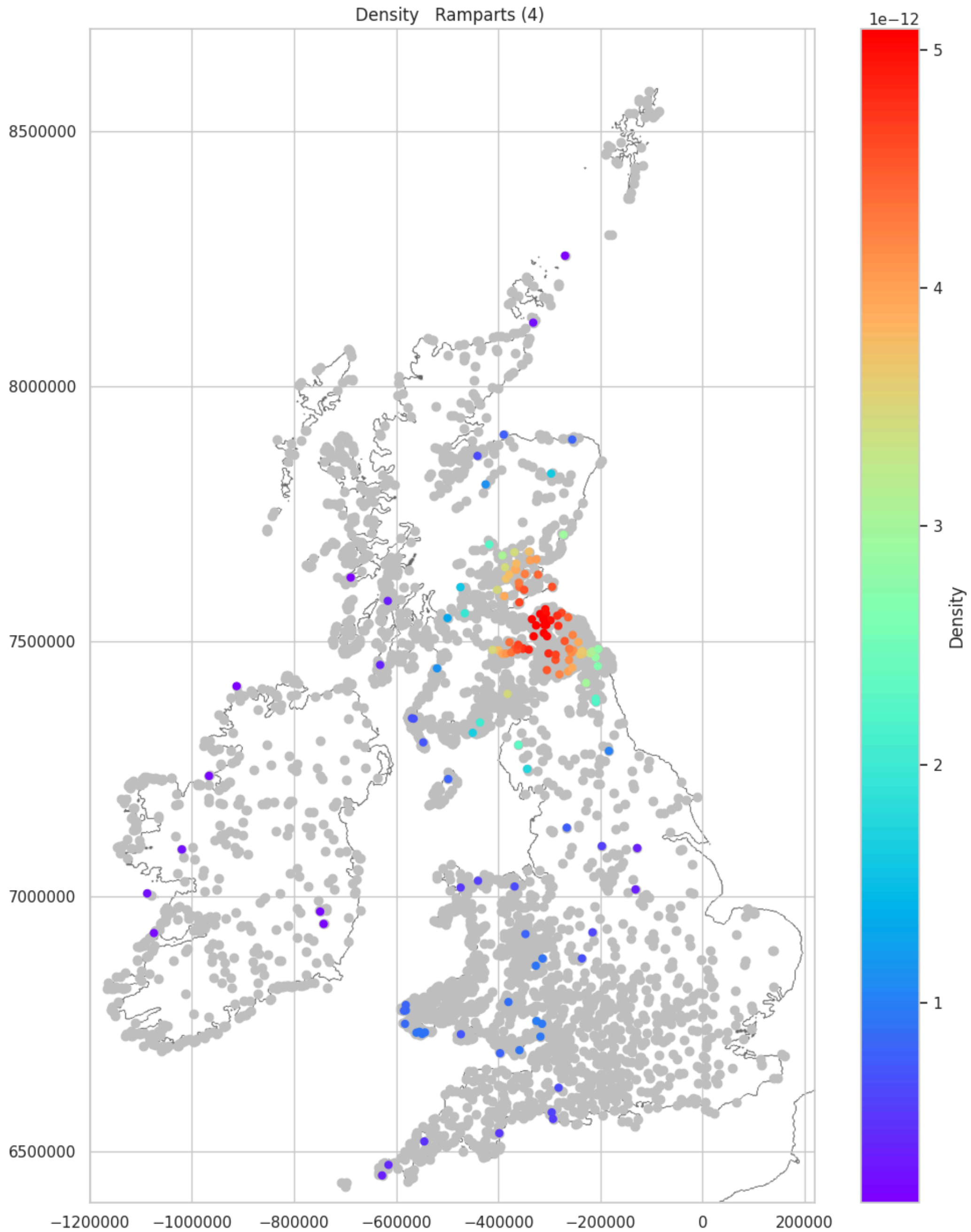



```
3.4%
```

Ramparts Density Mapped (4)

The focus for four rampart hillforts is in the Northeast over East Lothian.

```
In [ ]: plot_density_over_grey(four_ramparts_stats, 'Ramparts (4)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ramparts Mapped (4+ NE)

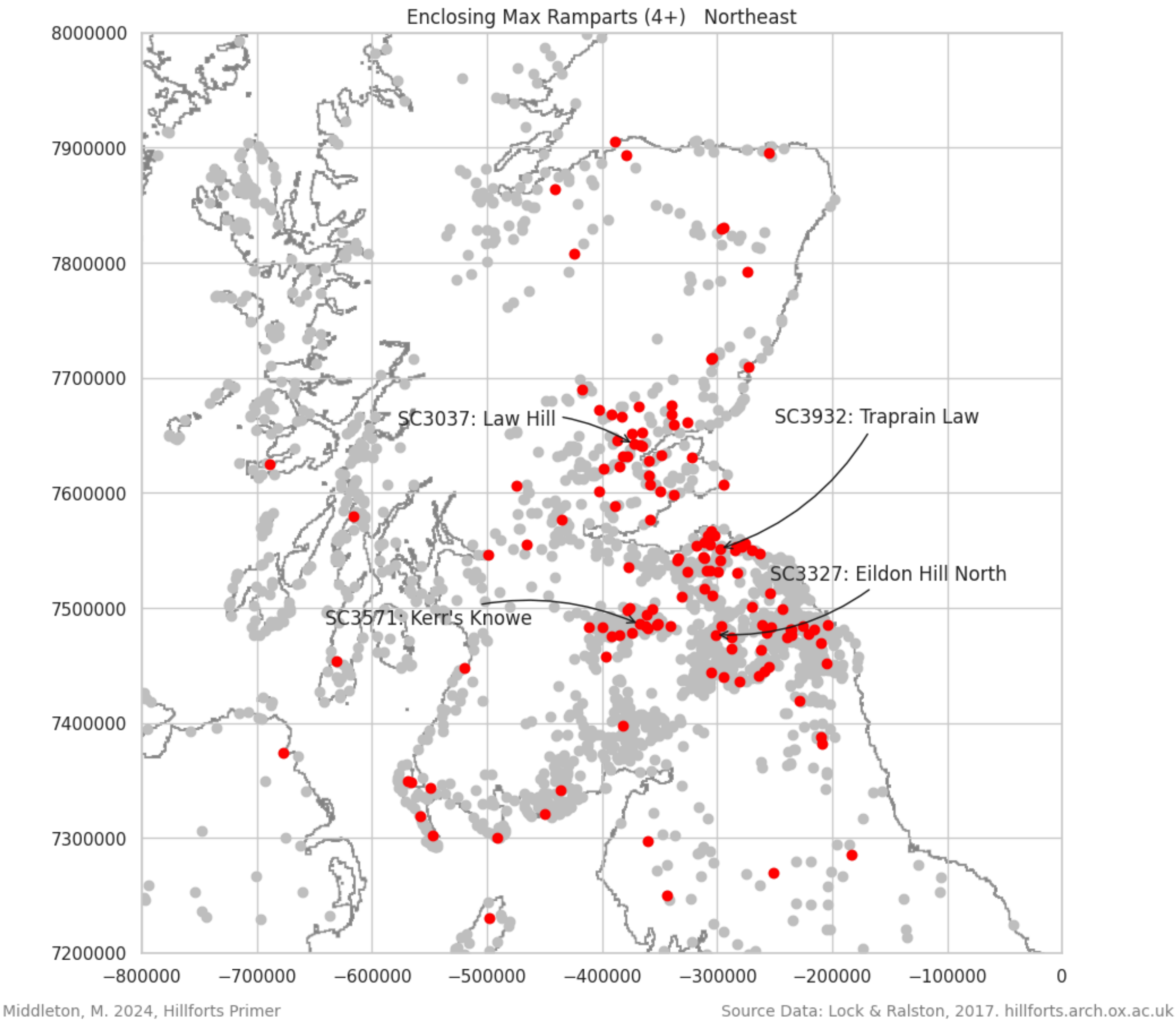
In the Northeast, hillforts with four or more ramparts cluster along the eastern fringe of the Southern Uplands, up and across western Fife and on into Perthshire, around Law Hill. There is also a cluster in the vicinity of Kerr's Hill, in south central Scotland. An interesting observation is that this class includes two of the more significant hillforts in southern Scotland, Traprain Law and Eildon Hill North. In East Lothian, the focus of the main cluster is around Traprain Law. It is important to note that this is an area which has undergone intensive aerial survey (See: Part 2: Cropmark Mapped) and there is thus a significant survey bias in this area.

In [ ]:

```
location_enclosing_data_ne = \
location_enclosing_data[location_enclosing_data['Location_Y'] > 7070000].copy()
location_enclosing_data_ne = \
location_enclosing_data_ne\
 [location_enclosing_data_ne['Location_X'] > -800000].copy()
outlier_ramparts_ne = \
location_enclosing_data_ne\
 [location_enclosing_data_ne['Enclosing_Max_Ramparts'] > 3].copy()
outlier_ramparts_ne['Enclosing_Max_Ramparts'] = "Yes"
```

In [ ]:

```
outlier_ramparts_stats_ne = \
plot_over_grey_north(outlier_ramparts_ne, 'Enclosing_Max_Ramparts', 'Yes', \
                     '(4+) - Northeast', 'Traprain')
```



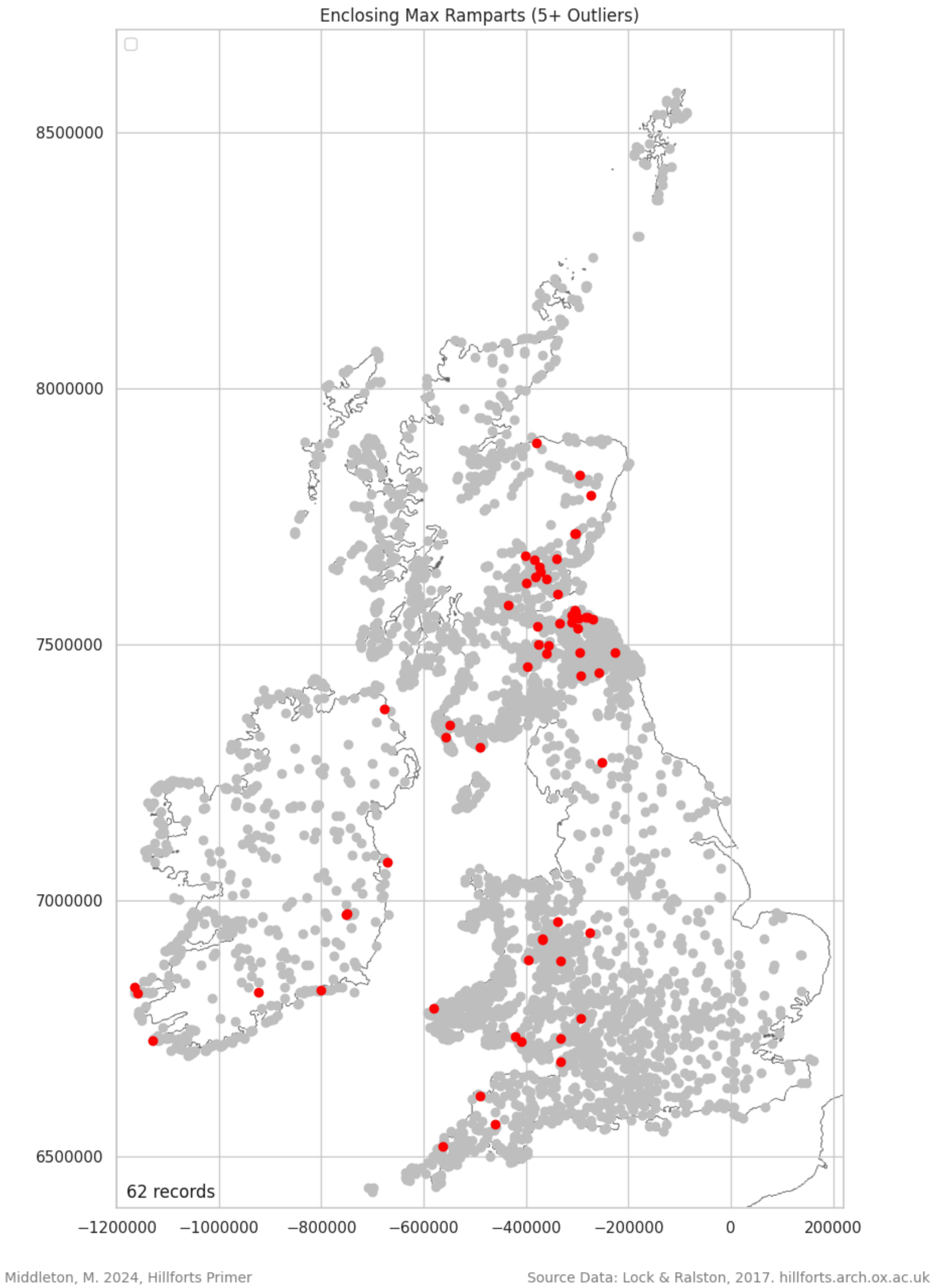
See: Ditches Mapped (4+ NE)

In [ ]:
```
# This code can be used to get details of hillforts within certain x and y coordinate ranges
# To use this code, first run the document using Runtime > Run all, then remove the '#' from the lines
# starting temp below. Once removed press the Run cell button, on this cell, to the left.
# Update the 'Location_X' & 'Location_Y' values as required.
# temp = pd.merge(name_and_number, outlier_ramparts_ne, left_index=True, right_index=True)
# temp = temp[temp['Location_X'].between(-300000, -200000)]
# temp = temp[temp['Location_Y'].between(7700000, 7800000)]
# temp
```

Ramparts Mapped (5+ Outliers)

West-Town, Waterfortd, in southeast Ireland, is the only Hillfort recorded as having 10 ramparts. Only 62 hillforts are recorded as having five or more ramparts and most are in the Northeast.

In [ ]:
```
outlier_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']>4].copy()
outlier_ramparts['Enclosing_Max_Ramparts'] = "Yes"
outlier_ramparts_stats = \
plot_over_grey(outlier_ramparts, 'Enclosing_Max_Ramparts', 'Yes', '(5+ Outliers)')
```




```
1.5%
```

In [ ]:

```
most_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']==10].copy()
most_ramparts = \
pd.merge(name_and_number, most_ramparts, left_index=True, right_index=True)
most_ramparts[["Main_Atlas_Number","Main_Display_Name","Enclosing_Area_1",\
               "Enclosing_Max_Ramparts","Enclosing_Ditches_Number"]]
```

Out[ ]:

	Main_Atlas_Number	Main_Display_Name	Enclosing_Area_1	Enclosing_Max_Ramparts	Enclosing_Ditches_Number
1015	1043	West-Town, Waterford (Great Island)	0.71	10.0	1.0

Ramparts by Region

Most hillforts have one to two ramparts. In the north of Ireland this is most likely to be one and, in the Northeast, it is most likely to range between one to three.

In [ ]:
```
location_enclosing_data_ne = \
pd.merge(north_east.reset_index(), enclosing_numeric_data, left_on='uid', \
         right_index=True)
location_enclosing_data_ne = \
pd.merge(name_and_number, location_enclosing_data_ne, left_index=True, \
         right_on='uid')
```

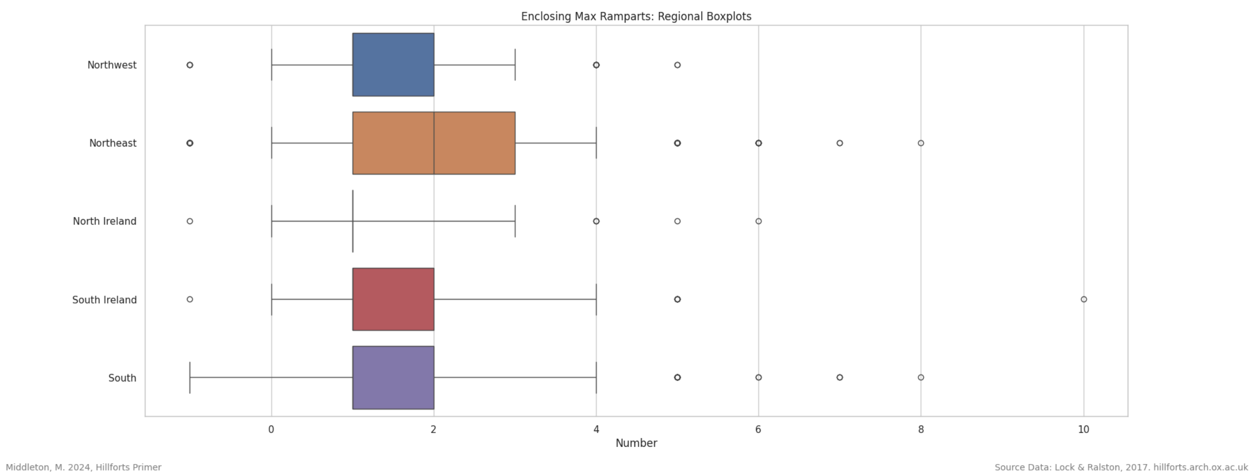
In [ ]:
```
location_enclosing_data_nw = \
pd.merge(north_west.reset_index(), enclosing_numeric_data, left_on='uid', \
         right_index=True)
location_enclosing_data_nw = \
pd.merge(name_and_number, location_enclosing_data_nw, left_index=True, \
         right_on='uid')
```

In [ ]:
```
location_enclosing_data_ireland_n = \
pd.merge(north_ireland.reset_index(), enclosing_numeric_data, left_on='uid', \
         right_index=True)
location_enclosing_data_ireland_n = \
pd.merge(name_and_number, location_enclosing_data_ireland_n, left_index=True, \
         right_on='uid')
```

In [ ]:
```
location_enclosing_data_ireland_s = \
pd.merge(south_ireland.reset_index(), enclosing_numeric_data, left_on='uid', \
         right_index=True)
location_enclosing_data_ireland_s = \
pd.merge(name_and_number, location_enclosing_data_ireland_s, left_index=True, \
         right_on='uid')
```

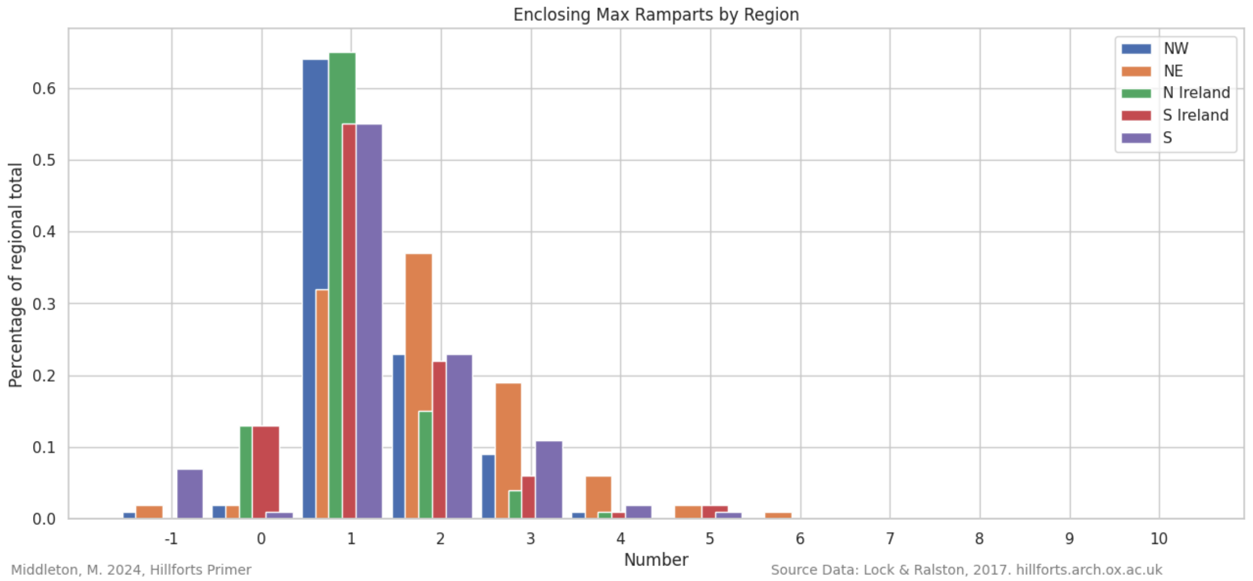
In [ ]:
```
location_enclosing_data_south = \
pd.merge(south, enclosing_numeric_data, left_on='uid', right_index=True)
location_enclosing_data_south = \
pd.merge(name_and_number, location_enclosing_data_south, left_index=True, \
         right_on='uid')
```

In [ ]:
```
regional_dict = \
 {'Northwest': location_enclosing_data_nw\
  ['Enclosing_Max_Ramparts'], 'Northeast': location_enclosing_data_ne\
   ['Enclosing_Max_Ramparts'], 'North Ireland': location_enclosing_data_ireland_n\
    ['Enclosing_Max_Ramparts'], 'South Ireland': location_enclosing_data_ireland_s\
     ['Enclosing_Max_Ramparts'], 'South': location_enclosing_data_south\
      ['Enclosing_Max_Ramparts']}
plot_data = pd.DataFrame.from_dict(regional_dict)
plt.figure(figsize=(20,8))
ax = sns.boxplot(data=plot_data, orient="h" , whis=[2.2, 97.8], showfliers=True);
add_annotation_plot(ax)
ax.set_xlabel('Number')
title = 'Enclosing_Max_Ramparts: Regional Boxplots'
plt.title(get_print_title(title))
save_fig(title)
plt.show()
```



The Northeast is noticeable in that hillforts with a single rampart are proportionally far less than in other areas. Similarly, the Northeast is more likely to have forts with two, three or four ramparts than other regions. Hillforts in the remaining regions have quite similar proportions of ramparts apart from forts with no ramparts, which are more common in Ireland.

In [ ]:
```
plot_feature_by_region(location_enclosing_data_nw,
                       location_enclosing_data_ne,
                       location_enclosing_data_ireland_n,
                       location_enclosing_data_ireland_s,
                       location_enclosing_data_south,
                       'Enclosing_Max_Ramparts',
                       'Enclosing Max Ramparts by Region', 12)
```

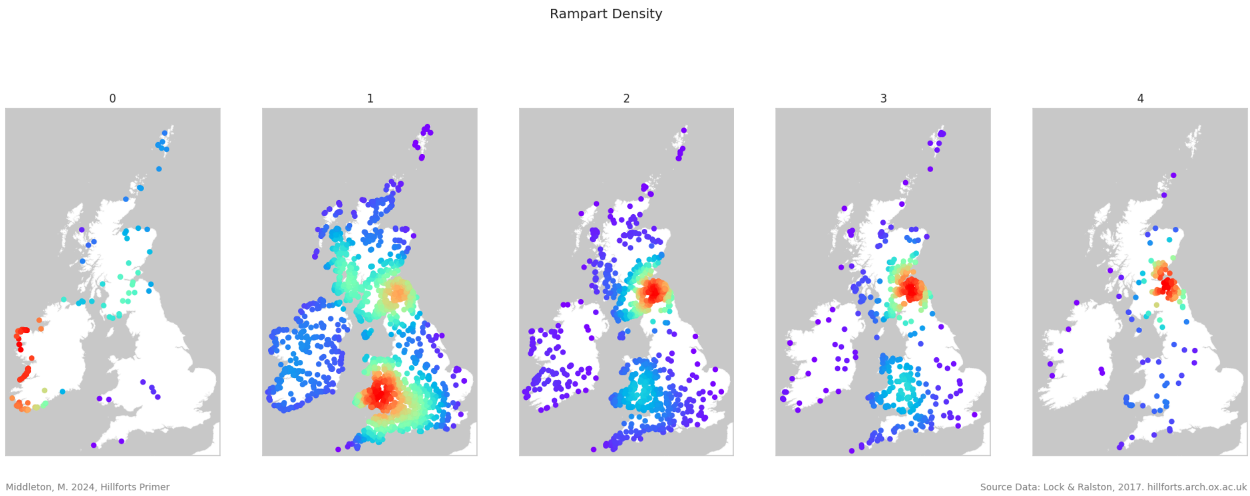


Ramparts Summary

Ramparts show four distinct clusters:

- Hillforts without ramparts along the west coast of Ireland
- Hillforts with one rampart in the Northwest
- Hillforts with mostly one rampart in southern Wales and south-central England (occasionally two or three)
- Hillforts with one or more ramparts in the Northeast

In [ ]:
```
plot_density_over_grey_five(zero_ramparts, one_rampart, two_ramparts, \
                            three_ramparts, four_ramparts, 'Rampart Density')
```



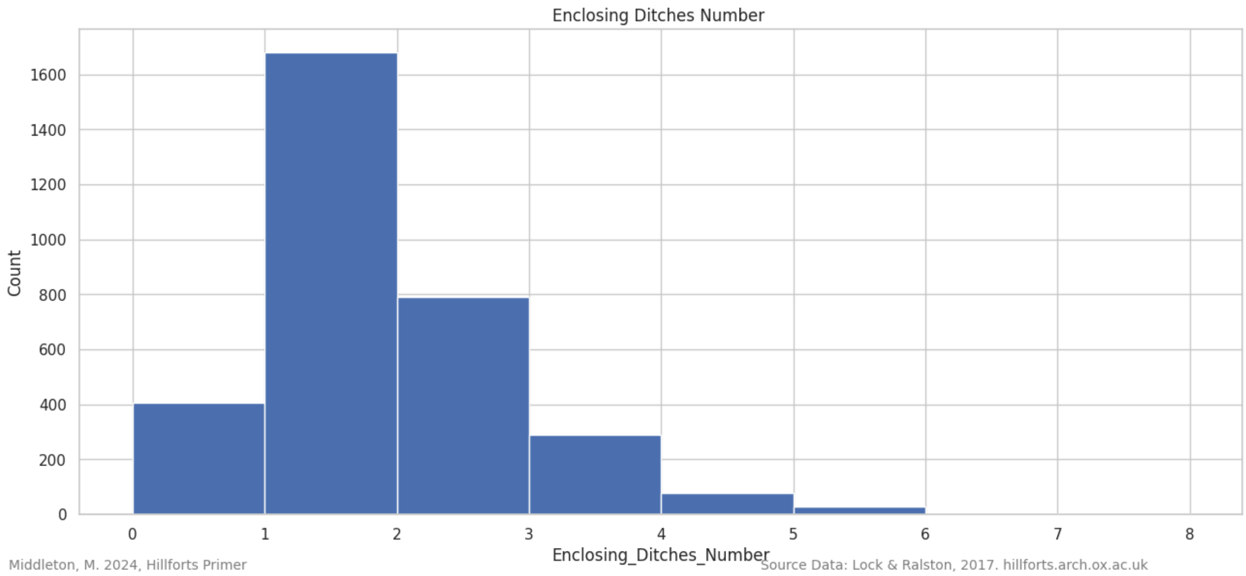
Ditches Plotted

868 hillforts (20.93%) have no information regarding ditches. These are mostly in Scotland. Of those that are recorded, 1681 (40.54%) have one ditch; 789 (19.3%) have two ditches and 406 (9.79%) no ditches. Because of the lack of recording in Scotland and what looks like a survey bias in the forts with no recorded ditches, caution should be taken when interpreting these distributions. The fort with the most ditches is Trevelgue Head in Cornwall which has eight.

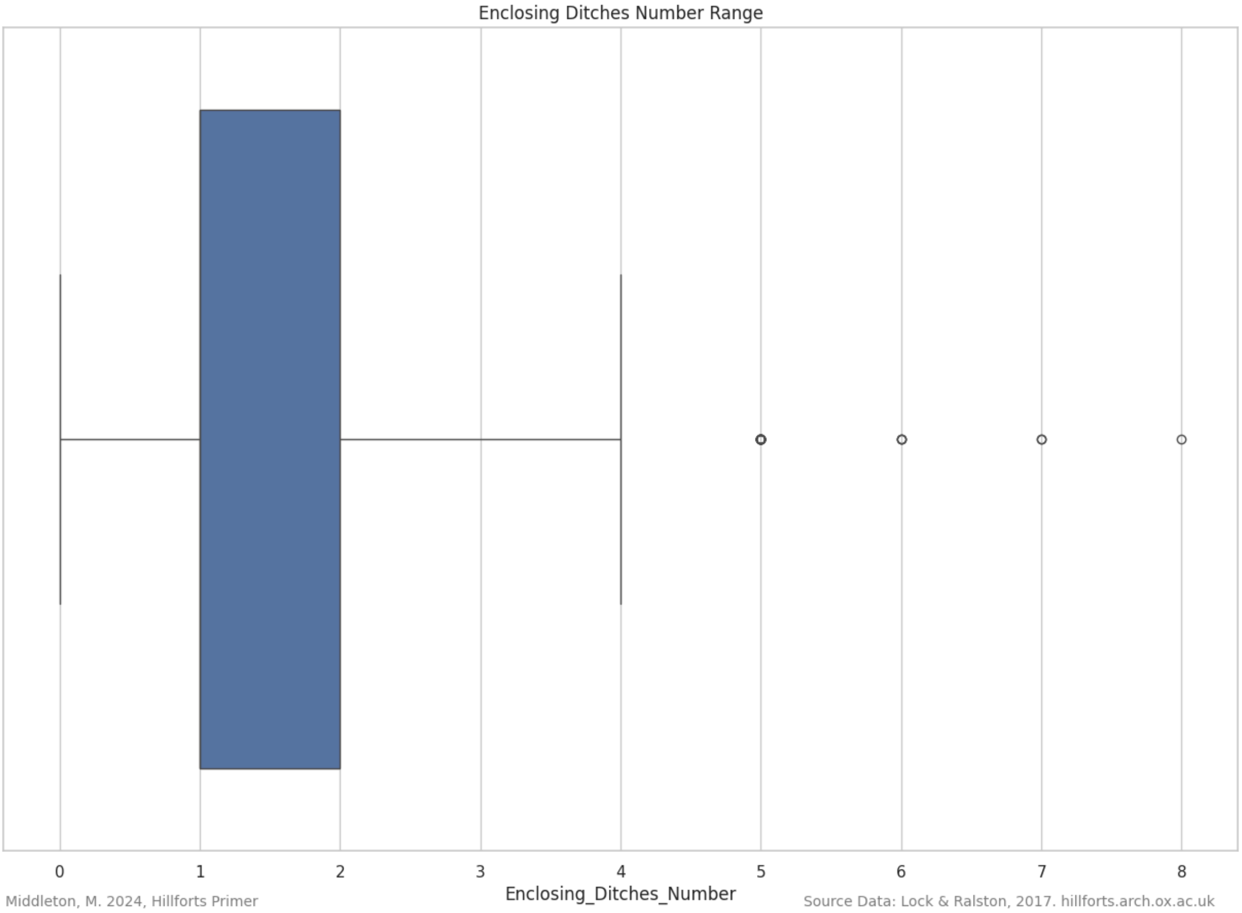
In [ ]:
```
ditches_location_enc_data = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']>=0]
ditches_location_enc_data['Enclosing_Ditches_Number'].value_counts().sort_index()
```

Out[ ]:
```
0.0     406
1.0    1681
2.0     789
3.0     289
4.0      79
5.0      29
6.0       3
7.0       2
8.0       1
Name: Enclosing_Ditches_Number, dtype: int64
```

In [ ]:
```
plot_bar_chart_numeric(ditches_location_enc_data, 1, \
                       'Enclosing_Ditches_Number', 'Count', \
                       'Enclosing_Ditches_Number', \
                       int(ditches_location_enc_data\
                        ['Enclosing_Ditches_Number'].max()))
```



In [ ]:
```
ditches_data = \
plot_data_range(ditches_location_enc_data['Enclosing_Ditches_Number'].\
                reset_index(drop = True), 'Enclosing_Ditches_Number', "h")
```



In [ ]:
```
ditches_data
```

Out[ ]:
```
[0.0, 1.0, 1.0, 2.0, 4.0]
```

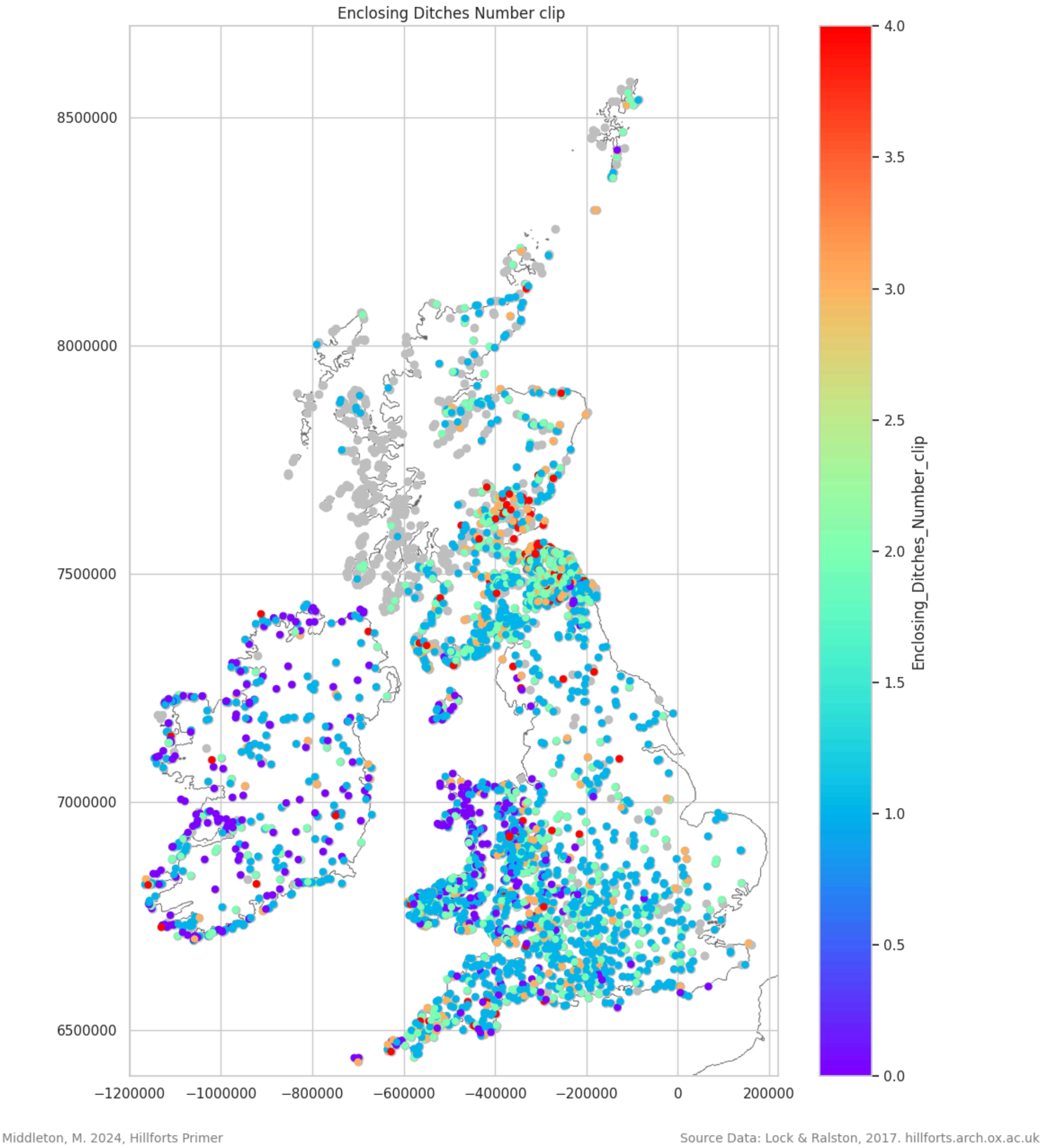
Ditches Clipped Mapped

As with ramparts, the combined plot is difficult to read so each value will be reviewed individually. There is a noticeable survey bias in the data across Scotland. There are very few records in the Northwest.

In [ ]:
```
ditches_clip = ditches_location_enc_data.copy()
ditches_clip['Enclosing_Ditches_Number_clip'] = \
ditches_clip['Enclosing_Ditches_Number'].\
clip(ditches_clip['Enclosing_Ditches_Number'], ditches_data[-1], axis=0)
ditches_clip['Enclosing_Ditches_Number_clip'].value_counts().sort_index()
```

Out[ ]:
```
0.0     406
1.0    1681
2.0     789
3.0     289
4.0     114
Name: Enclosing_Ditches_Number_clip, dtype: int64
```

In [ ]:
```
plot_type_values(ditches_clip, \
                 'Enclosing_Ditches_Number_clip', \
                 'Enclosing_Ditches_Number_clip')
```

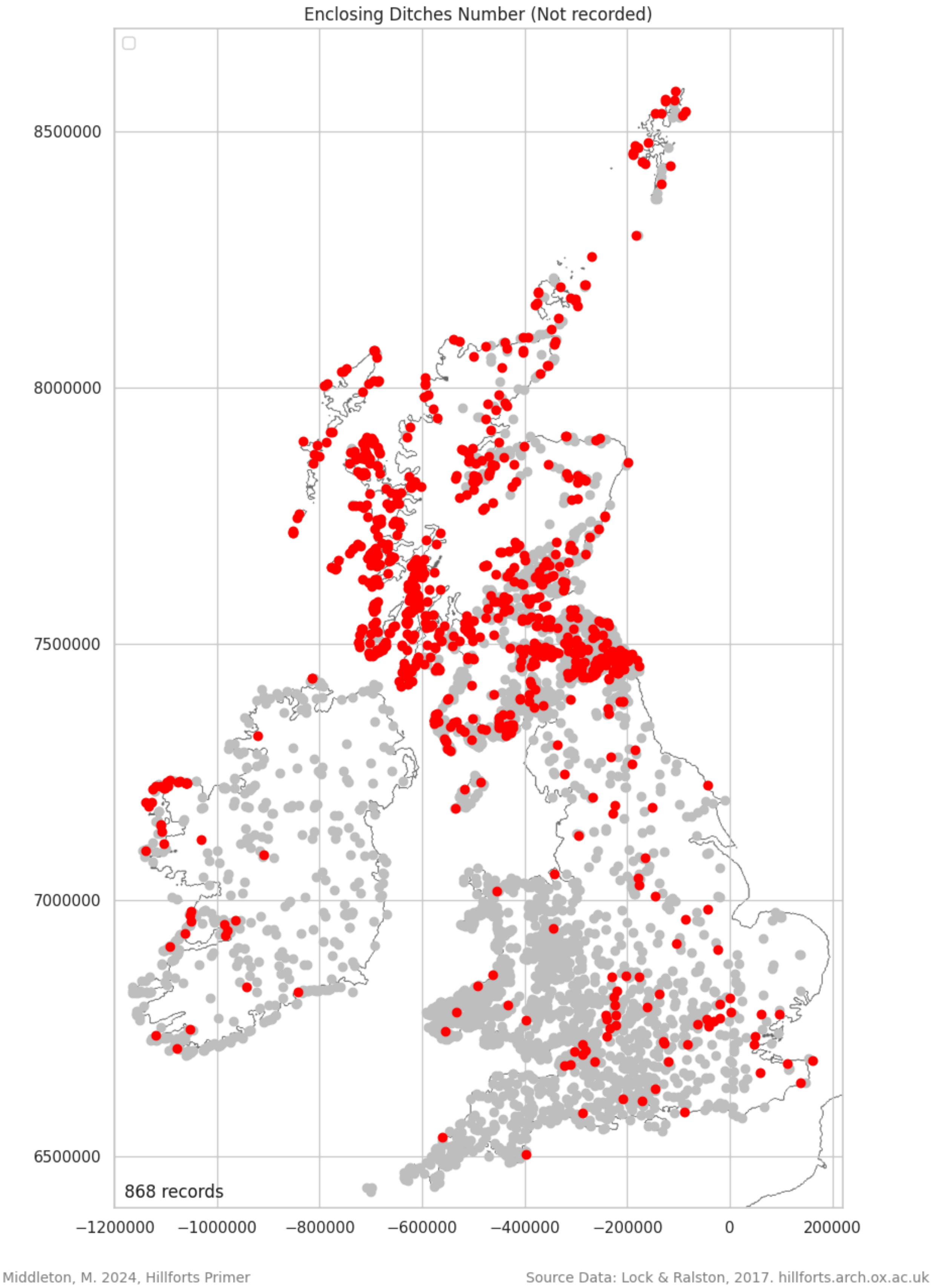


See: Enclosing Ditches Density Mapped

Ditches Mapped (Not Recorded)

A remarkable 868 (20.93%) of hillforts have no record of the presence of a ditch. Most of these are in Scotland. This is a survey bias in the data. It is likely that this is partly due to a practice of not recording diches being used as a shorthand for there not being ditches. The hard geology, of the north, and surveyors thinking it is obvious combining to create ambiguous records. This would be an obvious area where a study could rapidly improve this section of the atlas.

In [ ]:
```
nan_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']==-1].copy()
nan_ditches['Enclosing_Ditches_Number'] = "Yes"
nan_ditches_stats = plot_over_grey(nan_ditches, 'Enclosing_Ditches_Number', \
                                   'Yes', '(Not recorded)')
```



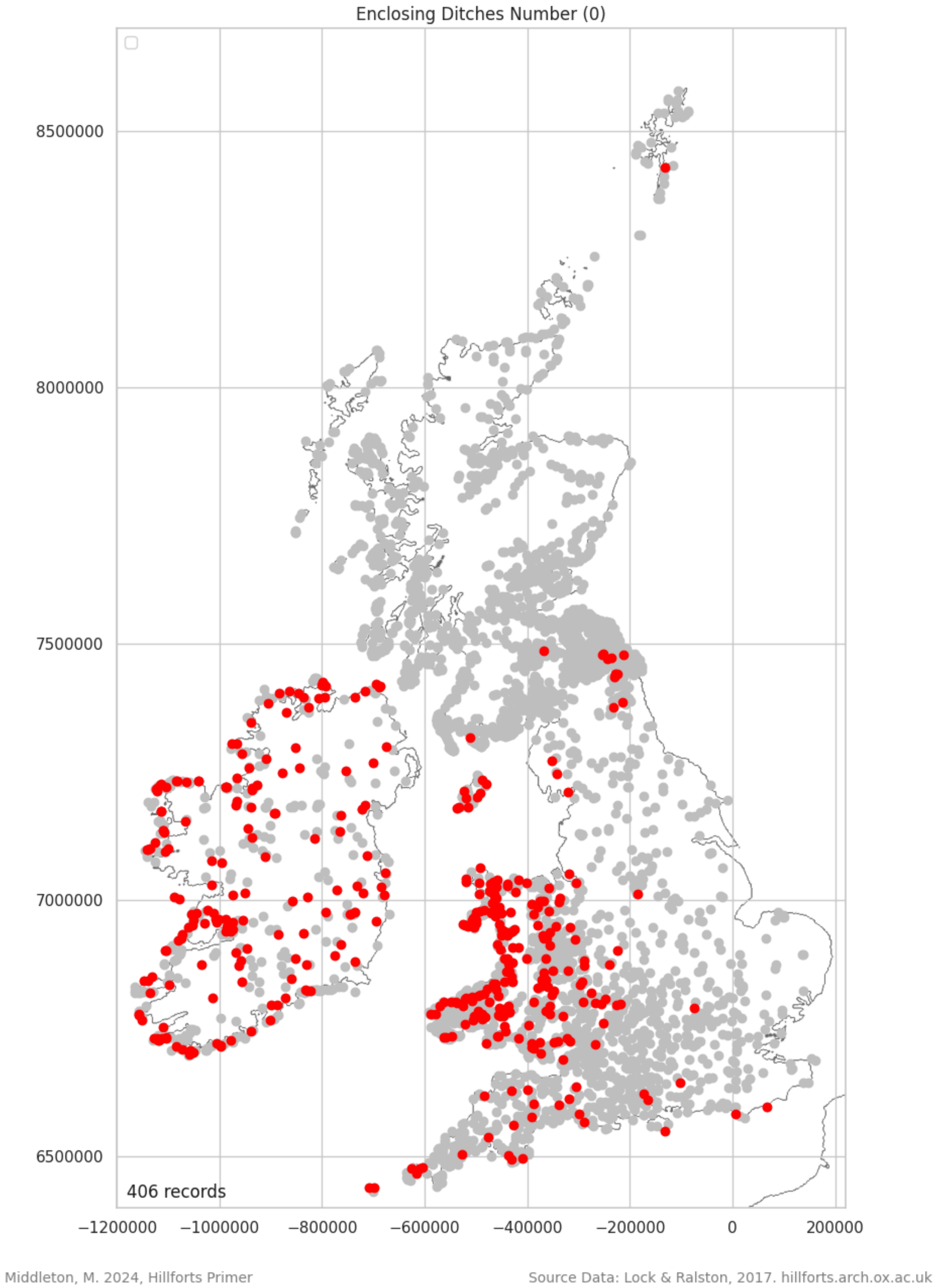

```
20.93%
```

Ditches Mapped (0)

With the survey bias across Scotland in mind, see: Ditches Mapped (Not Recorded), the distribution of forts with no diches is very much over Wales and Ireland.

In [ ]:
```
zero_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']==0].copy()
zero_ditches['Enclosing_Ditches_Number'] = "Yes"
```

```
zero_ditches_stats = plot_over_grey(zero_ditches, 'Enclosing_Ditches_Number', \
                                    'Yes', '(0)')
```

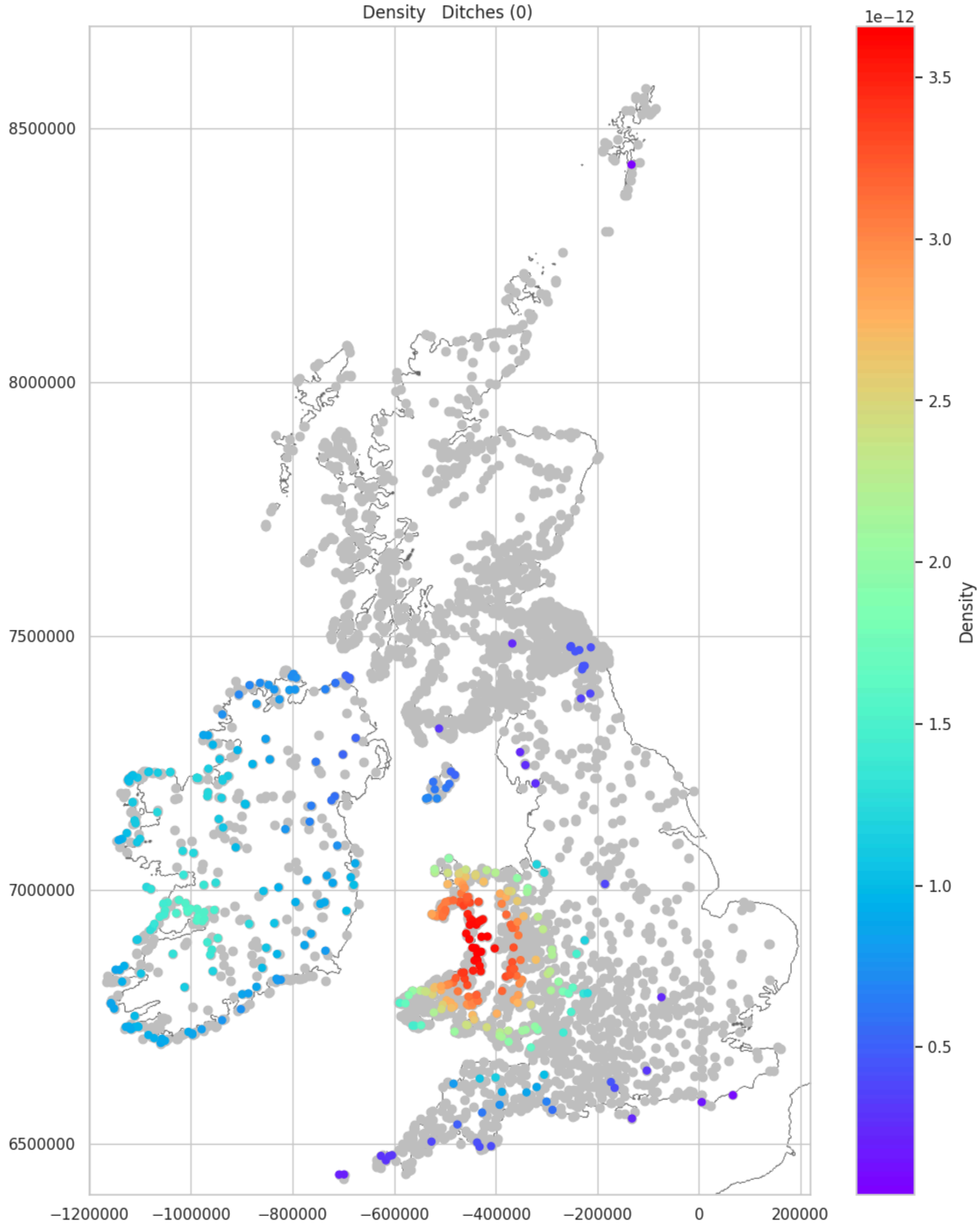



```
9.79%
```

Ditches Density Mapped (0)

The most intense cluster, of hillforts with no ditches, is over the western fringe of the Cambrian Mountains.

In [ ]:
```
plot_density_over_grey(zero_ditches_stats, 'Ditches (0)')
```



Middleton, M. 2024, Hillforts Primer

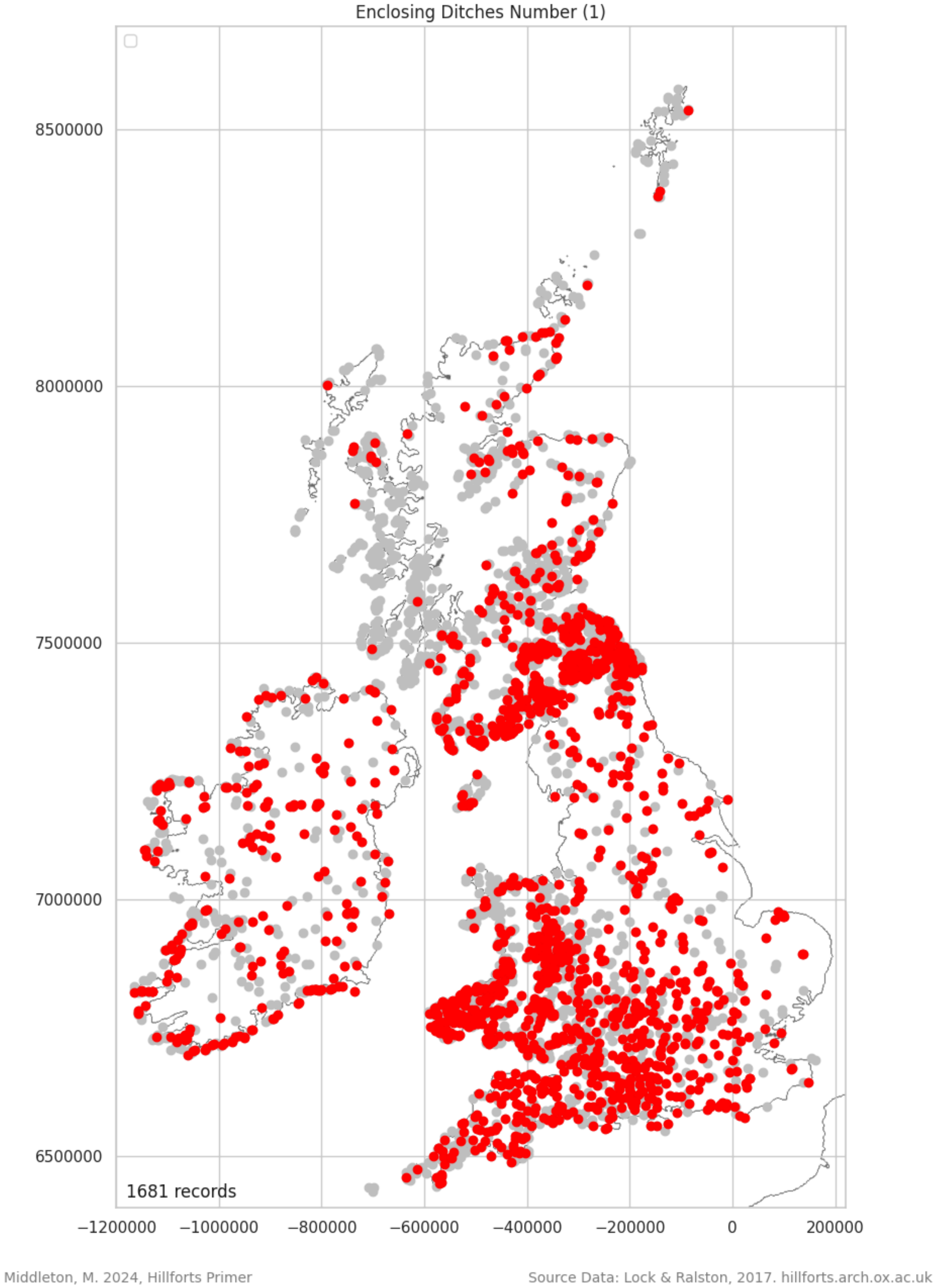
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ditchs Mapped (1)

The distribution of single ditch hillforts is much more uniform. 1681 (40.84%) of hillforts fall into this class. Again, see Ditches Mapped (Not Recorded).

In [ ]:

```
one_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']==1].copy()
one_ditches['Enclosing_Ditches_Number'] = "Yes"
one_ditches_stats = plot_over_grey(one_ditches, 'Enclosing_Ditches_Number', \
                                   'Yes', '(1)')
```

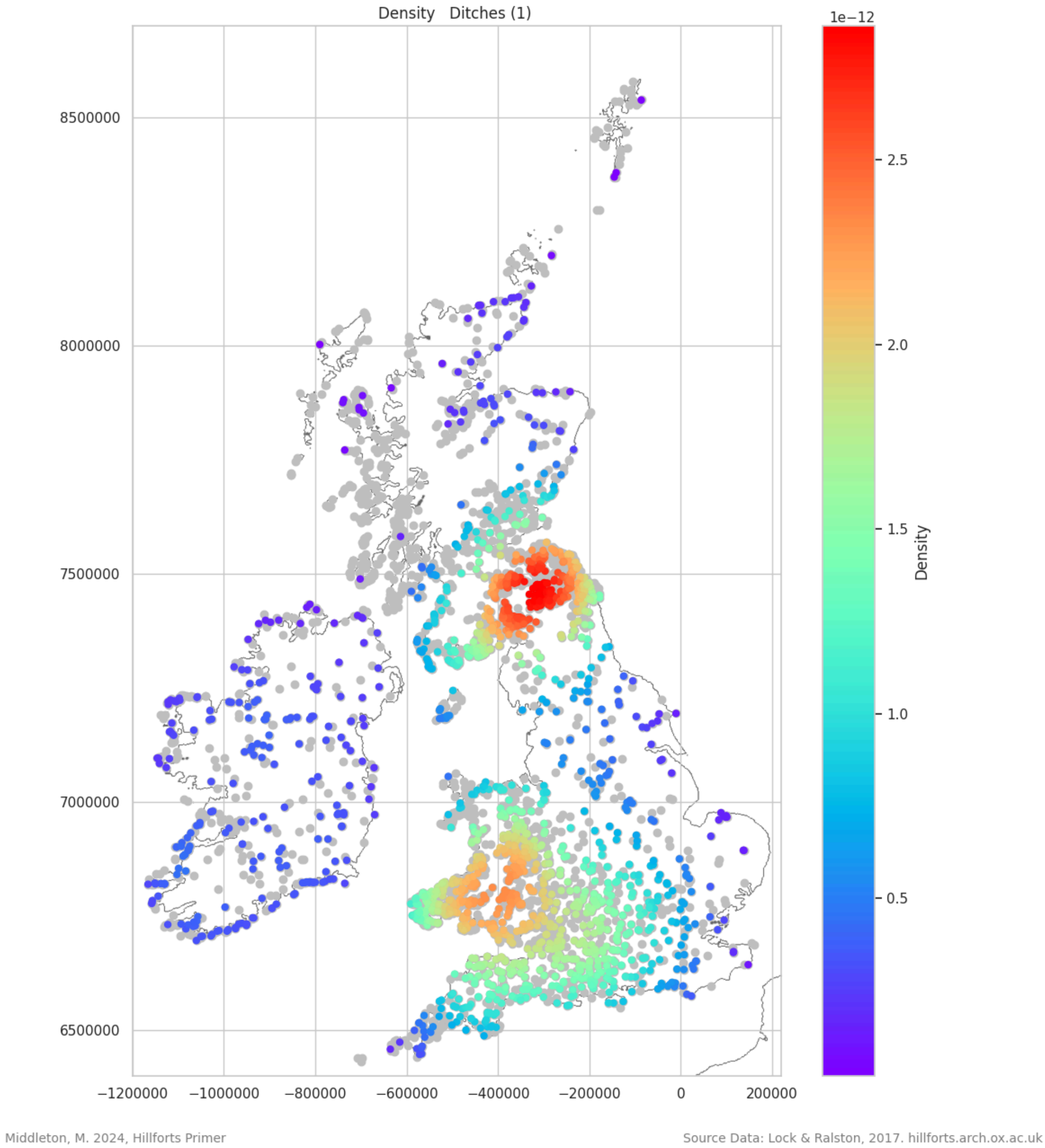



```
40.54%
```

Ditchs Density Mapped (1)

The density of hillforts with one ditch is split between two clusters. The most intense is over the Northeast with the other focussed over the southern end of the Cambrian Mountains and into south, central England. It is interesting to compare this with Ramparts Density Mapped (1) where the main focus was far more intense over Wales and far less intense over the Northeast.

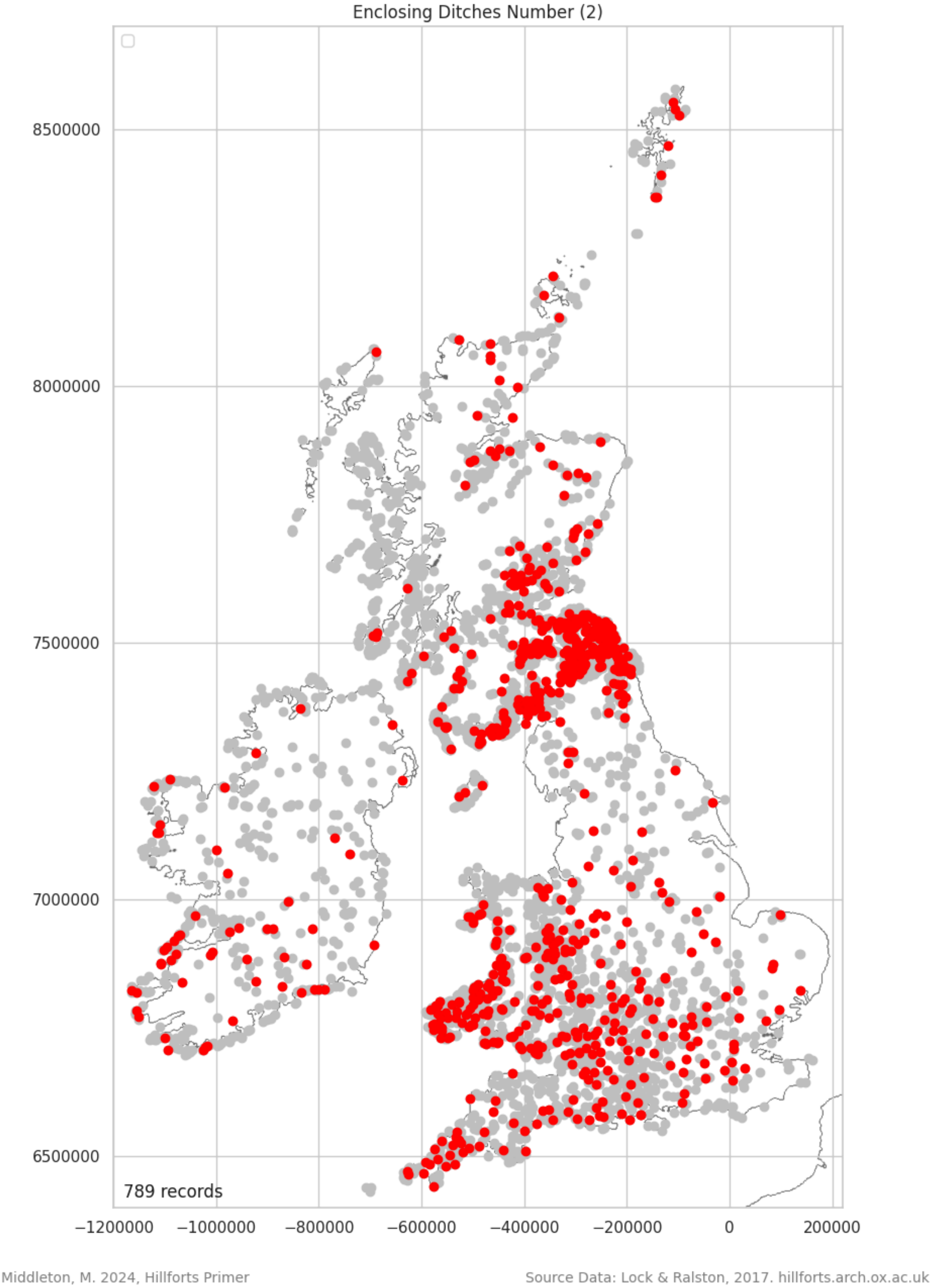
In [ ]:
```
plot_density_over_grey(one_ditches_stats, 'Ditches (1)')
```



Ditches Mapped (2)

789 (19.03%) of hillforts are recorded as having two ditches. These are mostly distributed over the South, Northeast and southern Ireland. Note Ditches Mapped (Not Recorded).

In [ ]:
```
two_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']==2].copy()
two_ditches['Enclosing_Ditches_Number'] = "Yes"
two_ditches_stats = plot_over_grey(two_ditches, 'Enclosing_Ditches_Number', \
                                   'Yes', '(2)')
```

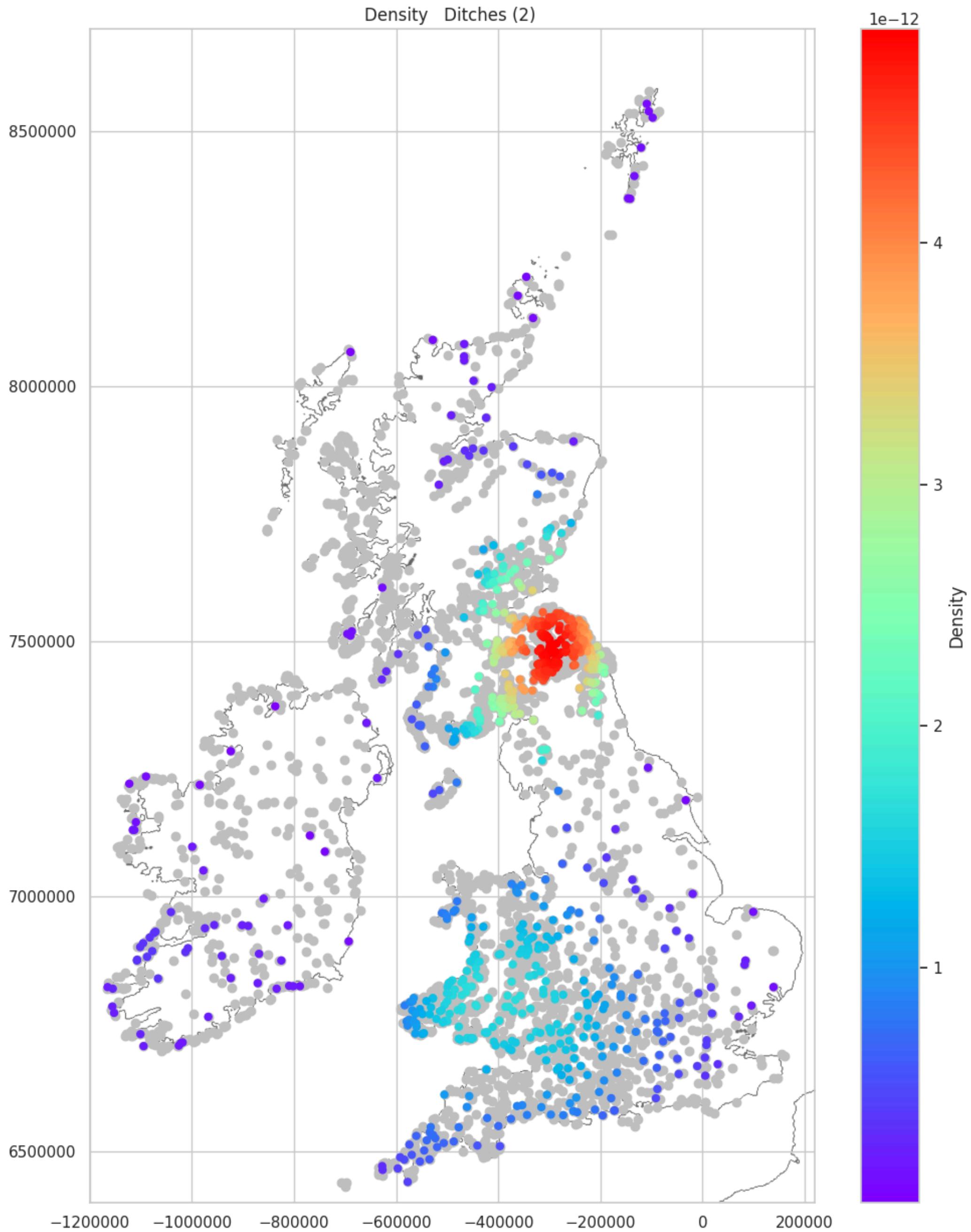



```
19.03%
```

Ditches Density Mapped (2)

As was seen with ramparts with more than one rampart, the main cluster of forts with more than one ditch is in the Northeast. A secondary cluster can be seen over the southern Cambrian Mountains and there is a peppering of these forts over southern and western Ireland.

In [ ]:
```
plot_density_over_grey(two_ditches_stats, 'Ditches (2)')
```



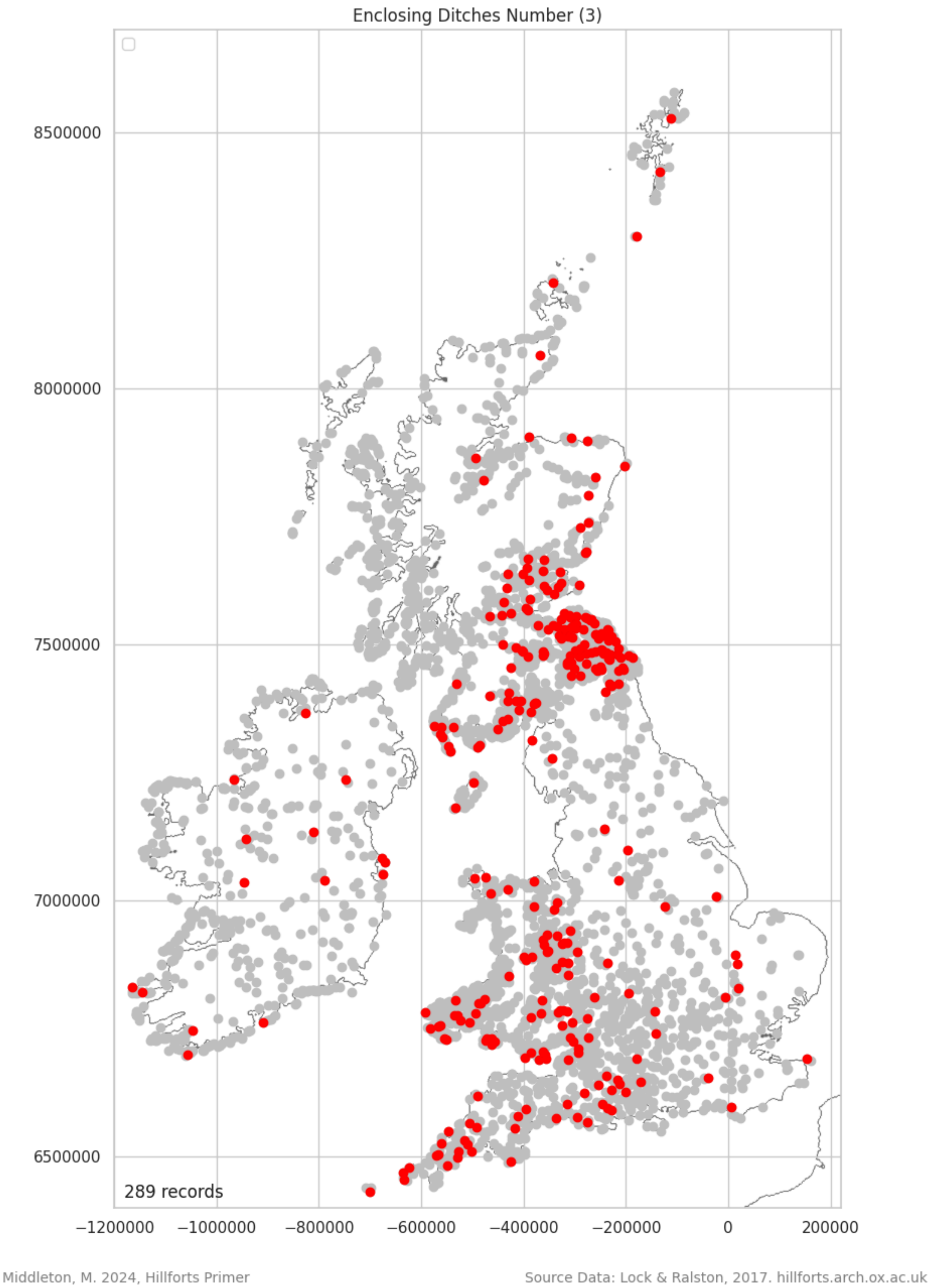
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ditches Mapped (3)

The distribution of hillforts with three ditches is focussed in the Northeast. Note Ditches Mapped (Not Recorded).

In [ ]:
```
three_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']==3].copy()
three_ditches['Enclosing_Ditches_Number'] = "Yes"
three_ditches_stats = plot_over_grey(three_ditches, 'Enclosing_Ditches_Number', \
                                     'Yes', '(3)')
```

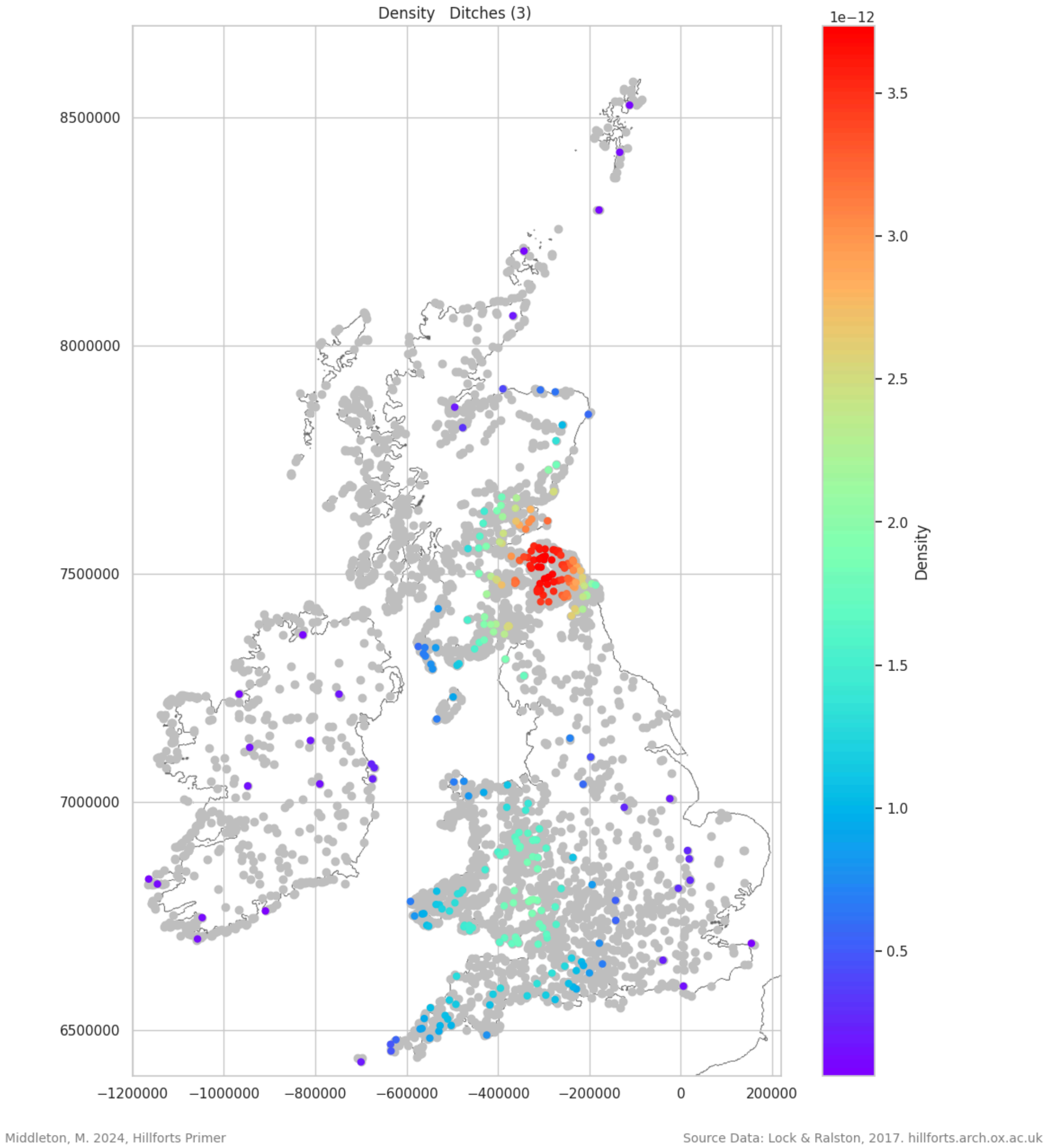



```
6.97%
```

Ditches Density Mapped (3)

The main cluster of three ditch hillforts is over the Northeast. A secondary cluster runs down the eastern fringe of the Cambrian Mountains.

In [ ]:
```
plot_density_over_grey(three_ditches_stats, 'Ditches (3)')
```

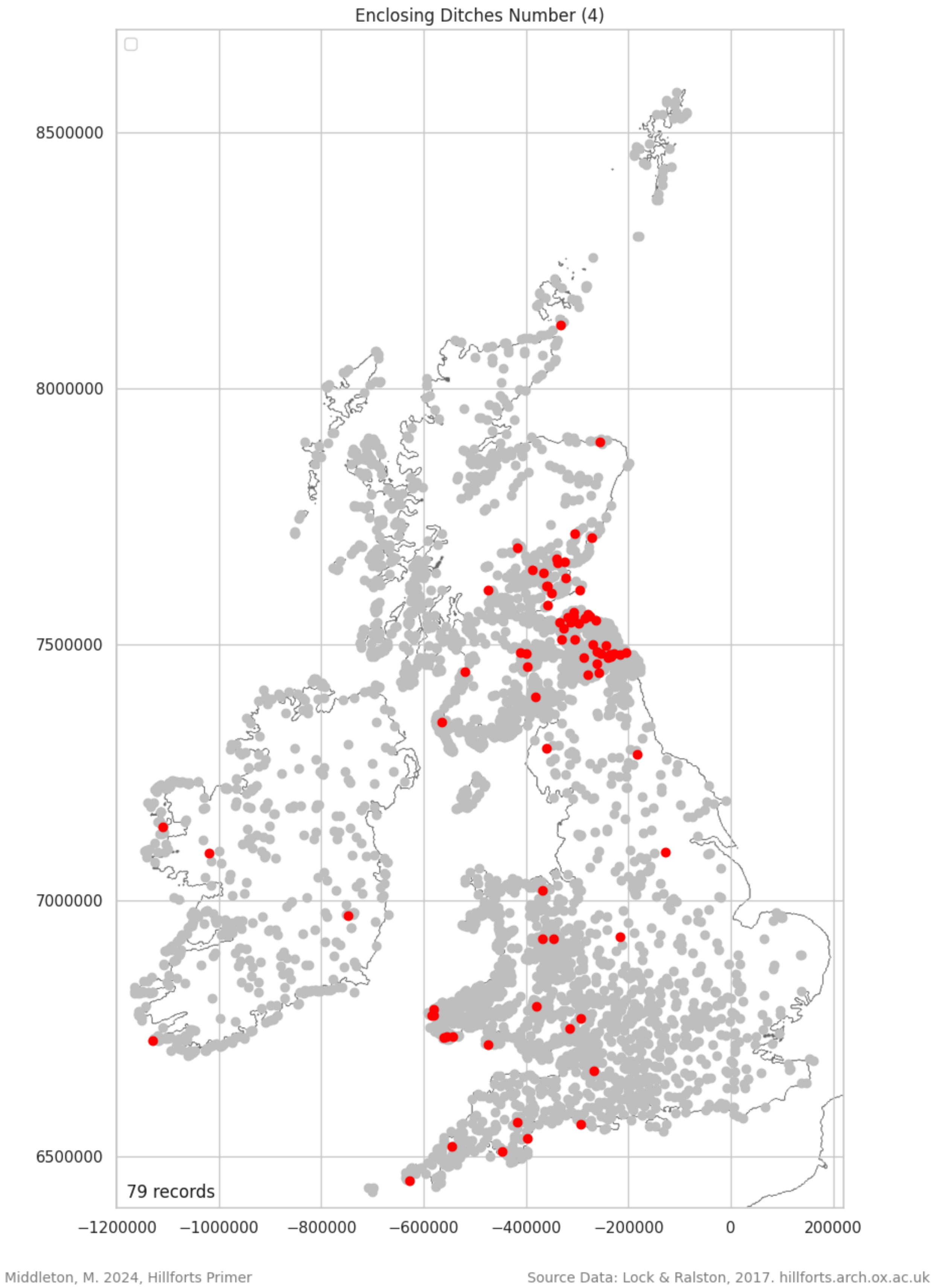


Ditches Mapped (4)

The focus for four ditch hillforts is in the Northeast over East Lothian. This is in line with what was seen for ramparts. See Ramparts Mapped (4). Note Ditches Mapped (Not Recorded).

In [ ]:

```
four_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']==4].copy()
four_ditches['Enclosing_Ditches_Number'] = "Yes"
four_ditches_stats = plot_over_grey(four_ditches, 'Enclosing_Ditches_Number', \
                                    'Yes', '(4)')
```




```
1.9%
```

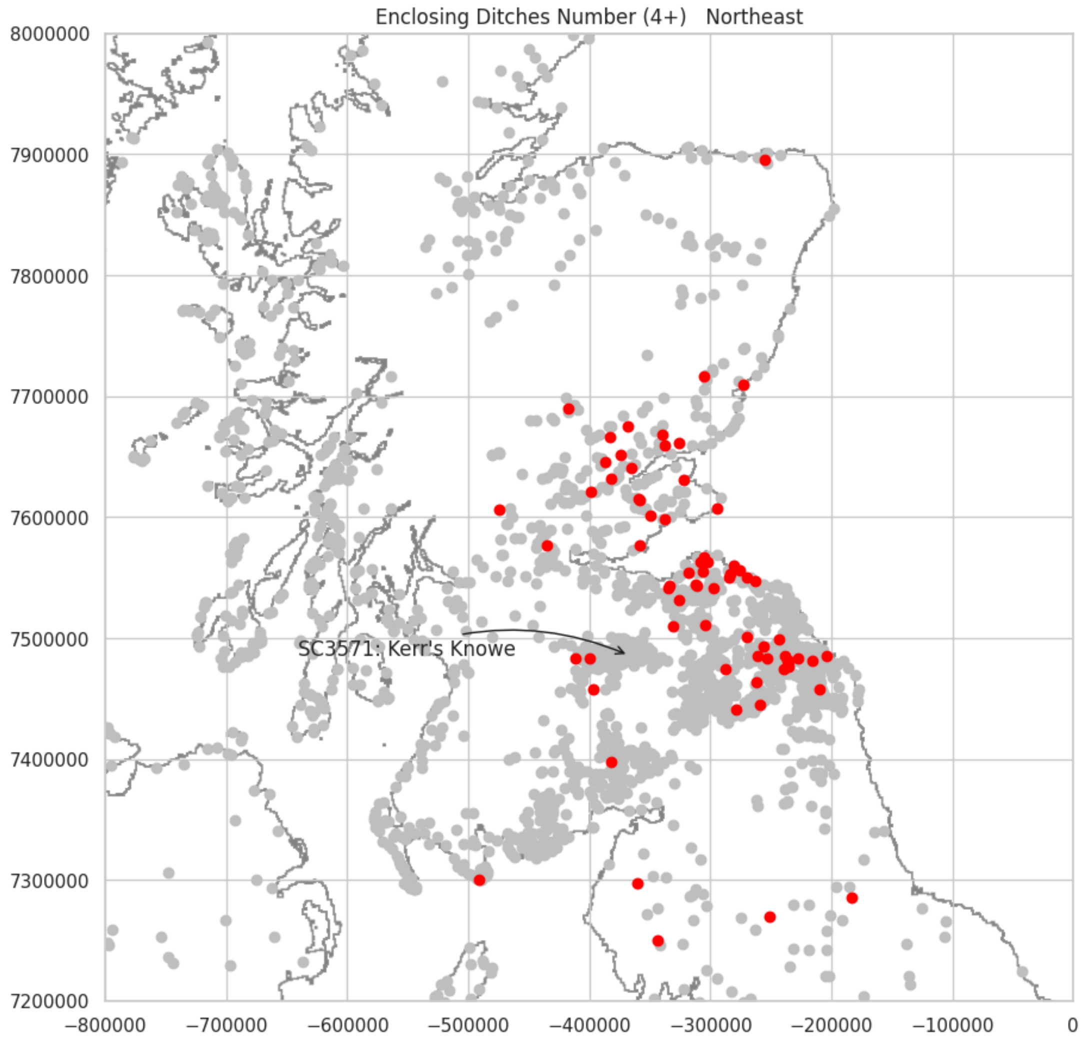
Ditches Mapped (4+ NE)

The concentration of hillforts, in the Northeast, with four or more ditches has a similar cluster to Ramparts Mapped (4+ NE), running along the eastern fringe of the Southern Uplands, up and across Fife and on into Perthshire and Angus. It does not include the cluster seen in the ramparts data around Kerr's Hill and neither Traprain Law or Eildon Hill North have four or more ditches.

In [ ]:
```
outlier_ditches_ne = \
location_enclosing_data_ne[location_enclosing_data_ne\
```

```
 ['Enclosing_Ditches_Number'] > 3].copy()
outlier_ditches_ne['Enclosing_Ditches_Number'] = "Yes"
```

In [ ]:
```
outlier_ditches_stats_ne = \
plot_over_grey_north(outlier_ditches_ne, 'Enclosing_Ditches_Number', 'Yes', \
                     '(4+) - Northeast', 'Kerr')
```



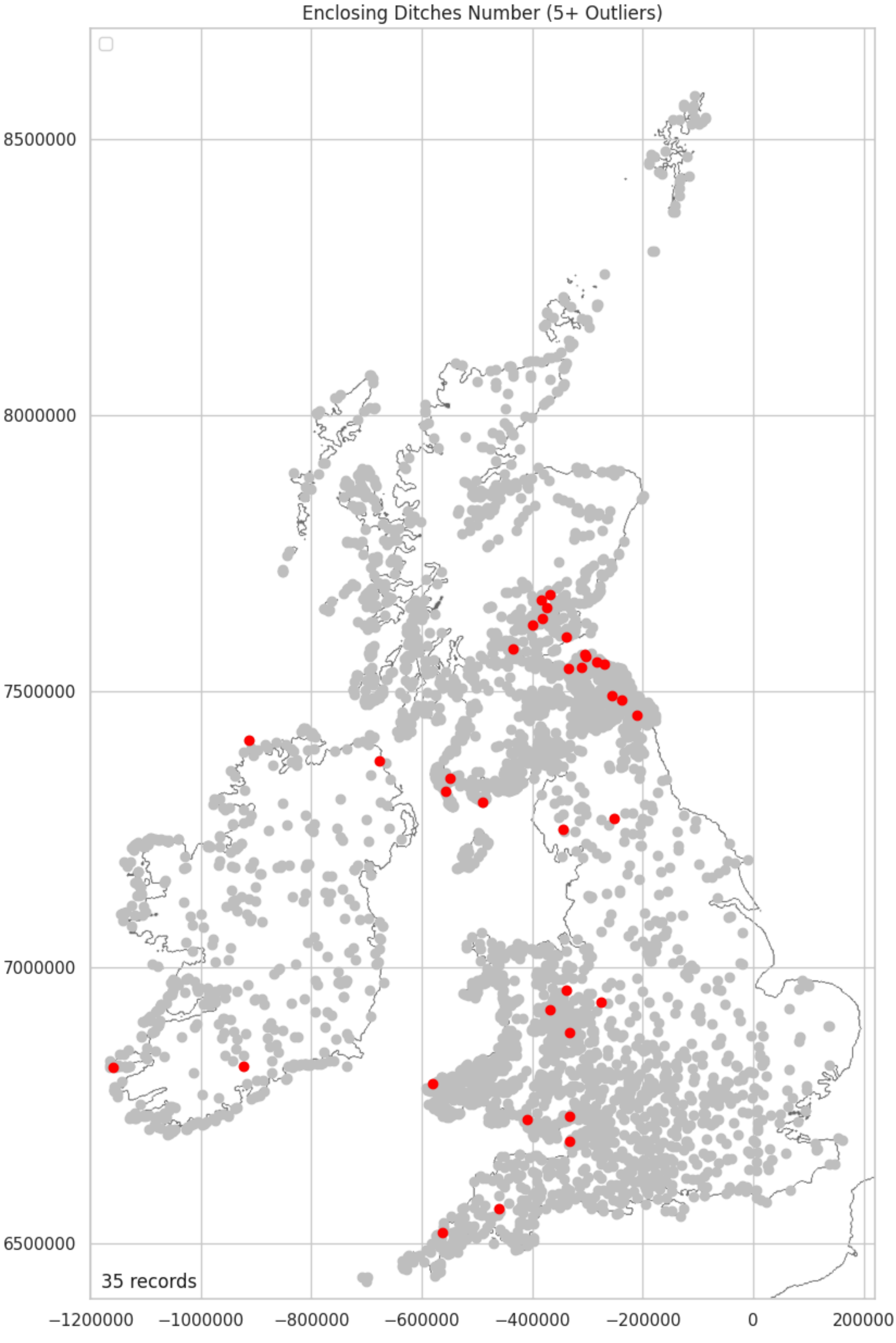
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Ditches Mapped (5+ Outliers)

There are 35 hillforts with five or more ditches. Note Ditches Mapped (Not Recorded).

In [ ]:
```
outlier_ditches = \
location_enclosing_data\
 [location_enclosing_data['Enclosing_Ditches_Number']>4].copy()
outlier_ditches['Enclosing_Ditches_Number'] = "Yes"
outlier_ditches_stats = plot_over_grey(outlier_ditches, \
                                       'Enclosing_Ditches_Number', 'Yes', \
                                       '(5+ Outliers)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

0.84%

In [ ]:

```
most_ditches = \
location_enclosing_data[location_enclosing_data\
 ['Enclosing_Ditches_Number']==8].copy()
most_ditches = \
pd.merge(name_and_number, most_ditches, left_index=True, right_index=True)
most_ditches[["Main_Atlas_Number","Main_Display_Name","Enclosing_Area_1",\
              "Enclosing_Max_Ramparts","Enclosing_Ditches_Number"]]
```

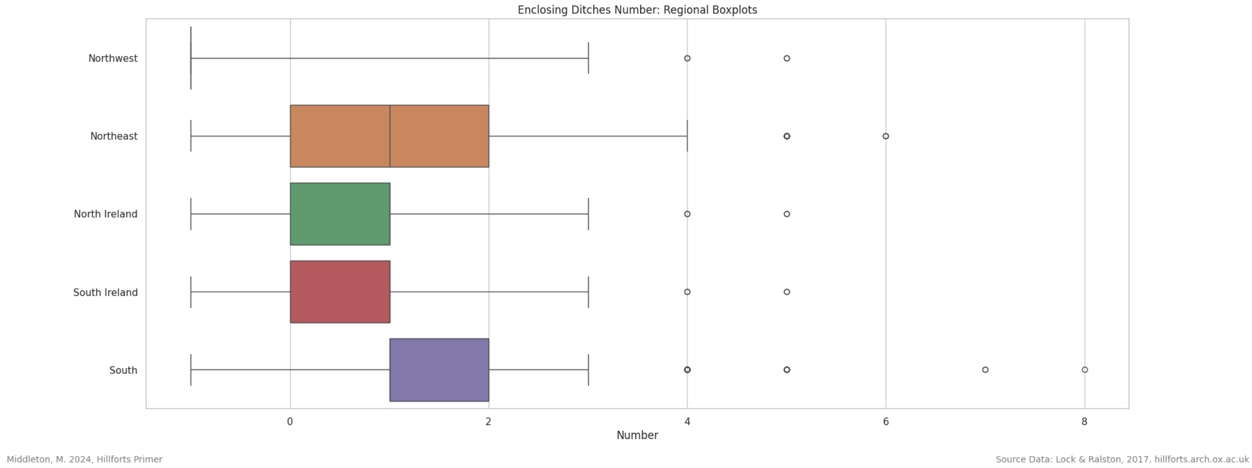
Out[ ]:

	Main_Atlas_Number	Main_Display_Name	Enclosing_Area_1	Enclosing_Max_Ramparts	Enclosing_Ditches_Number
634	656	Trevelgue Head, Cornwall	3.2	8.0	8.0

Ditches by Region

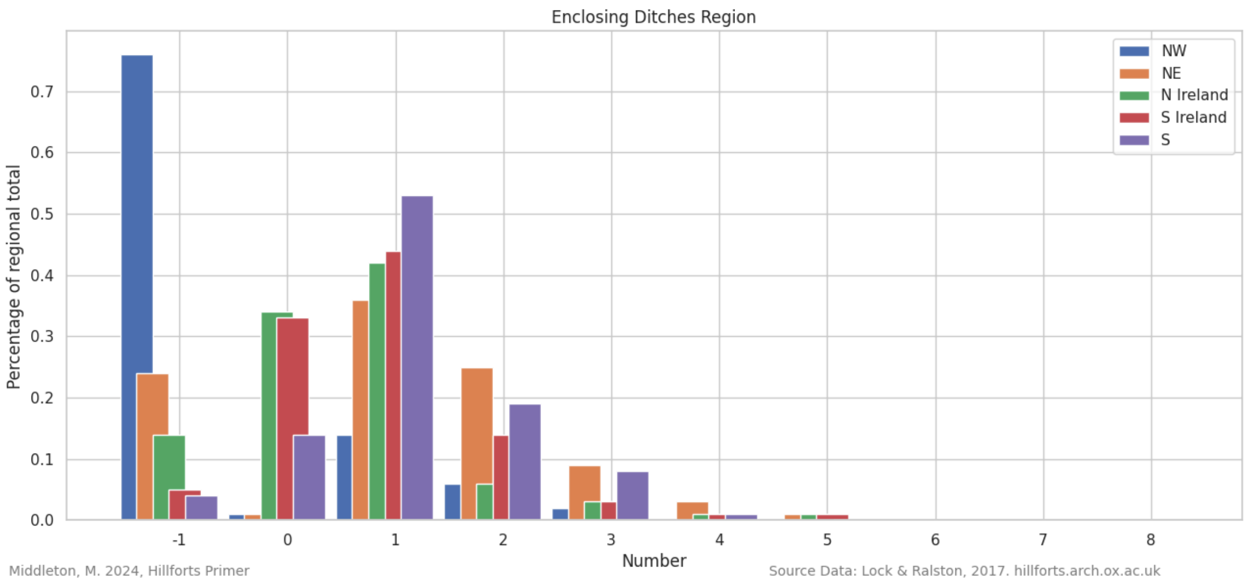
Hillforts in Ireland are most likely to have zero or one ditch; In the Northeast, zero to two ditches and in the South, one to two ditches. It is not possible to say anything about the Northwest because of the survey bias seen in Ditches Mapped (Not Recorded).

In [ ]:
```
regional_dict = \
 {'Northwest': location_enclosing_data_nw['Enclosing_Ditches_Number'], \
  'Northeast': location_enclosing_data_ne['Enclosing_Ditches_Number'], \
  'North Ireland': location_enclosing_data_ireland_n['Enclosing_Ditches_Number'], \
  'South Ireland': location_enclosing_data_ireland_s['Enclosing_Ditches_Number'], \
  'South': location_enclosing_data_south['Enclosing_Ditches_Number']}
plot_data = pd.DataFrame.from_dict(regional_dict)
plt.figure(figsize=(20,8))
ax = sns.boxplot(data=plot_data, orient="h" , whis=[2.2, 97.8], showfliers=True);
add_annotation_plot(ax)
ax.set_xlabel('Number')
title = 'Enclosing_Ditches_Number: Regional Boxplots'
plt.title(get_print_title(title))
save_fig(title)
plt.show()
```



Overall, hillforts are most likely to have a single ditch. The proportions are roughly similar across all areas apart from Ireland, where forts are more likely to have no ditch. The large number of hillforts where ditches have not been recorded (-1) shows the data from the Northwest, Northeast and North Ireland is particularly susceptible to the survey bias noted in Ditches Mapped (Not Recorded).

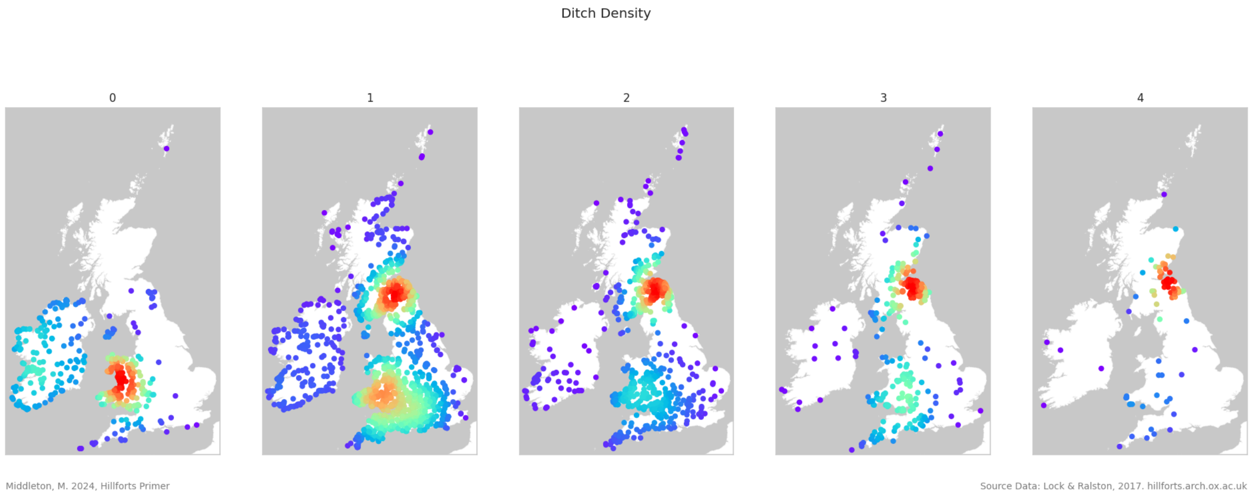
In [ ]:
```
plot_feature_by_region(location_enclosing_data_nw,
                       location_enclosing_data_ne,
                       location_enclosing_data_ireland_n,
                       location_enclosing_data_ireland_s,
                       location_enclosing_data_south,
                       'Enclosing_Ditches_Number',
                       'Enclosing Ditches Region',10)
```



Ditches Summary

The focus for hillforts without ditches is the upland areas of Wales. There is a smaller concentration of these forts in Ireland. It is important to note the bias in recording ditches seen in Ditches Mapped (Not Recorded). An absence of recording may indicate an absence of ditches in this area, and this may suggest there is a third cluster in the Northwest. Work needs to be done to confirm this either way. Hillforts with a single ditch cluster into two groups. One in the Southern Uplands and the second over the southern Cambrian Mountains and into south, central England. Hillforts with two or more ditches tend to cluster to the east of the Southern Uplands although there are also small clusters of these in Wales.

```
In [ ]: plot_density_over_grey_five(zero_ditches, one_ditches, two_ditches, \
                                    three_ditches, four_ditches, 'Ditch Density')
```



Quadrant Data

The commentary for the quadrant data will be summarised at the top of each class. Individual commentary will not be provided for each orientation.

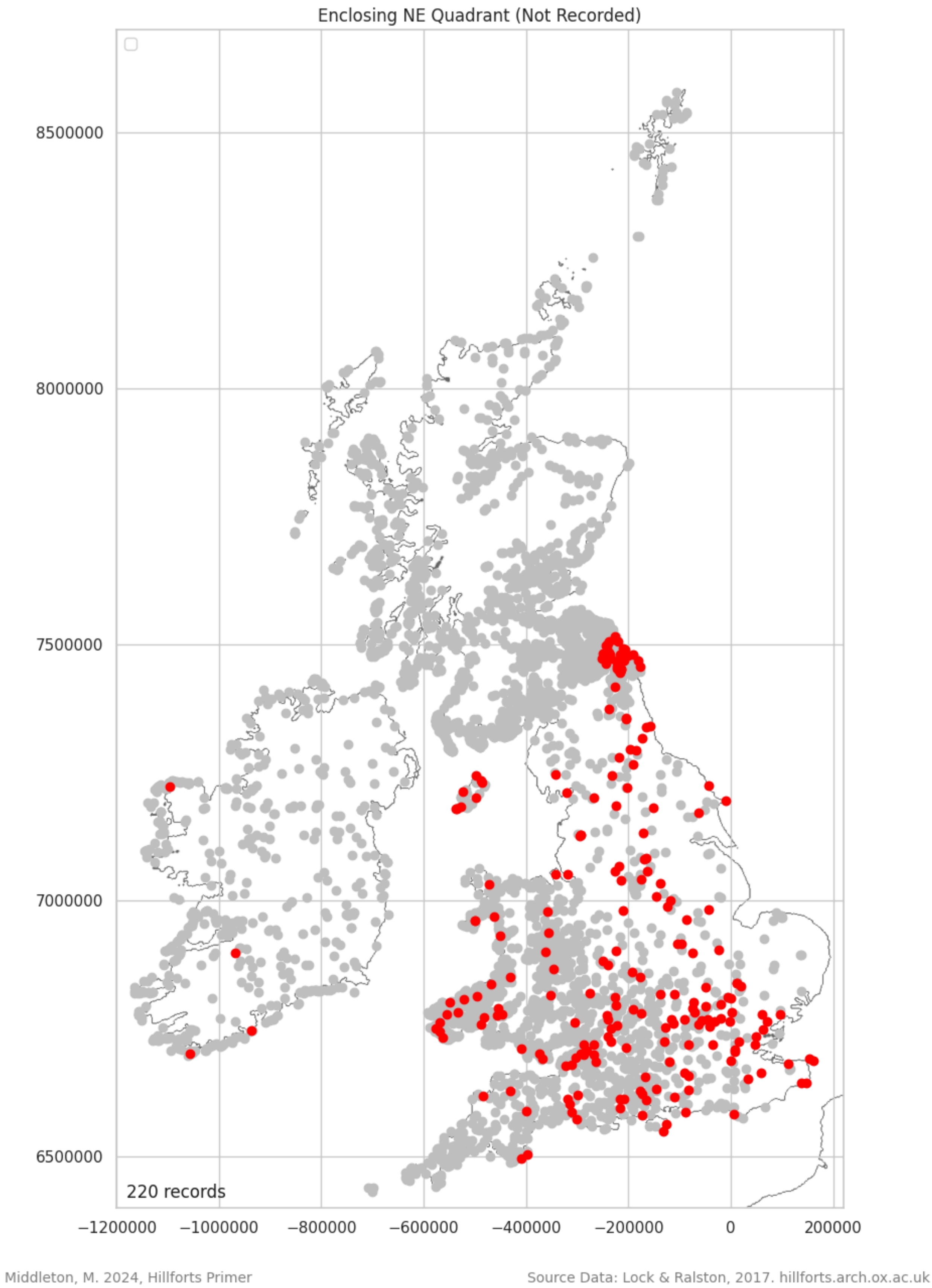
Quadrant Data Mapped (Not Recorded)

Only between 220 (NE) to 251 (SW) hillforts have not had quadrant data recorded. Almost all of these are in England and Wales.

NE Quadrant Data Mapped (Not Recorded)

```
In [ ]: all_ramparts = \
location_enclosing_data[location_enclosing_data['Enclosing_Max_Ramparts']>-1]
all_ditches = \
location_enclosing_data[location_enclosing_data['Enclosing_Ditches_Number']>-1]
ne_quadrant_data = \
location_enclosing_data[location_enclosing_data['Enclosing_NE_Quadrant']>-1]
se_quadrant_data = \
location_enclosing_data[location_enclosing_data['Enclosing_SE_Quadrant']>-1]
sw_quadrant_data = \
location_enclosing_data[location_enclosing_data['Enclosing_SW_Quadrant']>-1]
nw_quadrant_data = \
location_enclosing_data[location_enclosing_data['Enclosing_NW_Quadrant']>-1]
```

```
In [ ]: nan_ne = \
location_enclosing_data\
 [location_enclosing_data['Enclosing_NE_Quadrant']==-1].copy()
nan_ne['Enclosing_NE_Quadrant'] = "Yes"
nan_ne_stats = plot_over_grey(nan_ne, 'Enclosing_NE_Quadrant', \
                              'Yes', '(Not Recorded)')
```

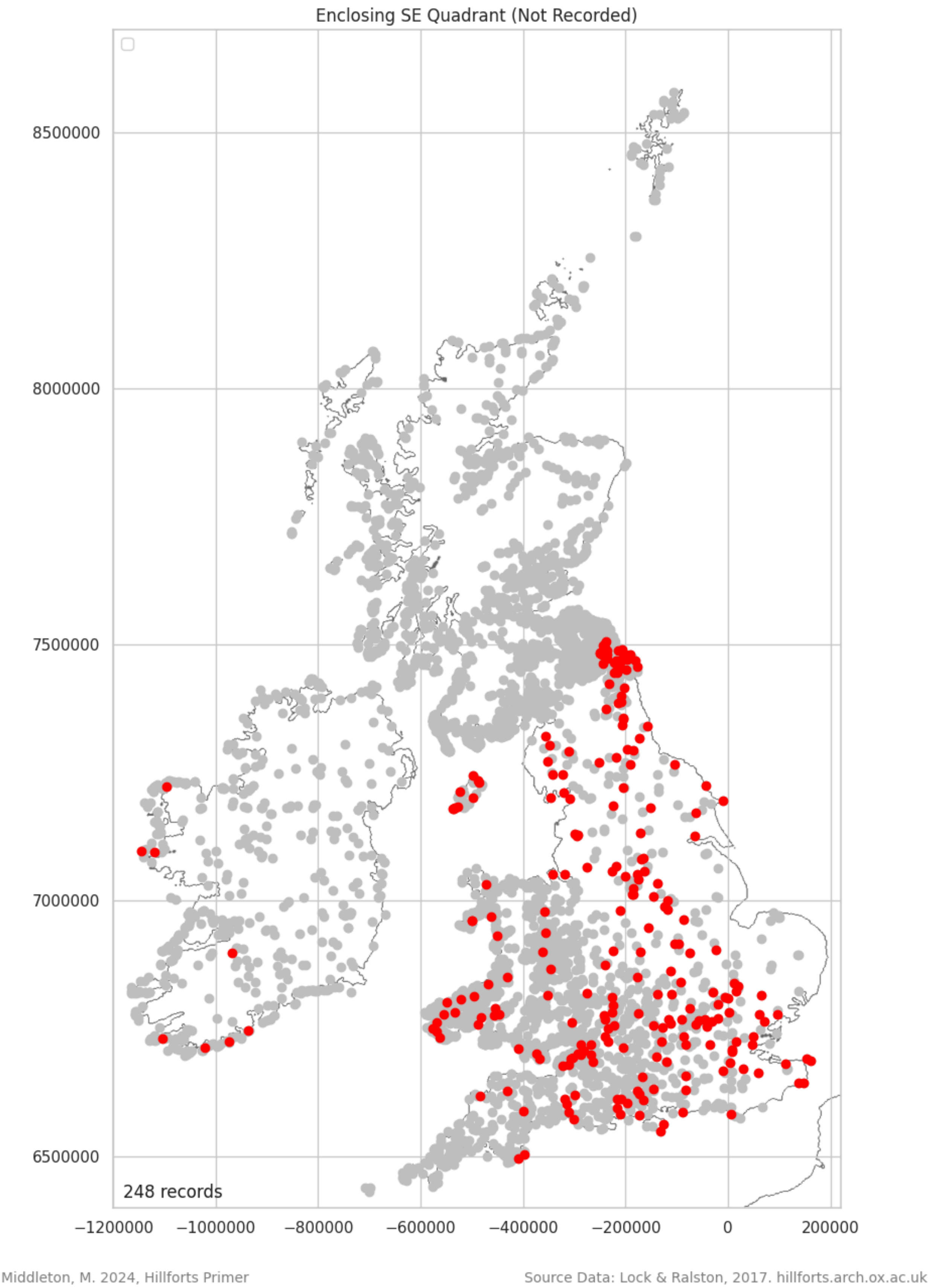



```
5.31%
```

SE Quadrant Data Mapped (Not Recorded)

In [ ]:

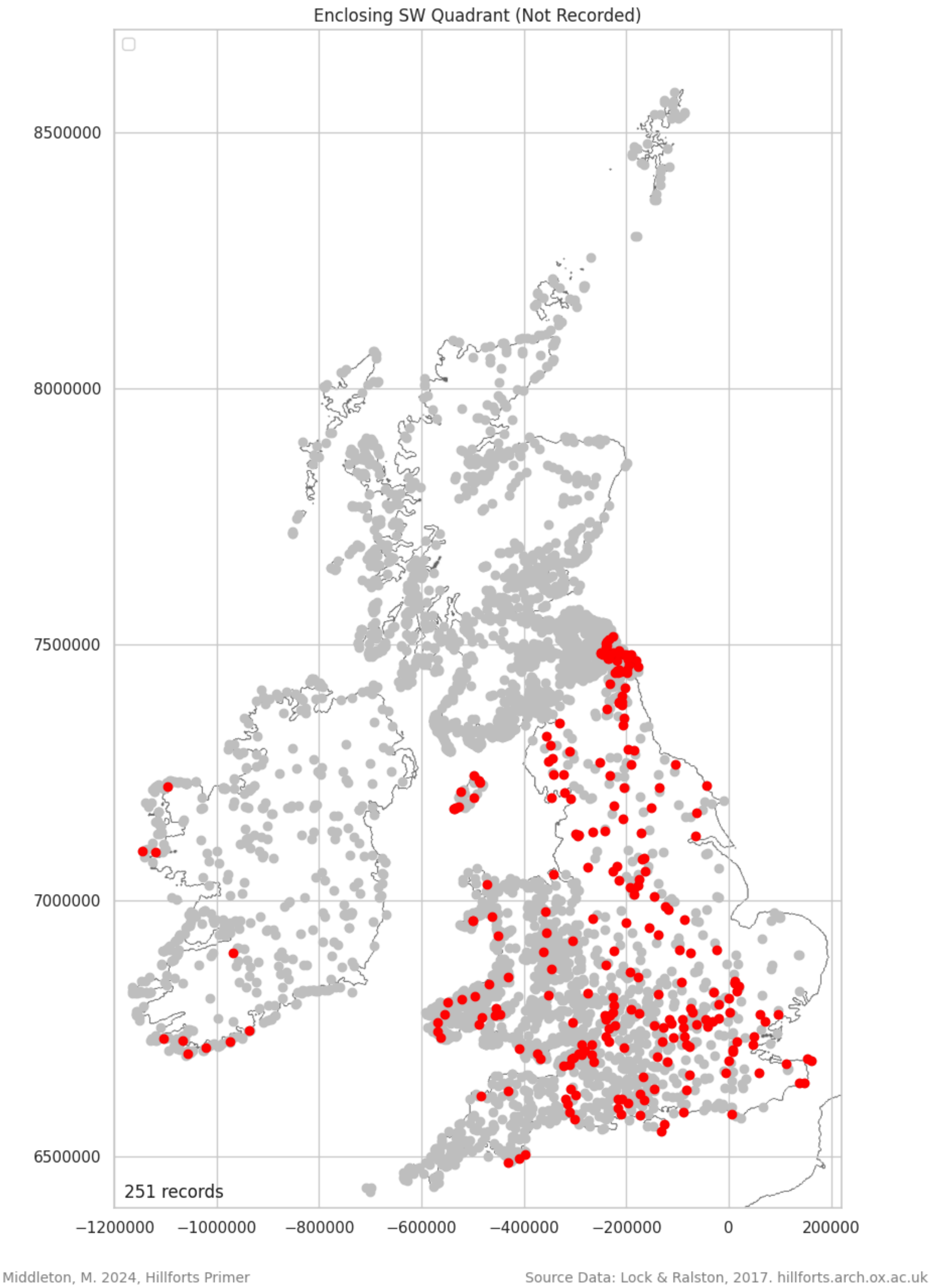
```
nan_se = \
location_enclosing_data[location_enclosing_data['Enclosing_SE_Quadrant']==-1].copy()
nan_se['Enclosing_SE_Quadrant'] = "Yes"
nan_se_stats = plot_over_grey(nan_se, 'Enclosing_SE_Quadrant', 'Yes', \
                              '(Not Recorded)')
```




```
5.98%
```

SW Quadrant Data Mapped (Not Recorded)

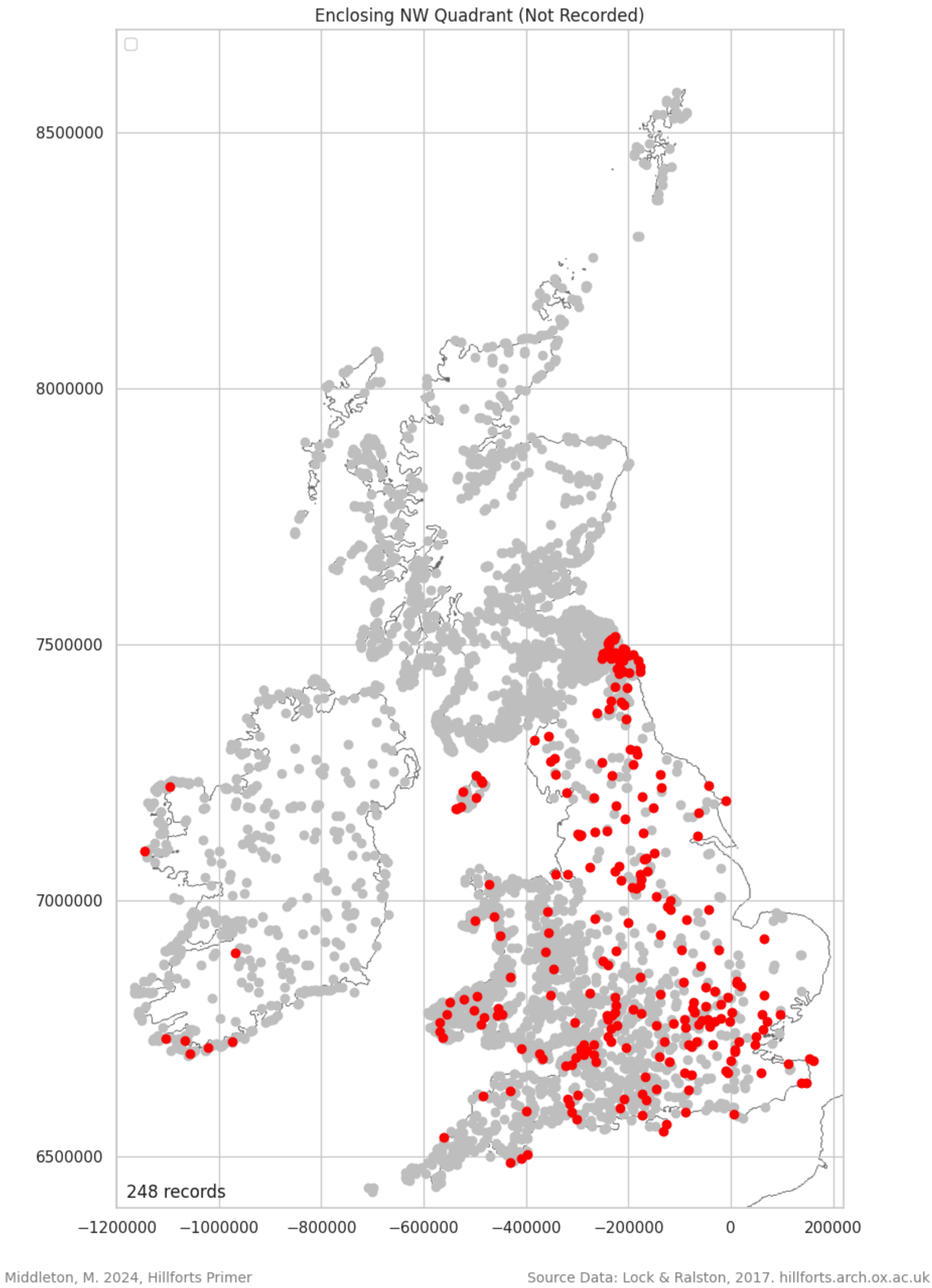
In [ ]:
```
nan_sw = \
location_enclosing_data[location_enclosing_data['Enclosing_SW_Quadrant']==-1].copy()
nan_sw['Enclosing_SW_Quadrant'] = "Yes"
nan_sw_stats = plot_over_grey(nan_sw, 'Enclosing_SW_Quadrant', 'Yes', \
                              '(Not Recorded)')
```




```
6.05%
```

NW Quadrant Data Mapped (Not Recorded)

In [ ]:
```
nan_nw = \
location_enclosing_data\
 [location_enclosing_data['Enclosing_NW_Quadrant']==-1].copy()
nan_nw['Enclosing_NW_Quadrant'] = "Yes"
nan_nw_stats = plot_over_grey(nan_nw, 'Enclosing_NW_Quadrant', 'Yes', \
                              '(Not Recorded)')
```



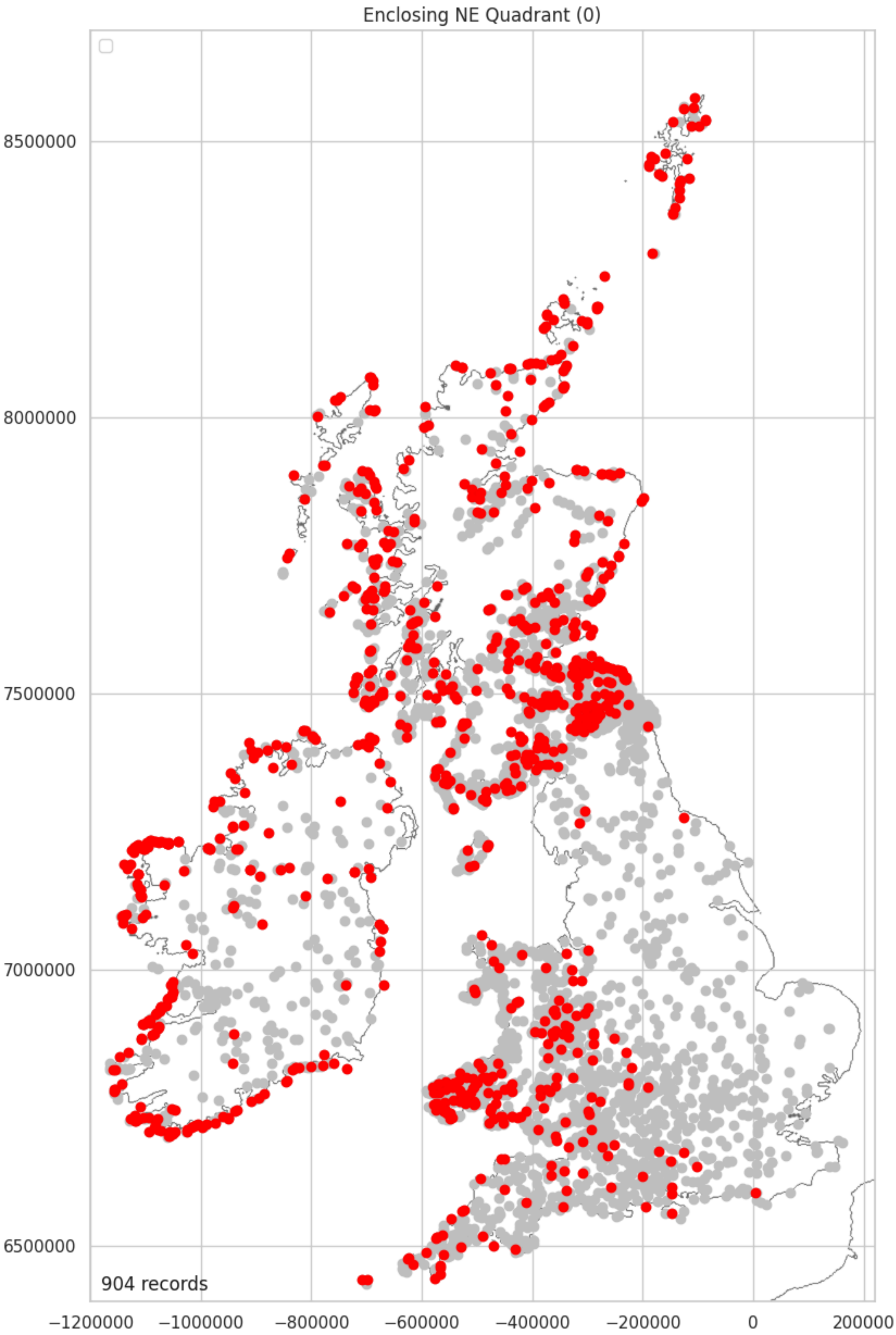

```
5.98%
```

Quadrant Data Mapped (0)

Where there are no ramparts, these show an influence from the local topography. Irish coastal forts on the west coast show a cluster of forts with no ramparts facing southwest and northwest - toward the sea. Similarly, forts on the Pembrokeshire peninsula show more intense clusters of forts with no ramparts to the southwest and northwest. Again, facing the sea. As this data is most likely to reflect the macro topographic situation of each fort, large scale regional analysis of this data is likely to provide limited insight. Commentary over the remainder of this section will be brief.

NE Quadrant Data Mapped (0)

```
In [ ]: zero_ne = ne_quadrant_data[ne_quadrant_data['Enclosing_NE_Quadrant']==0].copy()
        zero_ne['Enclosing_NE_Quadrant'] = "Yes"
        zero_ne_stats = plot_over_grey(zero_ne, 'Enclosing_NE_Quadrant', 'Yes', '(0)')
```



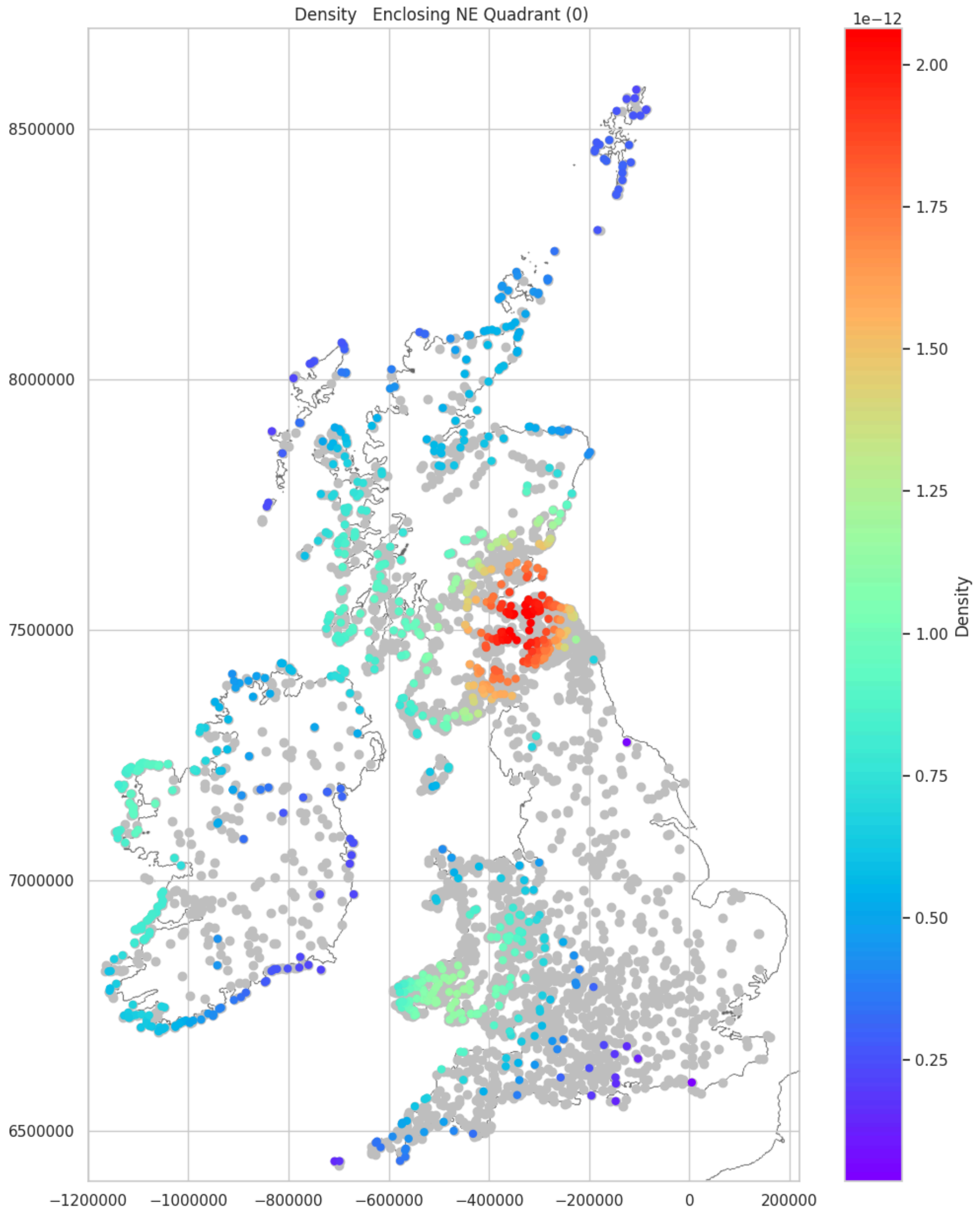
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
21.8%
```

NE Quadrant Data Density Mapped (0)

```
In [ ]: plot_density_over_grey(zero_ne_stats, 'Enclosing_NE_Quadrant (0)')
```



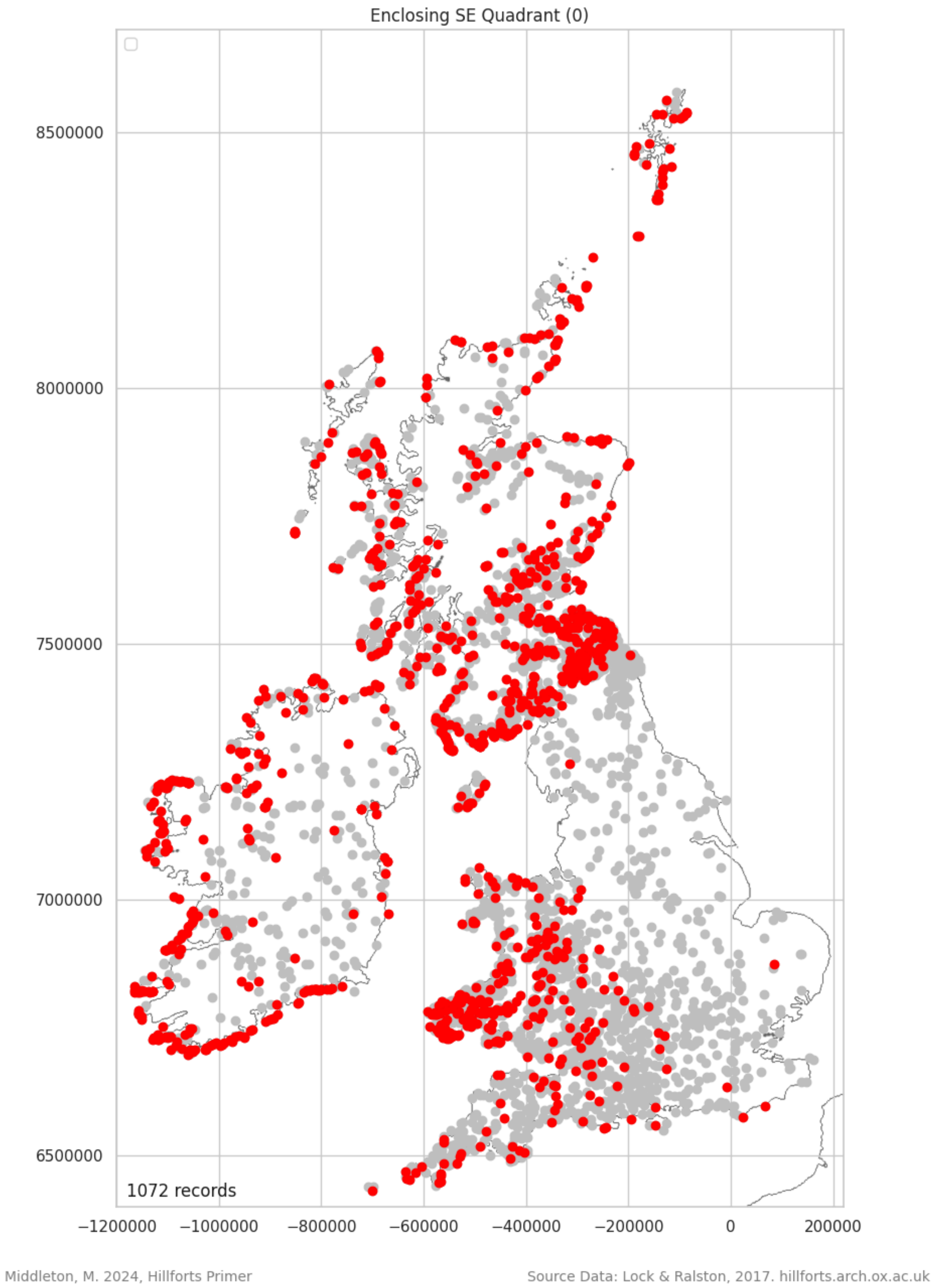
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

SE Quadrant Data Mapped (0)

In [ ]:

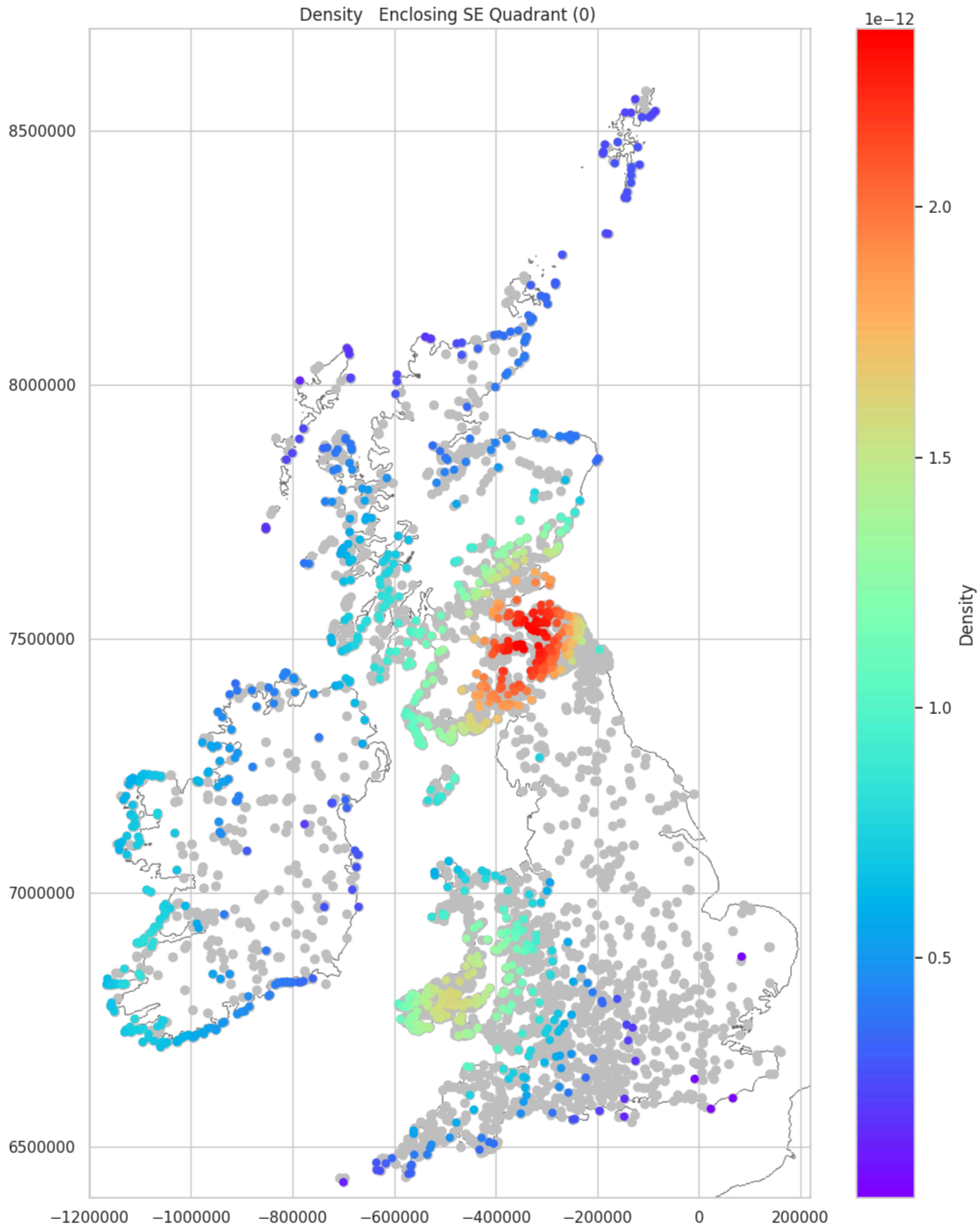
```
zero_se = se_quadrant_data[se_quadrant_data['Enclosing_SE_Quadrant']==0].copy()
zero_se['Enclosing_SE_Quadrant'] = "Yes"
zero_se_stats = plot_over_grey(zero_se, 'Enclosing_SE_Quadrant', 'Yes', '(0)')
```




```
25.85%
```

NE Quadrant Data Density Mapped (0)

In [ ]:
```
plot_density_over_grey(zero_se_stats, 'Enclosing_SE_Quadrant (0)')
```

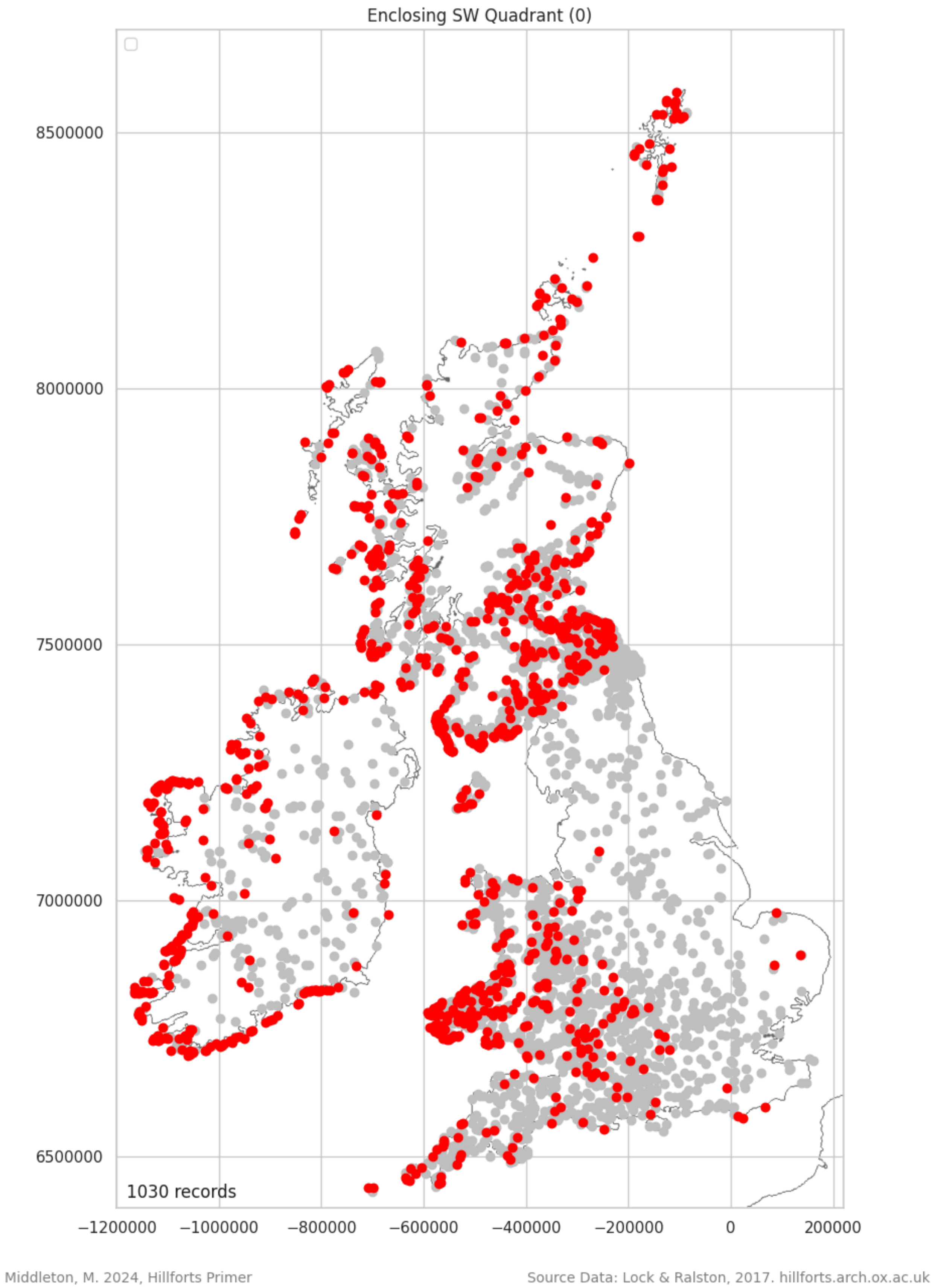


Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

SW Quadrant Data Mapped (0)

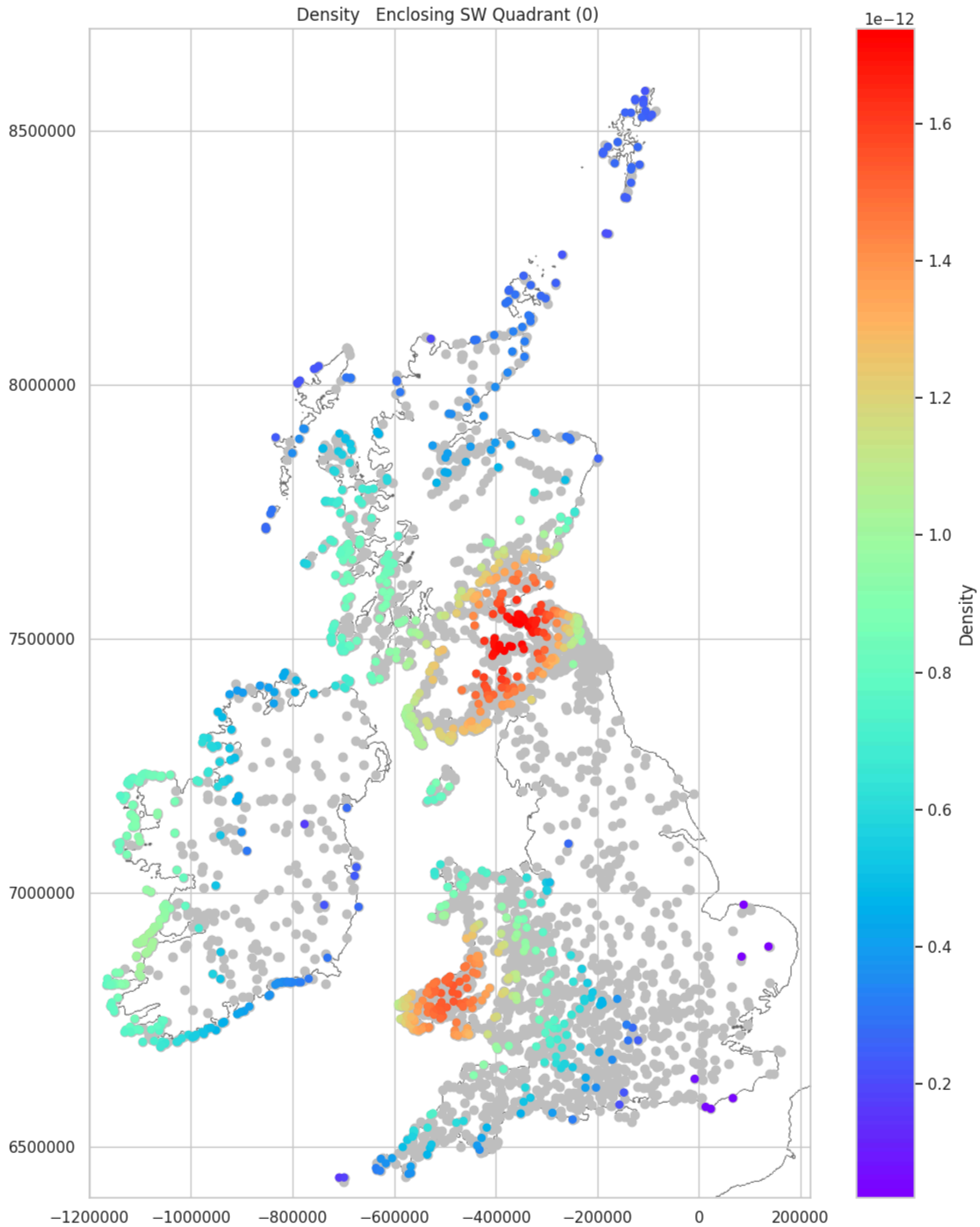
In [ ]:
```
zero_sw = sw_quadrant_data[sw_quadrant_data['Enclosing_SW_Quadrant']==0].copy()
zero_sw['Enclosing_SW_Quadrant'] = "Yes"
zero_sw_stats = plot_over_grey(zero_sw, 'Enclosing_SW_Quadrant', 'Yes', '(0)')
```



24.84%

SW Quadrant Data Density Mapped (0)

In [ ]:
```
plot_density_over_grey(zero_sw_stats, 'Enclosing_SW_Quadrant (0)')
```

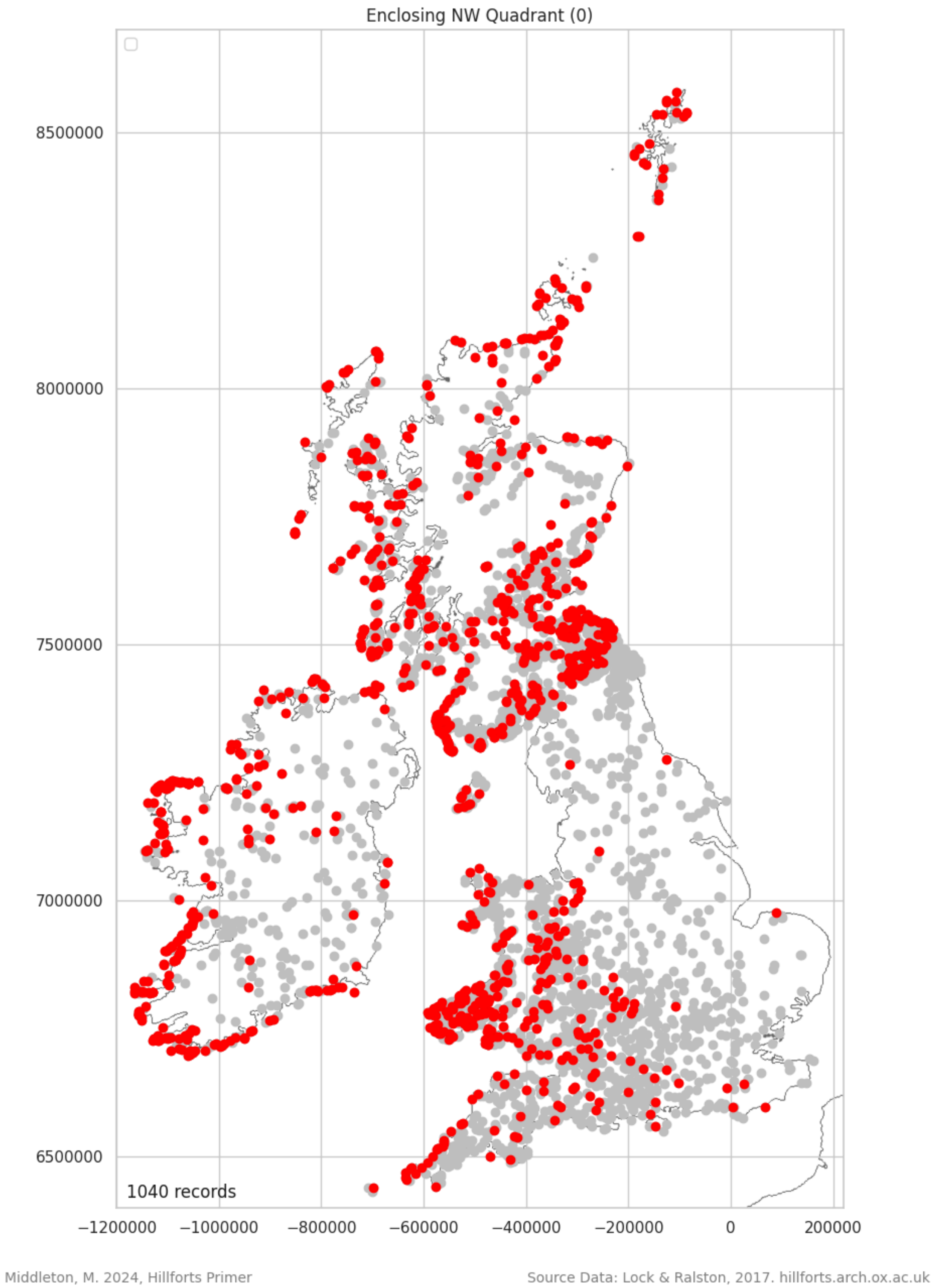


Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

NW Quadrant Data Mapped (0)

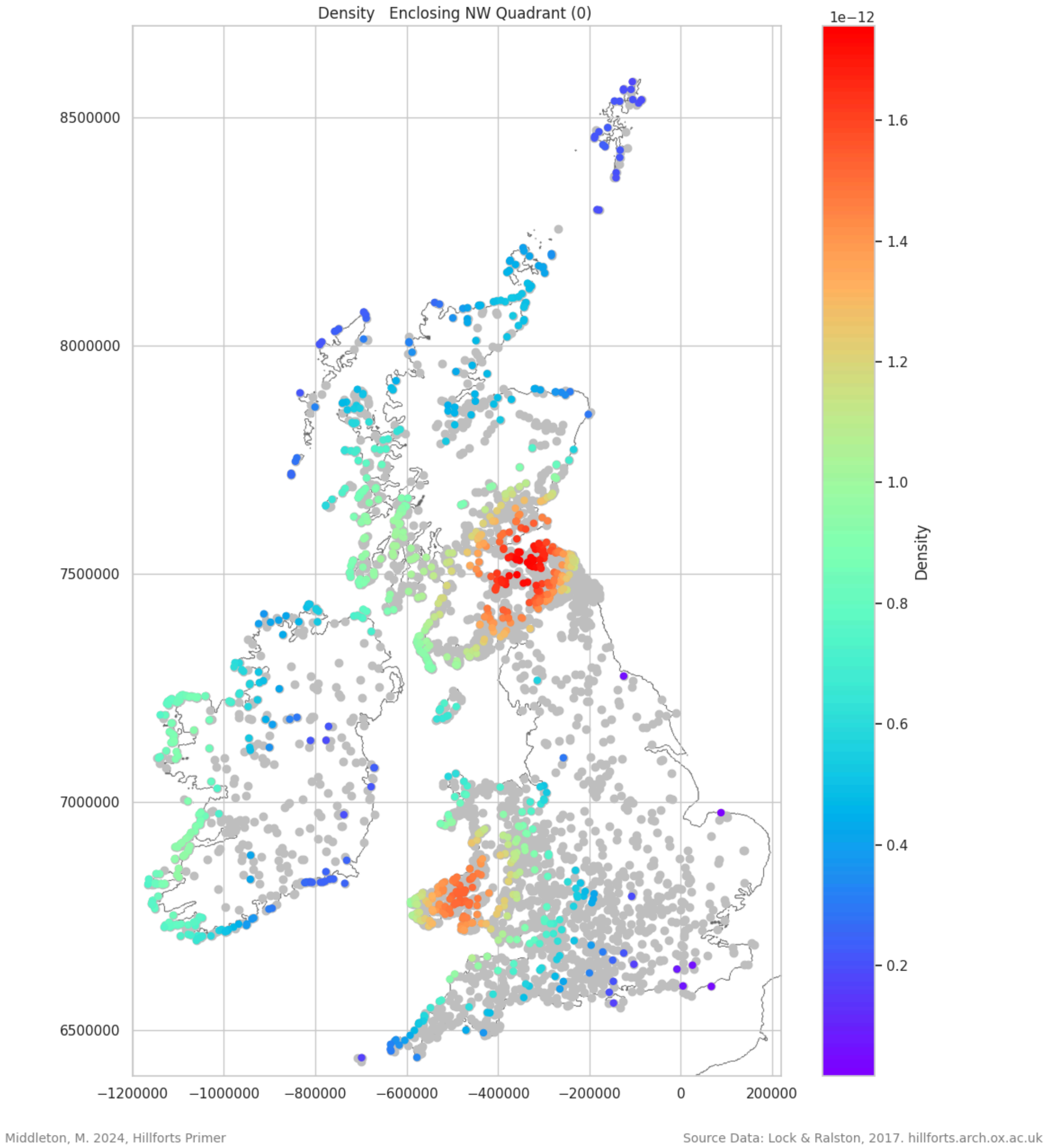
In [ ]:
```
zero_nw = nw_quadrant_data[nw_quadrant_data['Enclosing_NW_Quadrant']==0].copy()
zero_nw['Enclosing_NW_Quadrant'] = "Yes"
zero_nw_stats = plot_over_grey(zero_nw, 'Enclosing_NW_Quadrant', 'Yes', '(0)')
```




```
25.08%
```

NW Quadrant Data Density Mapped (0)

```
In [ ]: plot_density_over_grey(zero_nw_stats, 'Enclosing_NW_Quadrant (0)')
```



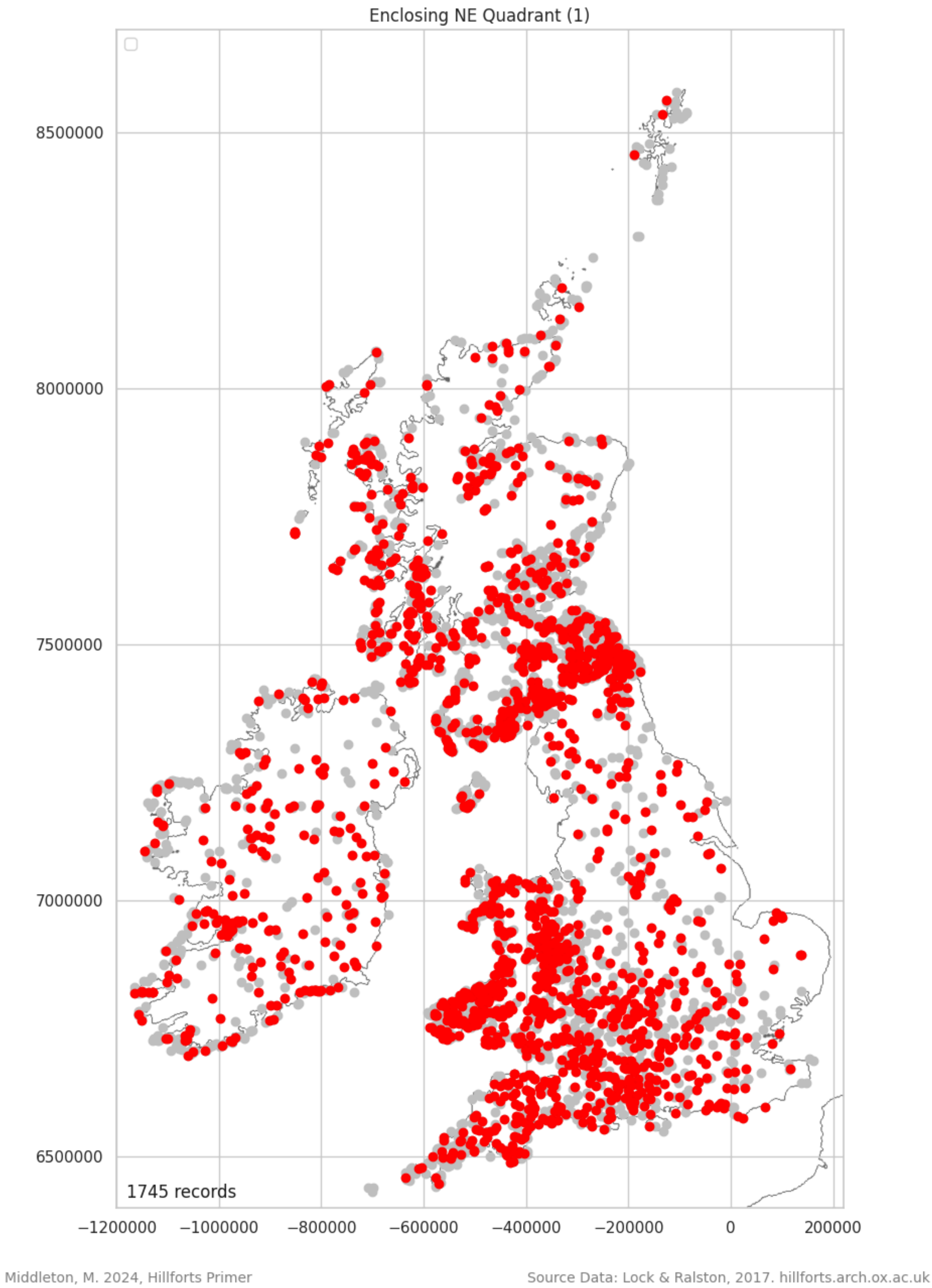
Quadrant Data Mapped (1)

The wide spread of forts with a single rampart reflects the distributions and clusters discussed in the ramparts section above. The general intensity of the clusters is as would be anticipated except for along the eastern fringe of the Cambrian Mountains where there is a slight reduction in the concentration of forts with ramparts facing northwest.

NE Quadrant Data Mapped (1)

In [ ]:

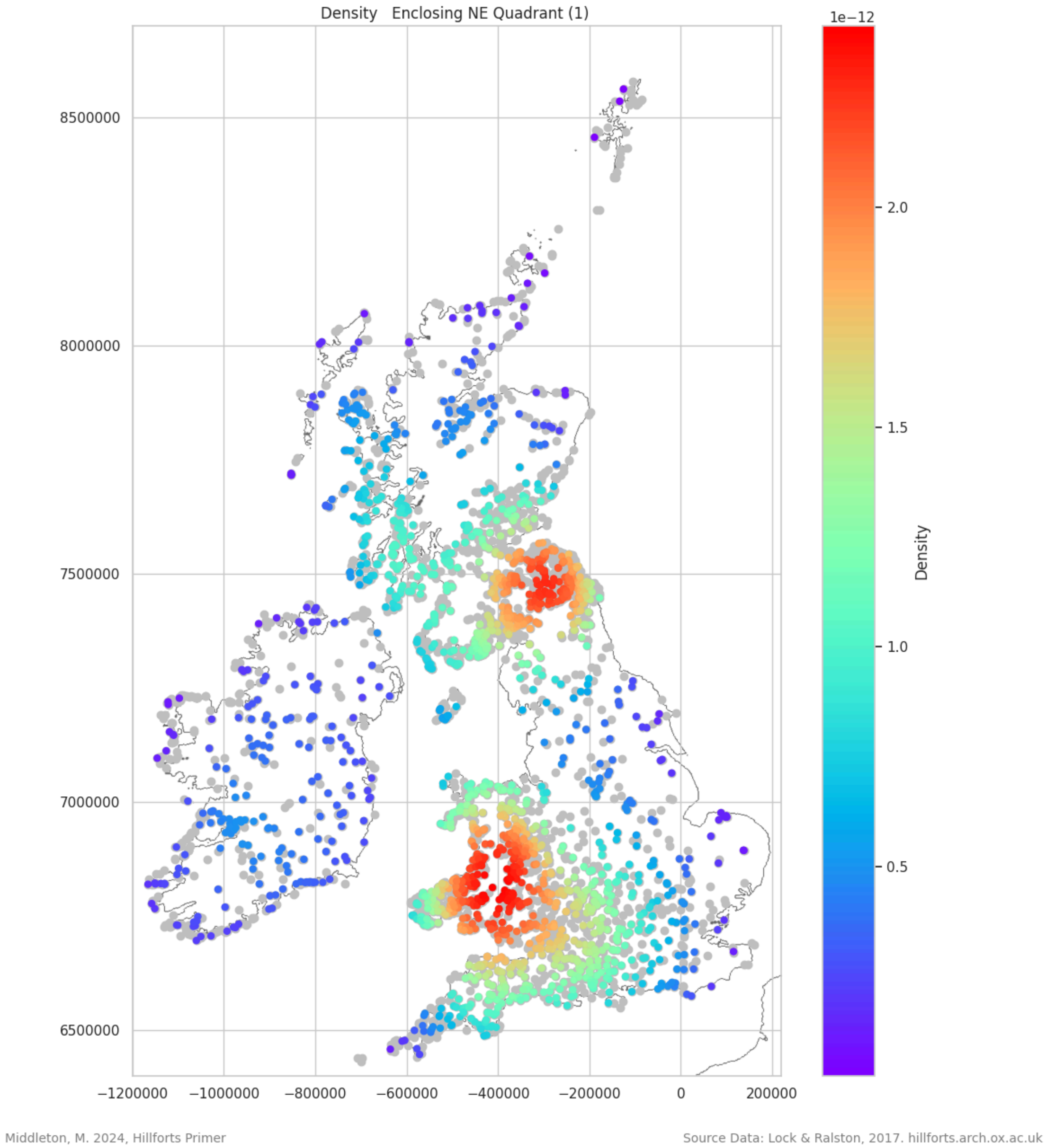
```
one_ne = ne_quadrant_data[ne_quadrant_data['Enclosing_NE_Quadrant']==1].copy()
one_ne['Enclosing_NE_Quadrant'] = "Yes"
one_ne_stats = plot_over_grey(one_ne, 'Enclosing_NE_Quadrant', 'Yes', '(1)')
```




```
42.08%
```

NE Quadrant Data Density Mapped (1)

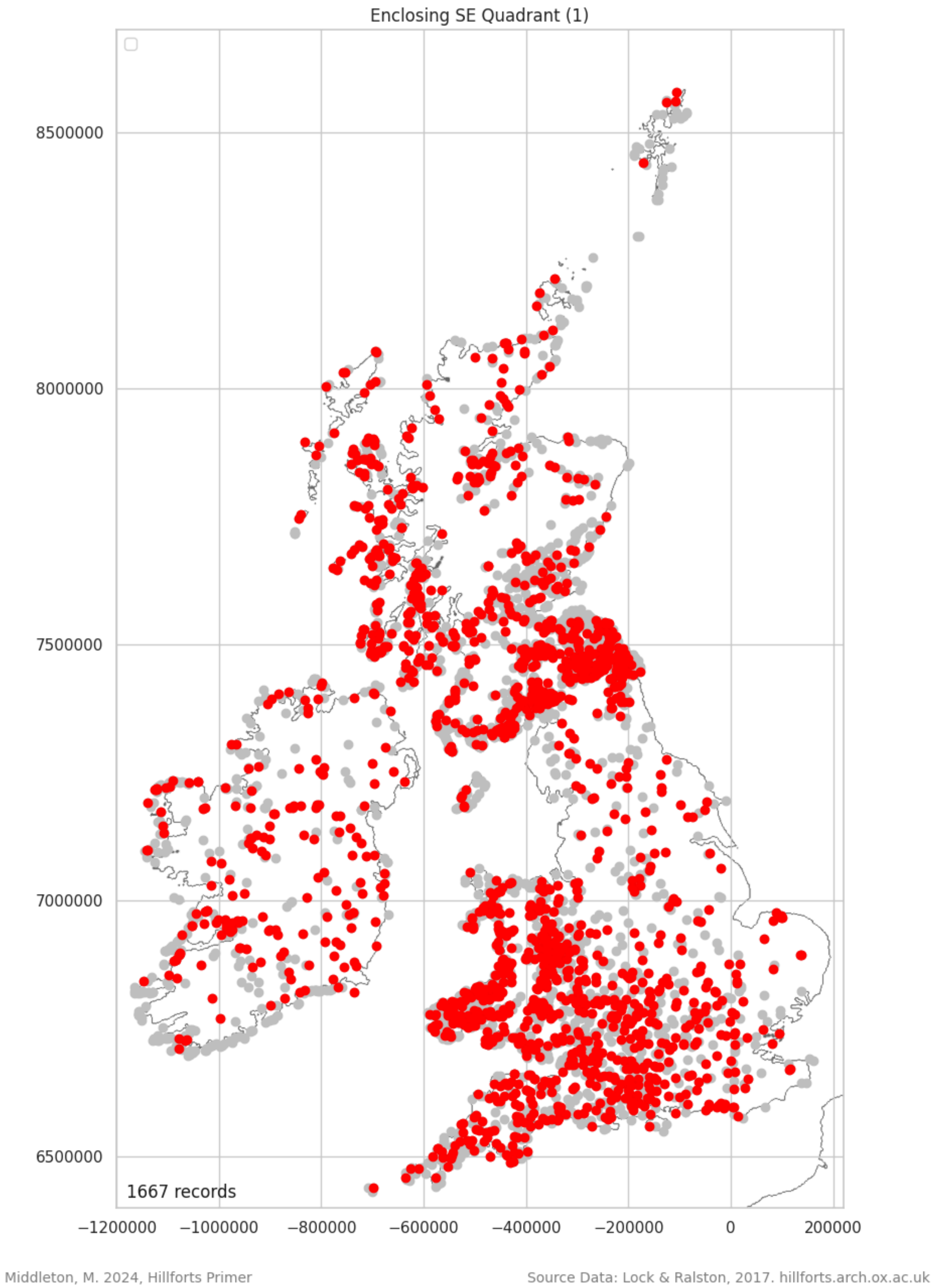
```
In [ ]: plot_density_over_grey(one_ne_stats, 'Enclosing_NE_Quadrant (1)')
```



SE Quadrant Data Mapped (1)

In [ ]:

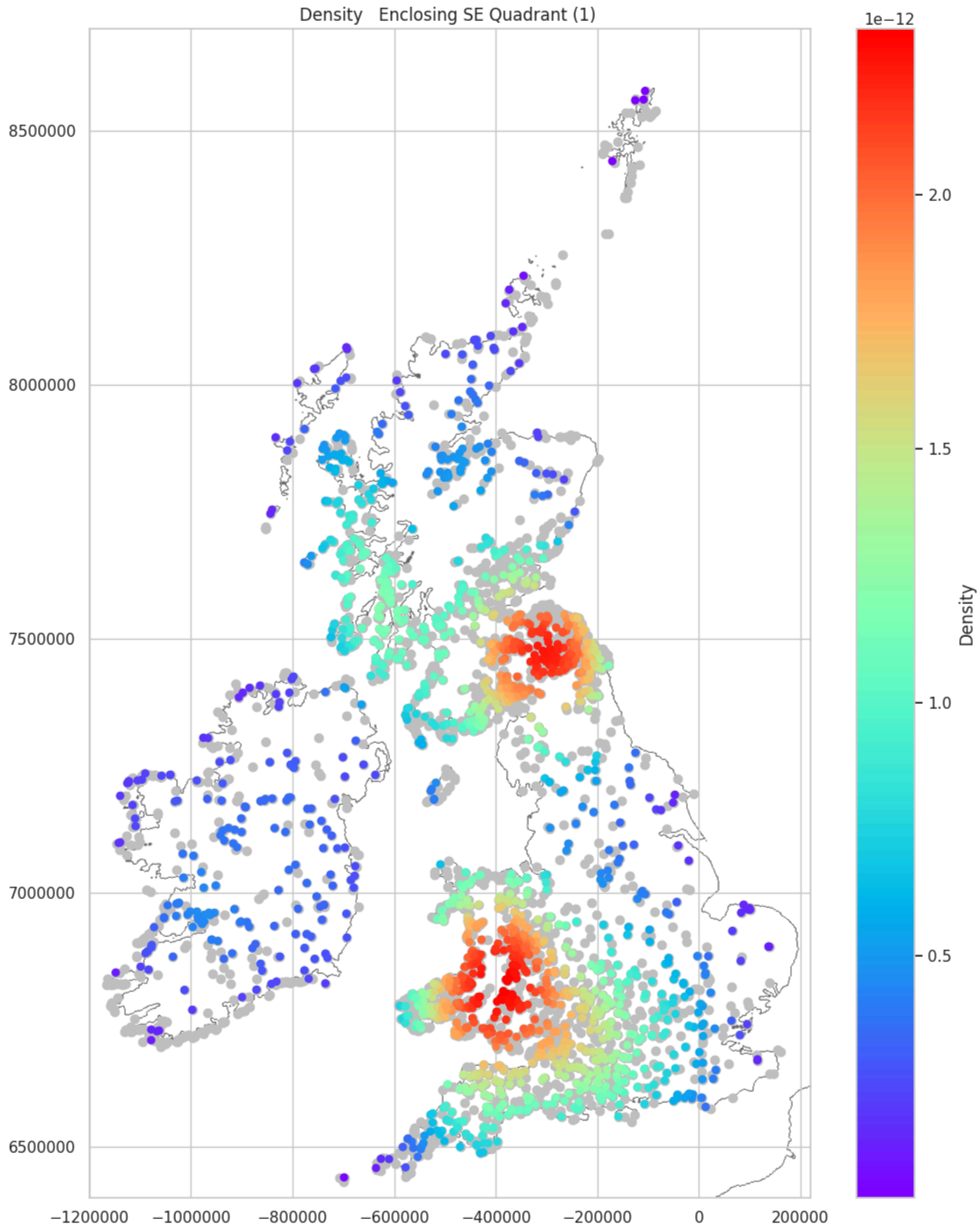
```
one_se = se_quadrant_data[se_quadrant_data['Enclosing_SE_Quadrant']==1].copy()
one_se['Enclosing_SE_Quadrant'] = "Yes"
one_se_stats = plot_over_grey(one_se, 'Enclosing_SE_Quadrant', 'Yes', '(1)')
```




```
40.2%
```

SE Quadrant Data Density Mapped (1)

```
In [ ]: plot_density_over_grey(one_se_stats, 'Enclosing_SE_Quadrant (1)')
```



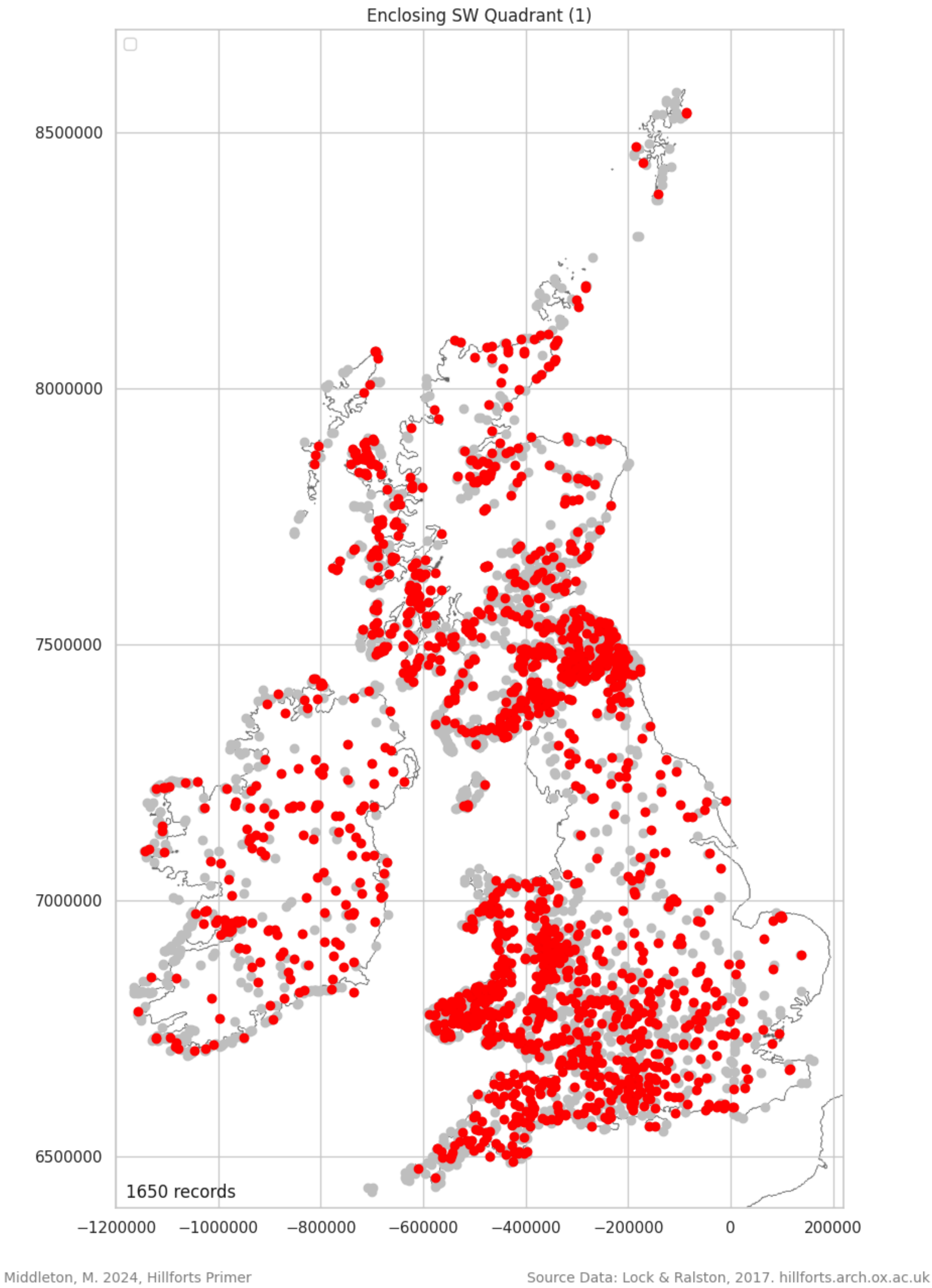
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

SW Quadrant Data Mapped (1)

In [ ]:

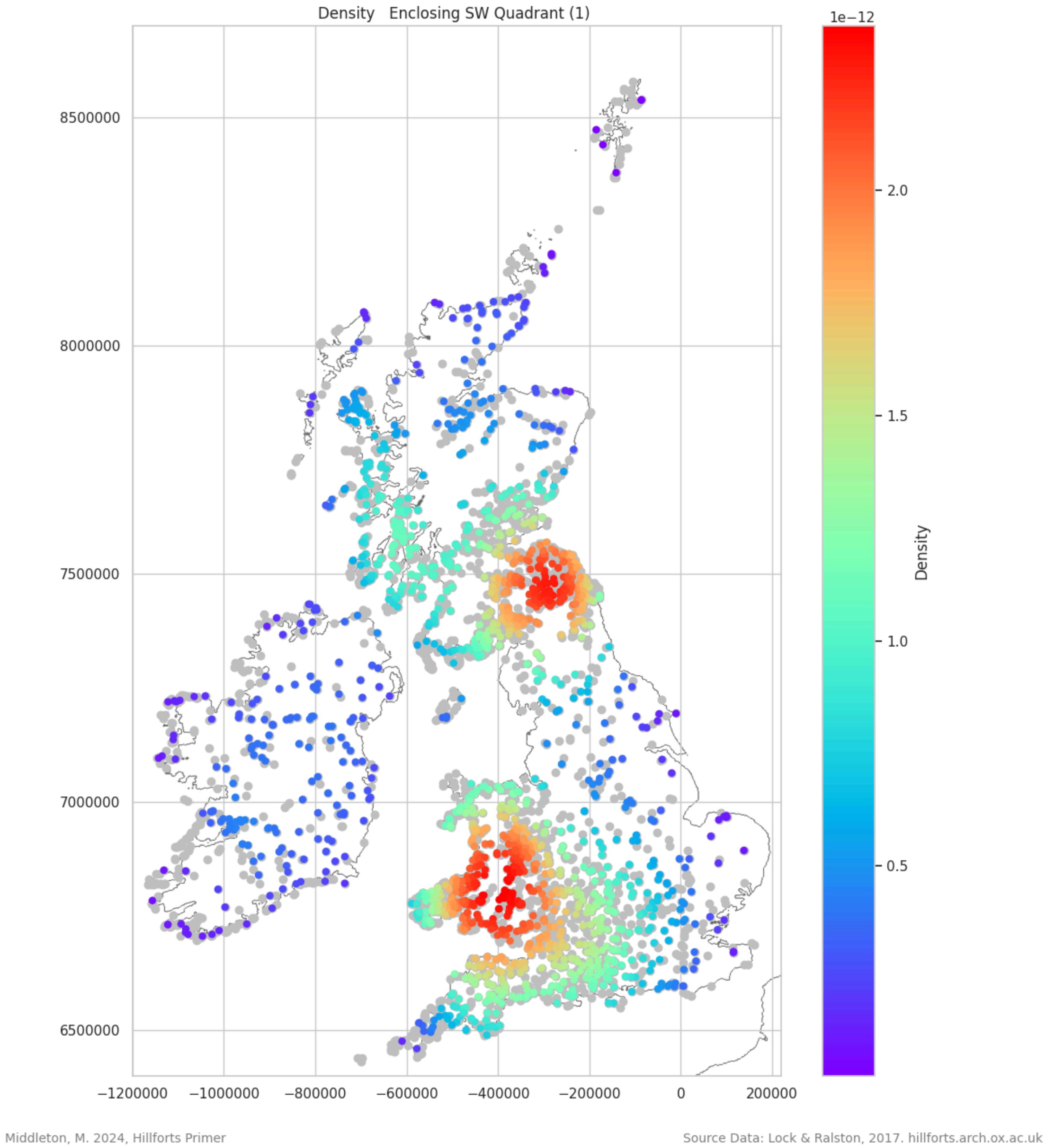
```
one_sw = sw_quadrant_data[sw_quadrant_data['Enclosing_SW_Quadrant']==1].copy()
one_sw['Enclosing_SW_Quadrant'] = "Yes"
one_sw_stats = plot_over_grey(one_sw, 'Enclosing_SW_Quadrant', 'Yes', '(1)')
```




```
39.79%
```

SW Quadrant Data Density Mapped (1)

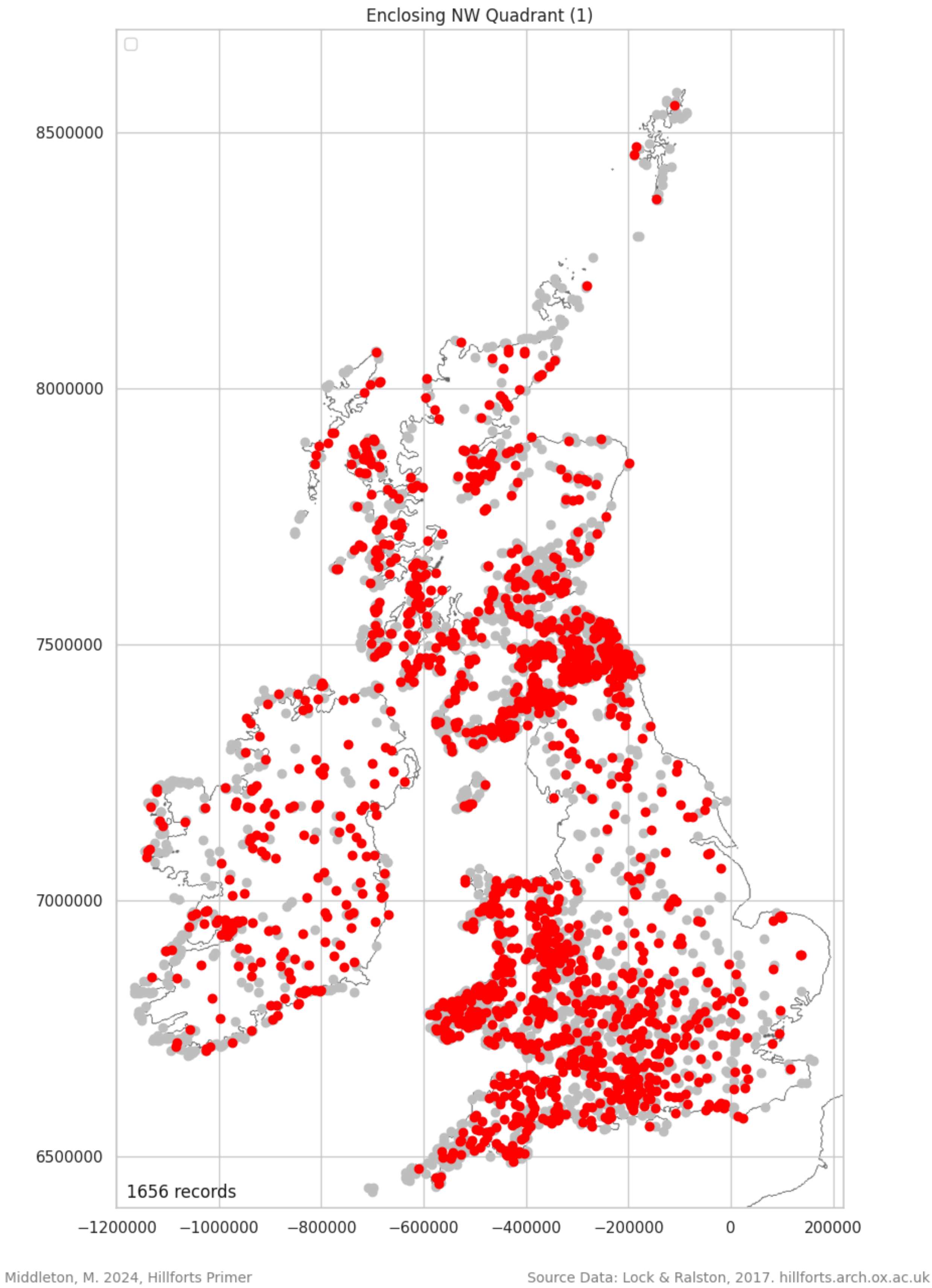
In [ ]:
```
plot_density_over_grey(one_sw_stats, 'Enclosing_SW_Quadrant (1)')
```



NW Quadrant Data Mapped (1)

In [ ]:

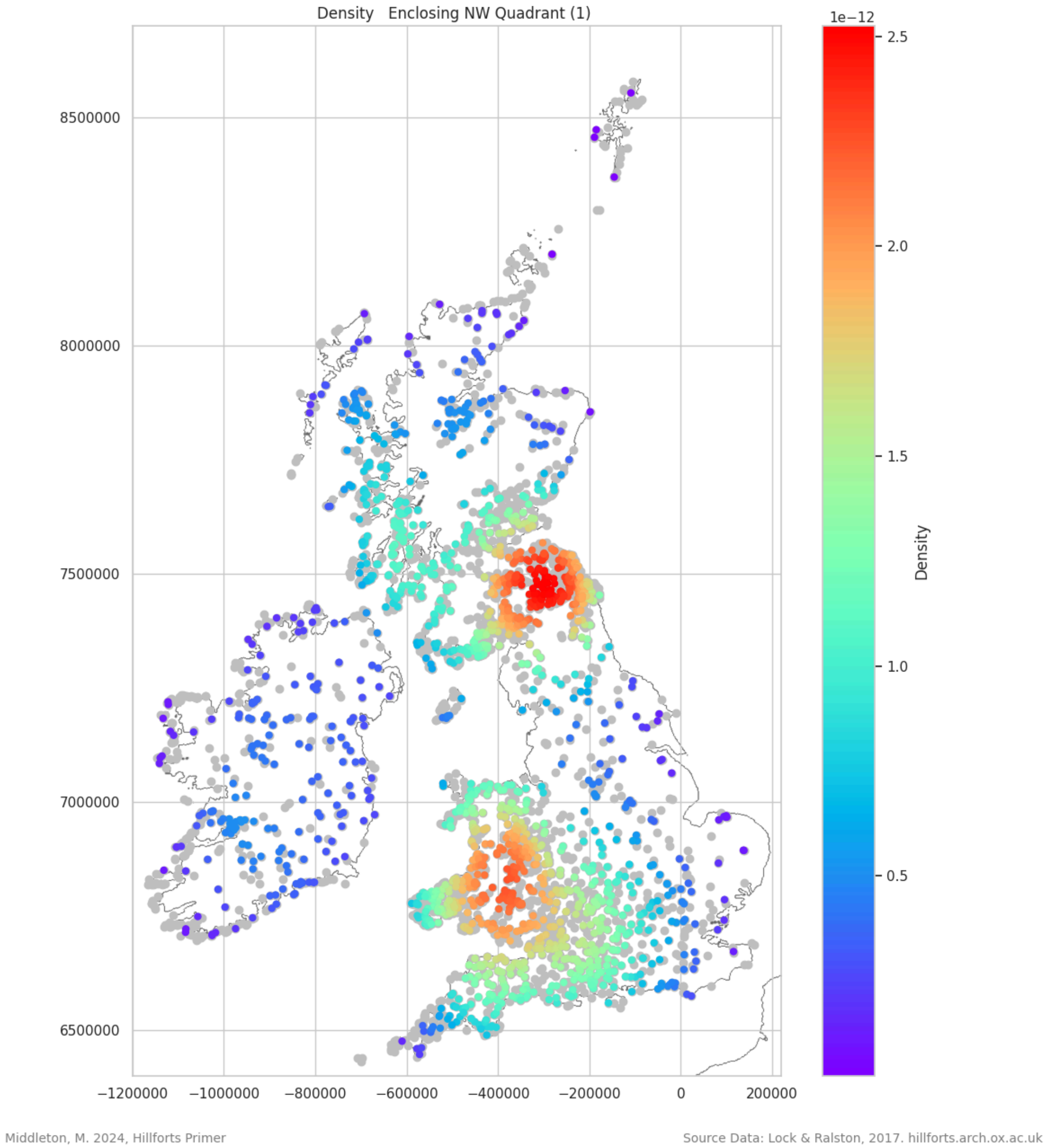
```
one_nw = nw_quadrant_data[nw_quadrant_data['Enclosing_NW_Quadrant']==1].copy()
one_nw['Enclosing_NW_Quadrant'] = "Yes"
one_nw_stats = plot_over_grey(one_nw, 'Enclosing_NW_Quadrant', 'Yes', '(1)')
```



39.93%

NW Quadrant Data Density Mapped (1)

In [ ]:
```
plot_density_over_grey(one_nw_stats, 'Enclosing_NW_Quadrant (1)')
```



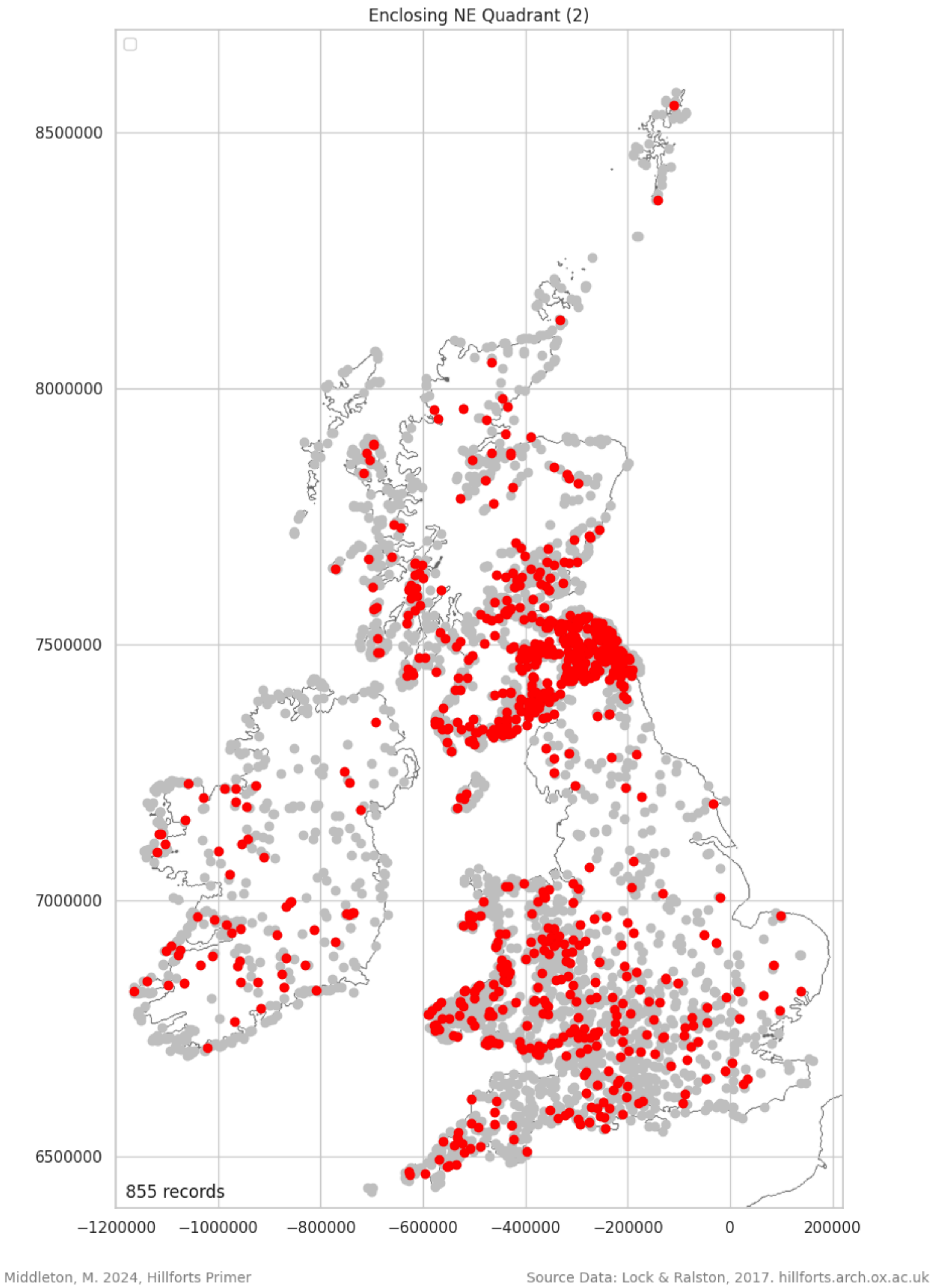
Quadrant Data Mapped (2)

There is little to be said about the quadrant data for two ramparts. Unsurprisingly, it is focussed over the Northeast. See Ramparts Mapped (2).

NE Quadrant Data Mapped (2)

In [ ]:

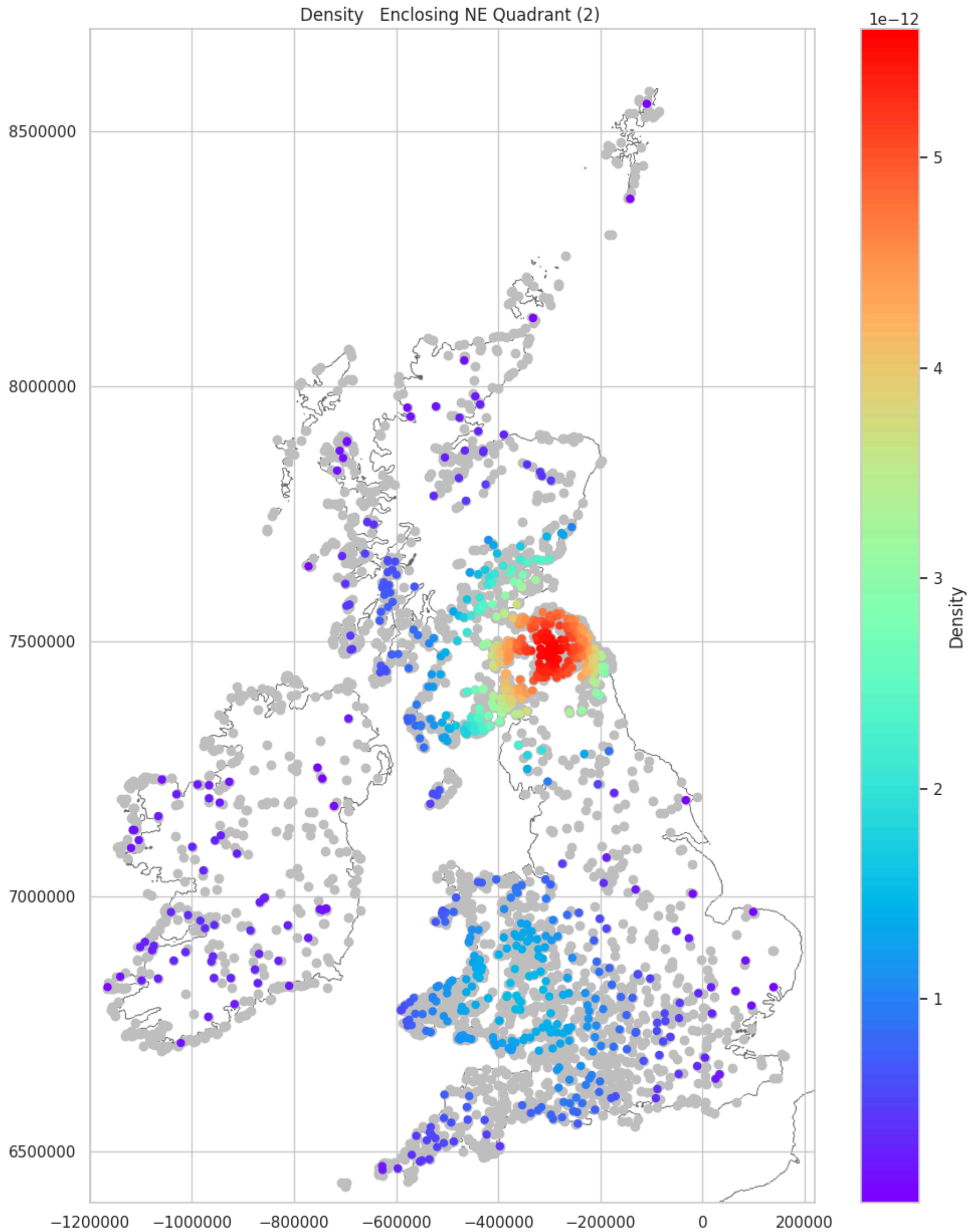
```
two_ne = ne_quadrant_data[ne_quadrant_data['Enclosing_NE_Quadrant']==2].copy()
two_ne['Enclosing_NE_Quadrant'] = "Yes"
two_ne_stats = plot_over_grey(two_ne, 'Enclosing_NE_Quadrant', 'Yes', '(2)')
```




```
20.62%
```

NE Quadrant Data Density Mapped (2)

```
In [ ]: plot_density_over_grey(two_ne_stats, 'Enclosing_NE_Quadrant (2)')
```

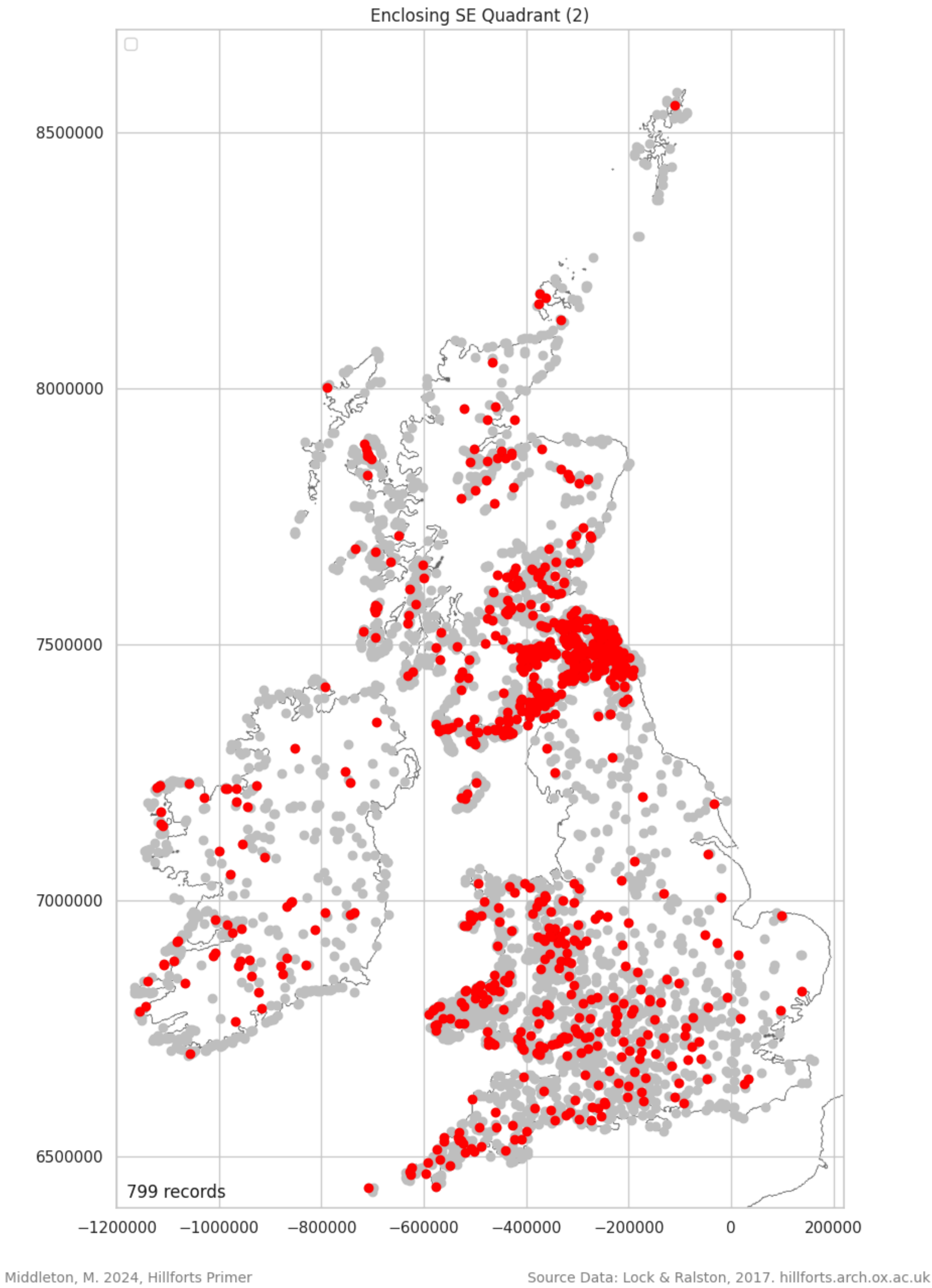


Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

SE Quadrant Data Mapped (2)

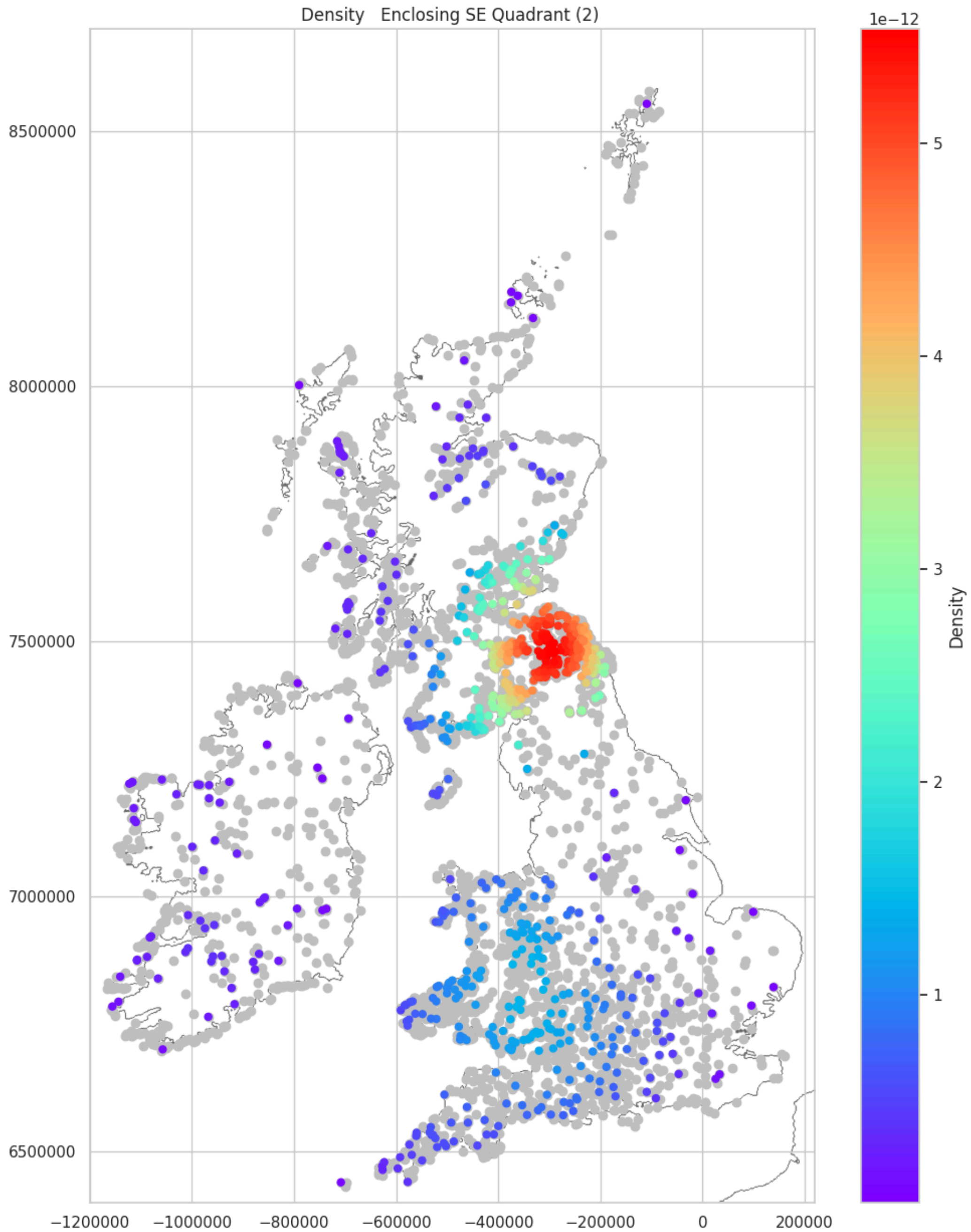
In [ ]:
```
two_se = se_quadrant_data[se_quadrant_data['Enclosing_SE_Quadrant']==2].copy()
two_se['Enclosing_SE_Quadrant'] = "Yes"
two_se_stats = plot_over_grey(two_se, 'Enclosing_SE_Quadrant', 'Yes', '(2)')
```




```
19.27%
```

SE Quadrant Data Density Mapped (2)

```
In [ ]: plot_density_over_grey(two_se_stats, 'Enclosing_SE_Quadrant (2)')
```

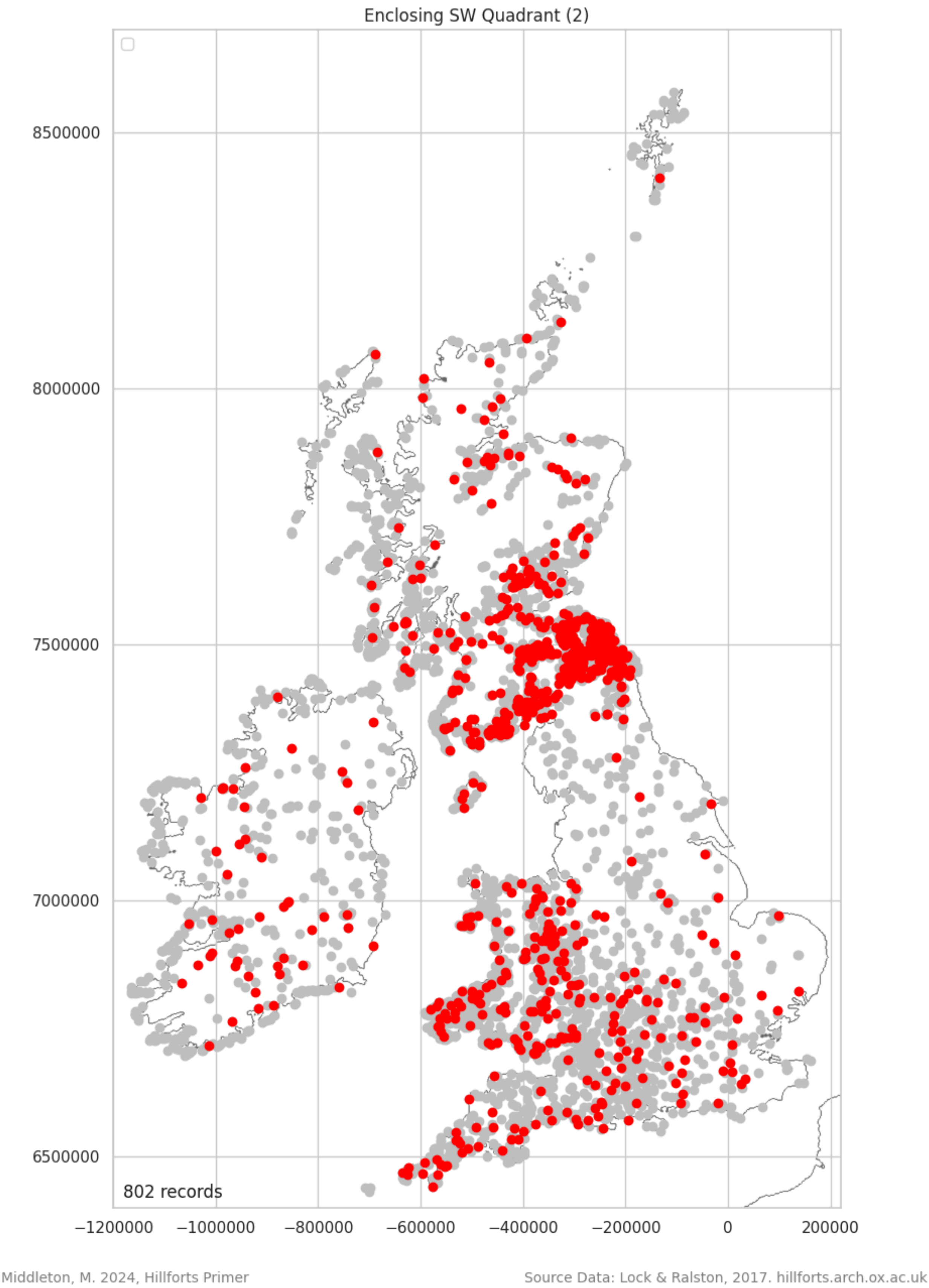


Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

SW Quadrant Data Mapped (2)

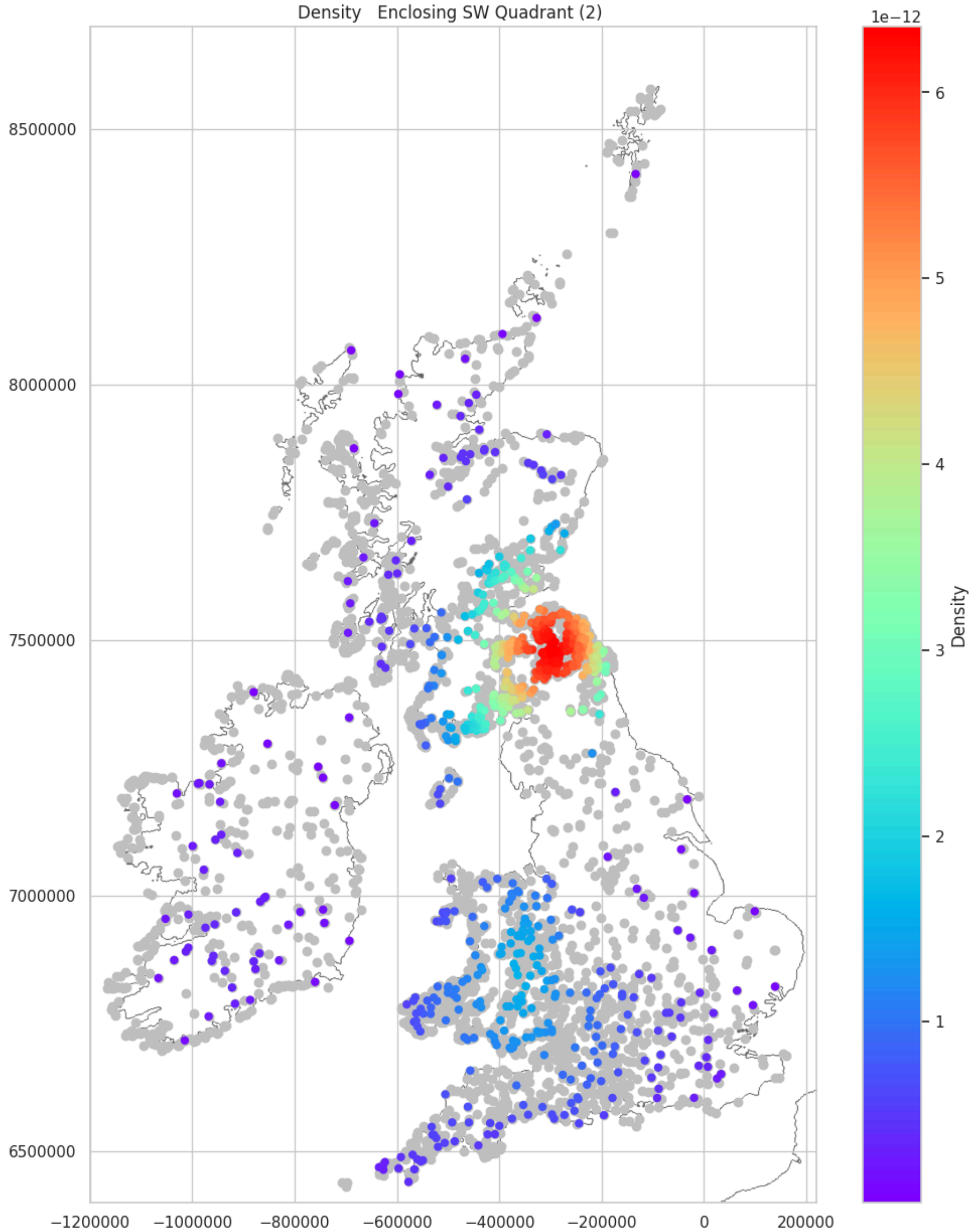
```
In [ ]: two_sw = sw_quadrant_data[sw_quadrant_data['Enclosing_SW_Quadrant']==2].copy()
        two_sw['Enclosing_SW_Quadrant'] = "Yes"
        two_sw_stats = plot_over_grey(two_sw, 'Enclosing_SW_Quadrant', 'Yes', '(2)')
```



19.34%

SW Quadrant Data Density Mapped (2)

In [ ]:
```
plot_density_over_grey(two_sw_stats, 'Enclosing_SW_Quadrant (2)')
```

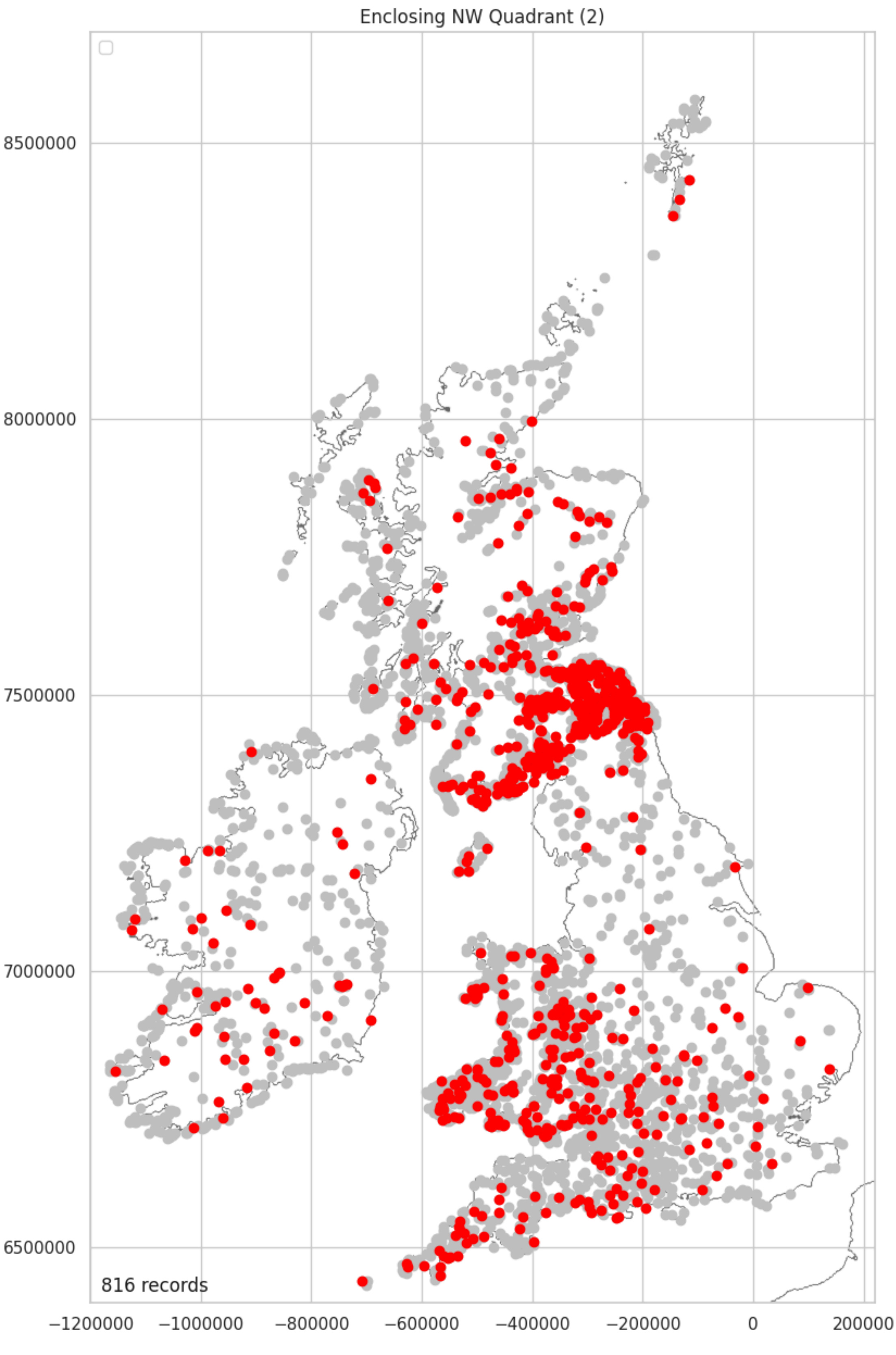


Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

NW Quadrant Data Mapped (2)

In [ ]:
```
two_nw = nw_quadrant_data[nw_quadrant_data['Enclosing_NW_Quadrant']==2].copy()
two_nw['Enclosing_NW_Quadrant'] = "Yes"
two_nw_stats = plot_over_grey(two_nw, 'Enclosing_NW_Quadrant', 'Yes', '(2)')
```



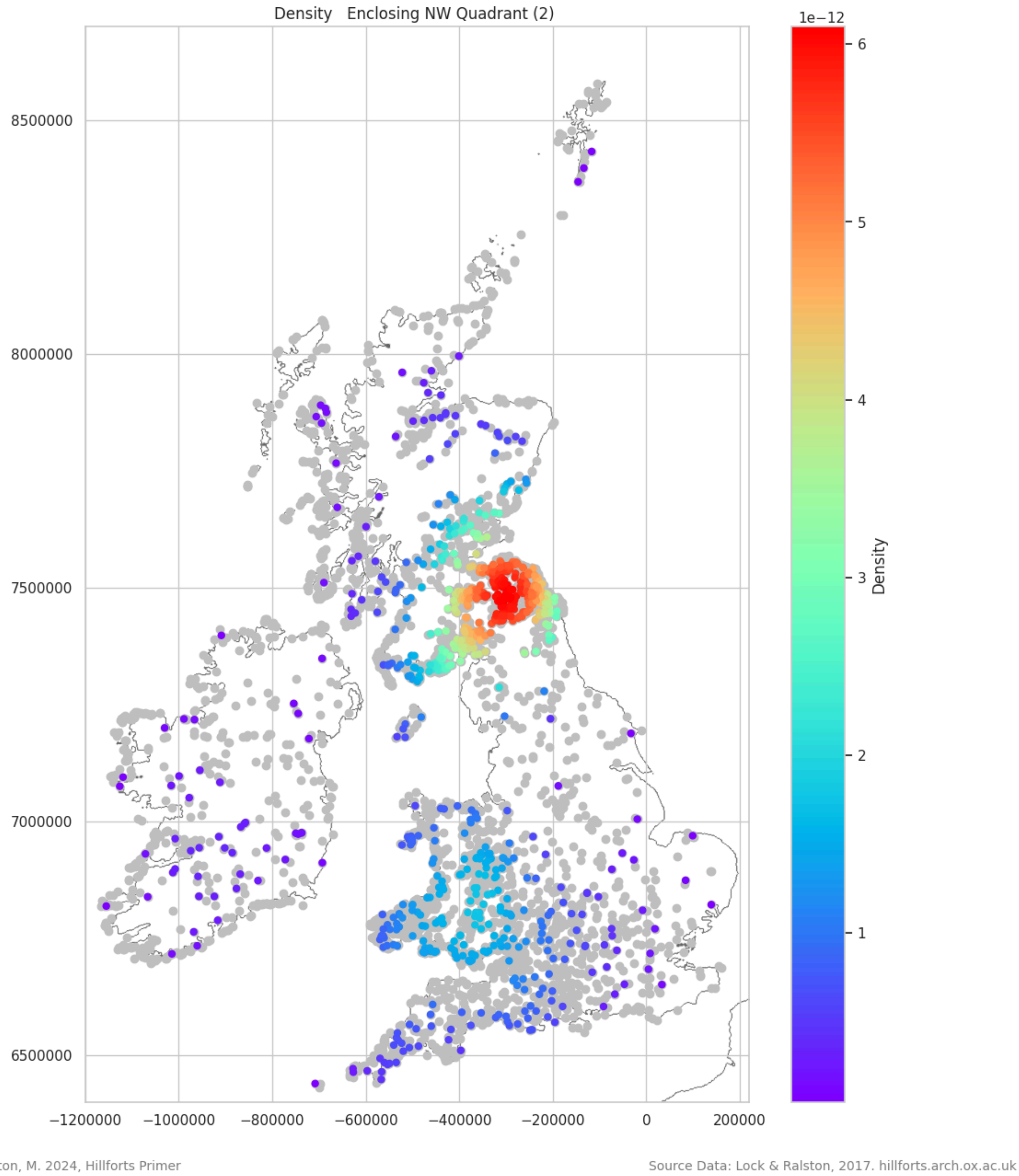
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
19.68%
```

NW Quadrant Data Density Mapped (2)

In [ ]:
```
plot_density_over_grey(two_nw_stats, 'Enclosing_NW_Quadrant (2)')
```

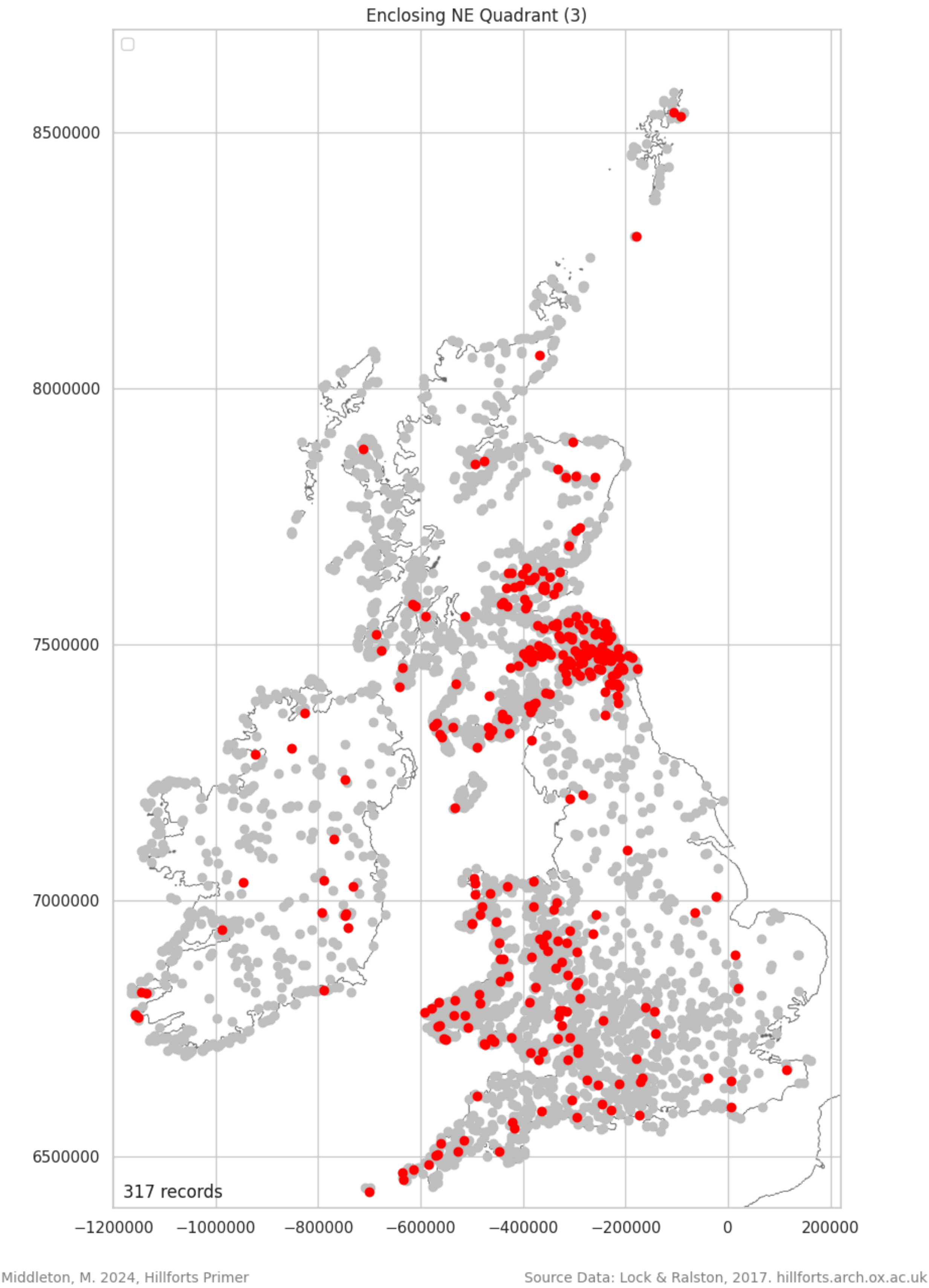


Quadrant Data Mapped (3)

There are no surprises in the quadrant data mapping three ramparts. As expected, it is focussed on the Northeast.

NE Quadrant Data Mapped (3)

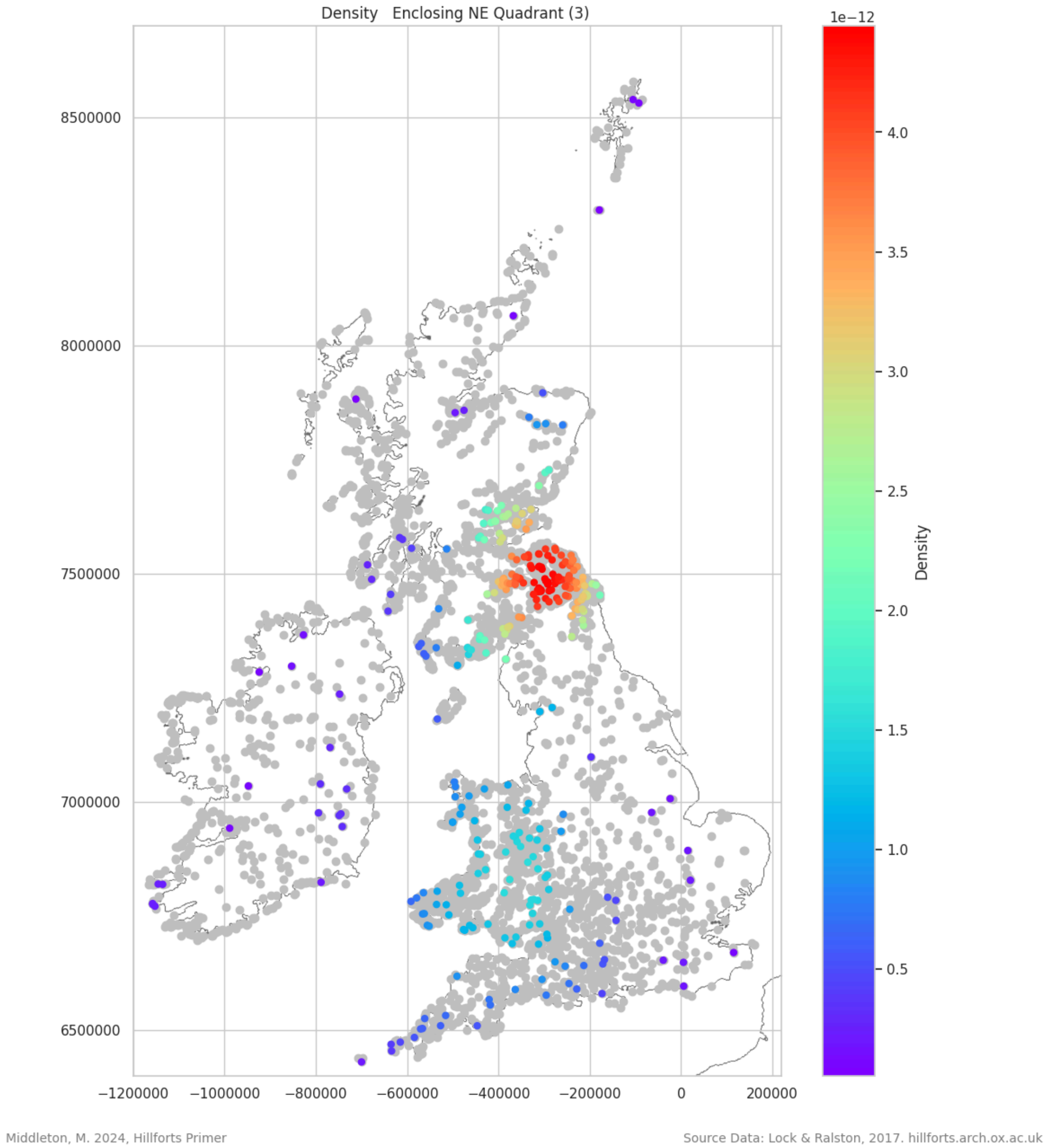
In [ ]:
```
three_ne = ne_quadrant_data[ne_quadrant_data['Enclosing_NE_Quadrant']==3].copy()
three_ne['Enclosing_NE_Quadrant'] = "Yes"
three_ne_stats = plot_over_grey(three_ne, 'Enclosing_NE_Quadrant', 'Yes', '(3)')
```




```
7.64%
```

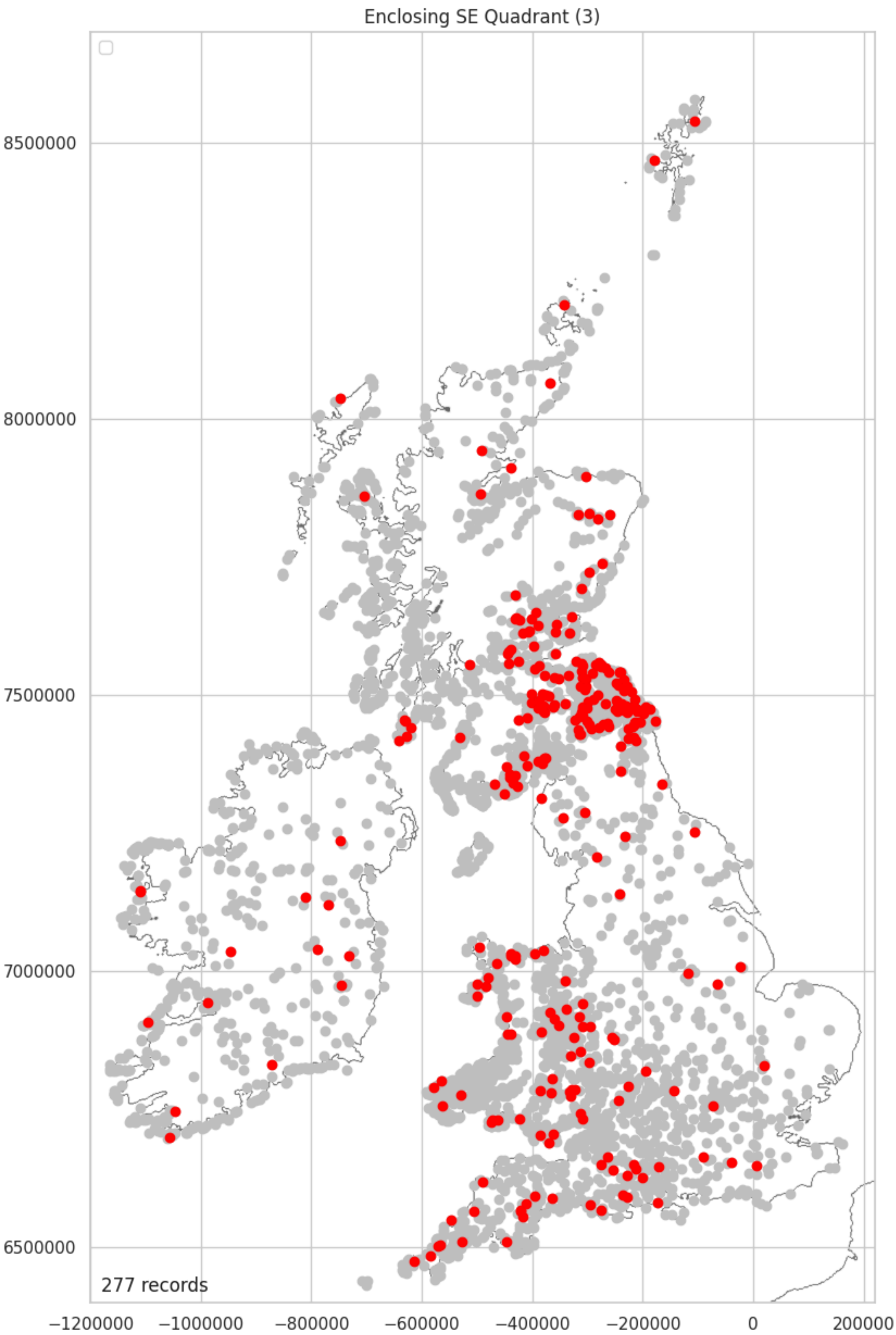
NE Quadrant Data Density Mapped (3)

```
In [ ]: plot_density_over_grey(three_ne_stats, 'Enclosing_NE_Quadrant (3)')
```



SE Quadrant Data Mapped (3)

In [ ]:
```
three_se = se_quadrant_data[se_quadrant_data['Enclosing_SE_Quadrant']==3].copy()
three_se['Enclosing_SE_Quadrant'] = "Yes"
three_se_stats = plot_over_grey(three_se, 'Enclosing_SE_Quadrant', 'Yes', '(3)')
```



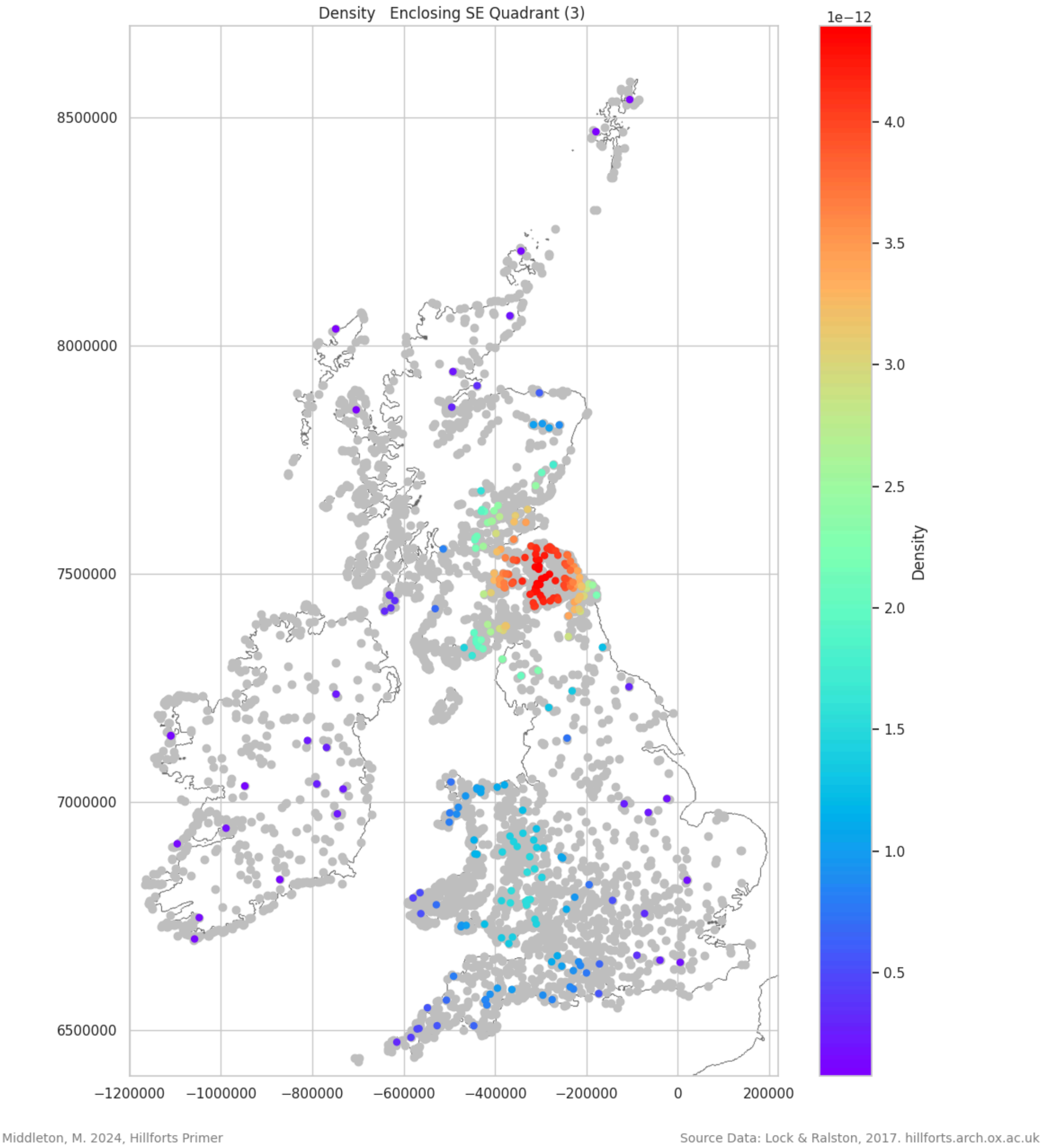
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
6.68%
```

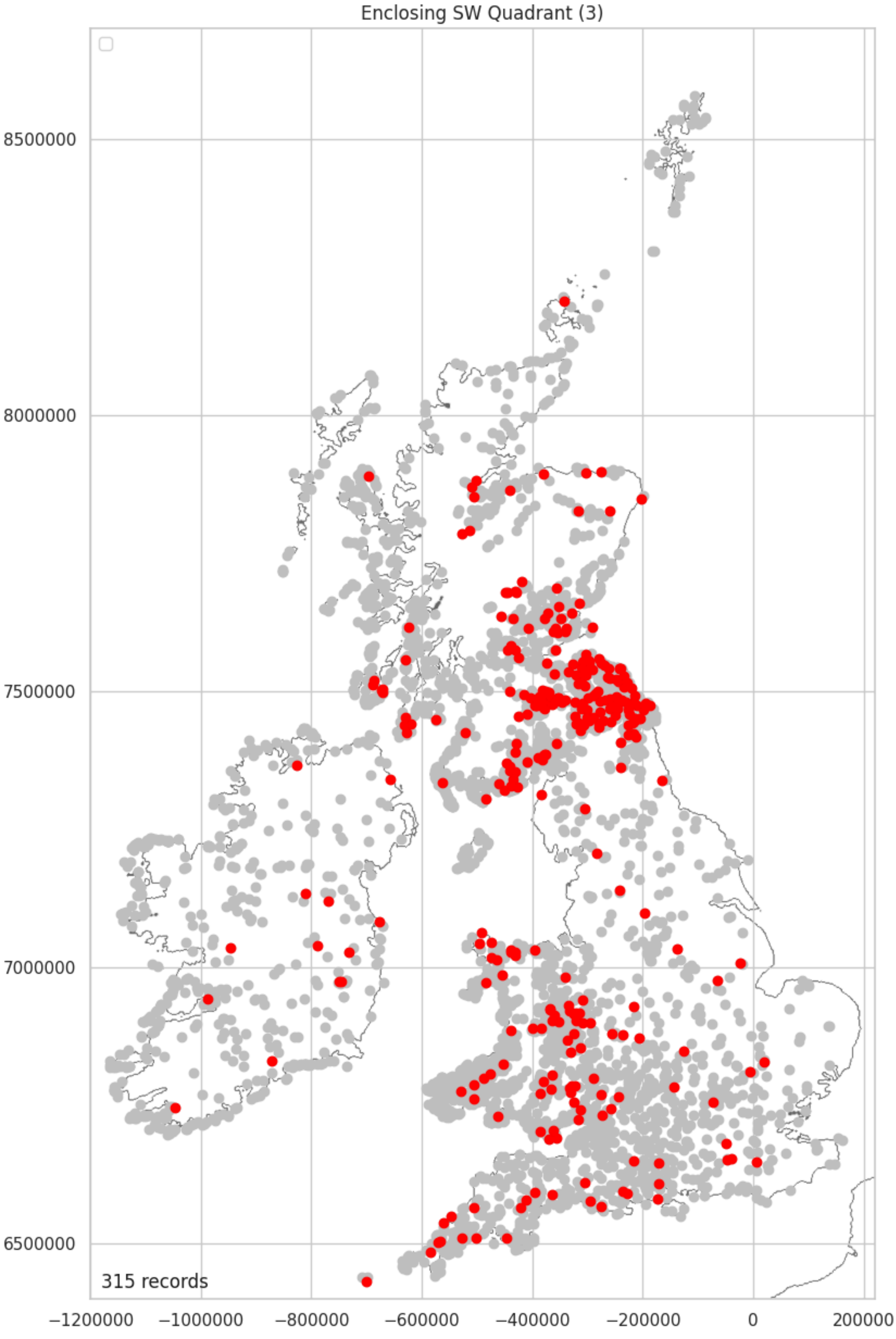
SE Quadrant Data Density Mapped (3)

```
In [ ]: plot_density_over_grey(three_se_stats, 'Enclosing_SE_Quadrant (3)')
```



SW Quadrant Data Mapped (3)

In [ ]:
```
three_sw = sw_quadrant_data[sw_quadrant_data['Enclosing_SW_Quadrant']==3].copy()
three_sw['Enclosing_SW_Quadrant'] = "Yes"
three_sw_stats = plot_over_grey(three_sw, 'Enclosing_SW_Quadrant', 'Yes', '(3)')
```



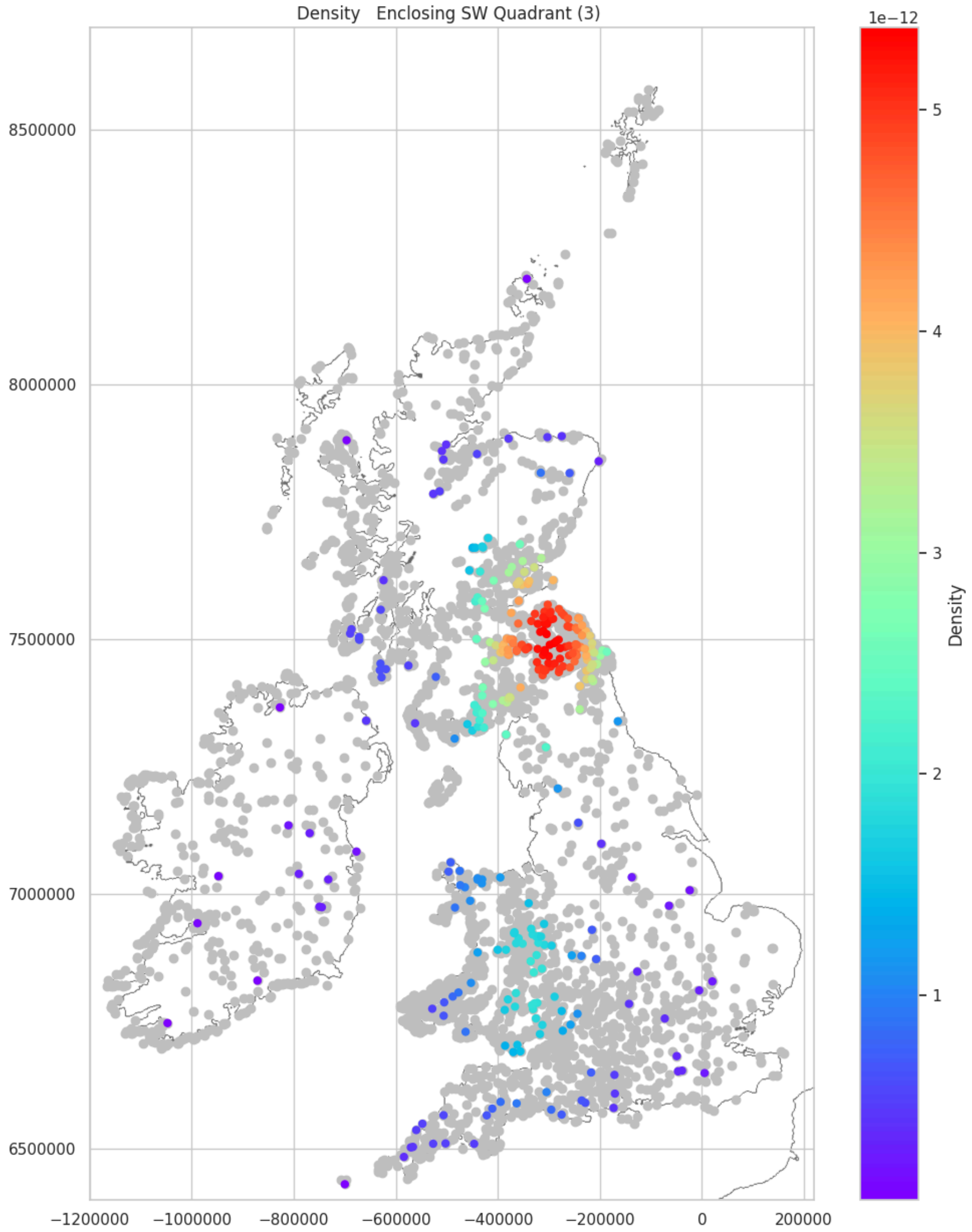
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

7.6%

SW Quadrant Data Density Mapped (3)

In [ ]:
```
plot_density_over_grey(three_sw_stats, 'Enclosing_SW_Quadrant (3)')
```

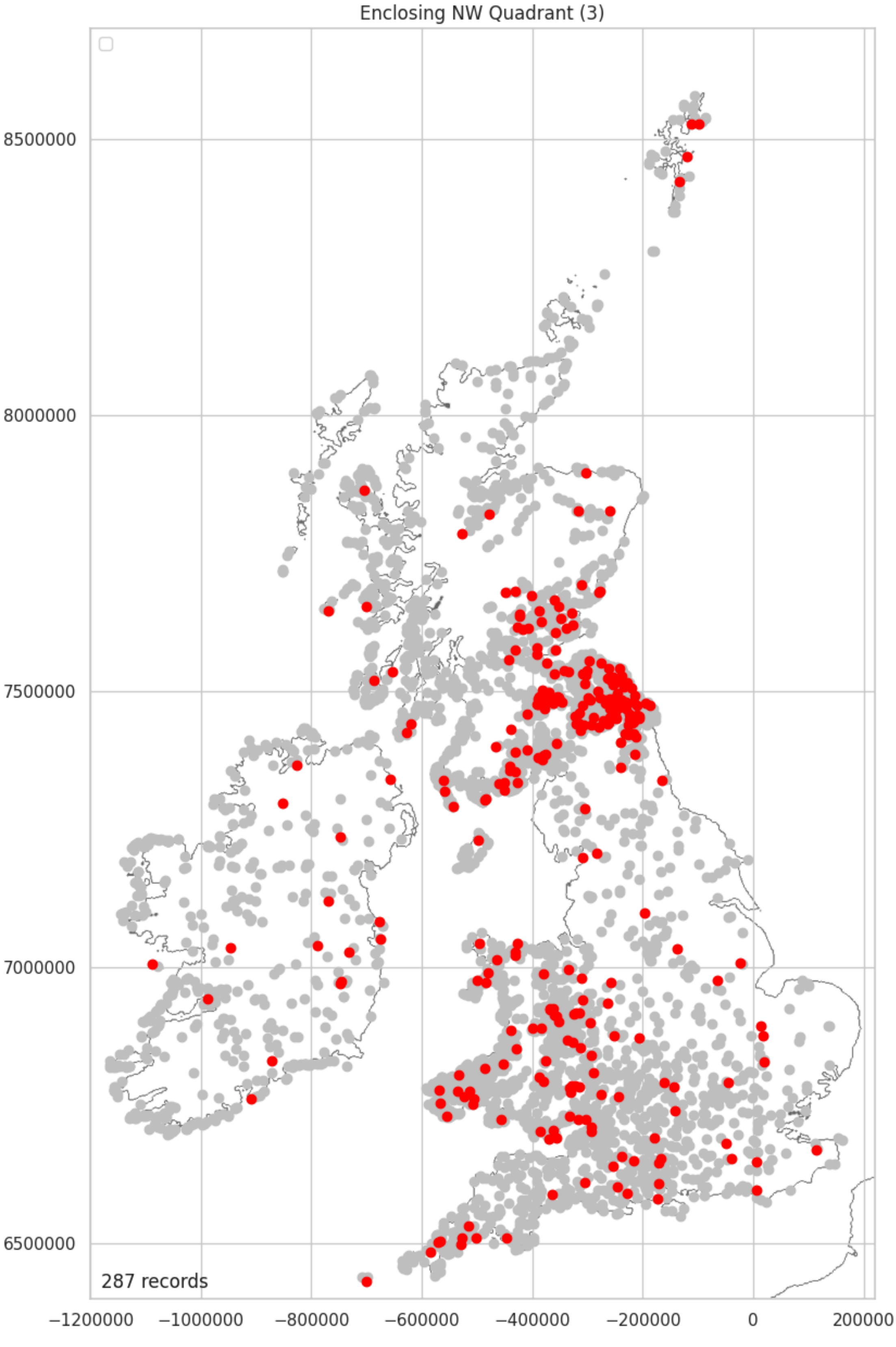


Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

NW Quadrant Data Mapped (3)

In [ ]:
```
three_nw = nw_quadrant_data[nw_quadrant_data['Enclosing_NW_Quadrant']==3].copy()
three_nw['Enclosing_NW_Quadrant'] = "Yes"
three_nw_stats = plot_over_grey(three_nw, 'Enclosing_NW_Quadrant', 'Yes', '(3)')
```



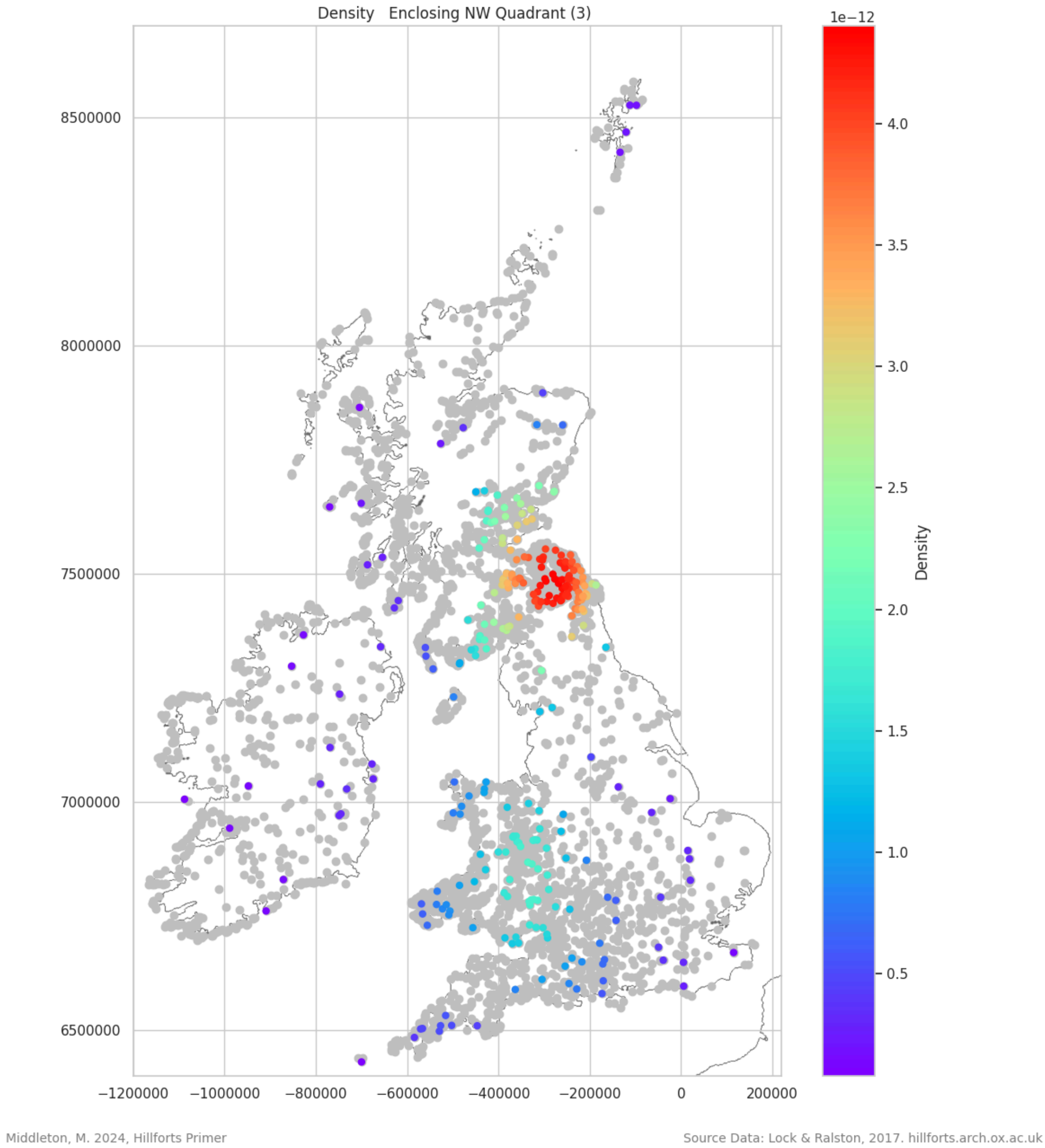
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
6.92%
```

NW Quadrant Data Density Mapped (3)

```
In [ ]: plot_density_over_grey(three_nw_stats, 'Enclosing_NW_Quadrant (3)')
```

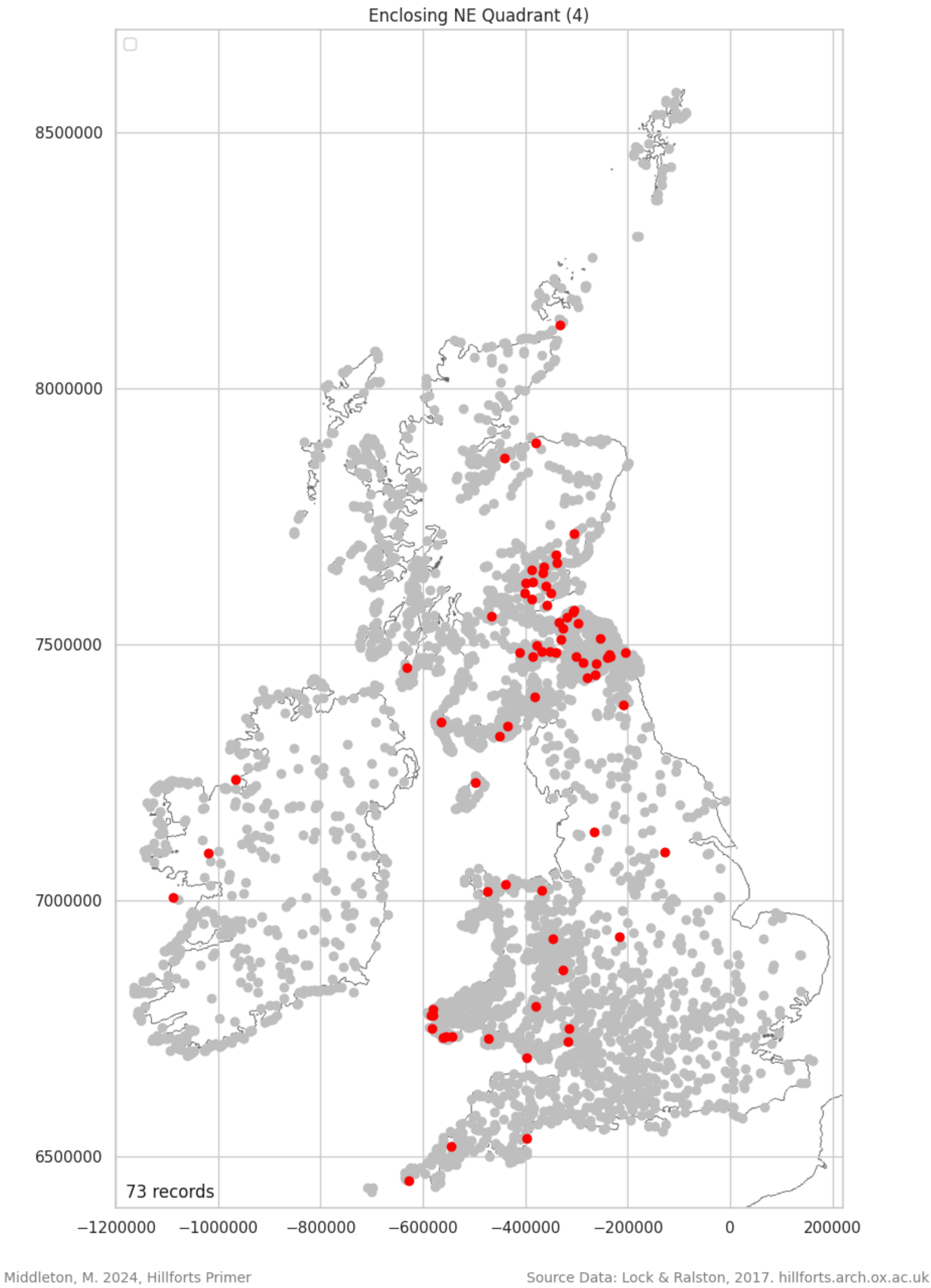


Quadrant Data Mapped (4)

As expected, the quadrant data mapping four ramparts is concentrated in the Northeast.

NE Quadrant Data Mapped (4)

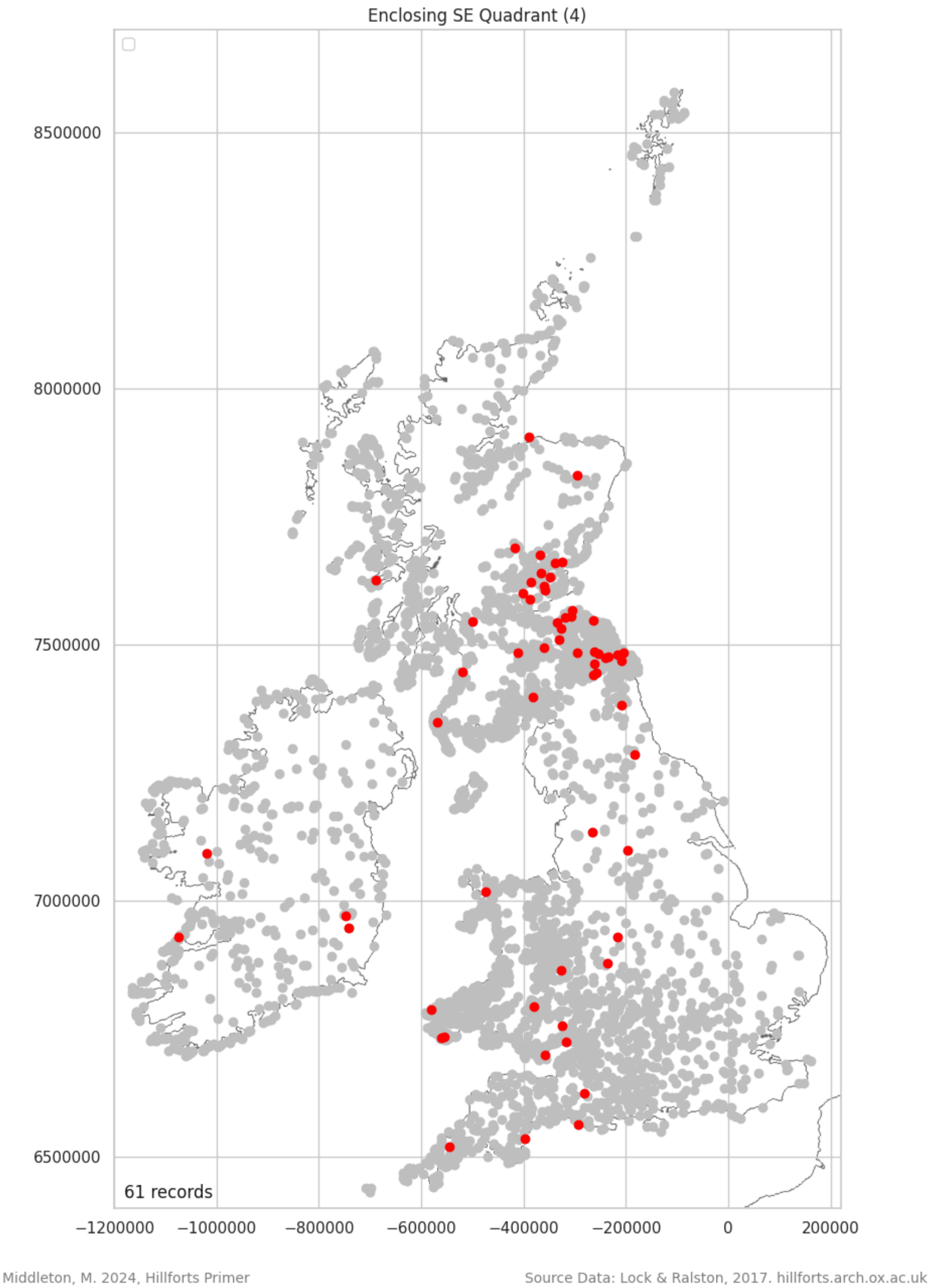
In [ ]:
```
four_ne = ne_quadrant_data[ne_quadrant_data['Enclosing_NE_Quadrant']==4].copy()
four_ne['Enclosing_NE_Quadrant'] = "Yes"
four_ne_stats = plot_over_grey(four_ne, 'Enclosing_NE_Quadrant', 'Yes', '(4)')
```




```
1.76%
```

SE Quadrant Data Mapped (4)

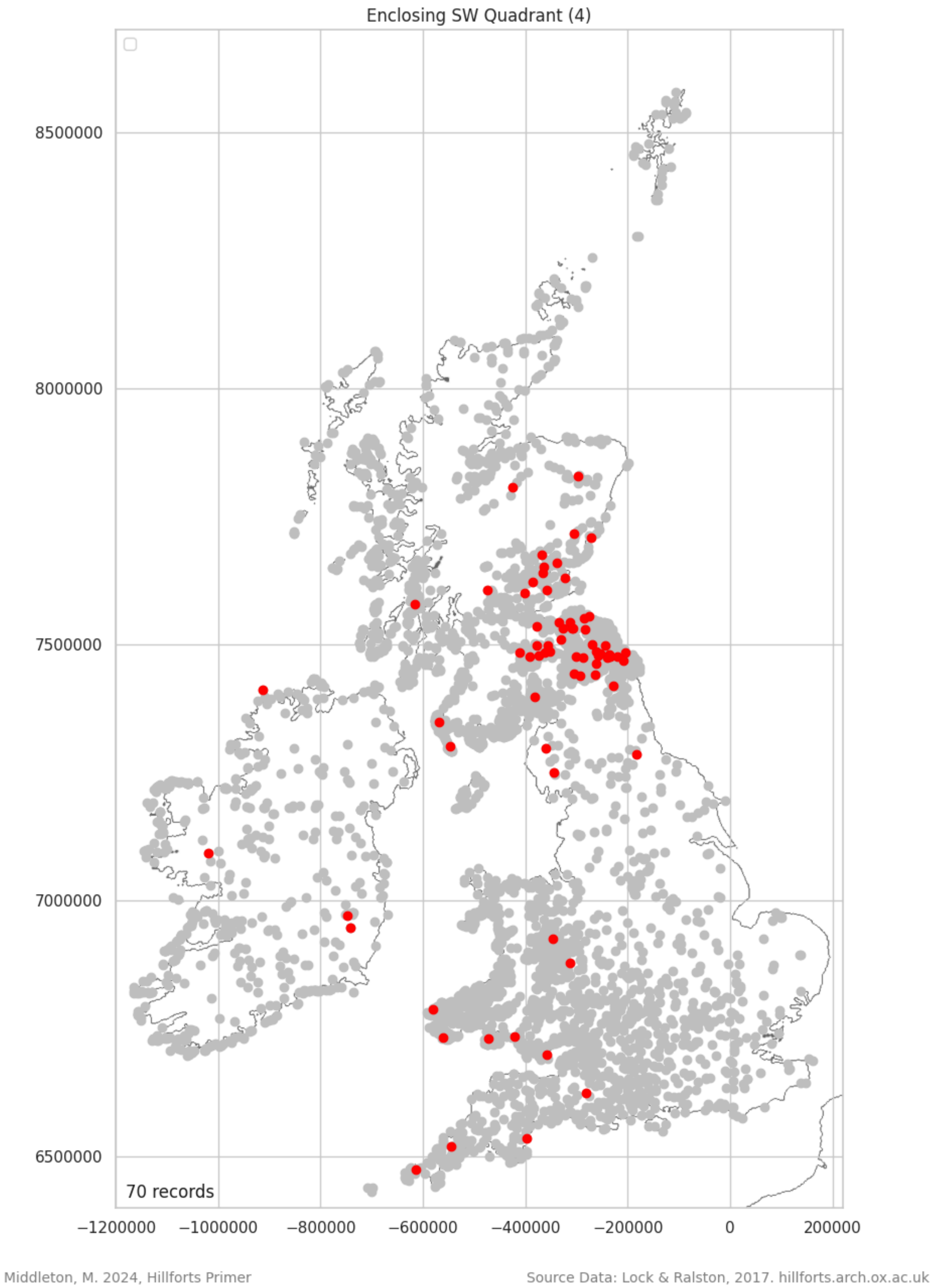
```
In [ ]: four_se = se_quadrant_data[se_quadrant_data['Enclosing_SE_Quadrant']==4].copy()
        four_se['Enclosing_SE_Quadrant'] = "Yes"
        four_se_stats = plot_over_grey(four_se, 'Enclosing_SE_Quadrant', 'Yes', '(4)')
```



1.47%

SW Quadrant Data Mapped (4)

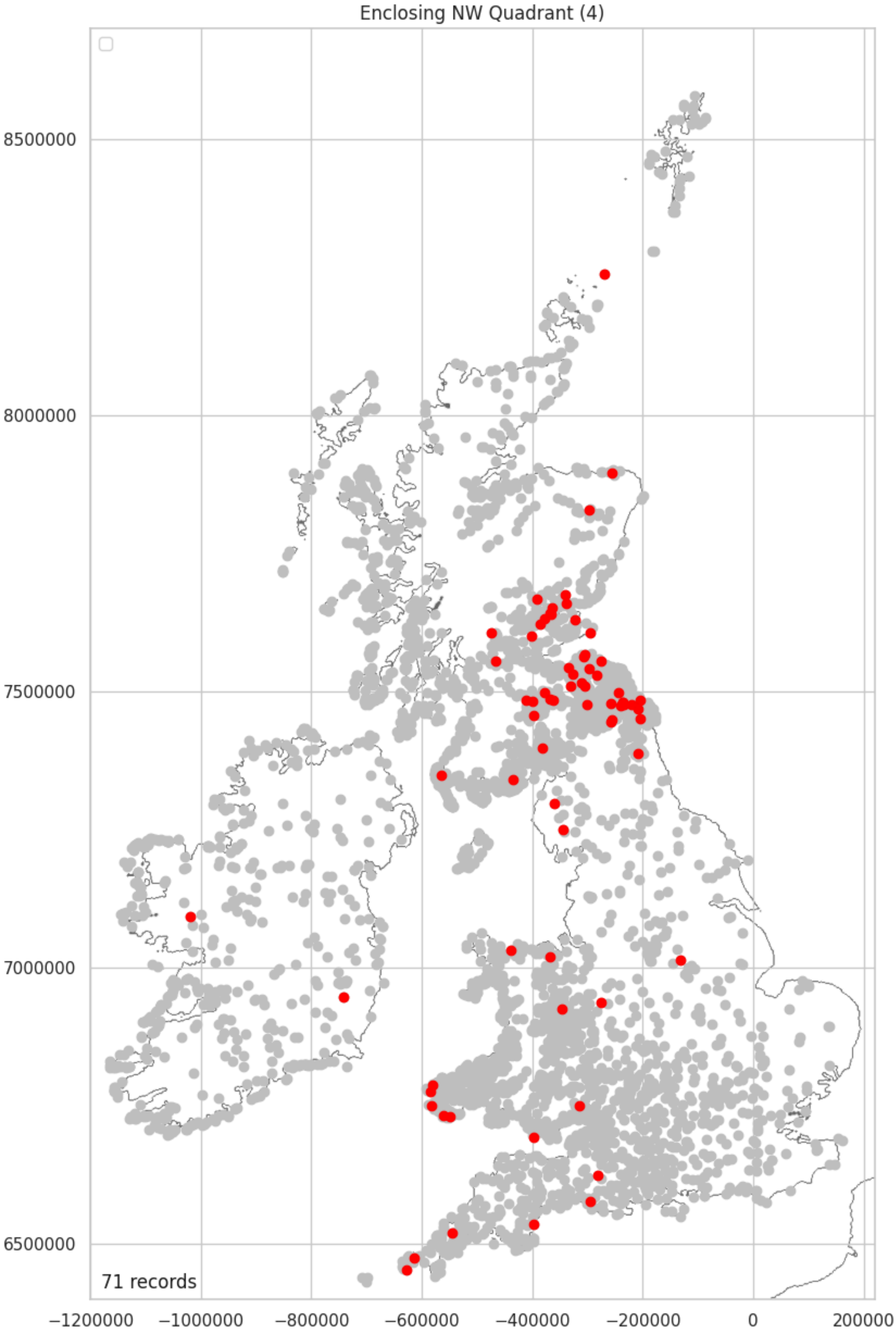
In [ ]:
```
four_sw = sw_quadrant_data[sw_quadrant_data['Enclosing_SW_Quadrant']==4].copy()
four_sw['Enclosing_SW_Quadrant'] = "Yes"
four_sw_stats = plot_over_grey(four_sw, 'Enclosing_SW_Quadrant', 'Yes', '(4)')
```




```
1.69%
```

NW Quadrant Data Mapped (4)

```
In [ ]: four_nw = nw_quadrant_data[nw_quadrant_data['Enclosing_NW_Quadrant']==4].copy()
        four_nw['Enclosing_NW_Quadrant'] = "Yes"
        four_nw_stats = plot_over_grey(four_nw, 'Enclosing_NW_Quadrant', 'Yes', '(4)')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
1.71%
```

Quadrant Data Mapped (5+)

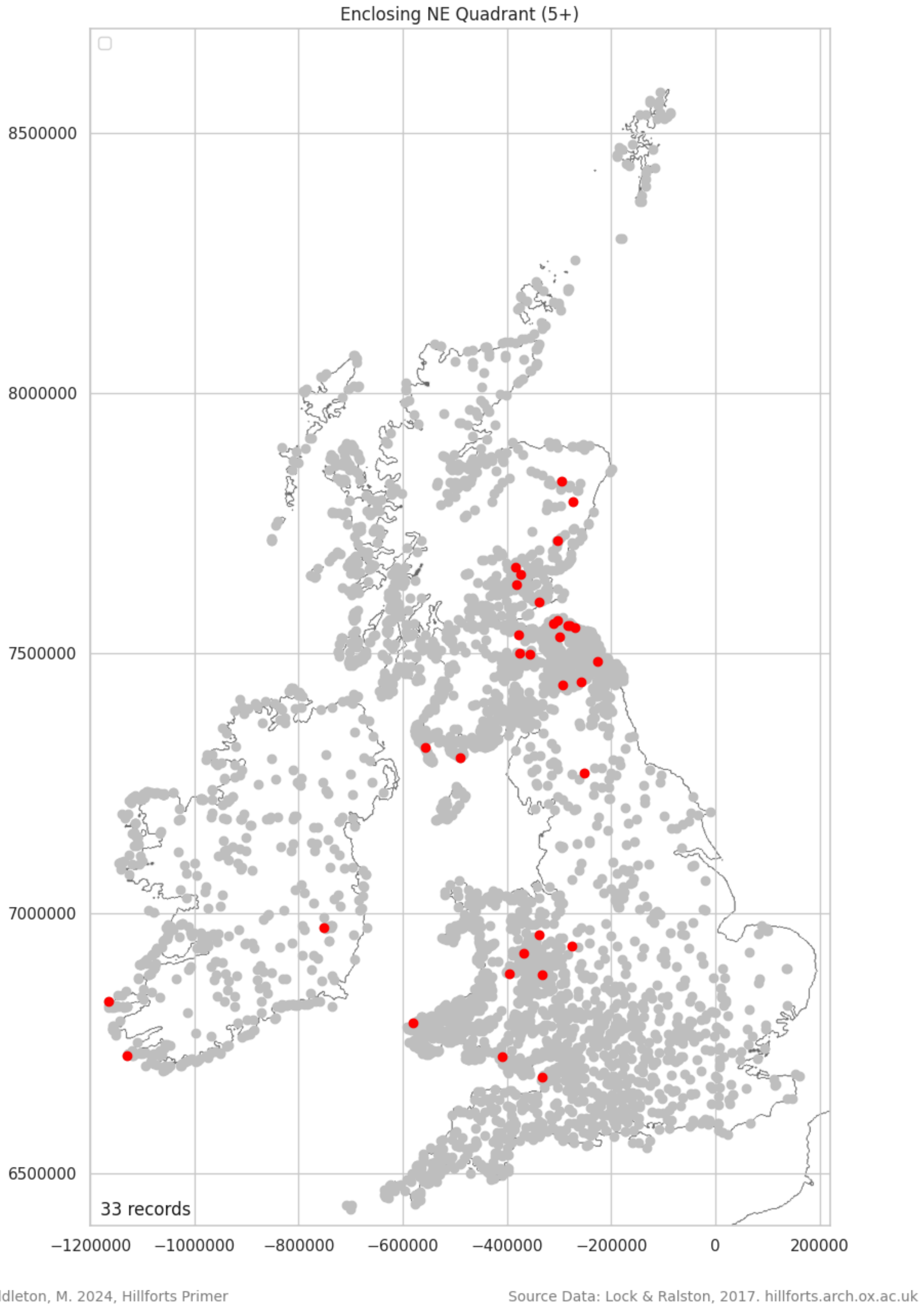
As expected, the quadrant data mapping five plus ramparts is concentrated in the Northeast.

NE Quadrant Data Mapped (5+)

In [ ]:

```
outliers_ne = \
ne_quadrant_data[ne_quadrant_data['Enclosing_NE_Quadrant']>4].copy()
outliers_ne['Enclosing_NE_Quadrant'] = "Yes"
```

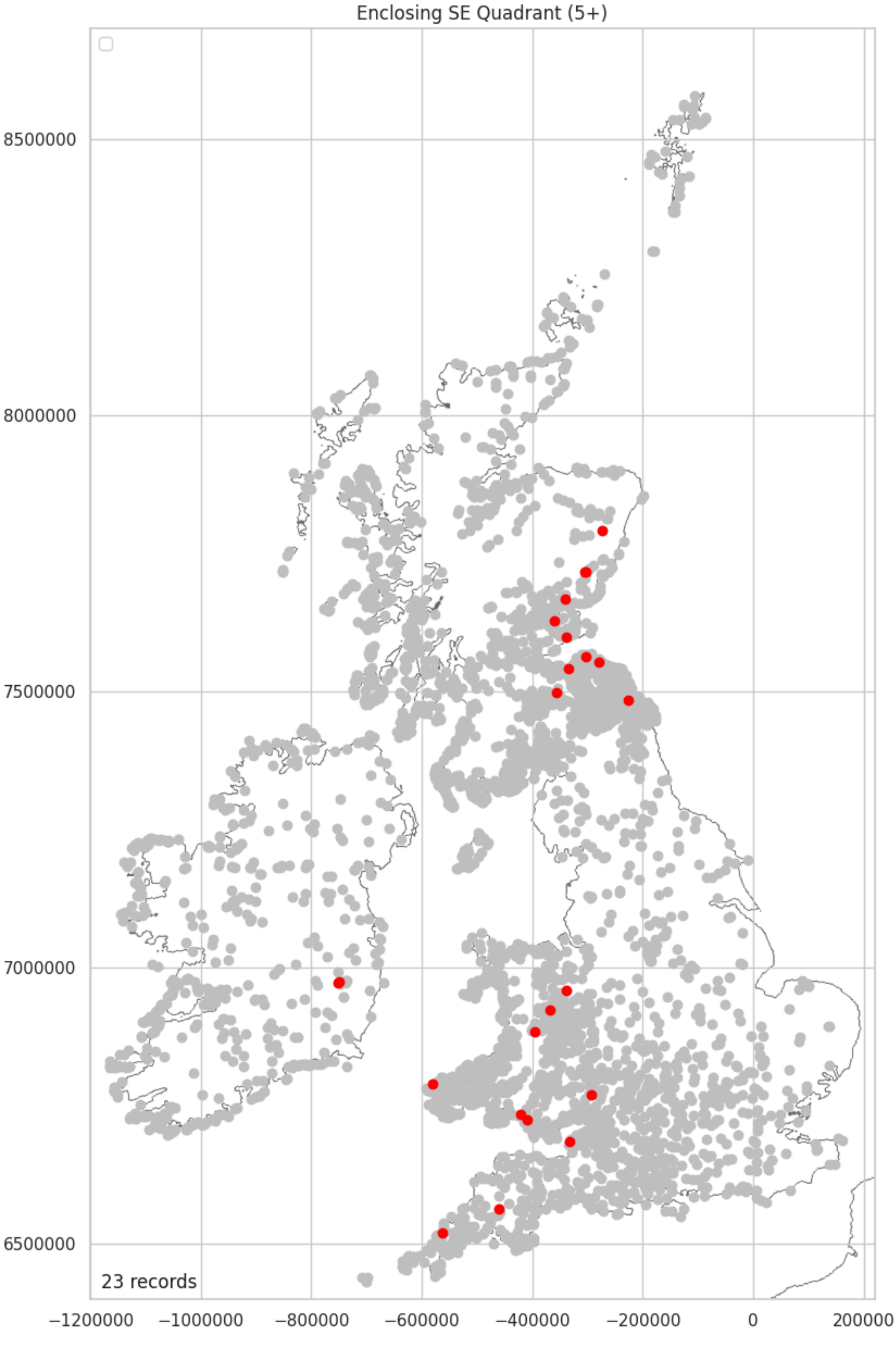
```
outliers_ne_stats = plot_over_grey(outliers_ne, 'Enclosing_NE_Quadrant', \
                                   'Yes', '(5+)')
```




```
0.8%
```

SE Quadrant Data Mapped (5+)

In [ ]:
```
outliers_se = \
se_quadrant_data[se_quadrant_data['Enclosing_SE_Quadrant']>4].copy()
outliers_se['Enclosing_SE_Quadrant'] = "Yes"
outliers_se_stats = plot_over_grey(outliers_se, 'Enclosing_SE_Quadrant', \
                                   'Yes', '(5+)')
```



Middleton, M. 2024, Hillforts Primer

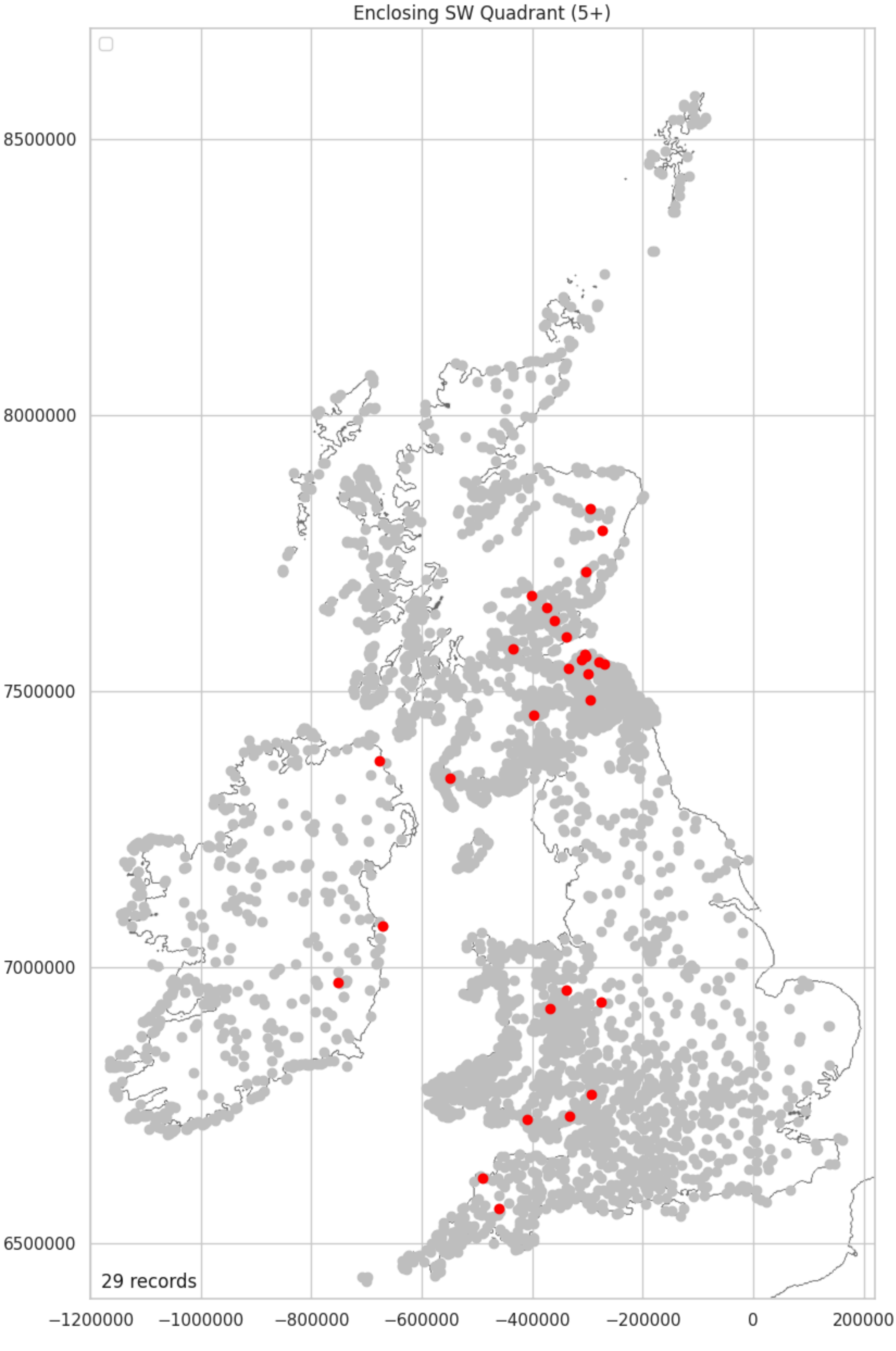
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
0.55%
```

SW Quadrant Data Mapped (5+)

In [ ]:

```
outliers_sw = \
sw_quadrant_data[sw_quadrant_data['Enclosing_SW_Quadrant']>4].copy()
outliers_sw['Enclosing_SW_Quadrant'] = "Yes"
outliers_sw_stats = plot_over_grey(outliers_sw, 'Enclosing_SW_Quadrant', \
                                   'Yes', '(5+)')
```



Middleton, M. 2024, Hillforts Primer

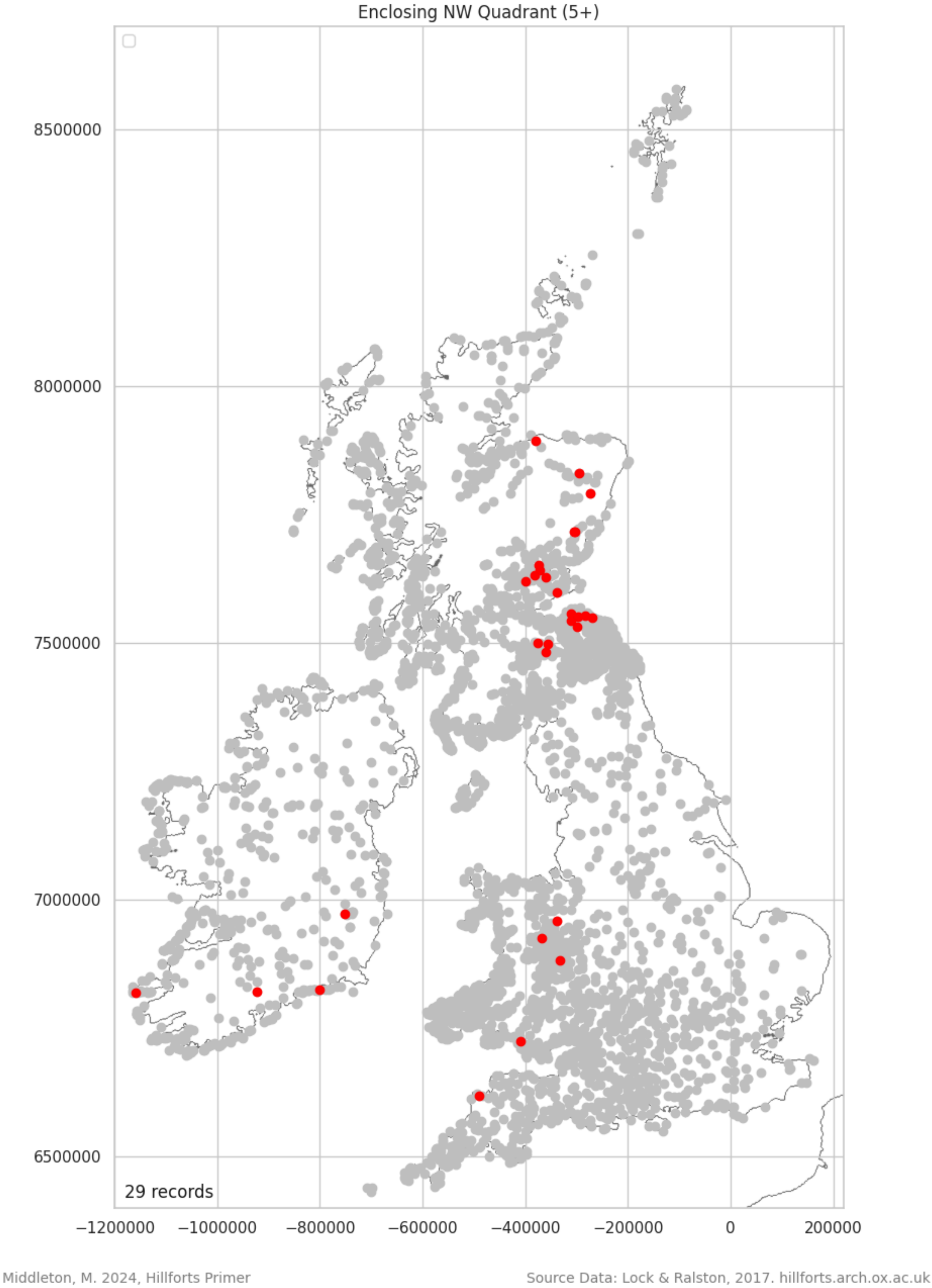
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

0.7%

NW Quadrant Data Mapped (5+)

In [ ]:

```
outliers_nw = \
nw_quadrant_data[nw_quadrant_data['Enclosing_NW_Quadrant']>4].copy()
outliers_nw['Enclosing_NW_Quadrant'] = "Yes"
outliers_nw_stats = plot_over_grey(outliers_nw, 'Enclosing_NW_Quadrant', \
                                   'Yes', '(5+)')
```

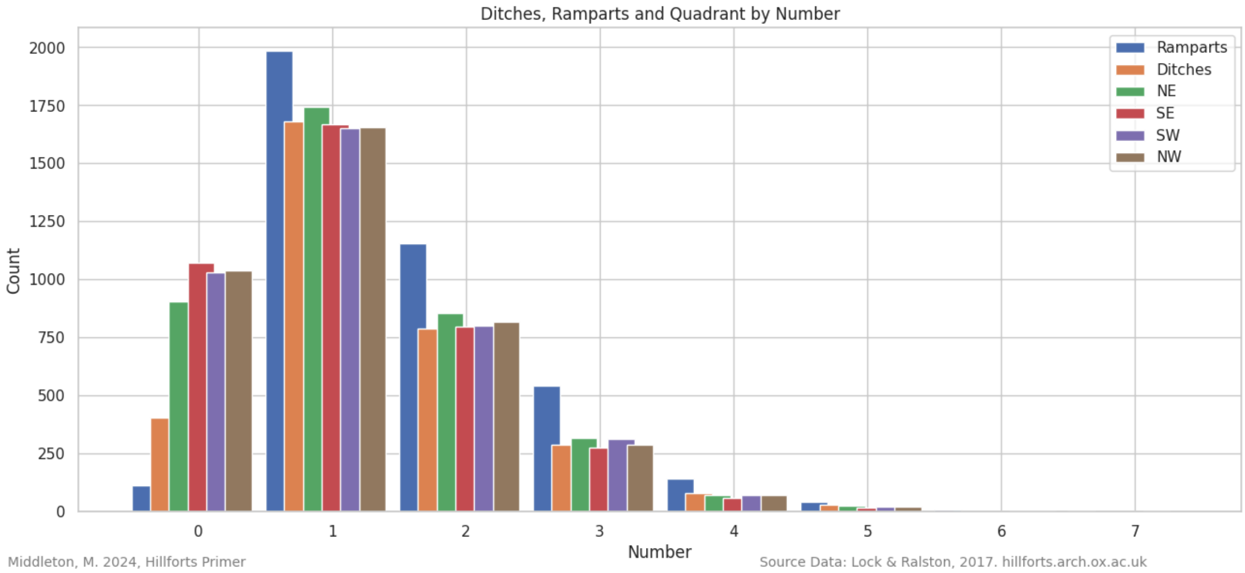



```
0.7%
```

Quadrant Data Plotted Against Ditches and Ramparts

As would be expected, the number of ramparts by quadrant roughly follows the distributions seen in the ramparts and ditches sections above.

```
In [ ]: plot_quadrents(all_ramparts,all_ditches,ne_quadrant_data,\
                       se_quadrant_data,sw_quadrant_data,nw_quadrant_data)
```



For the specific plots relating to ramparts and ditches see:

- Ramparts Plotted
- Ditches Plotted

In [ ]:
```
ne_quadrant_data['Enclosing_NE_Quadrant'].value_counts().sort_index()
```

Out[ ]:
```
0.0     904
1.0    1745
2.0     855
3.0     317
4.0      73
5.0      28
6.0       3
7.0       1
8.0       1
Name: Enclosing_NE_Quadrant, dtype: int64
```

In [ ]:
```
se_quadrant_data['Enclosing_SE_Quadrant'].value_counts().sort_index()
```

Out[ ]:
```
0.0    1072
1.0    1667
2.0     799
3.0     277
4.0      61
5.0      17
6.0       3
7.0       2
8.0       1
Name: Enclosing_SE_Quadrant, dtype: int64
```

In [ ]:
```
sw_quadrant_data['Enclosing_SW_Quadrant'].value_counts().sort_index()
```

Out[ ]:
```
0.0    1030
1.0    1650
2.0     802
3.0     315
4.0      70
5.0      20
6.0       5
7.0       4
Name: Enclosing_SW_Quadrant, dtype: int64
```

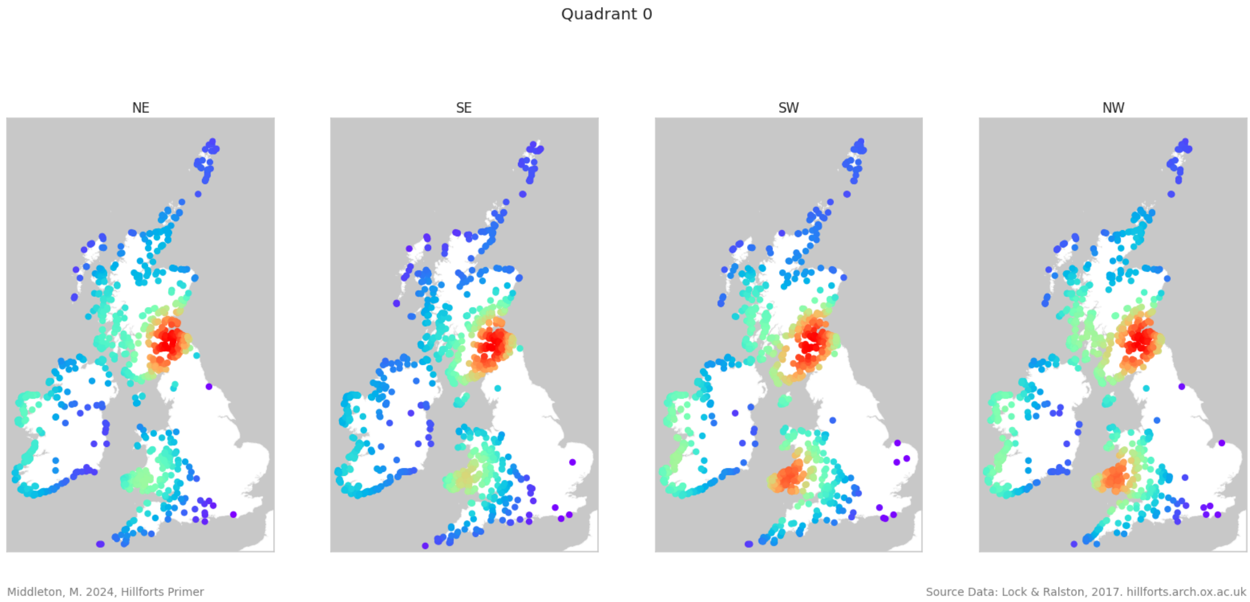
In [ ]:
```
nw_quadrant_data['Enclosing_NW_Quadrant'].value_counts().sort_index()
```

Out[ ]:
```
0.0     1040
1.0     1656
2.0      816
3.0      287
4.0       71
5.0       23
6.0        4
7.0        1
10.0       1
Name: Enclosing_NW_Quadrant, dtype: int64
```

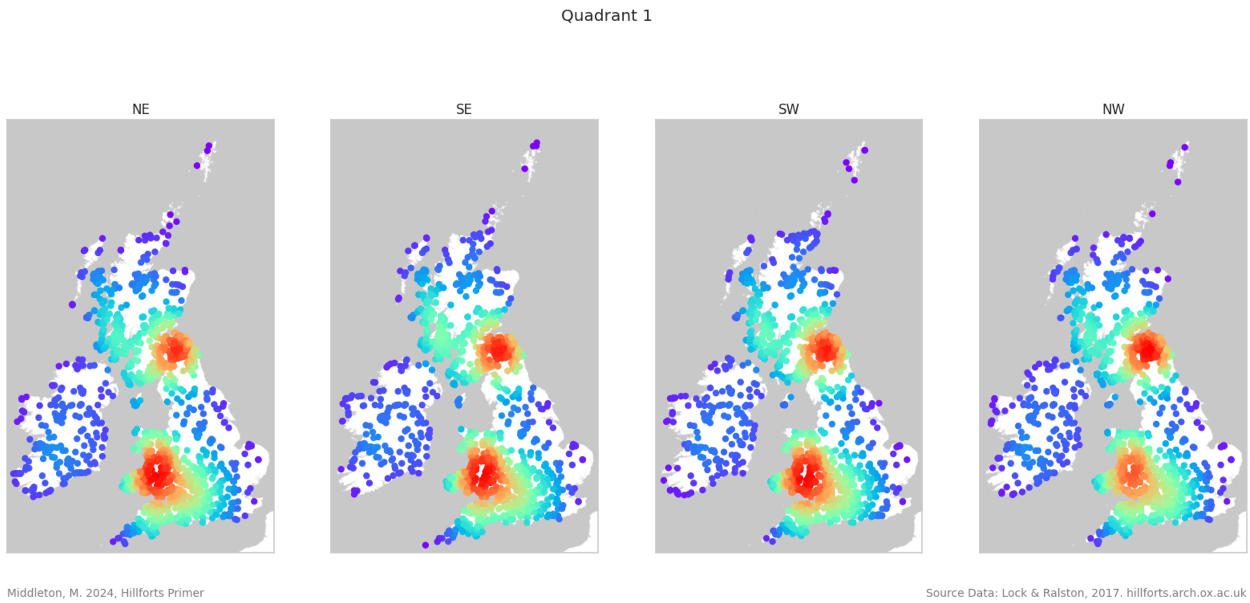
Quadrant Summary

Quadrant data is most influenced by local topography. This can be seen in Irish coastal forts, forts on the Pembrokeshire peninsula and the Northwestern hillforts all having less ramparts on their western, coastal sides. Other than this, large scale regional analysis provides little additional insight beyond that already discussed for ramparts and ditches above.

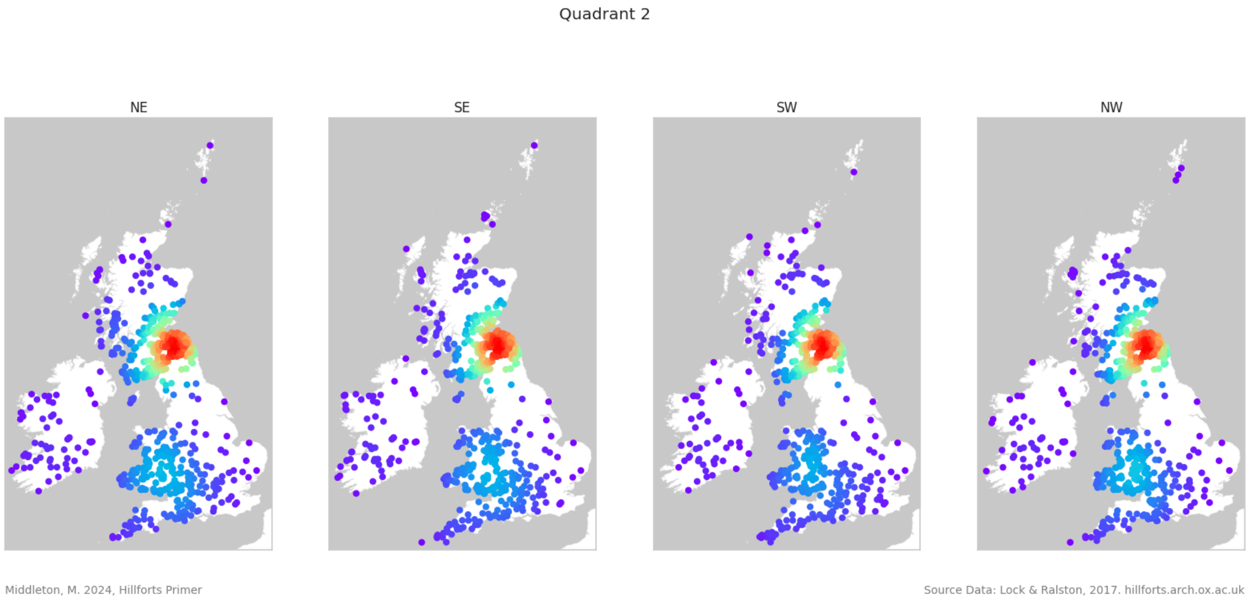
```
In [ ]: plot_density_over_grey_four(zero_ne_stats, zero_se_stats, zero_sw_stats, \
                                    zero_nw_stats, 'Quadrant 0')
```



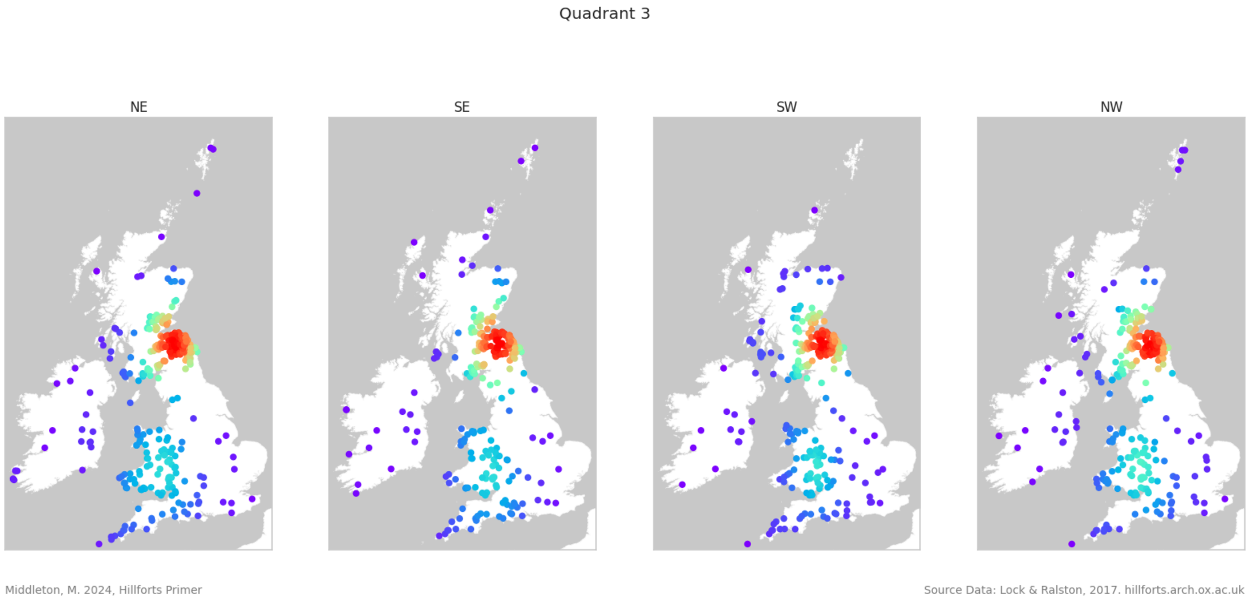

```
In [ ]: plot_density_over_grey_four(one_ne_stats, one_se_stats, one_sw_stats, \
                                    one_nw_stats, 'Quadrant 1')
```




```
In [ ]: plot_density_over_grey_four(two_ne_stats, two_se_stats, two_sw_stats, \
                                    two_nw_stats, 'Quadrant 2')
```



In [ ]:
```
plot_density_over_grey_four(three_ne_stats, three_se_stats, three_sw_stats, \
                            three_nw_stats, 'Quadrant 3')
```



Enclosing Text Data

There are eight Enclosing text features. All contain null values.

In [ ]:
```
enclosing_text_features = [
 'Enclosing_Summary',
 'Enclosing_Multiperiod_Comments',
 'Enclosing_Circuit_Comments',
 'Enclosing_Quadrant_Comments',
 'Enclosing_Surface_Comments',
 'Enclosing_Excavation_Comments',
 'Enclosing_Gang_Working_Comments',
 'Enclosing_Ditches_Comments']

enclosing_text_data = enclosing_data[enclosing_text_features].copy()
enclosing_text_data.head()
```

Out[ ]:

	Enclosing_Summary	Enclosing_Multiperiod_Comments	Enclosing_Circuit_Comments	Enclosing_Quadrant_Comments	Enclosing_Surface_Co
0	Univallate hillfort with complete circuit, but...	Univallate hillfort with complete circuit.	Single rampart continues around circuit.	NaN	Little surface evi features and
1	Defined differentially by single rampart to 5....	NaN	The ramparts are irregular which makes assessm...	NaN	Bank possibly Counterscarp bank
2	Three ramparts and ditches on the N. Although ...	NaN	Ramparts damaged and discontinuous. On the W t...	NaN	
3	Steep natural scarp artificially scarped with ...	Area not exact.	The ramparts are slight but complete the circuit.	NaN	Possible earthen bar berm to 7.
4	In Phase I, c. 3ha were enclosed by a slight b...	The site is very long and sinuous. Phased cons...	The ramparts of the Phase II overall enclosure...	NaN	Surface evidence of an bank and

In [ ]:
```
enclosing_text_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 8 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   Enclosing_Summary                4138 non-null   object
 1   Enclosing_Multiperiod_Comments   1016 non-null   object
 2   Enclosing_Circuit_Comments       1201 non-null   object
 3   Enclosing_Quadrant_Comments      58 non-null     object
 4   Enclosing_Surface_Comments       1236 non-null   object
 5   Enclosing_Excavation_Comments    526 non-null    object
 6   Enclosing_Gang_Working_Comments  48 non-null     object
 7   Enclosing_Ditches_Comments       1499 non-null   object
dtypes: object(8)
memory usage: 259.3+ KB
```

Entrance Text Data - Resolve Null Values

Test for 'NA'.

In [ ]:
```
test_cat_list_for_NA(enclosing_text_data, enclosing_text_features)
```

```
Enclosing_Summary 0
Enclosing_Multiperiod_Comments 0
Enclosing_Circuit_Comments 0
Enclosing_Quadrant_Comments 0
Enclosing_Surface_Comments 0
Enclosing_Excavation_Comments 0
Enclosing_Gang_Working_Comments 0
Enclosing_Ditches_Comments 0
```

Fill null values with 'NA'.

In [ ]:
```
enclosing_text_data = \
update_cat_list_for_NA(enclosing_text_data, enclosing_text_features)
enclosing_text_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 8 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   Enclosing_Summary                4147 non-null   object
 1   Enclosing_Multiperiod_Comments   4147 non-null   object
 2   Enclosing_Circuit_Comments       4147 non-null   object
 3   Enclosing_Quadrant_Comments      4147 non-null   object
 4   Enclosing_Surface_Comments       4147 non-null   object
 5   Enclosing_Excavation_Comments    4147 non-null   object
 6   Enclosing_Gang_Working_Comments  4147 non-null   object
 7   Enclosing_Ditches_Comments       4147 non-null   object
dtypes: object(8)
memory usage: 259.3+ KB
```

Enclosing Encodeable Data

There are 44 Enclosing encodable features. Non contain null values.

In [ ]:
```python
enclosing_encodeable_features = [
 'Enclosing_Multiperiod',
 'Enclosing_Circuit',
 'Enclosing_Current_Part_Uni',
 'Enclosing_Current_Uni',
 'Enclosing_Current_Part_Bi',
 'Enclosing_Current_Bi',
 'Enclosing_Current_Part_Multi',
 'Enclosing_Current_Multi',
 'Enclosing_Current_Unknown',
 'Enclosing_Period_Part_Uni',
 'Enclosing_Period_Uni',
 'Enclosing_Period_Part_Bi',
 'Enclosing_Period_Bi',
 'Enclosing_Period_Part_Multi',
 'Enclosing_Period_Multi',
 'Enclosing_Surface_None',
 'Enclosing_Surface_Bank',
 'Enclosing_Surface_Wall',
 'Enclosing_Surface_Rubble',
 'Enclosing_Surface_Walk',
 'Enclosing_Surface_Timber',
 'Enclosing_Surface_Vitrification',
 'Enclosing_Surface_Burning',
 'Enclosing_Surface_Palisade',
 'Enclosing_Surface_Counter_Scarp',
 'Enclosing_Surface_Berm',
 'Enclosing_Surface_Unfinished',
 'Enclosing_Surface_Other',
 'Enclosing_Excavation_Nothing',
 'Enclosing_Excavation_Bank',
 'Enclosing_Excavation_Wall',
 'Enclosing_Excavation_Murus',
 'Enclosing_Excavation_Timber_Framed',
 'Enclosing_Excavation_Timber_Laced',
 'Enclosing_Excavation_Vitrification',
 'Enclosing_Excavation_Burning',
 'Enclosing_Excavation_Palisade',
 'Enclosing_Excavation_Counter_Scarp',
 'Enclosing_Excavation_Berm',
 'Enclosing_Excavation_Unfinished',
 'Enclosing_Excavation_No_Known',
 'Enclosing_Excavation_Other',
 'Enclosing_Gang_Working',
 'Enclosing_Ditches']

enclosing_encodeable_data = enclosing_data[enclosing_encodeable_features].copy()
enclosing_encodeable_data.head()
```

Out[ ]:

	Enclosing_Multiperiod	Enclosing_Circuit	Enclosing_Current_Part_Uni	Enclosing_Current_Uni	Enclosing_Current_Part_Bi	Enclosing_Current_
0	No	Yes	No	Yes	No	N
1	No	Yes	No	Yes	No	N
2	No	No	Yes	No	Yes	N
3	No	Yes	No	Yes	No	N
4	Yes	Yes	No	No	No	Y

In [ ]:
```python
enclosing_encodeable_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 44 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   Enclosing_Multiperiod            4147 non-null   object
 1   Enclosing_Circuit                4147 non-null   object
 2   Enclosing_Current_Part_Uni       4147 non-null   object
 3   Enclosing_Current_Uni            4147 non-null   object
 4   Enclosing_Current_Part_Bi        4147 non-null   object
 5   Enclosing_Current_Bi             4147 non-null   object
 6   Enclosing_Current_Part_Multi     4147 non-null   object
 7   Enclosing_Current_Multi          4147 non-null   object
 8   Enclosing_Current_Unknown        4147 non-null   object
 9   Enclosing_Period_Part_Uni        4147 non-null   object
 10  Enclosing_Period_Uni             4147 non-null   object
 11  Enclosing_Period_Part_Bi         4147 non-null   object
 12  Enclosing_Period_Bi              4147 non-null   object
 13  Enclosing_Period_Part_Multi      4147 non-null   object
 14  Enclosing_Period_Multi           4147 non-null   object
 15  Enclosing_Surface_None           4147 non-null   object
 16  Enclosing_Surface_Bank           4147 non-null   object
 17  Enclosing_Surface_Wall           4147 non-null   object
 18  Enclosing_Surface_Rubble         4147 non-null   object
 19  Enclosing_Surface_Walk           4147 non-null   object
 20  Enclosing_Surface_Timber         4147 non-null   object
 21  Enclosing_Surface_Vitrification  4147 non-null   object
 22  Enclosing_Surface_Burning        4147 non-null   object
 23  Enclosing_Surface_Palisade       4147 non-null   object
 24  Enclosing_Surface_Counter_Scarp  4147 non-null   object
 25  Enclosing_Surface_Berm           4147 non-null   object
 26  Enclosing_Surface_Unfinished     4147 non-null   object
 27  Enclosing_Surface_Other          4147 non-null   object
 28  Enclosing_Excavation_Nothing     4147 non-null   object
 29  Enclosing_Excavation_Bank        4147 non-null   object
 30  Enclosing_Excavation_Wall        4147 non-null   object
 31  Enclosing_Excavation_Murus       4147 non-null   object
 32  Enclosing_Excavation_Timber_Framed  4147 non-null   object
 33  Enclosing_Excavation_Timber_Laced  4147 non-null   object
 34  Enclosing_Excavation_Vitrification  4147 non-null   object
 35  Enclosing_Excavation_Burning     4147 non-null   object
 36  Enclosing_Excavation_Palisade    4147 non-null   object
 37  Enclosing_Excavation_Counter_Scarp  4147 non-null   object
 38  Enclosing_Excavation_Berm        4147 non-null   object
 39  Enclosing_Excavation_Unfinished  4147 non-null   object
 40  Enclosing_Excavation_No_Known    4147 non-null   object
 41  Enclosing_Excavation_Other       4147 non-null   object
 42  Enclosing_Gang_Working           4147 non-null   object
 43  Enclosing_Ditches                4147 non-null   object
dtypes: object(44)
memory usage: 1.4+ MB
```

Enclosing Multiperiod

528 hillforts (12.73%) are recorded as being multiperiod.

In [ ]:
```
multiperiod_counts = \
enclosing_encodeable_data['Enclosing_Multiperiod'].value_counts()
multiperiod_counts
```

Out[ ]:
```
No     3619
Yes     528
Name: Enclosing_Multiperiod, dtype: int64
```

In [ ]:
```
round(multiperiod_counts[1]/len(enclosing_encodeable_data)*100,2)
```

Out[ ]:
```
12.73
```

In [ ]:
```
location_enclosing_encodeable_data = \
pd.merge(location_numeric_data_short, enclosing_encodeable_data, \
         left_index=True, right_index=True)
```

In [ ]:
```
location_enclosing_encodeable_data_ne = \
pd.merge(north_east.reset_index(), enclosing_encodeable_data, \
         left_on='uid', right_index=True)
location_enclosing_encodeable_data_ne = \
pd.merge(name_and_number, location_enclosing_encodeable_data_ne, \
         left_index=True, right_on='uid')
```

In [ ]:
```
location_enclosing_encodeable_data_nw = \
pd.merge(north_west.reset_index(), enclosing_encodeable_data, \
         left_on='uid', right_index=True)
location_enclosing_encodeable_data_nw = \
```

```
pd.merge(name_and_number, location_enclosing_encodeable_data_nw, \
         left_index=True, right_on='uid')
```

In [ ]:
```
location_enclosing_encodeable_data_ireland_n = \
pd.merge(north_ireland.reset_index(), enclosing_encodeable_data, \
         left_on='uid', right_index=True)
location_enclosing_encodeable_data_ireland_n = \
pd.merge(name_and_number, location_enclosing_encodeable_data_ireland_n, \
         left_index=True, right_on='uid')
```

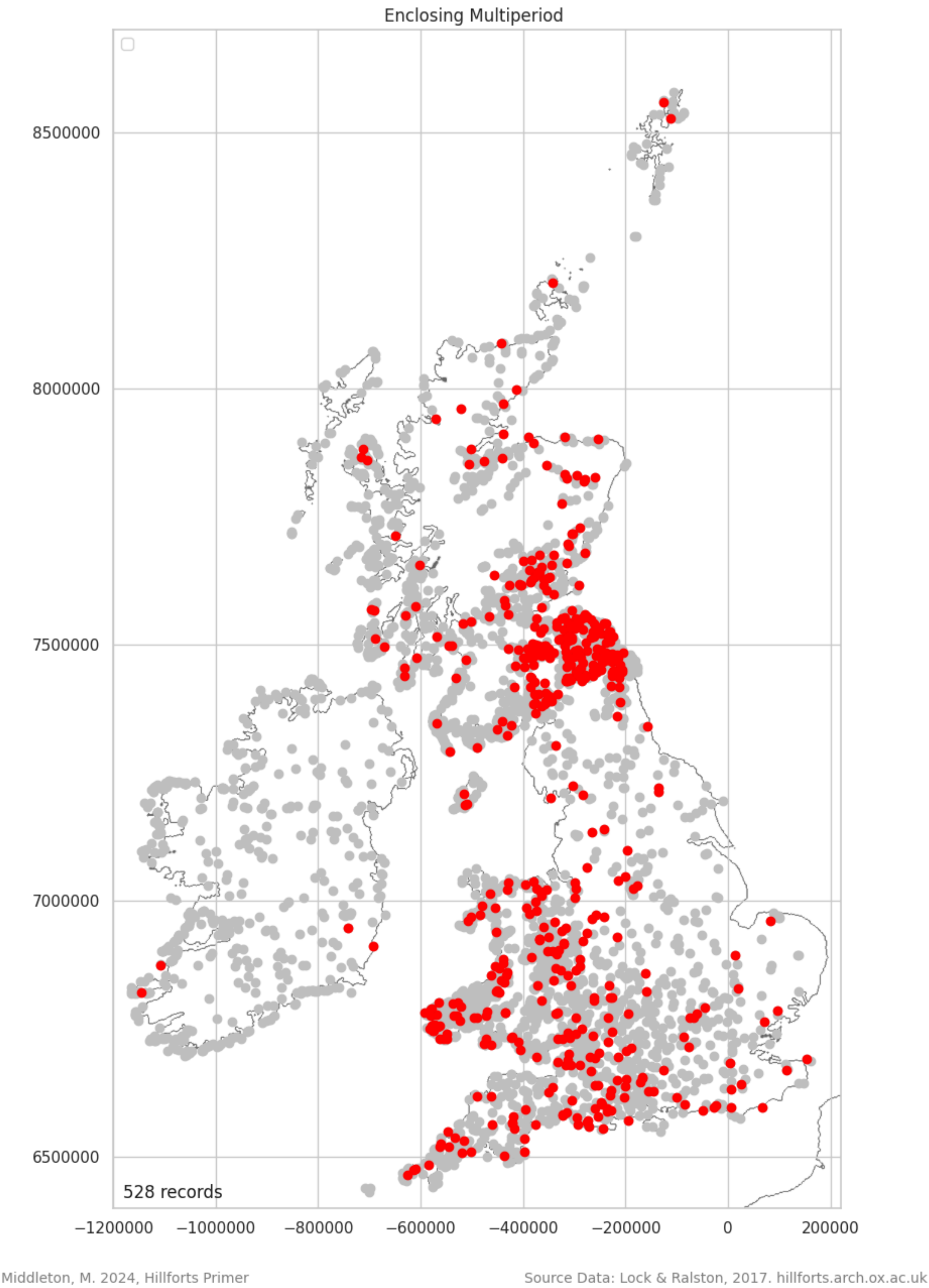
In [ ]:
```
location_enclosing_encodeable_data_ireland_s = \
pd.merge(south_ireland.reset_index(), enclosing_encodeable_data, \
         left_on='uid', right_index=True)
location_enclosing_encodeable_data_ireland_s = \
pd.merge(name_and_number, location_enclosing_encodeable_data_ireland_s, \
         left_index=True, right_on='uid')
```

In [ ]:
```
location_enclosing_encodeable_data_south = \
pd.merge(south, enclosing_encodeable_data, left_on='uid', right_index=True)
location_enclosing_encodeable_data_south = \
pd.merge(name_and_number, location_enclosing_encodeable_data_south, \
         left_index=True, right_on='uid')
```

Enclosing Multiperiod Mapped

There is an obvious recording bias in this data over Ireland. Having seen the spread of dating information – with the main phases of occupation being from 800 BC to AD400 (See: Part 3: Dating Data) – it is unlikely that only 12.73% of all forts have multiperiod occupation so, there is most likely, a recording bias across this entire class.

In [ ]:
```
multiperiod_data = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Multiperiod', \
               'Yes', '')
```

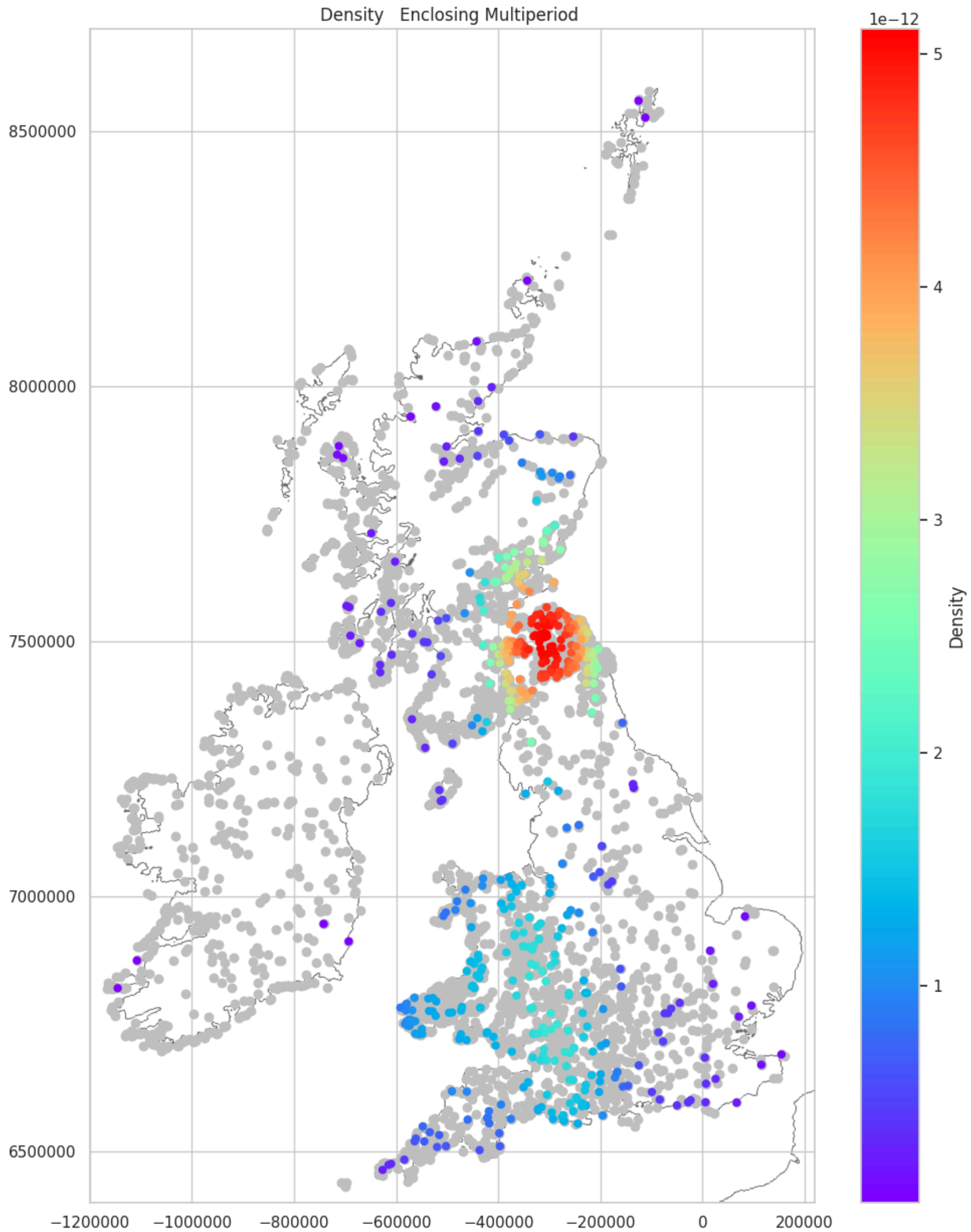



```
12.73%
```

Enclosing Multiperiod Density Mapped

Hillforts recorded as multiperiod cluster most intensely in the Northeast. There is a secondary cluster to the east of the Cambrian Mountains.

```
In [ ]: plot_density_over_grey(multiperiod_data, 'Enclosing_Multiperiod')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Circuit Mapped

There are 1891 (45.6%) of hillforts identified as having an Enclosing Circuit. It is assumed that Enclosing Circuit refers to hillforts having ramparts that form a completely enclosed ring.

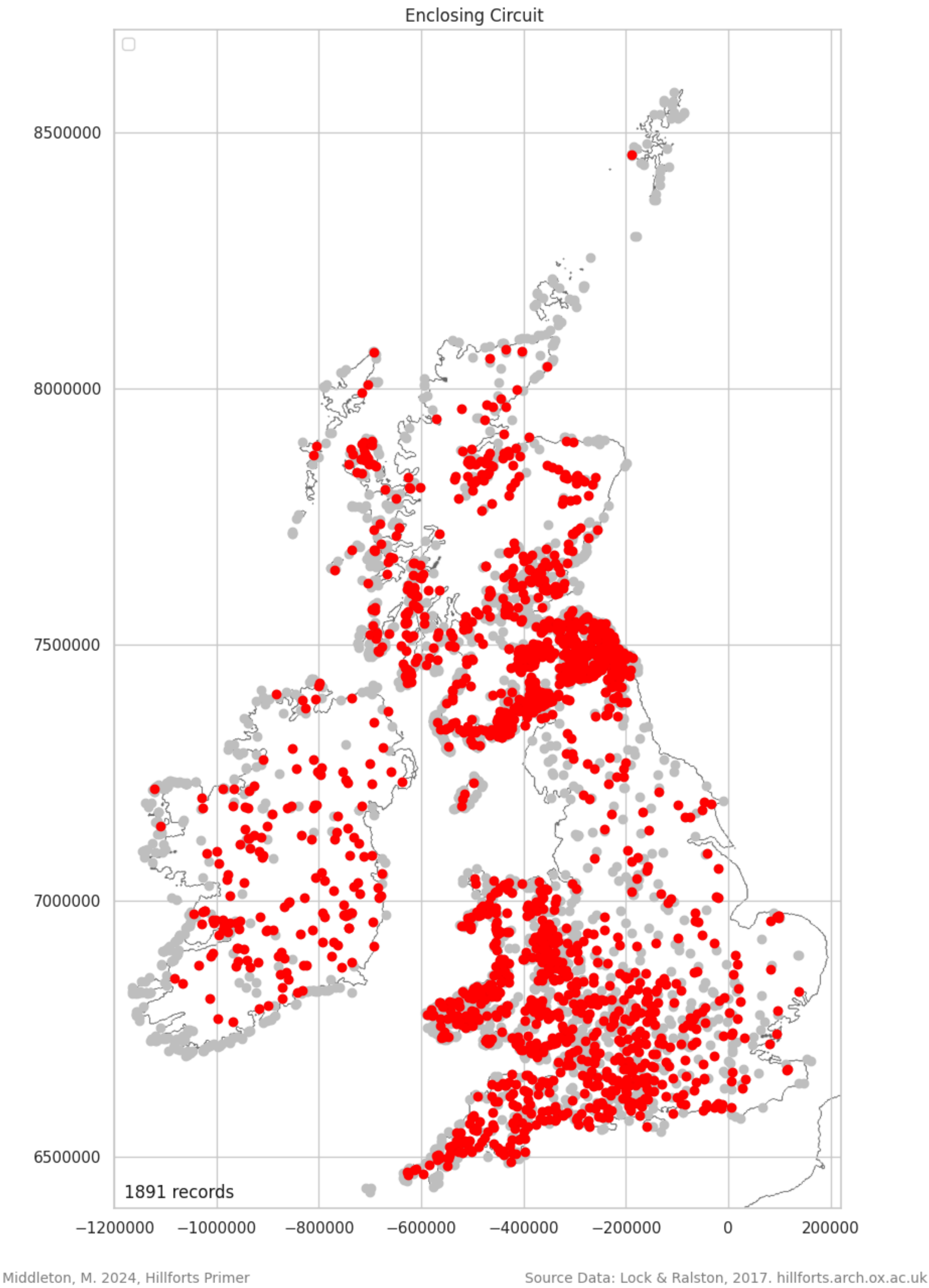
In [ ]:
```
circuit_counts = enclosing_encodeable_data['Enclosing_Circuit'].value_counts()
circuit_counts
```

Out[ ]:
```
No     2256
Yes    1891
Name: Enclosing_Circuit, dtype: int64
```

In [ ]:
```
print(f'{round(circuit_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
45.6%
```

In [ ]:
```
circuit_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Circuit', 'Yes', '')
```

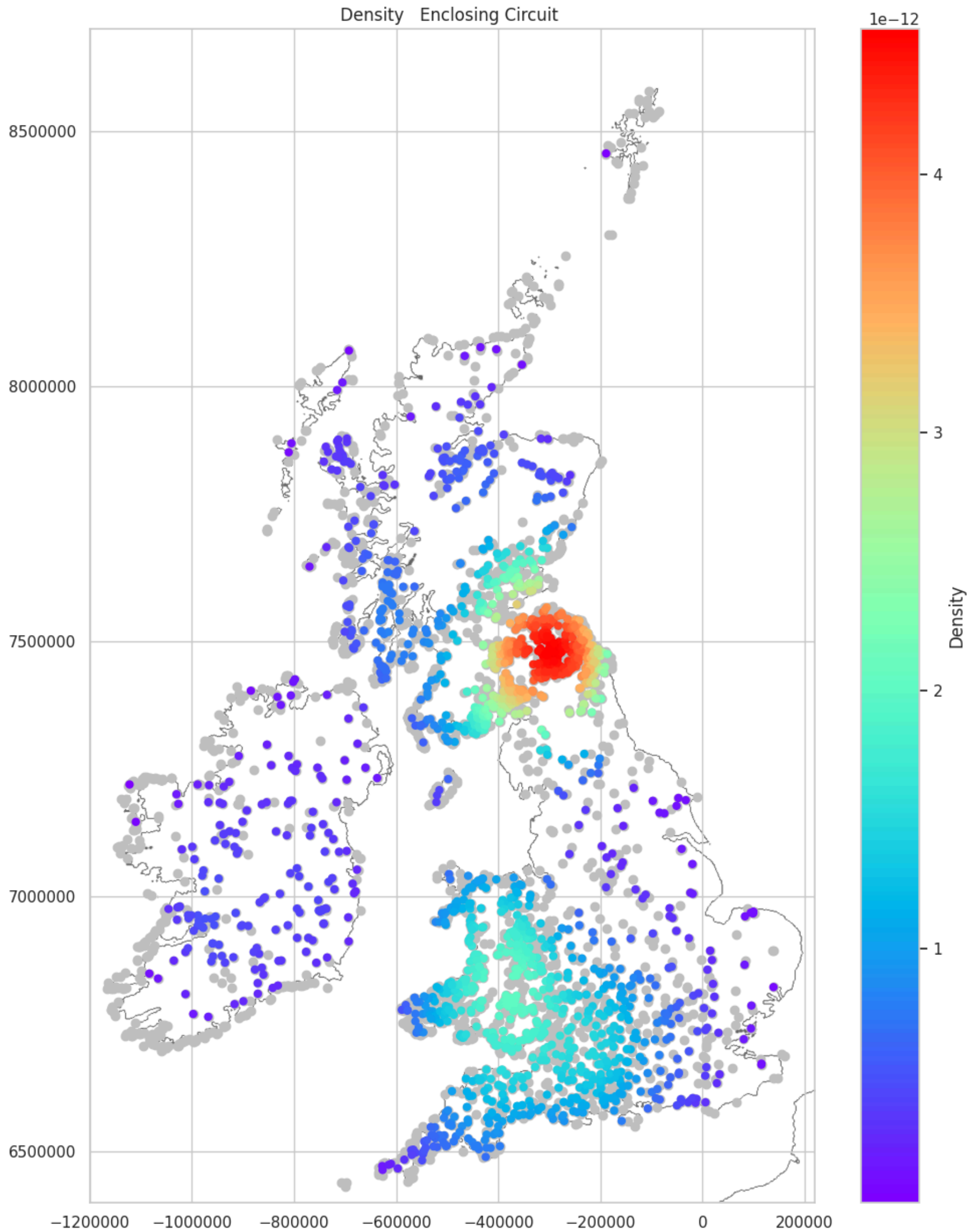



```
45.6%
```

Enclosing Circuit Density Mapped

The distribution is noticeably concentrated over the inland forts. There are two main concentrations, a strong cluster over the Northeast and a more subtle cluster to the east of the Cambrian Mountains. Unsurprisingly, coastal forts are less likely to have a fully enclosed rampart as they tend to incorporate naturally defensive features, such as cliffs, into their layout.

In [ ]:
```
plot_density_over_grey(circuit_data_yes, 'Enclosing_Circuit')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Current Part Univallate Mapped

1628 (39.26%) of hillforts are identified as Current Part Univallate. The distribution is relatively even across the atlas.

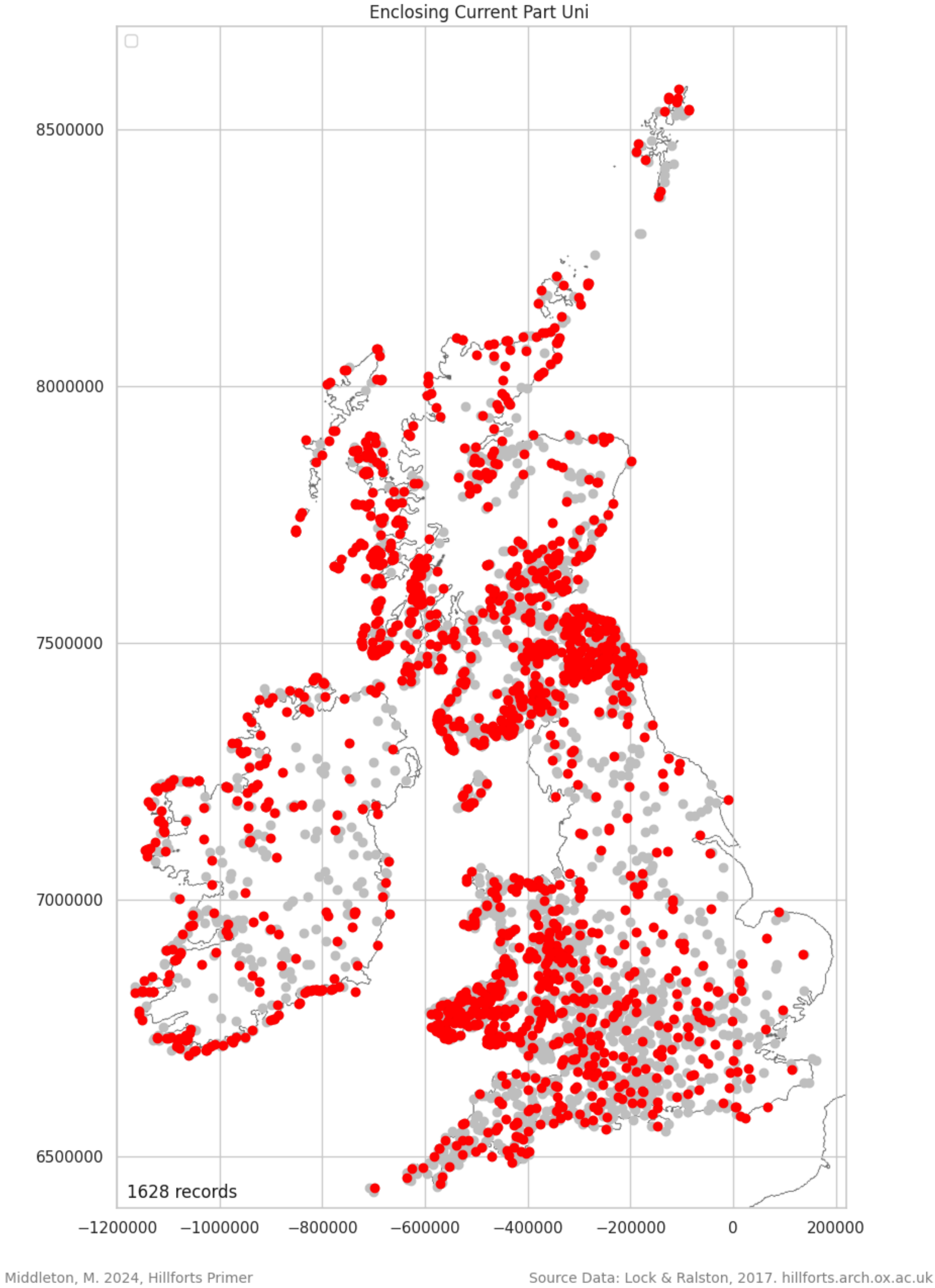
In [ ]:
```
current_part_uni_counts = \
enclosing_encodeable_data['Enclosing_Current_Part_Uni'].value_counts()
current_part_uni_counts
```

Out[ ]:
```
No      2519
Yes     1628
Name: Enclosing_Current_Part_Uni, dtype: int64
```

In [ ]:
```
print(f'{round(current_part_uni_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
39.26%
```

In [ ]:
```
current_part_uni_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Current_Part_Uni', \
               'Yes', '')
```

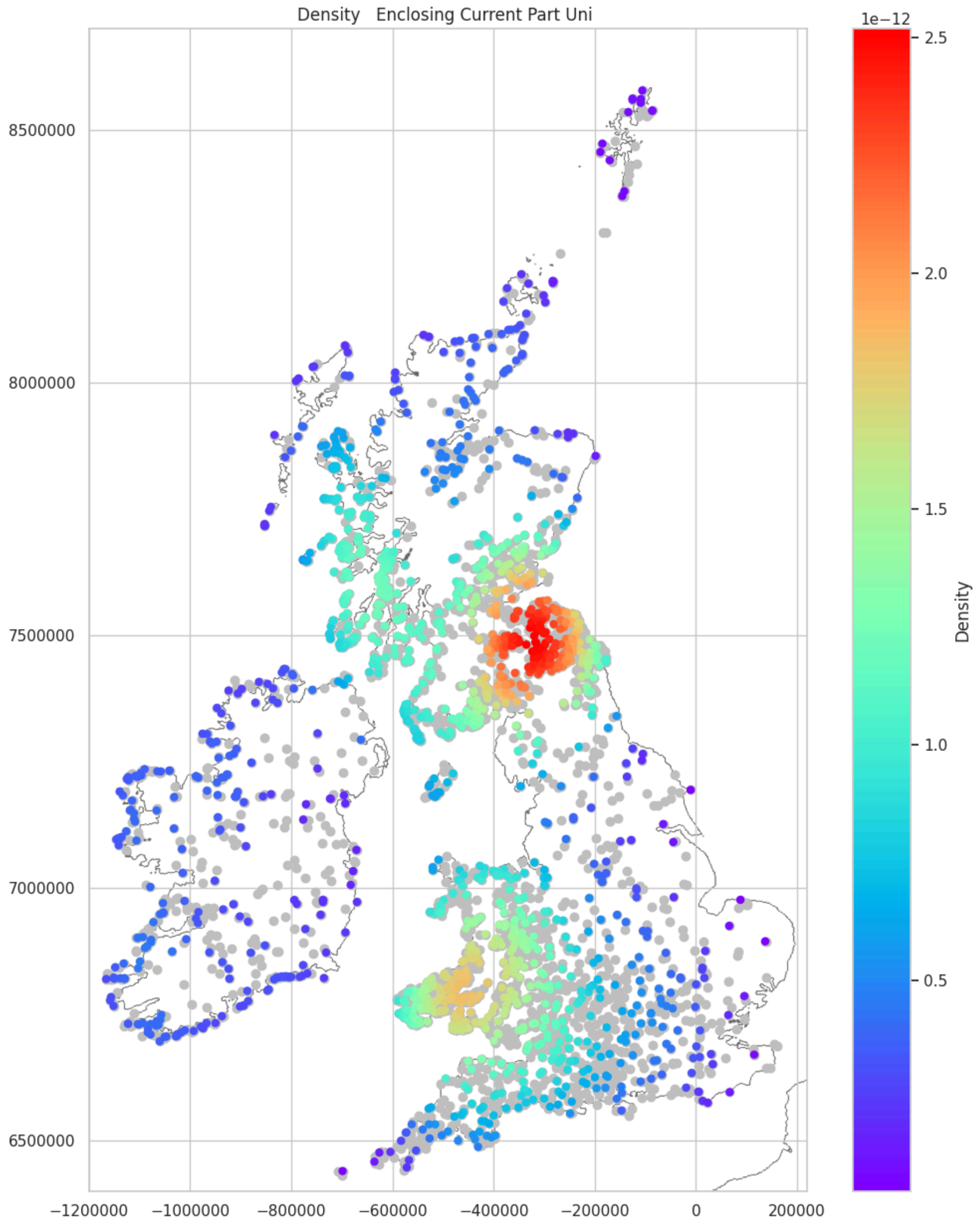



```
39.26%
```

Enclosing Current Part Univallate Density Mapped

The focus for this class is most intense over the Southern Uplands, southwest Wales and the Northwest.

In [ ]:
```
plot_density_over_grey(current_part_uni_data_yes, 'Enclosing_Current_Part_Uni')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Current Univallate Mapped

964 (23.25%) of hillforts are identified as Current Univallate. They are distributed right across the atlas.

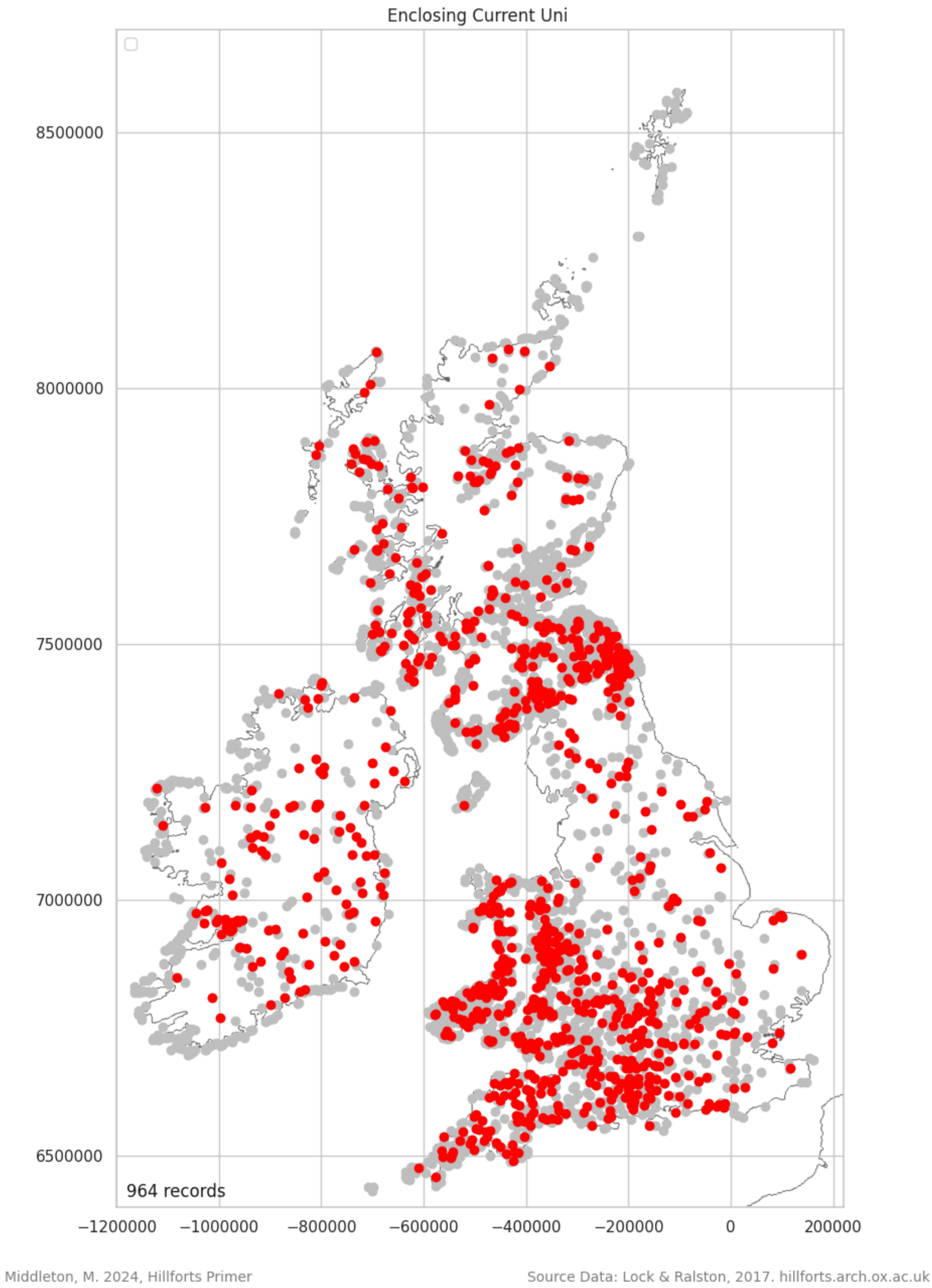
In [ ]:
```
current_uni_counts = \
enclosing_encodeable_data['Enclosing_Current_Uni'].value_counts()
current_uni_counts
```

Out[ ]:
```
No      3183
Yes      964
Name: Enclosing_Current_Uni, dtype: int64
```

In [ ]:
```
print(f'{round(current_uni_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
23.25%
```

In [ ]:
```
current_uni_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Current_Uni', 'Yes', '')
```

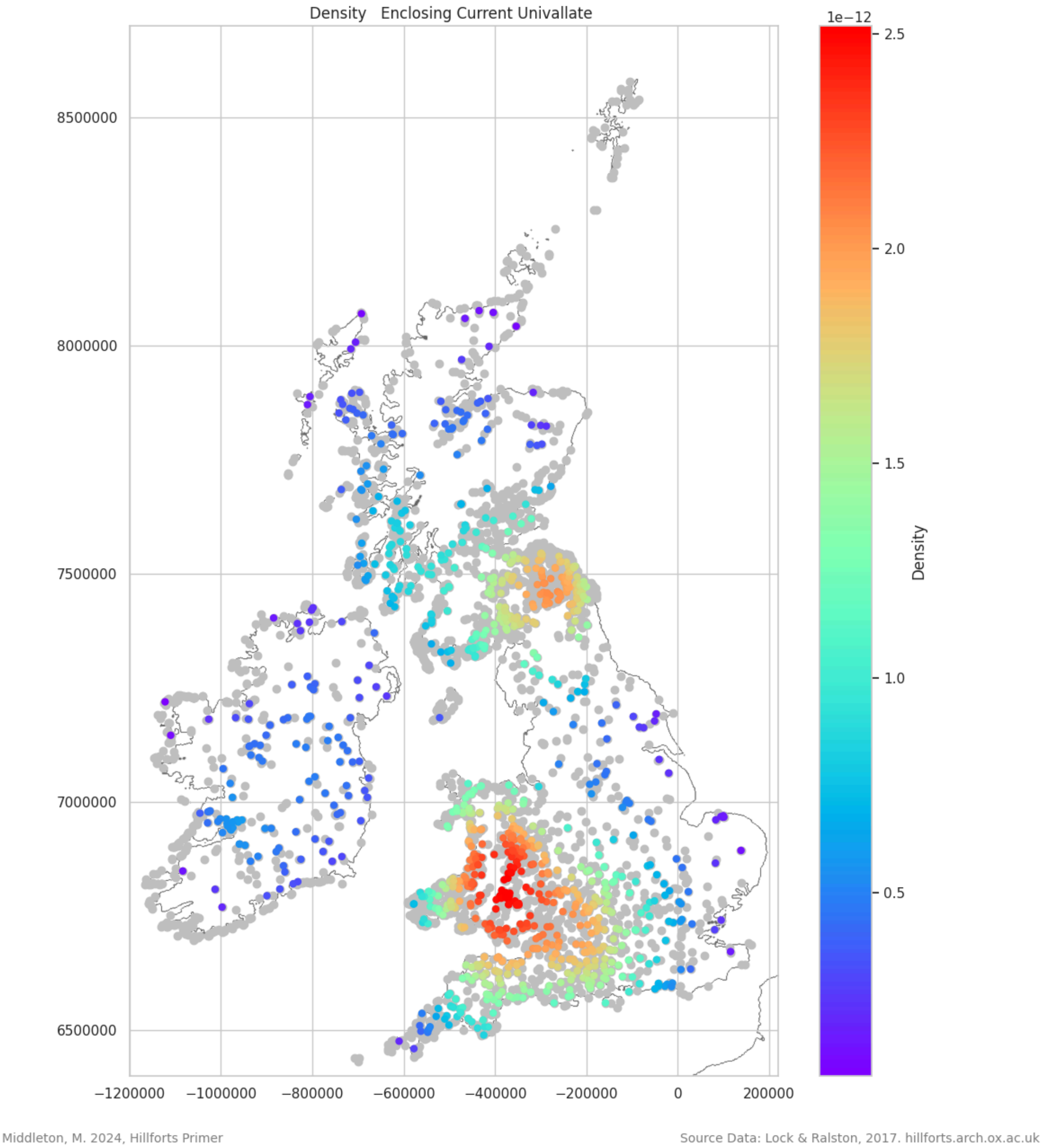



```
23.25%
```

Enclosing Current Univallate Density Mapped

Univallate hillforts cluster most in the South. The focus is noticeably east of the Cambrian Mountains and into South Central England. There is a secondary cluster in the Northeast and a third, much smaller cluster, in the Northwest.

In [ ]:
```
plot_density_over_grey(current_uni_data_yes, 'Enclosing_Current_Univallate')
```



Enclosing Current Part Bivallate Mapped

1058 (25.51%) of hillforts are identified as Current Part Bivallate. They are distributed right across the atlas. They are noticeably sparce across northeast Ireland.

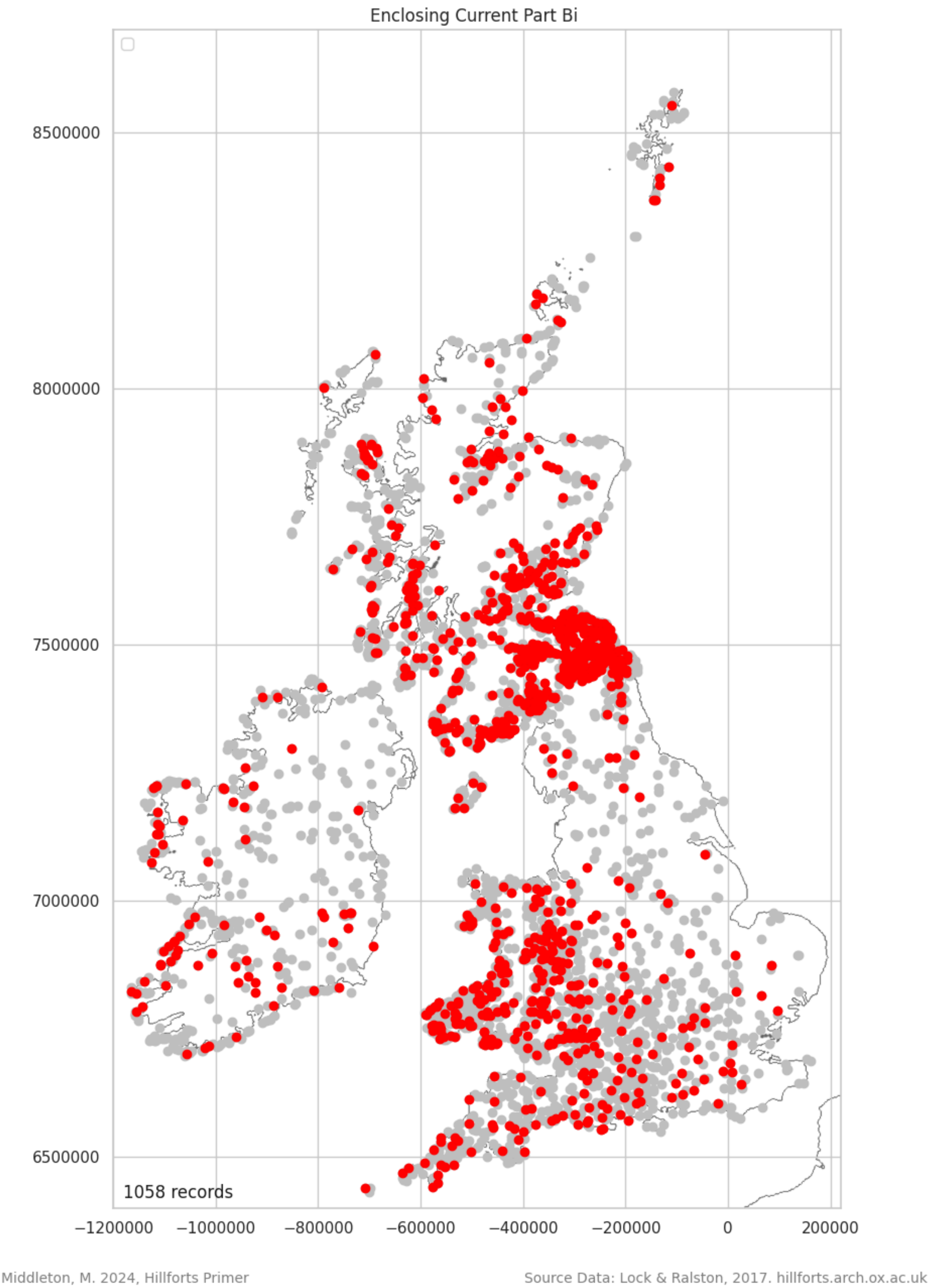
In [ ]:
```
current_part_bi_counts = \
enclosing_encodeable_data['Enclosing_Current_Part_Bi'].value_counts()
current_part_bi_counts
```

Out[ ]:
```
No      3089
Yes     1058
Name: Enclosing_Current_Part_Bi, dtype: int64
```

In [ ]:
```
print(f'{round(current_part_bi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
25.51%
```

In [ ]:
```
current_part_bi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Current_Part_Bi', \
               'Yes', '')
```

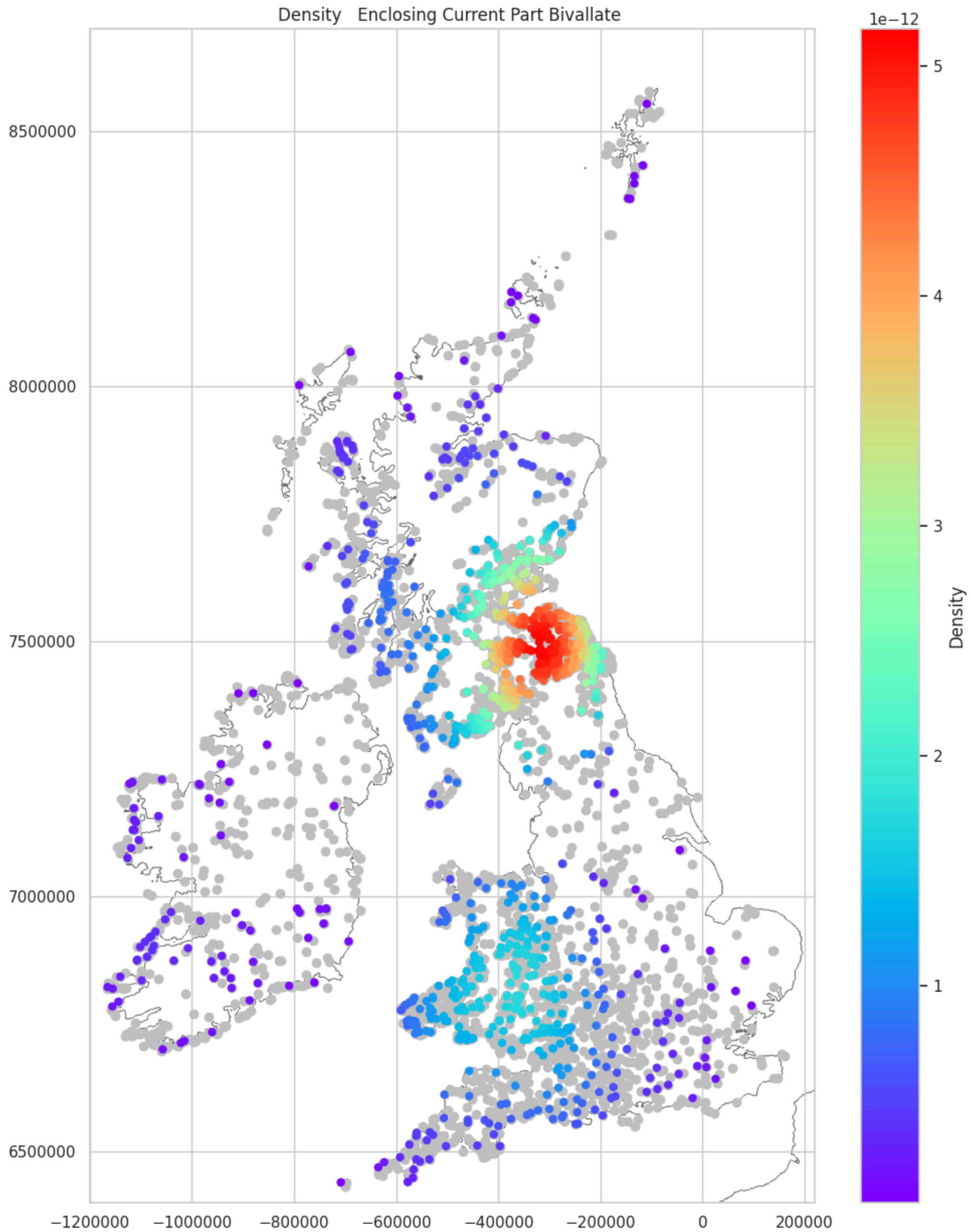


25.51%

Enclosing Current Part Bivallate Density Mapped

Current Part Bivallate forts cluster most intensively in the Northeast. There is a secondary, much more sparce cluster, over the Cambrian Mountains.

```
In [ ]: plot_density_over_grey(current_part_bi_data_yes, 'Enclosing_Current_Part_Bivallate')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Current Bivallate Mapped

395 (8.44%) of hillforts fall into the Current Bivallate class. They are noticeably more concentrated over the Northeast and away from the coasts.

In [ ]:

```
current_bi_counts = \
enclosing_encodeable_data['Enclosing_Current_Bi'].value_counts()
current_bi_counts
```

Out[ ]:

```
No     3797
Yes     350
Name: Enclosing_Current_Bi, dtype: int64
```

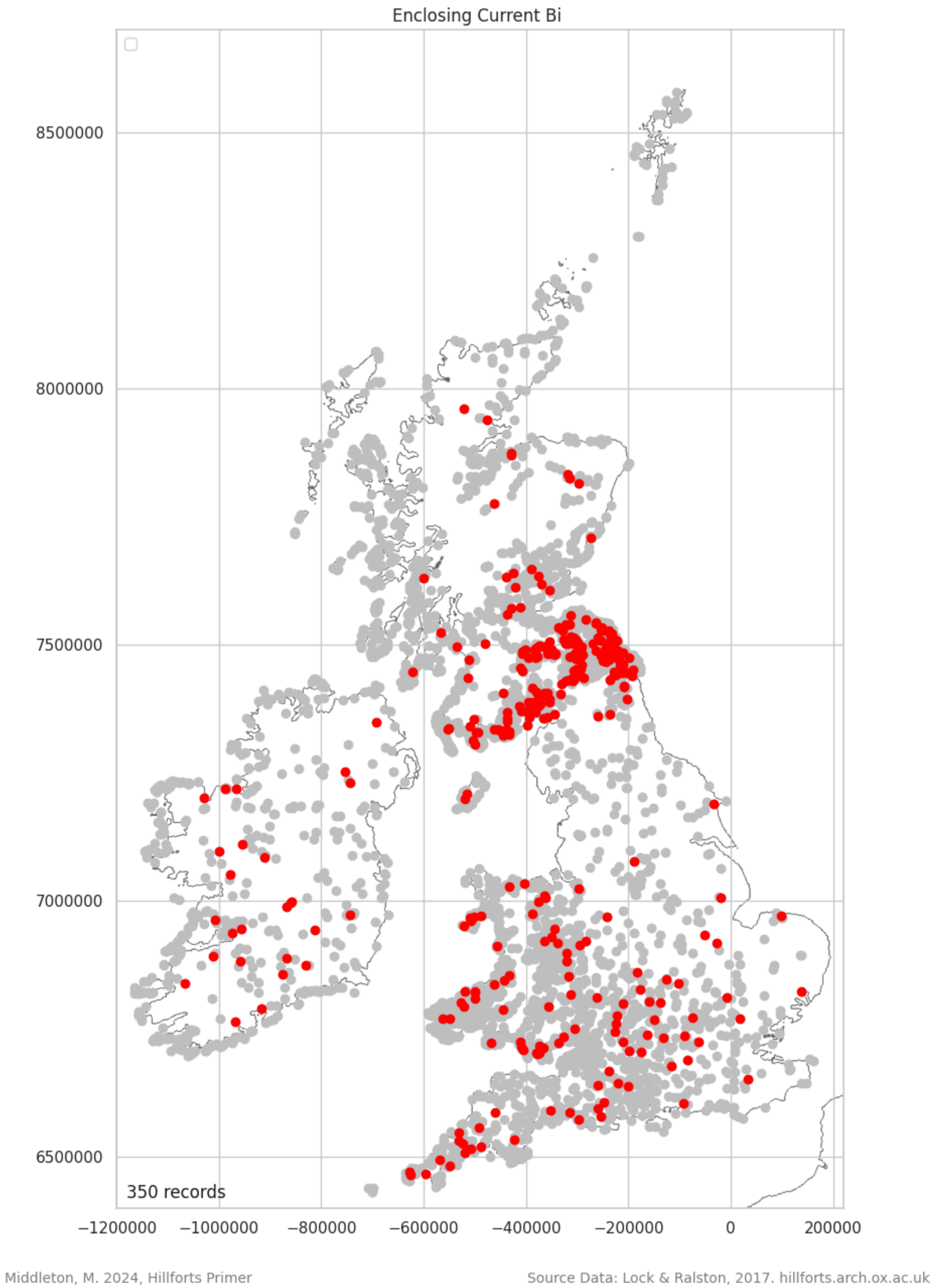
In [ ]:

```
print(f'{round(current_bi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
8.44%
```

In [ ]:

```
current_bi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Current_Bi', \
               'Yes', '')
```

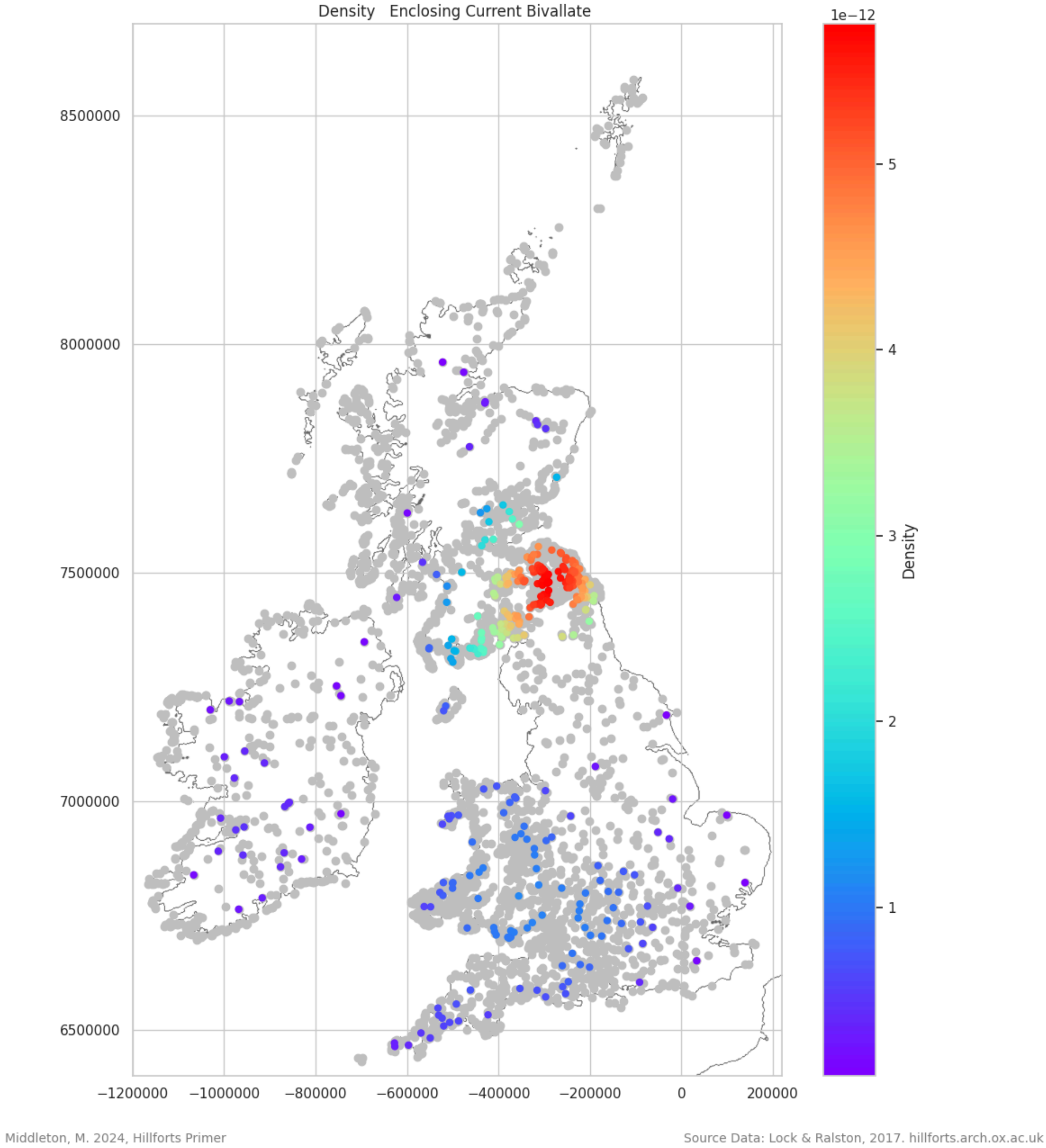



```
8.44%
```

Enclosing Current Bivallate Density Mapped

There is a single main cluster of Current Bivallate hillforts over the Northeast.

```
In [ ]: plot_density_over_grey(current_bi_data_yes, 'Enclosing_Current_Bivallate')
```



Enclosing Current Part Multivallate Mapped

596 (14.37%) of hillforts are classified as Current Part Multivallate.

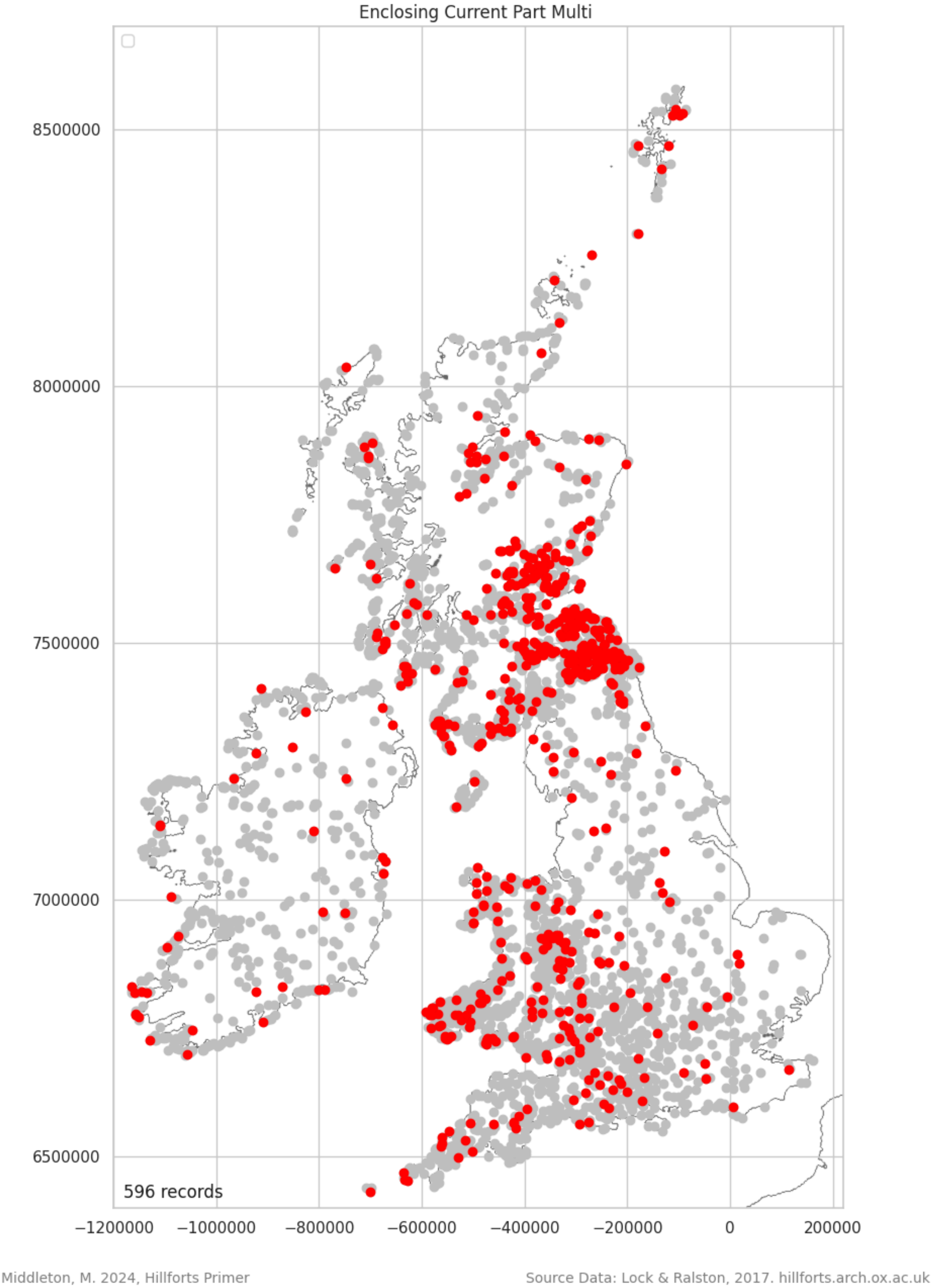
In [ ]:
```
current_part_multi_counts = \
enclosing_encodeable_data['Enclosing_Current_Part_Multi'].value_counts()
current_part_multi_counts
```

Out[ ]:
```
No      3551
Yes      596
Name: Enclosing_Current_Part_Multi, dtype: int64
```

In [ ]:
```
print(f'{round(current_part_multi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
14.37%
```

In [ ]:
```
current_part_multi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Current_Part_Multi', 'Yes', '')
```

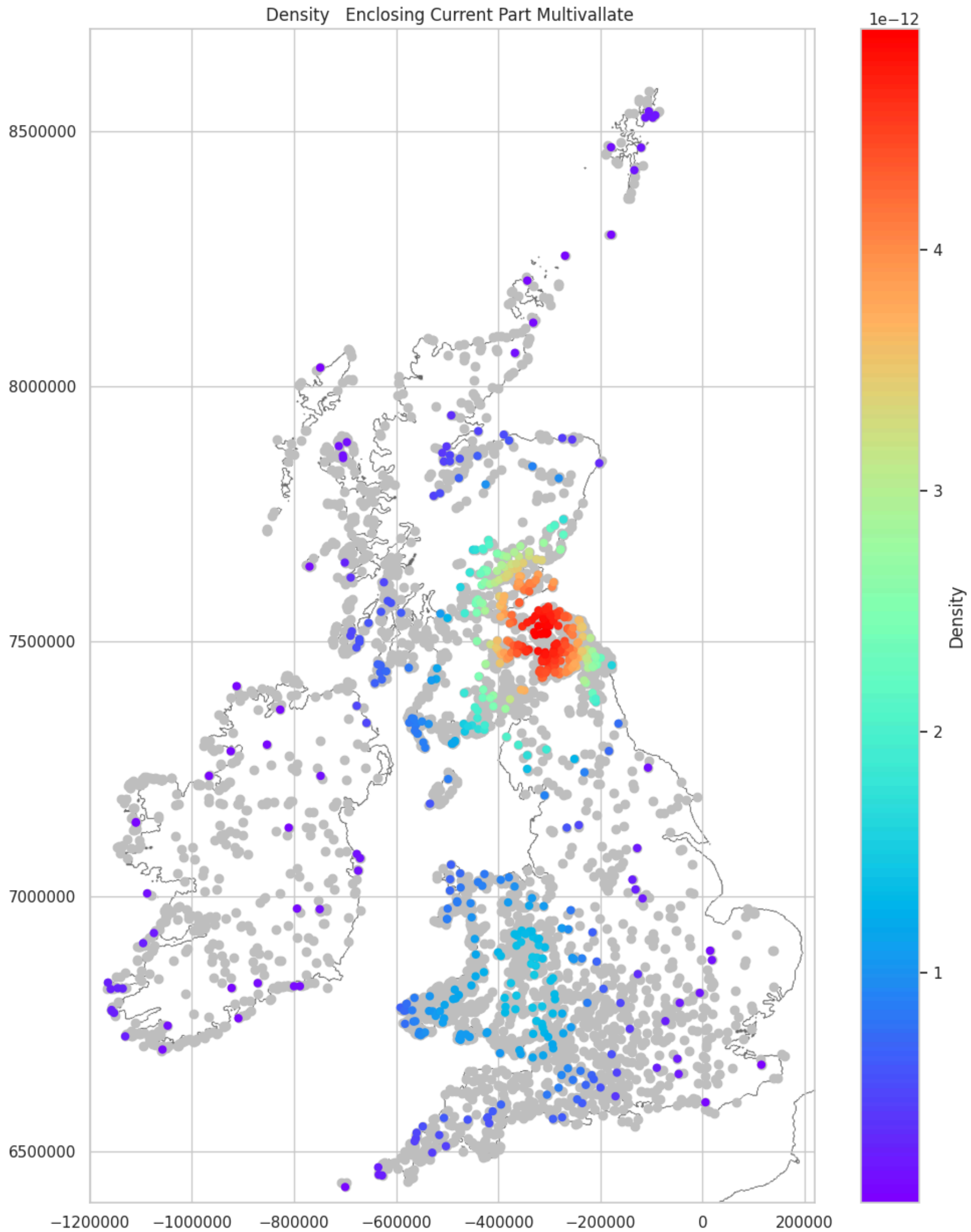



```
14.37%
```

Enclosing Current Part Multivallate Density Mapped

The main cluster of Current Part Multivallate hillforts is in the Northeast. There is a very sparce cluster in the South, east of the Cambrian Mountains.

In [ ]:
```
plot_density_over_grey(current_part_multi_data_yes, \
                       'Enclosing_Current_Part_Multivallate')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Current Multivallate Mapped

Just 149 (3.59%) of hillforts are identified as being current Multivallate.

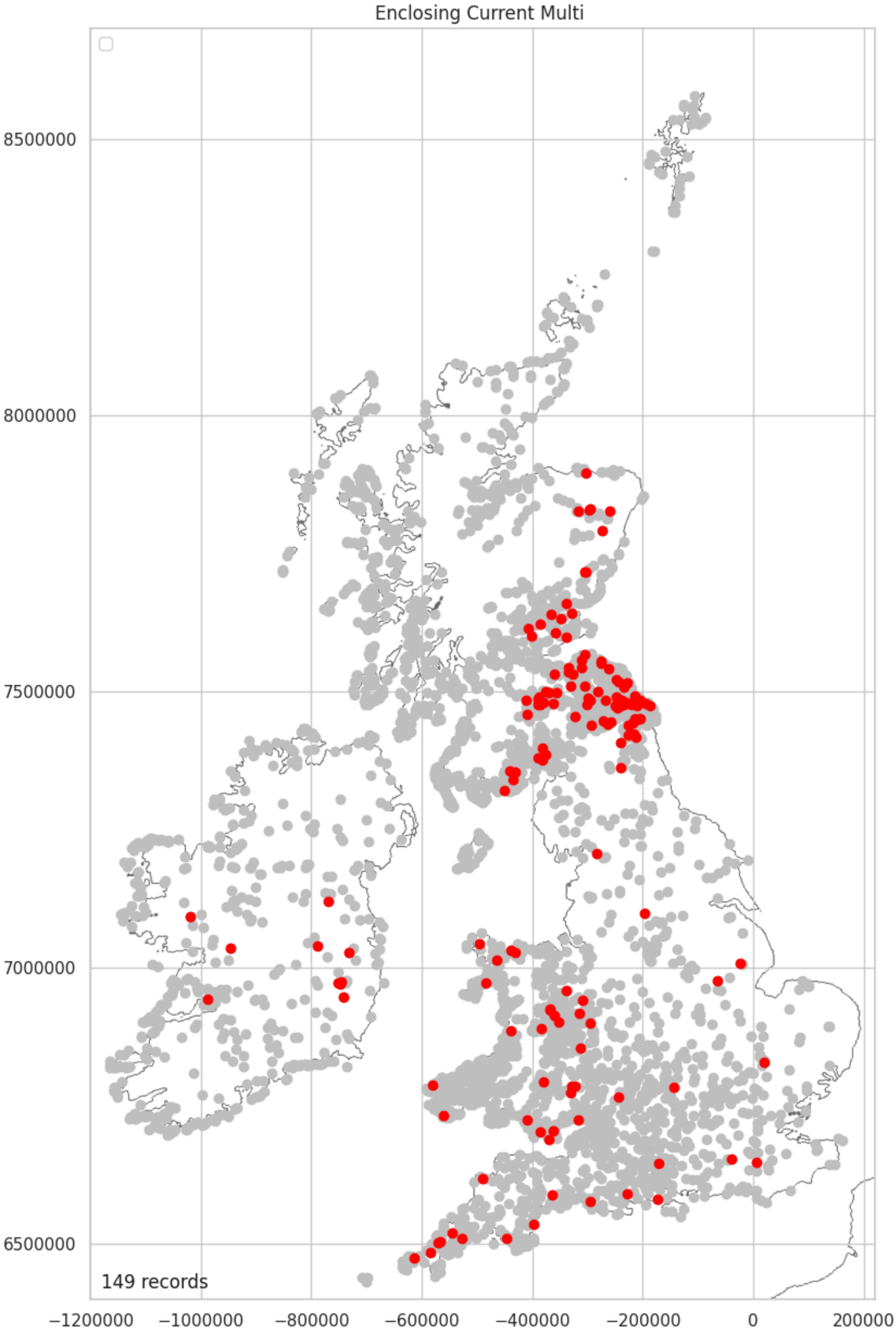
In [ ]:
```
current_multi_counts = \
enclosing_encodeable_data['Enclosing_Current_Multi'].value_counts()
current_multi_counts
```

Out[ ]:
```
No      3998
Yes      149
Name: Enclosing_Current_Multi, dtype: int64
```

In [ ]:
```
print(f'{round(current_multi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
3.59%
```

In [ ]:
```
current_multi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Current_Multi', \
               'Yes', '')
```



Middleton, M. 2024, Hillforts Primer

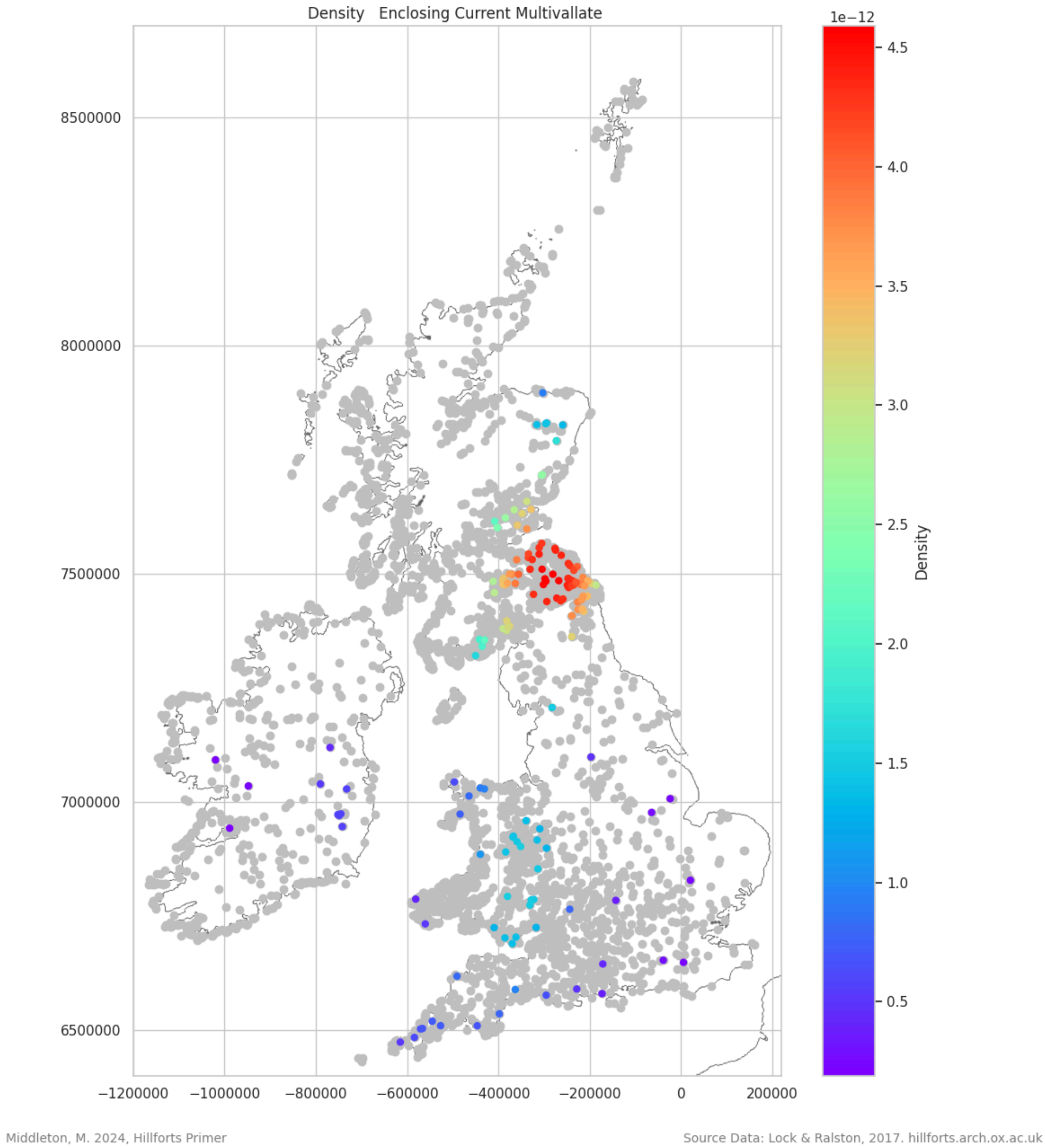
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
3.59%
```

Enclosing Current Multivallate Density Mapped

There is a main cluster of Current Multivallate hillforts in the Northeast and a very sparce second cluster along the eastern fringe of the Cambrian Mountains.

In [ ]:
```
plot_density_over_grey(current_multi_data_yes, 'Enclosing Current Multivallate')
```



Enclosing Current Unknown Mapped

256 (6.34%) of hillforts are identified as having an unknown current enclosing circuit.

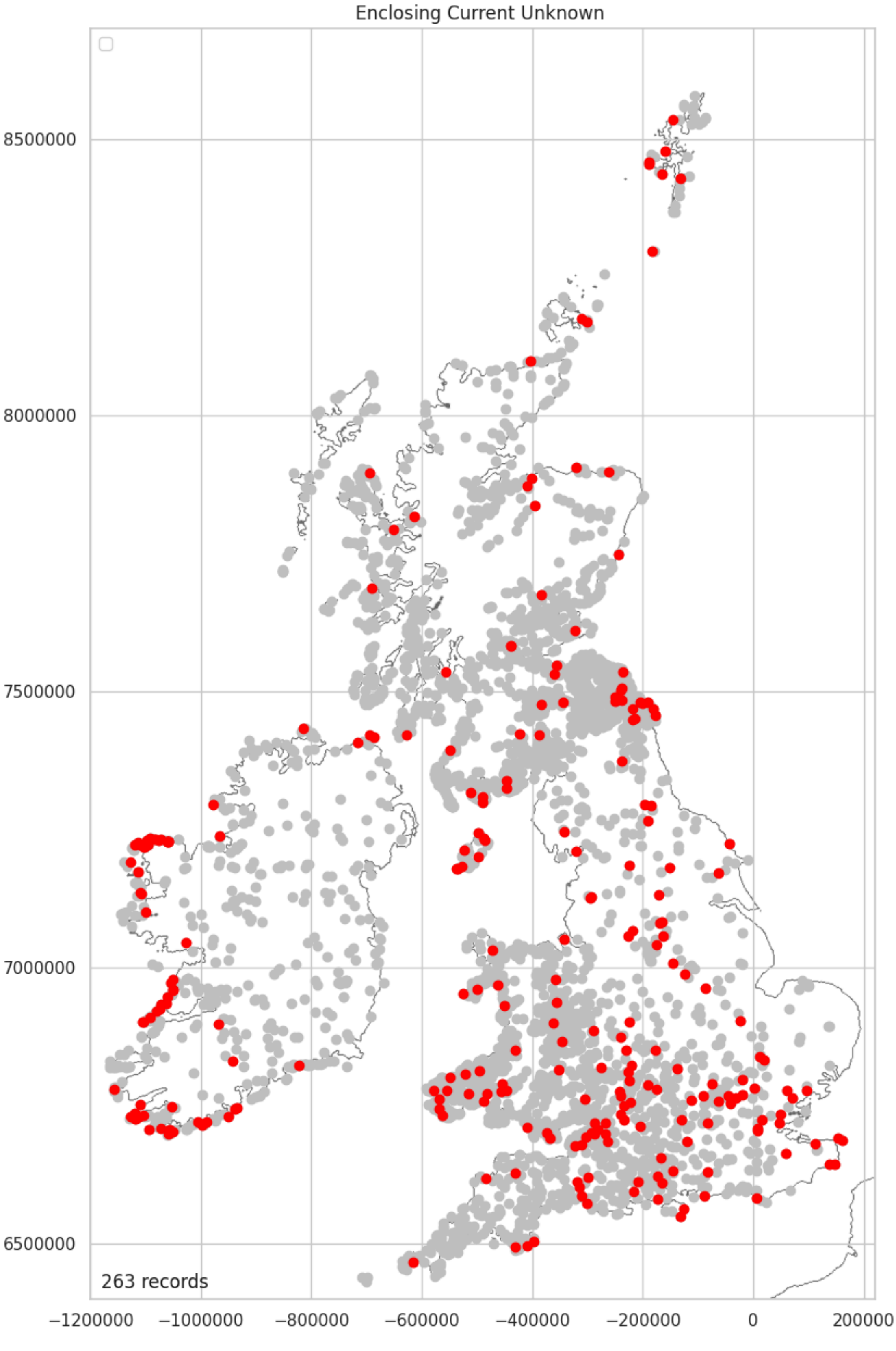
In [ ]:
```
current_uk_counts = \
enclosing_encodeable_data['Enclosing_Current_Unknown'].value_counts()
current_uk_counts
```

Out[ ]:
```
No     3884
Yes     263
Name: Enclosing_Current_Unknown, dtype: int64
```

In [ ]:
```
print(f'{round(current_uk_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
6.34%
```

In [ ]:
```
current_uk_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Current_Unknown', \
               'Yes', '')
```



Middleton, M. 2024, Hillforts Primer

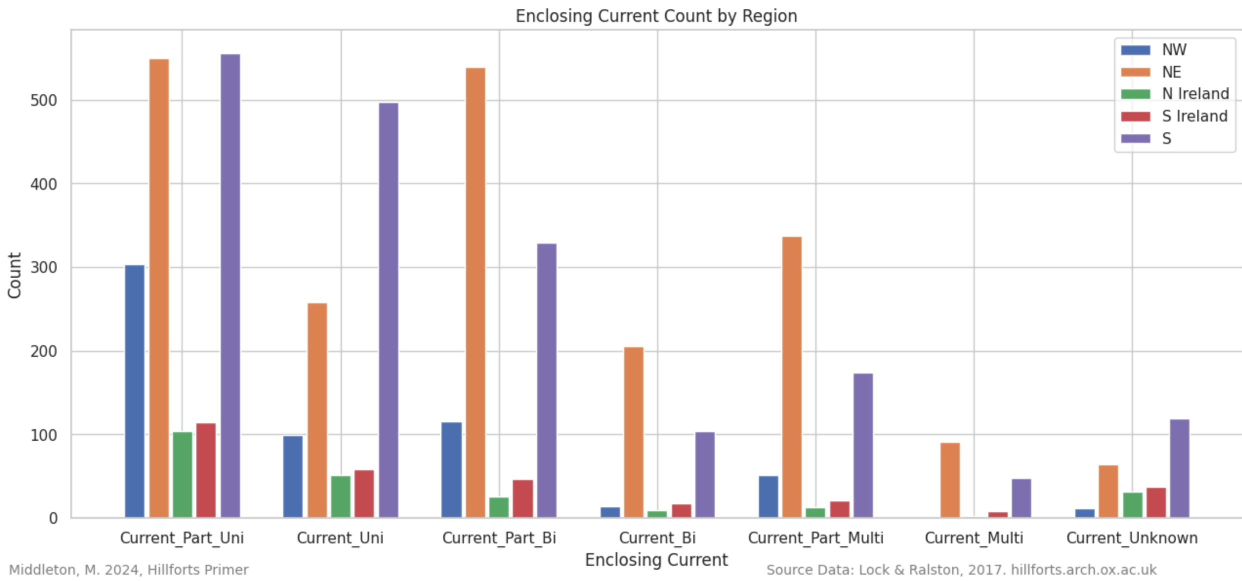
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
6.34%
```

Enclosing Current Plotted by Region (Count)

It is difficult to read the plot showing the count by current enclosing class as there are so many hillforts in the Northeast and South. It is simpler to look at all the proportions for an area and then compare these to the proportions across other areas. For instance, the South show strong returns at the bottom end of the range with high counts in Part Univallate, Univallate and Part Bivallate. In comparison, the Northeast has its three highest counts in Part Univallate, Part Bivallate and Part Multivallate.

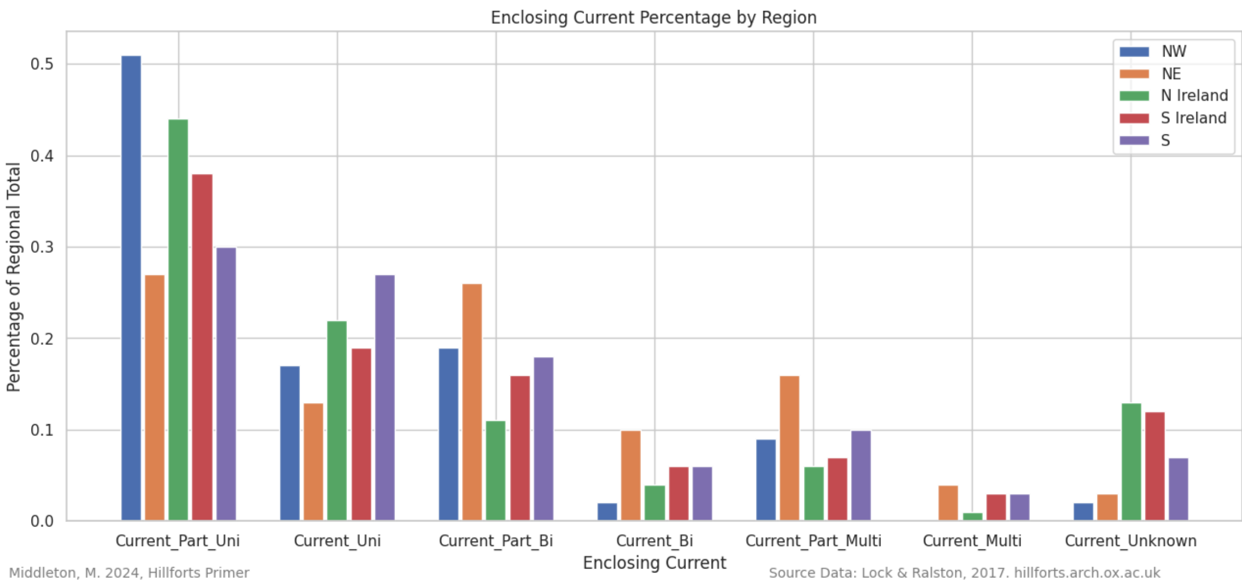
In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Current_Part_Uni','Enclosing_Current_Uni',\
                 'Enclosing_Current_Part_Bi',\
                  'Enclosing_Current_Bi','Enclosing_Current_Part_Multi',\
                  'Enclosing_Current_Multi',\
                  'Enclosing_Current_Unknown'],\
                 'Enclosing Current',\
                 'Enclosing_Current Count by Region', 1, 'Yes')
```



Enclosing Current Plotted by Region (Percentage)

It is revealing to look at this data proportionally. Looking at the data in this way, all the regions are relatively similar. All have a predominance of Part Univallate hillforts and secondary and tertiary clusters of Univallate and Part Bivallate. The Northeast is the outlier in that it is more likely to have Bivallate and Part Multivallate hillforts. The unknown are dominated by hillforts in Ireland.

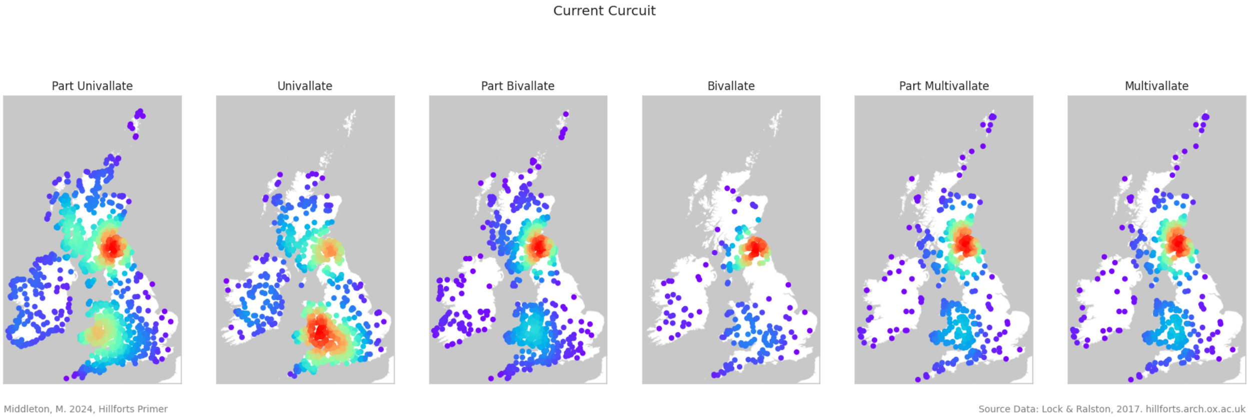
In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Current_Part_Uni','Enclosing_Current_Uni',\
                 'Enclosing_Current_Part_Bi',\
                  'Enclosing_Current_Bi','Enclosing_Current_Part_Multi',\
                  'Enclosing_Current_Multi',\
                  'Enclosing_Current_Unknown'],\
                 'Enclosing Current',\
                 'Enclosing_Current Percentage by Region', 1, 'Yes', True)
```



Current Curcuit Summary

All areas have a high proportion of Part Univallate hillforts. The main clusters are in the Northeast, south Wales and the Northwest. As a proportion by region, they are most common in the Northwest. Univallate forts cluster most densely in the South with smaller clusters in the Northeast and Northwest. Although the southern cluster is the most intense, within their own region, Univallate hillforts are almost half as common as Part Univallate forts. In all the remaining classes the Northeast has the most intense cluster with the South showing secondary, much less intense clusters.

In [ ]:
```
plot_density_over_grey_six(current_part_uni_data_yes, current_uni_data_yes, \
                           current_part_bi_data_yes, current_bi_data_yes, \
                           current_part_multi_data_yes, current_multi_data_yes, \
                           'Current Curcuit')
```



Enclosing Period

It is assumed that Enclosing Period refers to the morphology of the enclosing works at the time of construction. Very few hillforts have a Period Enclosing recorded. There is insufficient data to show any meaningful distributions. The majority of records are in the Northeast and this may indicate there is a recording bias toward area.

Enclosing Period Part Univallate Mapped

There are just 36 hillforts where Period Part Univallate has been recorded. The majority are in the Northeast.

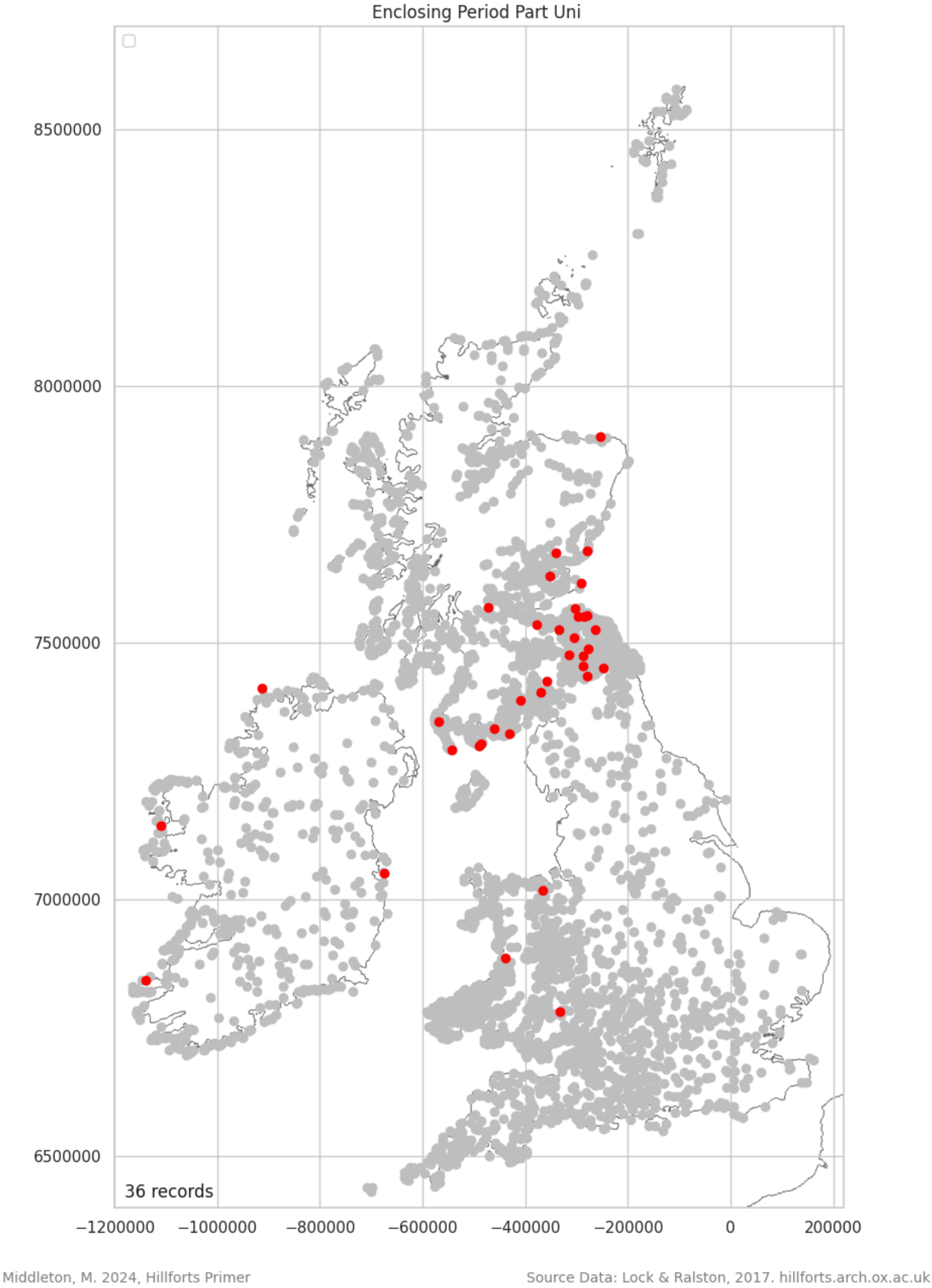
In [ ]:
```
period_part_uni_counts = \
enclosing_encodeable_data\
 ['Enclosing_Period_Part_Uni'].value_counts()
period_part_uni_counts
```

Out[ ]:
```
No     4111
Yes      36
Name: Enclosing_Period_Part_Uni, dtype: int64
```

In [ ]:
```
print(f'{round(period_part_uni_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.87%
```

In [ ]:
```
period_part_uni_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Period_Part_Uni', \
               'Yes', '')
```




```
0.87%
```

Enclosing Period Univallate Mapped

249 (6%) of hillforts have a Period Univallate classification.

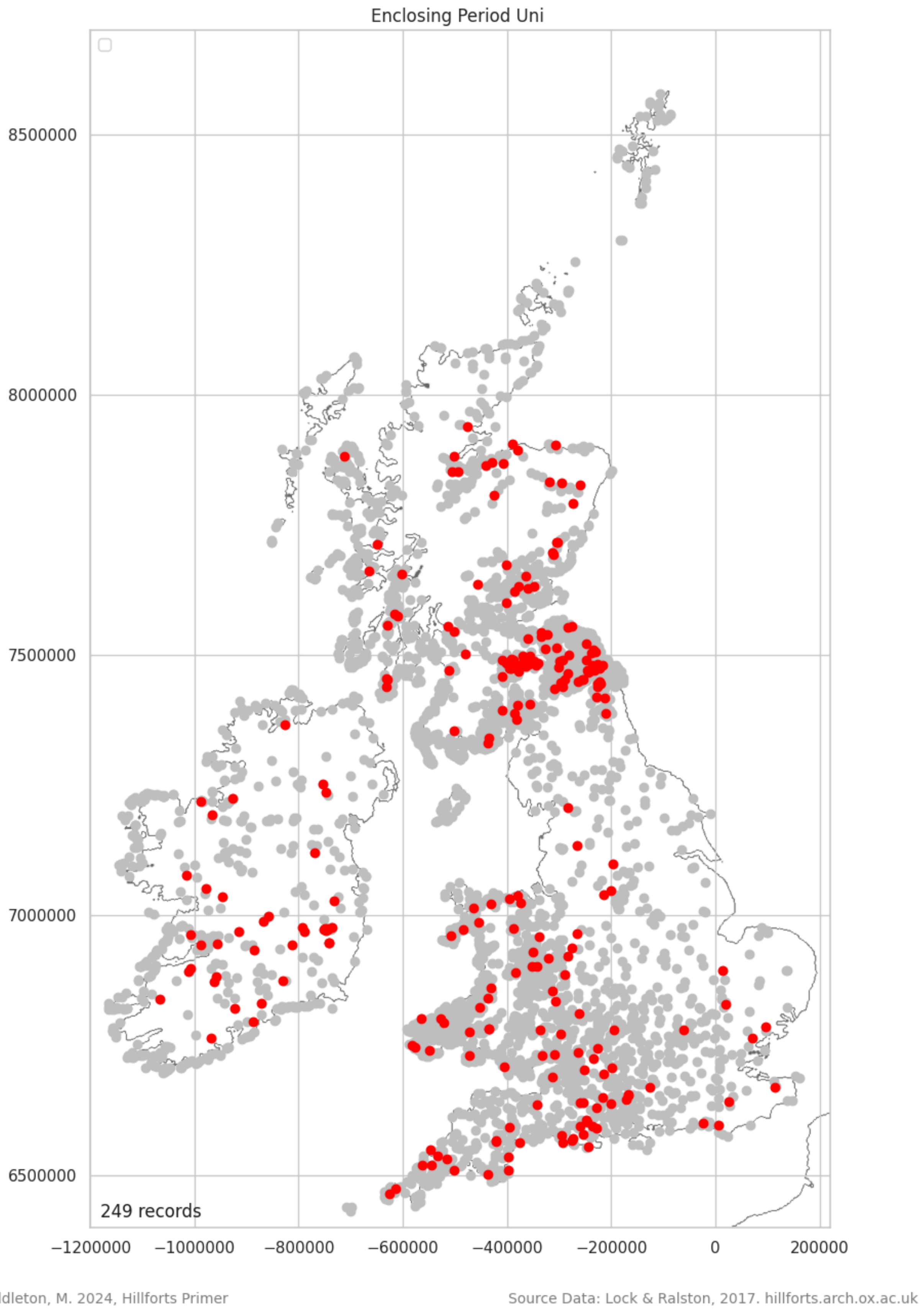
```
In [ ]: period_uni_counts = \
        enclosing_encodeable_data['Enclosing_Period_Uni'].value_counts()
        period_uni_counts
```

```
Out[ ]: No     3898
        Yes     249
        Name: Enclosing_Period_Uni, dtype: int64
```

```
In [ ]: print(f'{round(period_uni_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
6.0%
```

```
In [ ]: period_uni_data_yes = plot_over_grey(location_enclosing_encodeable_data, \
                                             'Enclosing_Period_Uni', 'Yes', '')
```

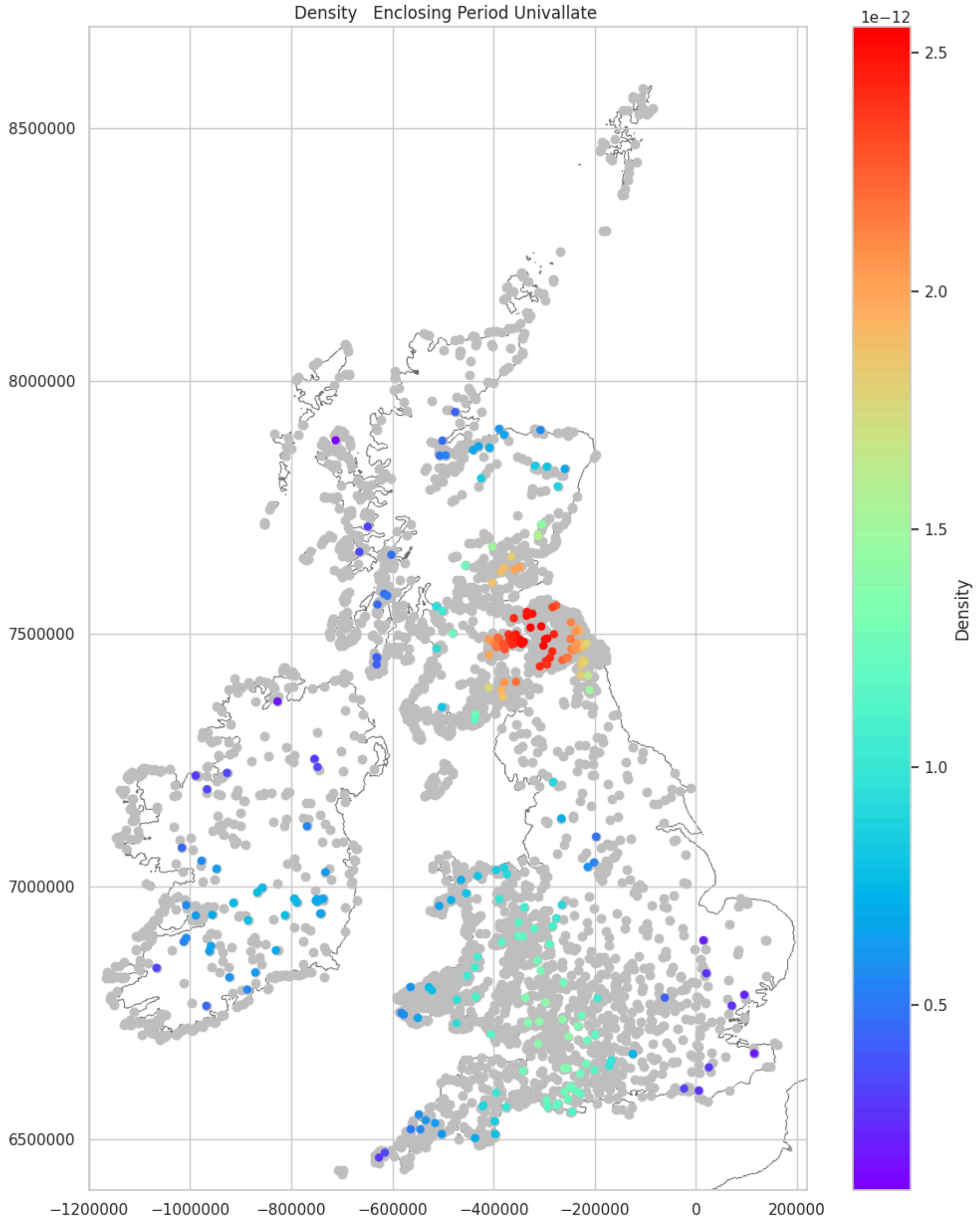



```
6.0%
```

Enclosing Period Univallate Density Mapped

The main cluster of Period Univallate forts is in the Northeast. There is a second, more diffuse cluster, over south-central England.

In [ ]:
```
plot_density_over_grey(period_uni_data_yes, 'Enclosing Period Univallate')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Period Part Bivallate Mapped

There are 35 (0.84%) Period Part Bivallate forts. Again, these are mostly in the Northeast.

In [ ]:
```
period_part_bi_counts = \
enclosing_encodeable_data['Enclosing_Period_Part_Bi'].value_counts()
period_part_bi_counts
```

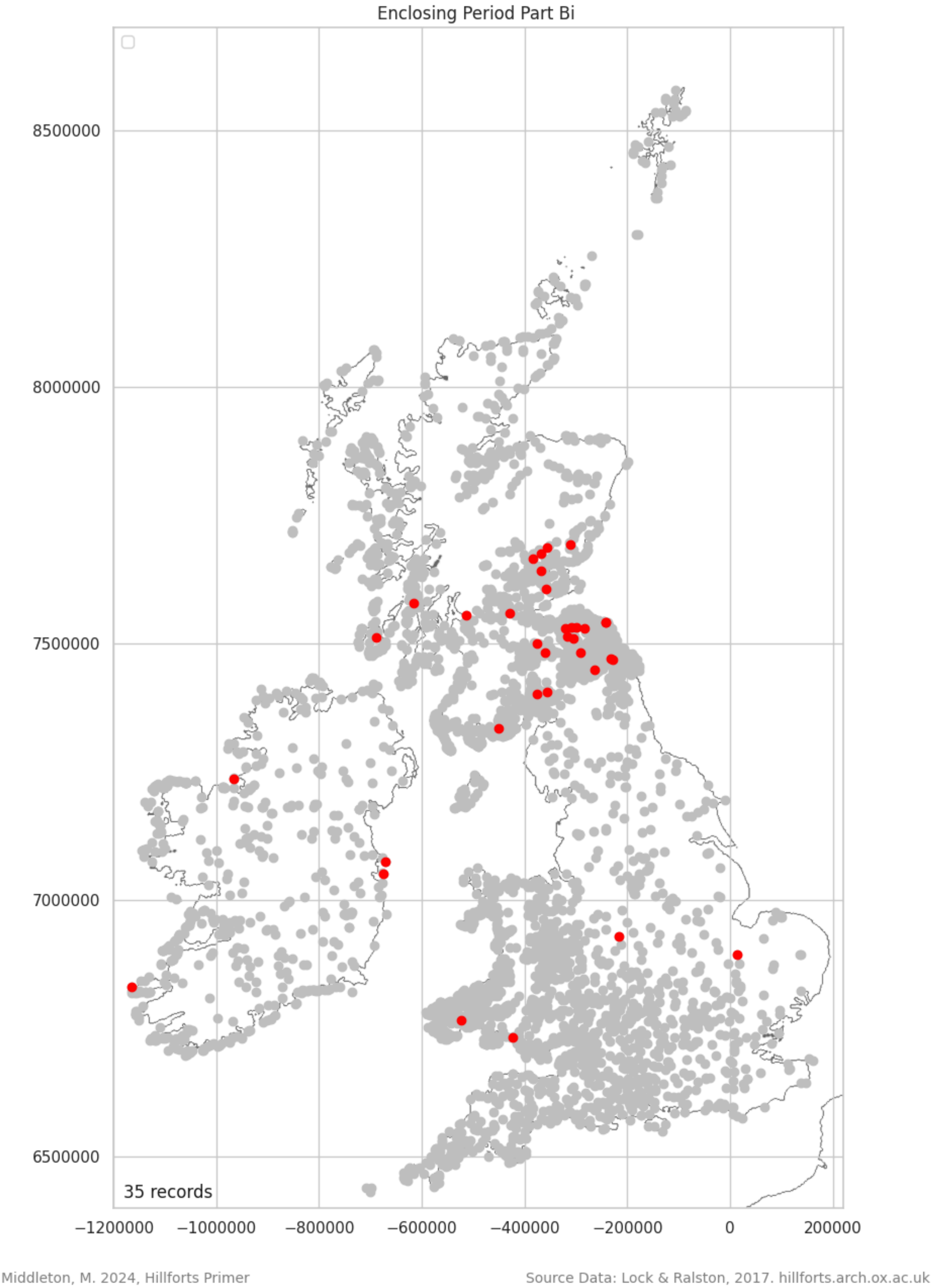
Out[ ]:
```
No     4112
Yes      35
Name: Enclosing_Period_Part_Bi, dtype: int64
```

In [ ]:
```
print(f'{round(period_part_bi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.84%
```

In [ ]:
```
period_part_bi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Period_Part_Bi', \
```

```
'Yes', '')
```




```
0.84%
```

Enclosing Period Bivallate Mapped

There are 55 (1.33%) Period Bivallate forts, also in the Northeast.

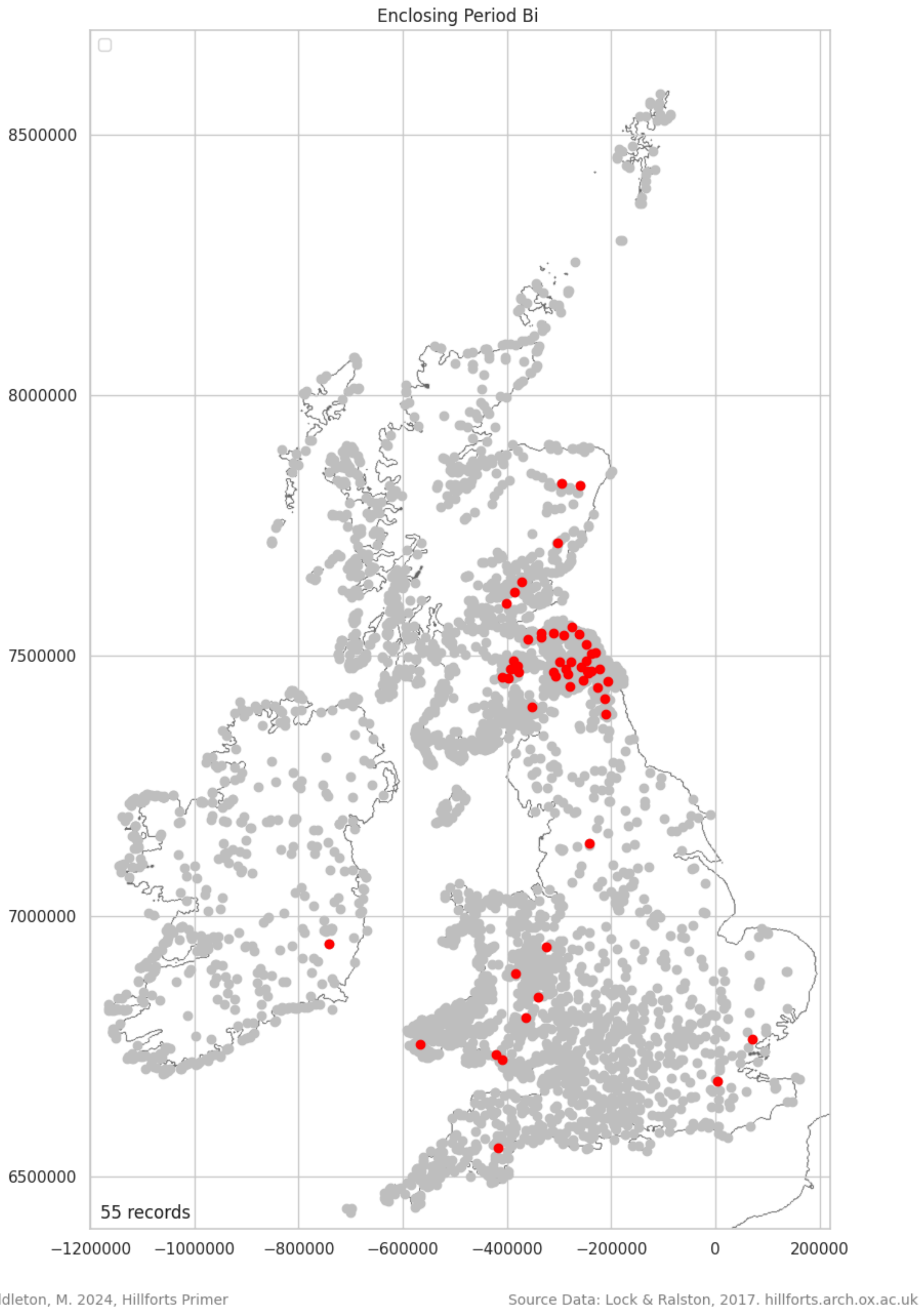
In [ ]:
```
period_bi_counts = \
enclosing_encodeable_data['Enclosing_Period_Bi'].value_counts()
period_bi_counts
```

```
Out[ ]: No      4092
        Yes       55
        Name: Enclosing_Period_Bi, dtype: int64
```

```
In [ ]: print(f'{round(period_bi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.33%
```

```
In [ ]: period_bi_data_yes = \
        plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Period_Bi', \
                       'Yes', '')
```




```
1.33%
```

Enclosing Period Part Multivallate Mapped

There are six (0.14%) Period Part Multivallate forts.

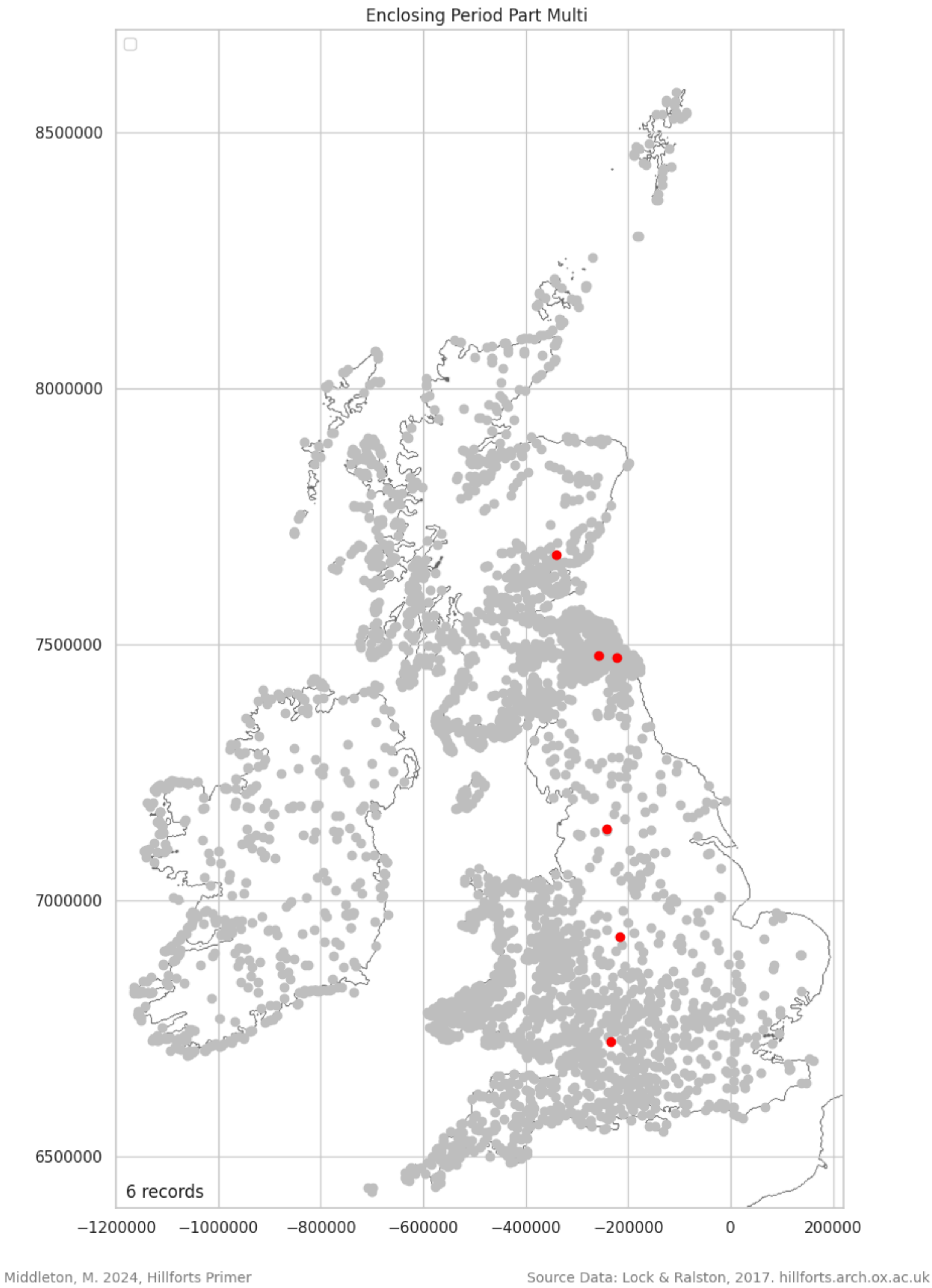
In [ ]:
```
period_part_multi_counts = \
enclosing_encodeable_data['Enclosing_Period_Part_Multi'].value_counts()
period_part_multi_counts
```

Out[ ]:
```
No     4141
Yes       6
Name: Enclosing_Period_Part_Multi, dtype: int64
```

In [ ]:
```
print(f'{round(period_part_multi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.14%
```

In [ ]:
```
period_part_multi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Period_Part_Multi', 'Yes', '')
```



0.14%

Enclosing Period Multivallate Mapped

There are 12 (0.29%) Period Multivallate forts.

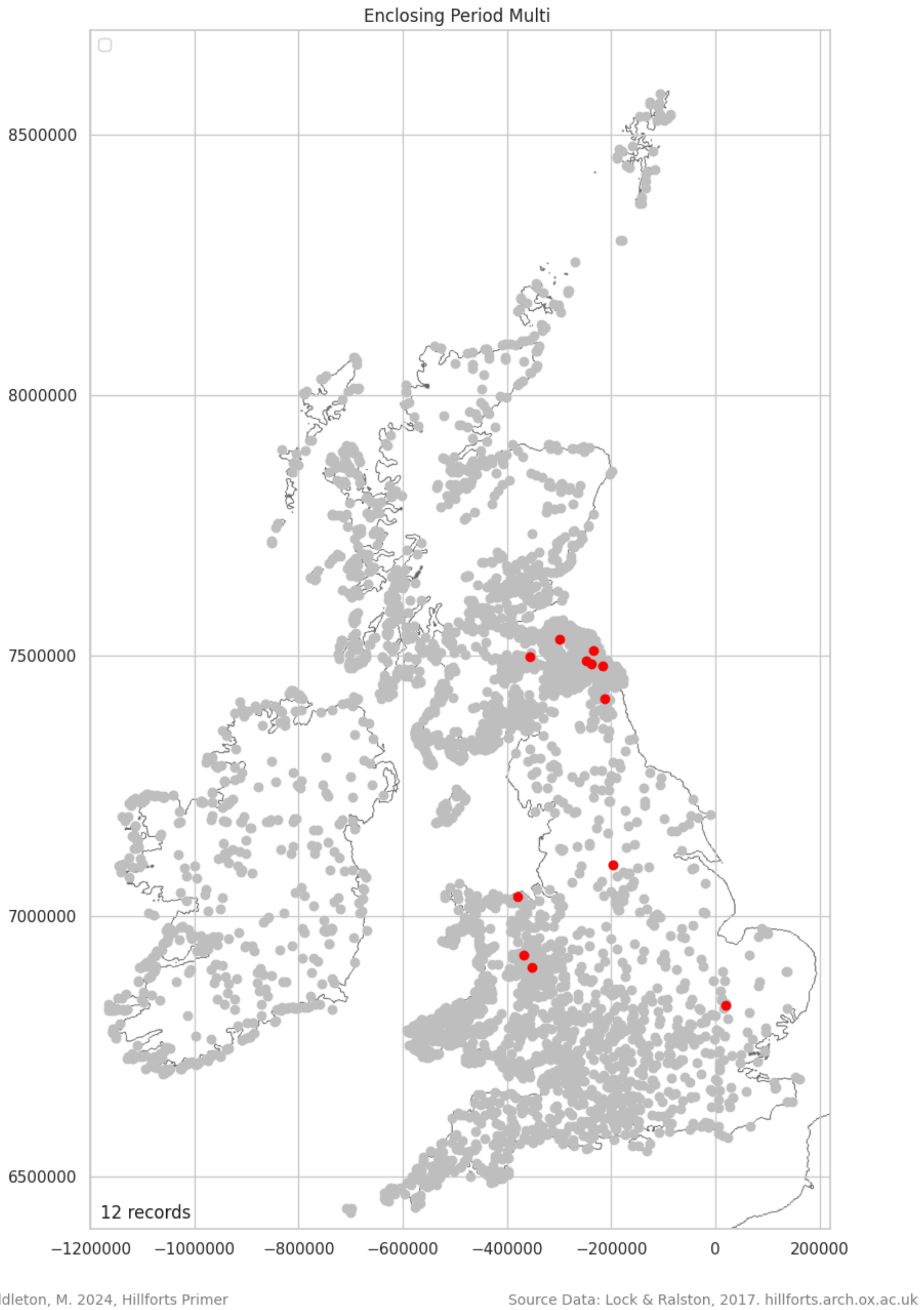
In [ ]:
```
period_multi_counts = \
enclosing_encodeable_data['Enclosing_Period_Multi'].value_counts()
period_multi_counts
```

Out[ ]:
```
No      4135
Yes       12
Name: Enclosing_Period_Multi, dtype: int64
```

In [ ]:
```
print(f'{round(period_multi_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

0.29%

In [ ]:
```
period_multi_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Period_Multi', 'Yes', '')
```



0.29%

Enclosing Surface

Enclosing Surface relates to the character of the enclosing circuit.

Enclosing Surface None Mapped

702 (16.93%) of hillforts have no information regarding the character of the enclosing circuit.

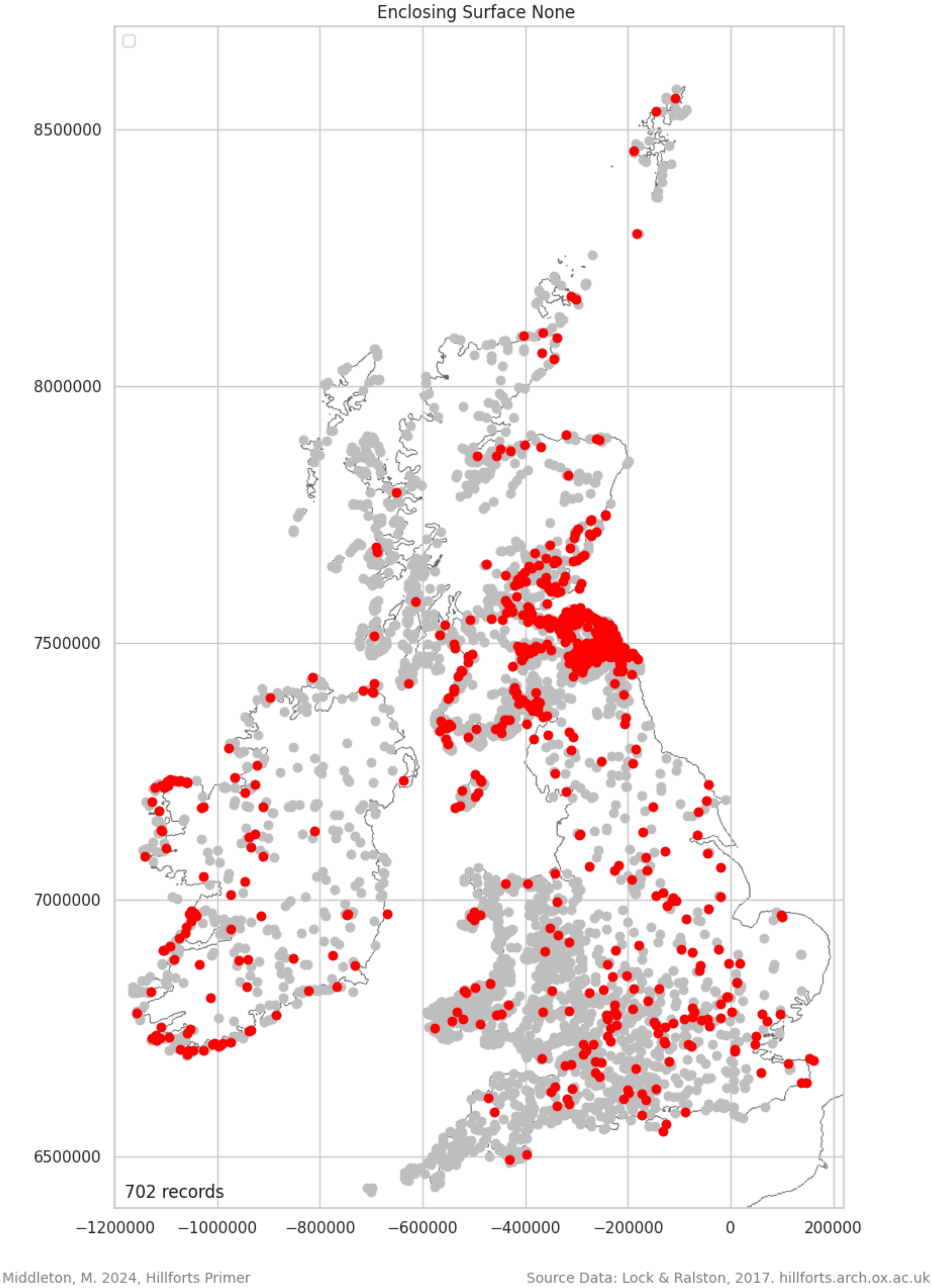
In [ ]:
```
surface_none_counts = \
enclosing_encodeable_data['Enclosing_Surface_None'].value_counts()
surface_none_counts
```

Out[ ]:
```
No     3445
Yes     702
Name: Enclosing_Surface_None, dtype: int64
```

In [ ]:
```
print(f'{round(surface_none_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
16.93%
```

In [ ]:
```
surface_none_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Surface_None', \
               'Yes', '')
```

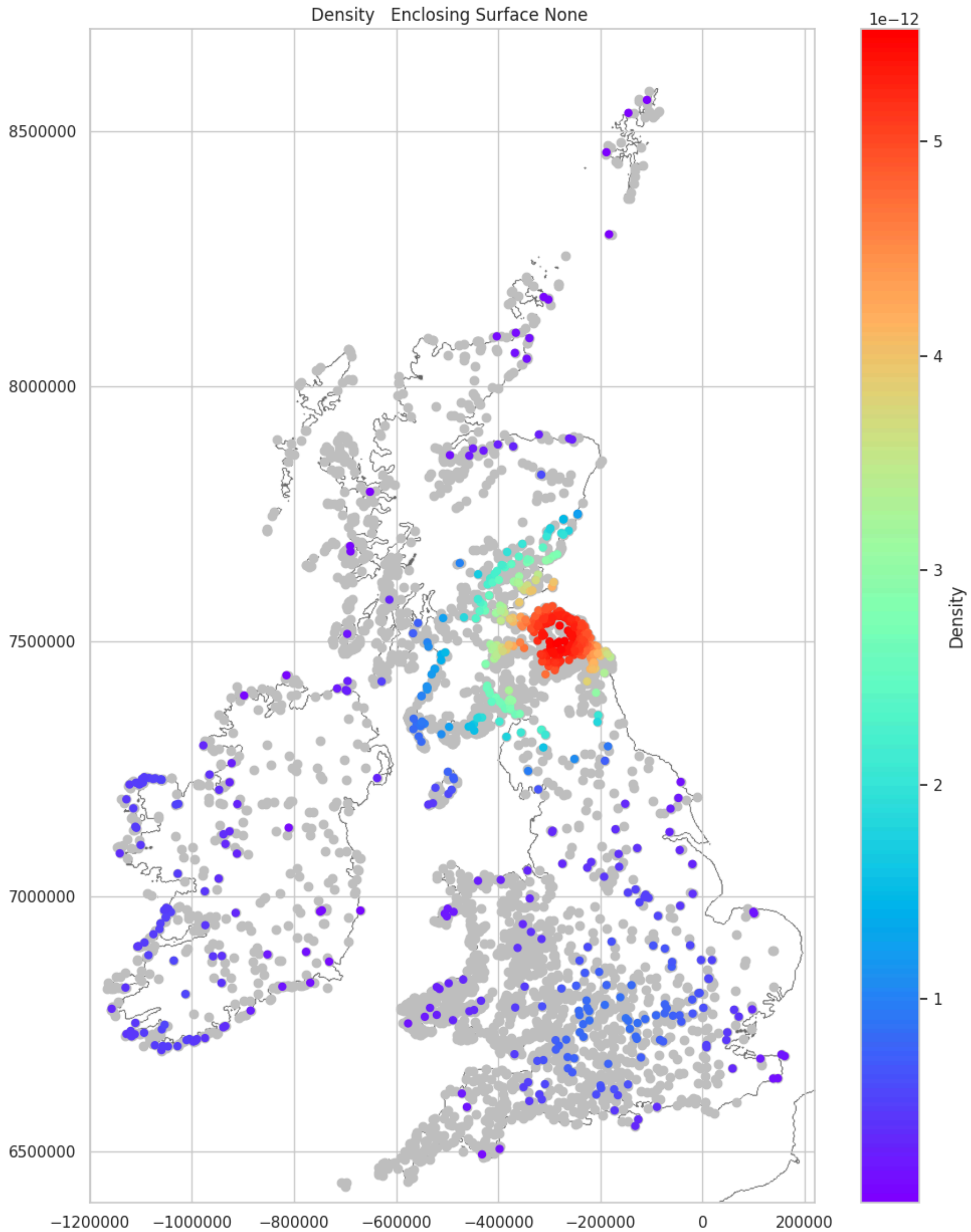



```
16.93%
```

Enclosing Surface None Density Mapped

Most hillforts, which have no information regarding the enclosing circuit, are in the Northeast.

In [ ]:
```
plot_density_over_grey(surface_none_data_yes, 'Enclosing_Surface_None')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Surface Bank Mapped

1782 (42.97%) of hillforts have an enclosing bank. What is most noticeable from this is how few forts, north and west of the Highland Boundary Fault, fall into this class.

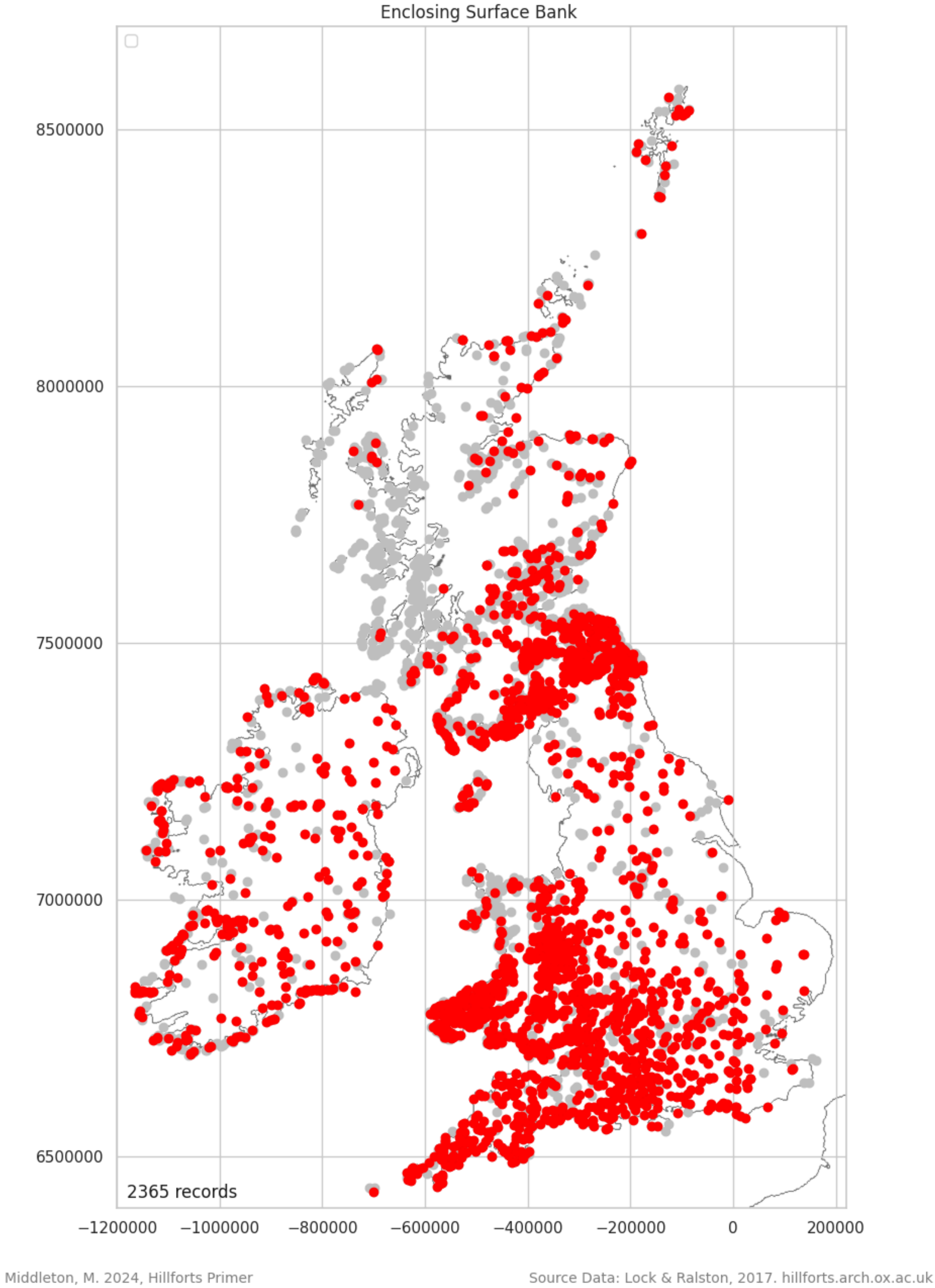
In [ ]:
```
surface_bank_counts = \
enclosing_encodeable_data['Enclosing_Surface_Bank'].value_counts()
surface_bank_counts
```

Out[ ]:
```
Yes     2365
No      1782
Name: Enclosing_Surface_Bank, dtype: int64
```

In [ ]:
```
print(f'{round(surface_bank_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
42.97%
```

In [ ]:
```
surface_bank_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Bank', 'Yes', '')
```

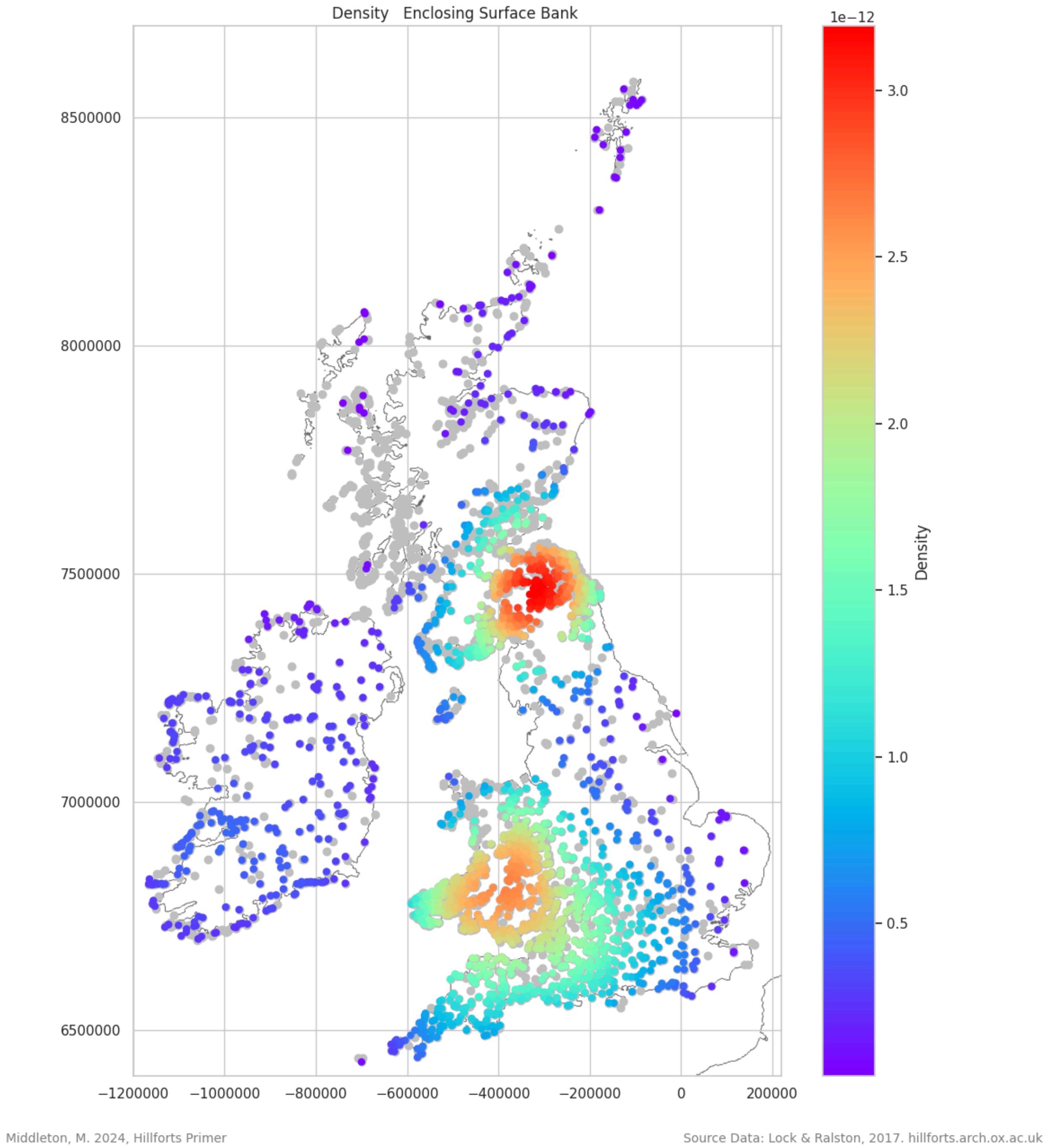



```
57.03%
```

Enclosing Surface Bank Density Mapped

There are two main clusters in this class. The most intense is in the Northeast while the second is to the southern end of the Cambrian Mountains. There looks to be a relatively even distribution of this class across the whole of Ireland.

In [ ]:
```
plot_density_over_grey(surface_bank_data_yes, 'Enclosing_Surface_Bank')
```



Enclosing Surface Wall Mapped

987 (23.8%) of hillforts have an enclosing wall. Unsurprisingly, these are located predominantly in the areas of hard, exposed geology.

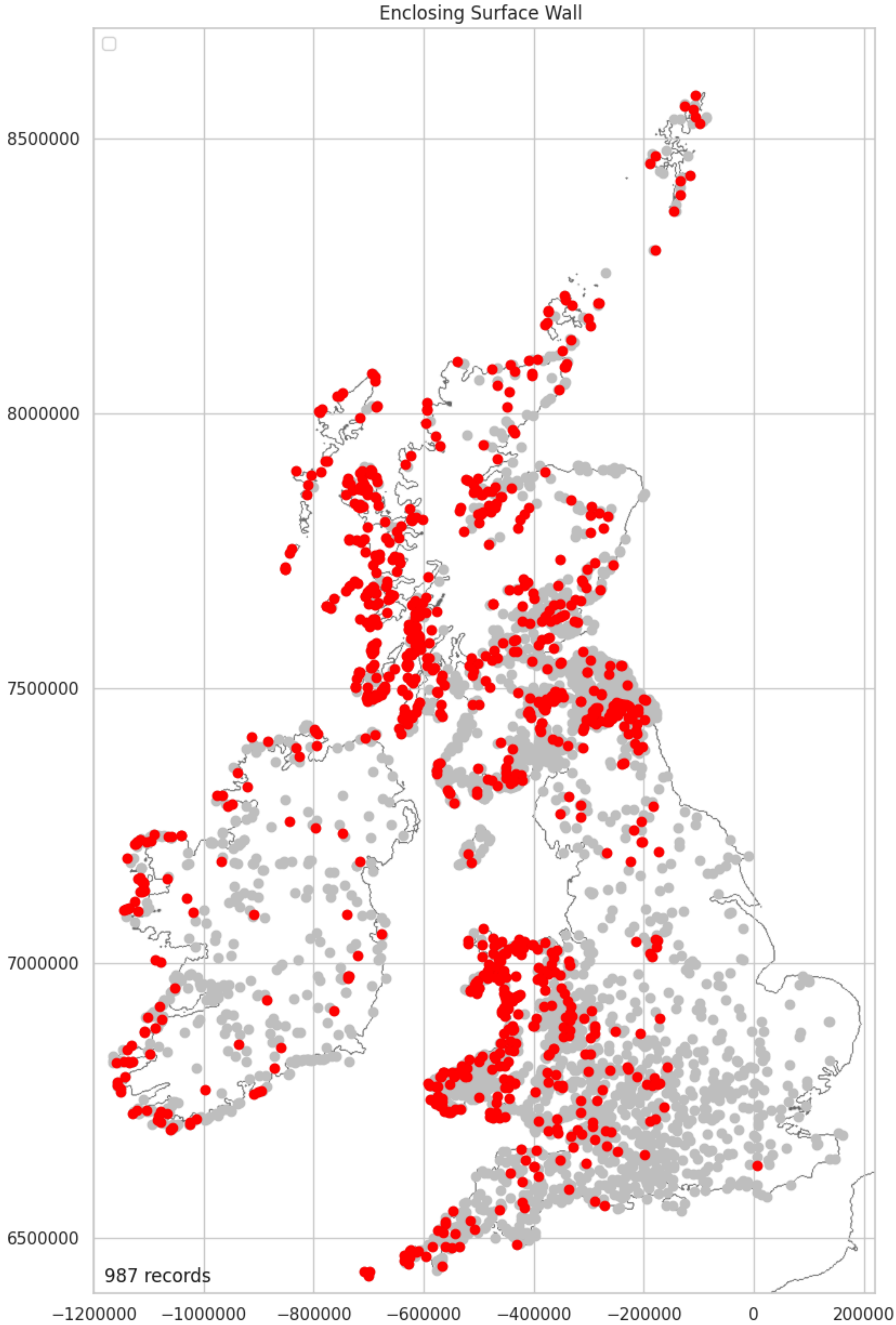
In [ ]:
```
surface_wall_counts = enclosing_encodeable_data['Enclosing_Surface_Wall'].value_counts()
surface_wall_counts
```

Out[ ]:
```
No     3160
Yes     987
Name: Enclosing_Surface_Wall, dtype: int64
```

In [ ]:
```
print(f'{round(surface_wall_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
23.8%
```

In [ ]:
```
surface_wall_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Wall', 'Yes', '')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

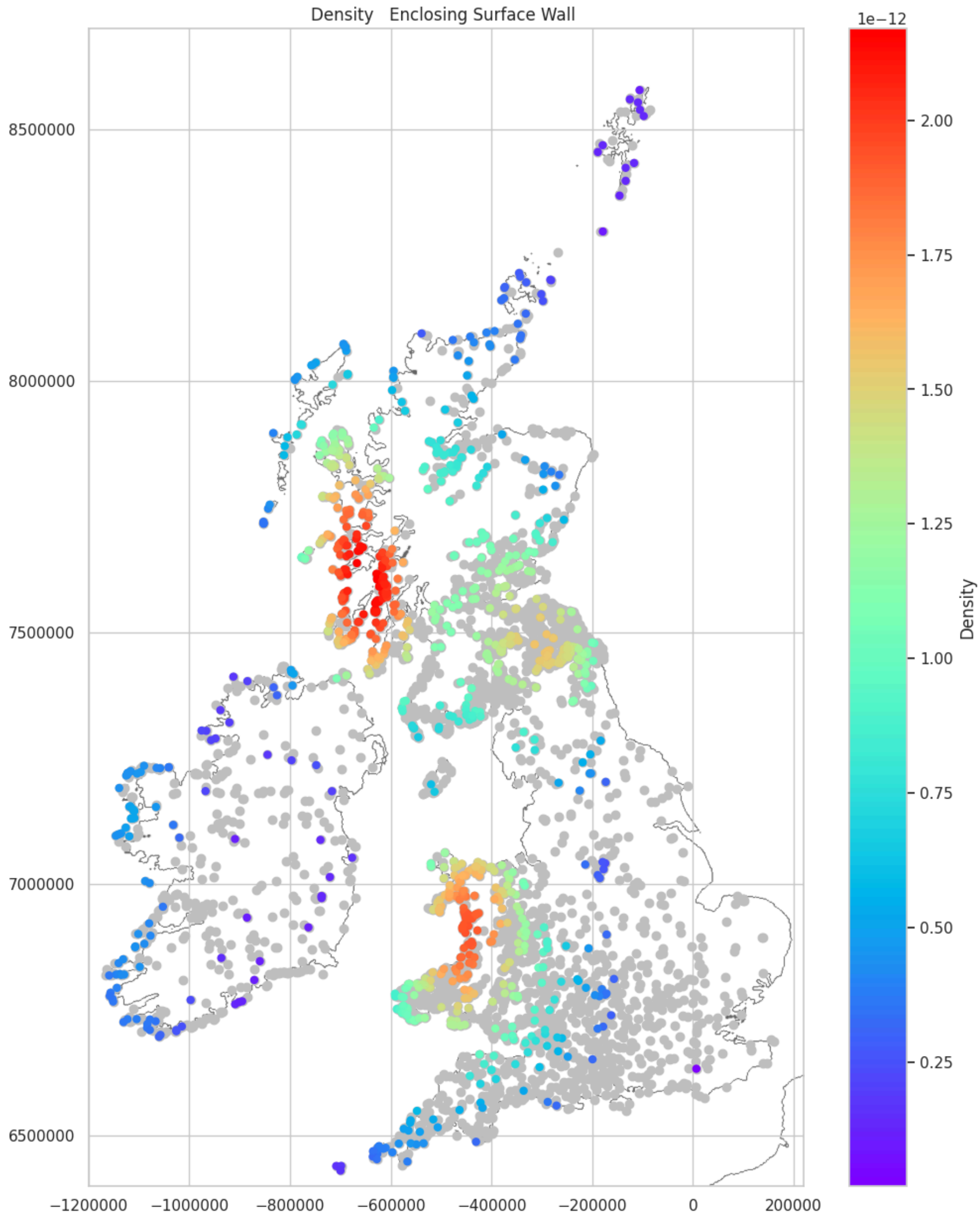
```
23.8%
```

Enclosing Surface Wall Density Mapped

Walls are focussed, most intensely, in the Northwest and in northwest Wales. There is a small cluster in the Northeast. In Ireland, coastal forts dominate the local distribution.

In [ ]:

```
plot_density_over_grey(surface_wall_data_yes, 'Enclosing_Surface_Wall')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Surface Rubble Mapped

659 (15.89%) of hillforts have a rubble enclosing circuit.

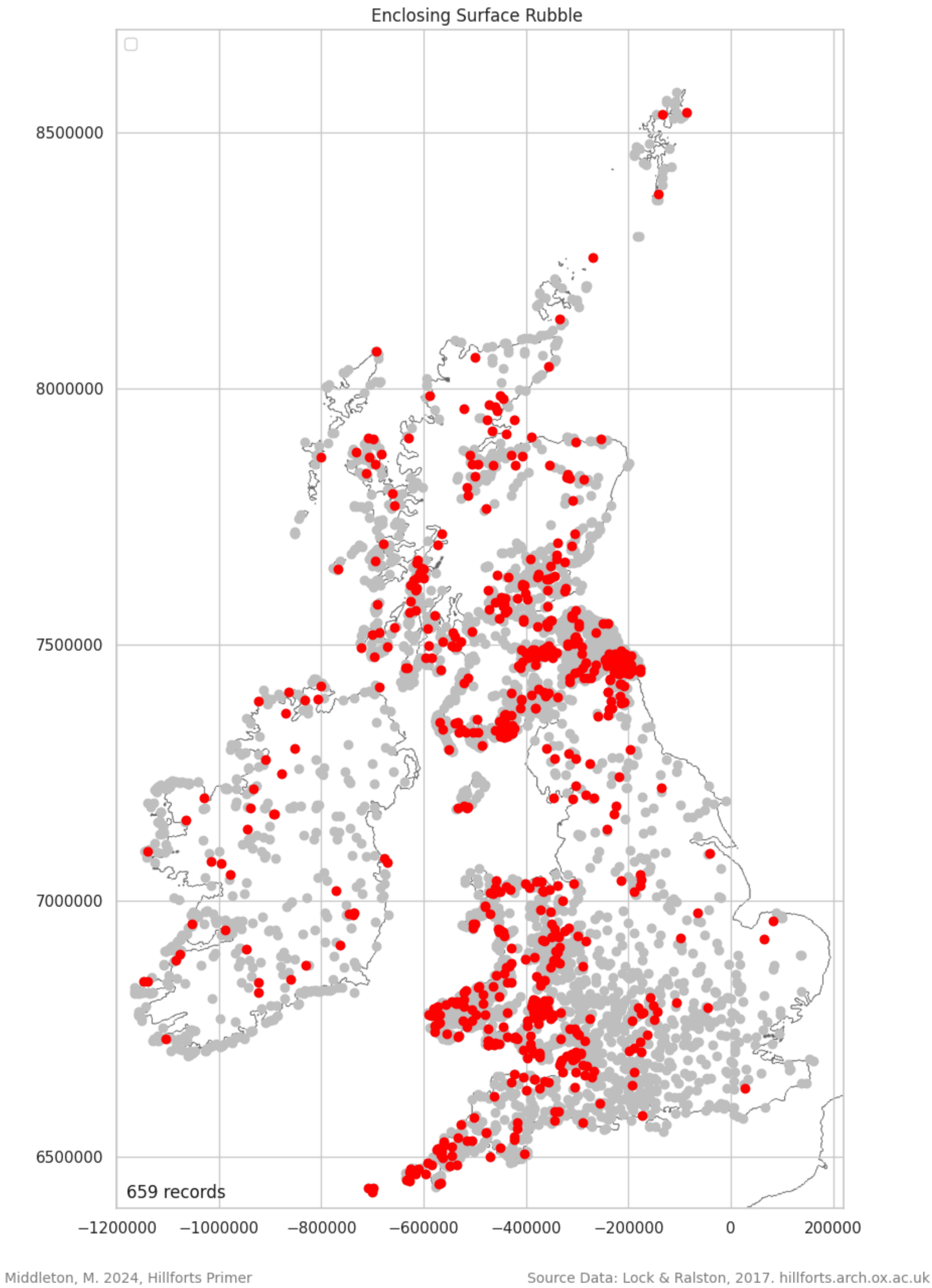
In [ ]:
```
surface_rubble_counts = \
enclosing_encodeable_data['Enclosing_Surface_Rubble'].value_counts()
surface_rubble_counts
```

Out[ ]:
```
No      3488
Yes      659
Name: Enclosing_Surface_Rubble, dtype: int64
```

In [ ]:
```
print(f'{round(surface_rubble_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
15.89%
```

In [ ]:
```
surface_rubble_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
                'Enclosing_Surface_Rubble', 'Yes', '')
```

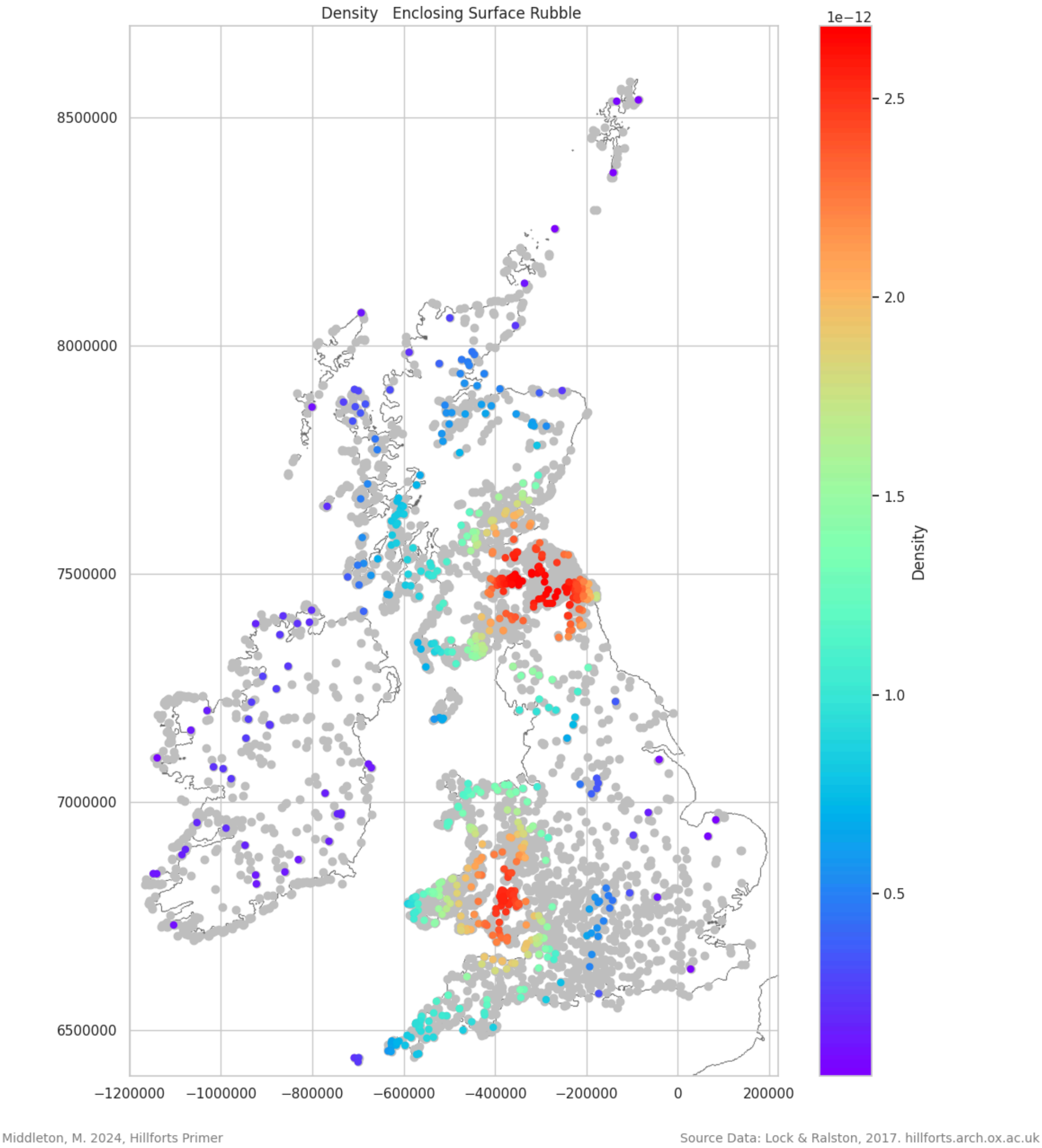


15.89%

Enclosing Surface Rubble Density Mapped

This class has two main clusters. The first in the Northeast and a second focussed over the Brecon Beacons, in the South.

In [ ]:
```
plot_density_over_grey(surface_rubble_data_yes, 'Enclosing_Surface_Rubble')
```



Enclosing Surface Walk Mapped

Just 15 (0.36%) hillforts have evidence for a Surface Walk.

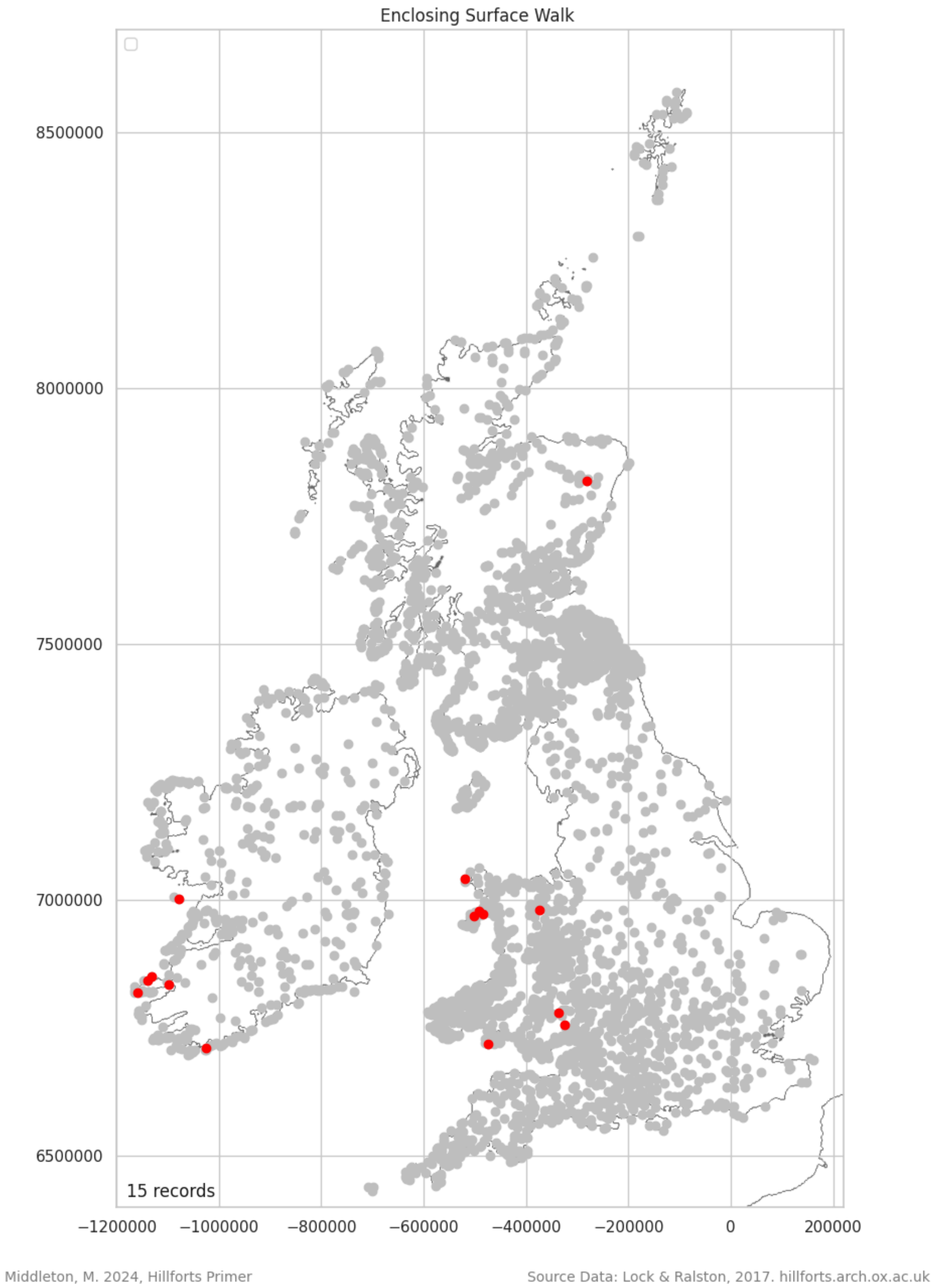
In [ ]:
```
surface_walk_counts = \
enclosing_encodeable_data['Enclosing_Surface_Walk'].value_counts()
surface_walk_counts
```

Out[ ]:
```
No      4132
Yes       15
Name: Enclosing_Surface_Walk, dtype: int64
```

In [ ]:
```
print(f'{round(surface_walk_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.36%
```

In [ ]:
```
surface_walk_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Walk', 'Yes', '')
```




```
0.36%
```

Enclosing Surface Timber Mapped

Only 2 hillforts have evidence for Surface Timber.

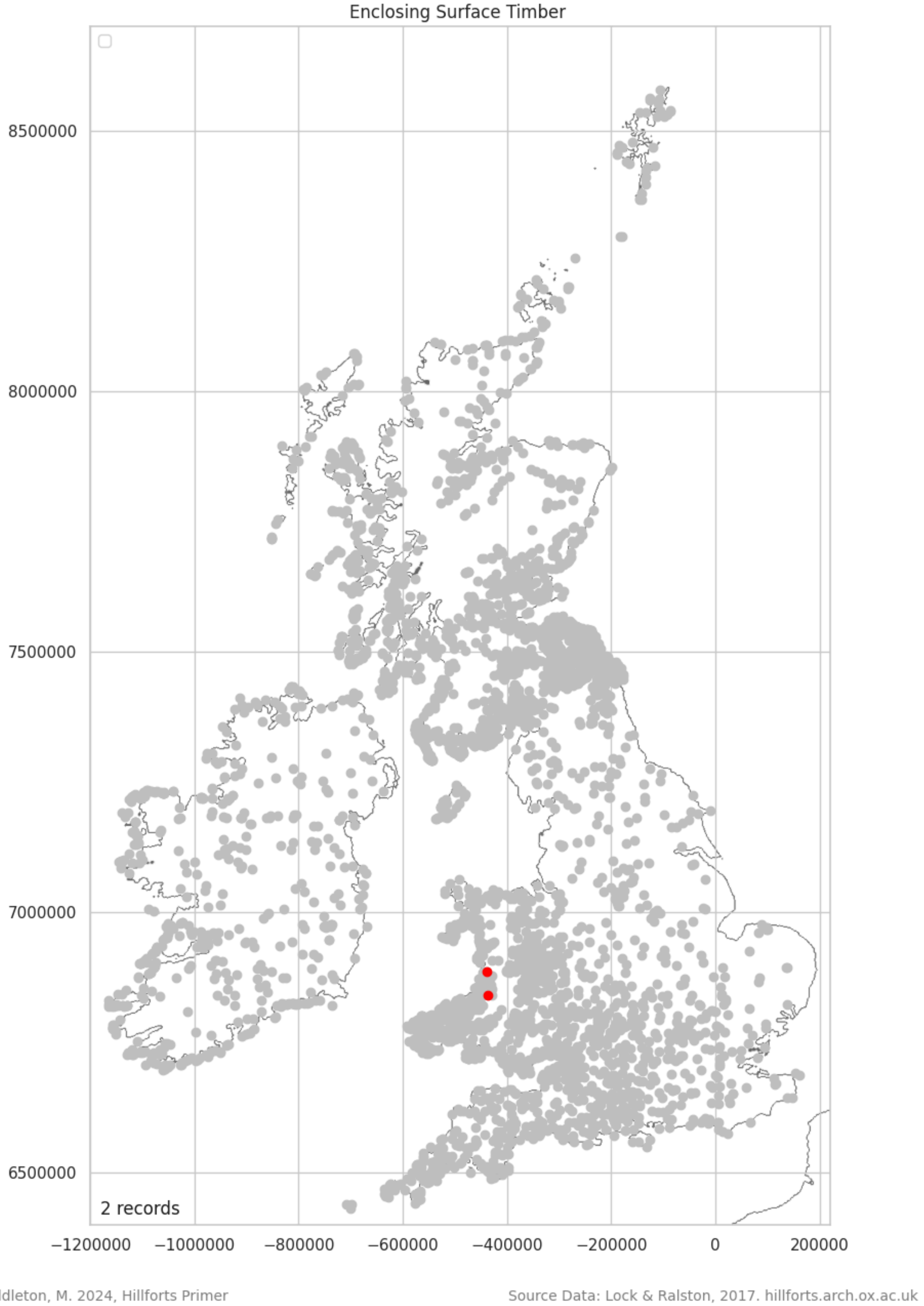
In [ ]:
```
surface_timber_counts = \
enclosing_encodeable_data['Enclosing_Surface_Timber'].value_counts()
surface_timber_counts
```

Out[ ]: No   4145
Yes   2
Name: Enclosing_Surface_Timber, dtype: int64

In [ ]:
```
print(f'{round(surface_timber_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

0.05%

In [ ]:
```
surface_timber_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Timber', 'Yes', '')
```



0.05%

Enclosing Surface Vitrification Mapped

88 (2.12%) hillforts show signs of vitrification. These are almost entirely in the North. See: Enclosing Excavation Vitrification Mapped

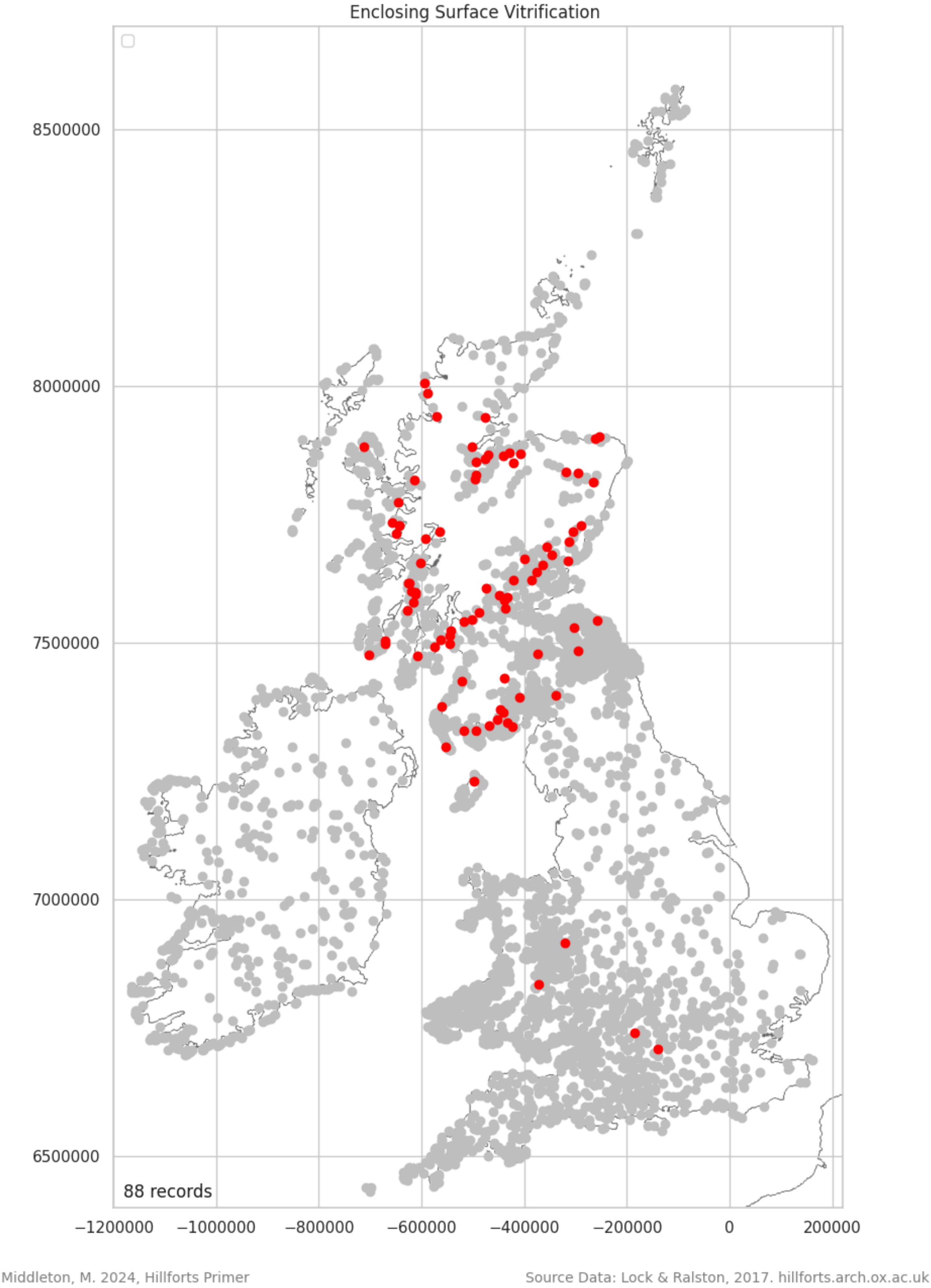
In [ ]:
```
surface_vitrification_counts = \
enclosing_encodeable_data['Enclosing_Surface_Vitrification'].value_counts()
surface_vitrification_counts
```

Out[ ]:
```
No     4059
Yes      88
Name: Enclosing_Surface_Vitrification, dtype: int64
```

In [ ]:
```
print(f'{round(surface_vitrification_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
2.12%
```

In [ ]:
```
surface_vitrification_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Vitrification', 'Yes', '')
```

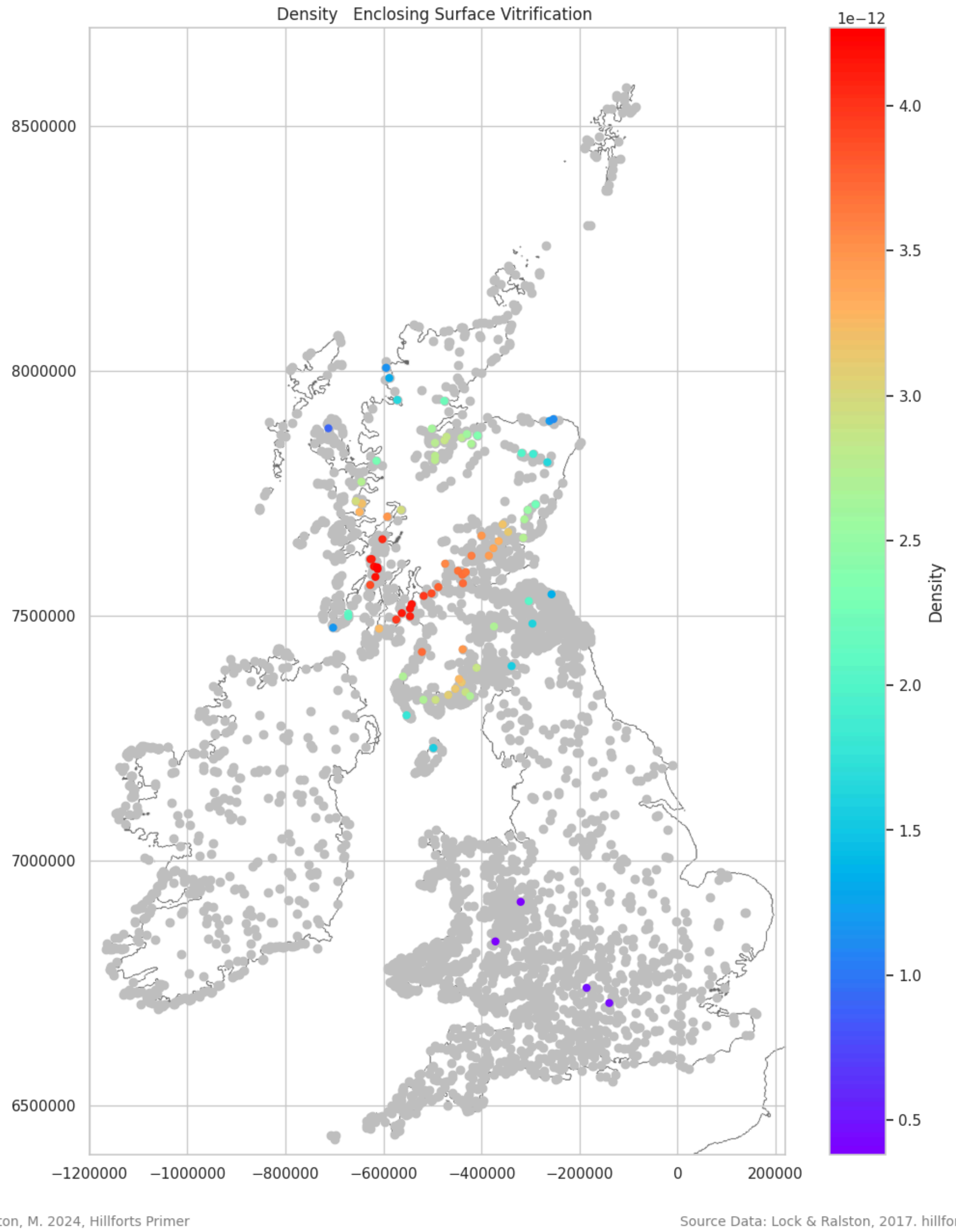



```
2.12%
```

Enclosing Surface Vitrification Density Mapped

The main concentration of vitrified hillforts is in the vicinity of Dunnad, along the Clyde Valley and up the Highland Boundary Fault. This density plot has been produced using very few records and extra caution should be taken in not over interpreting these results. This class is also likely to have a recording bias in that vitrification is notorious for being misidentified. See: Enclosing Excavation Vitrification Mapped.

In [ ]:
```
plot_density_over_grey(surface_vitrification_data_yes, \
                       'Enclosing_Surface_Vitrification')
```



Enclosing Surface Burning Mapped

Only eight (0.19%) hillforts have signs of 'Other Burning'.

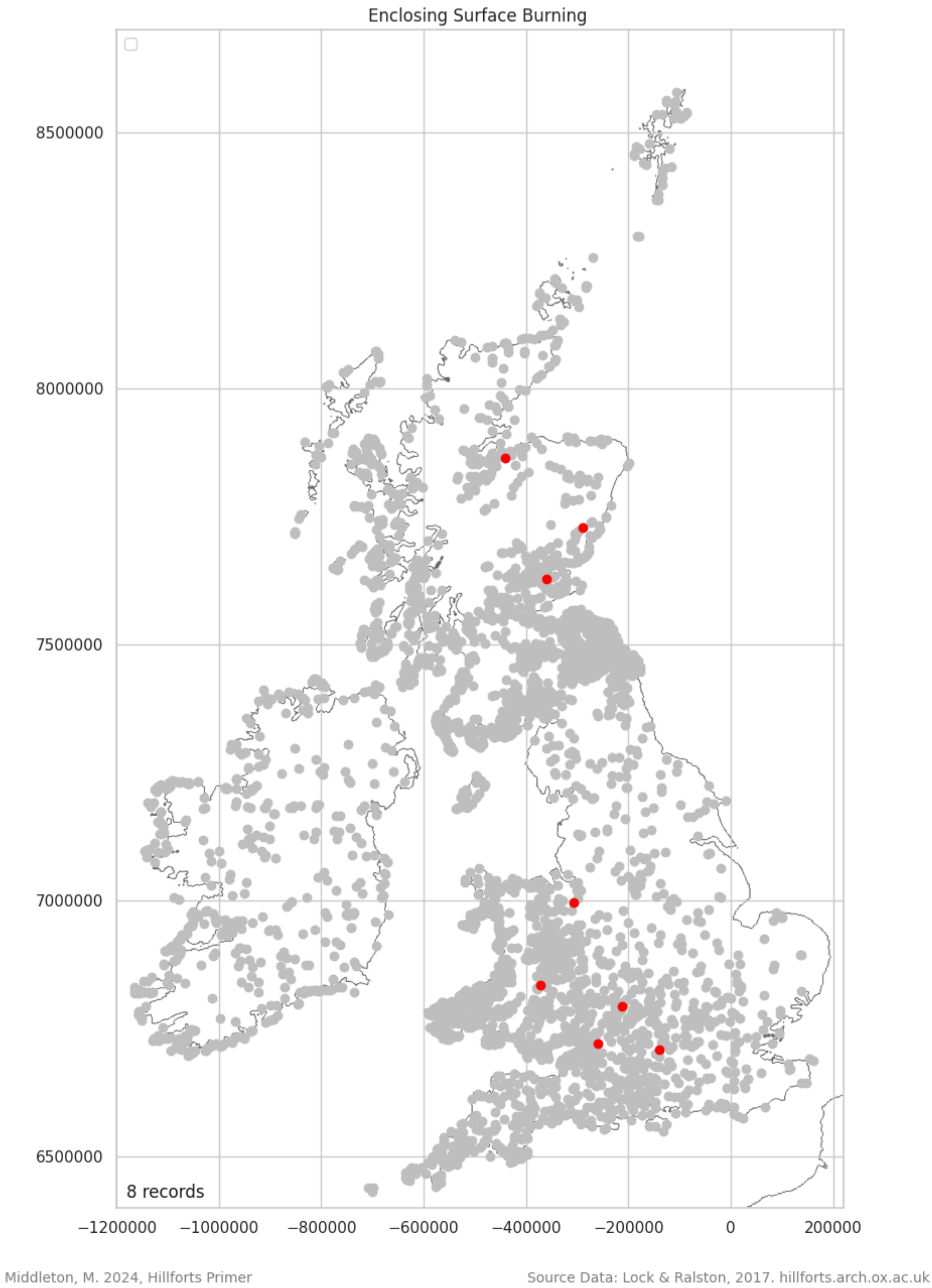
In [ ]:
```
surface_burning_counts = \
enclosing_encodeable_data['Enclosing_Surface_Burning'].value_counts()
surface_burning_counts
```

Out[ ]:
```
No     4139
Yes       8
Name: Enclosing_Surface_Burning, dtype: int64
```

In [ ]:
```
print(f'{round(surface_burning_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.19%
```

In [ ]:
```
surface_burning_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Burning', 'Yes', '')
```




```
0.19%
```

Enclosing Surface Palisade Mapped

135 (3.26%) of hillforts have recorded evidence for a palisade.

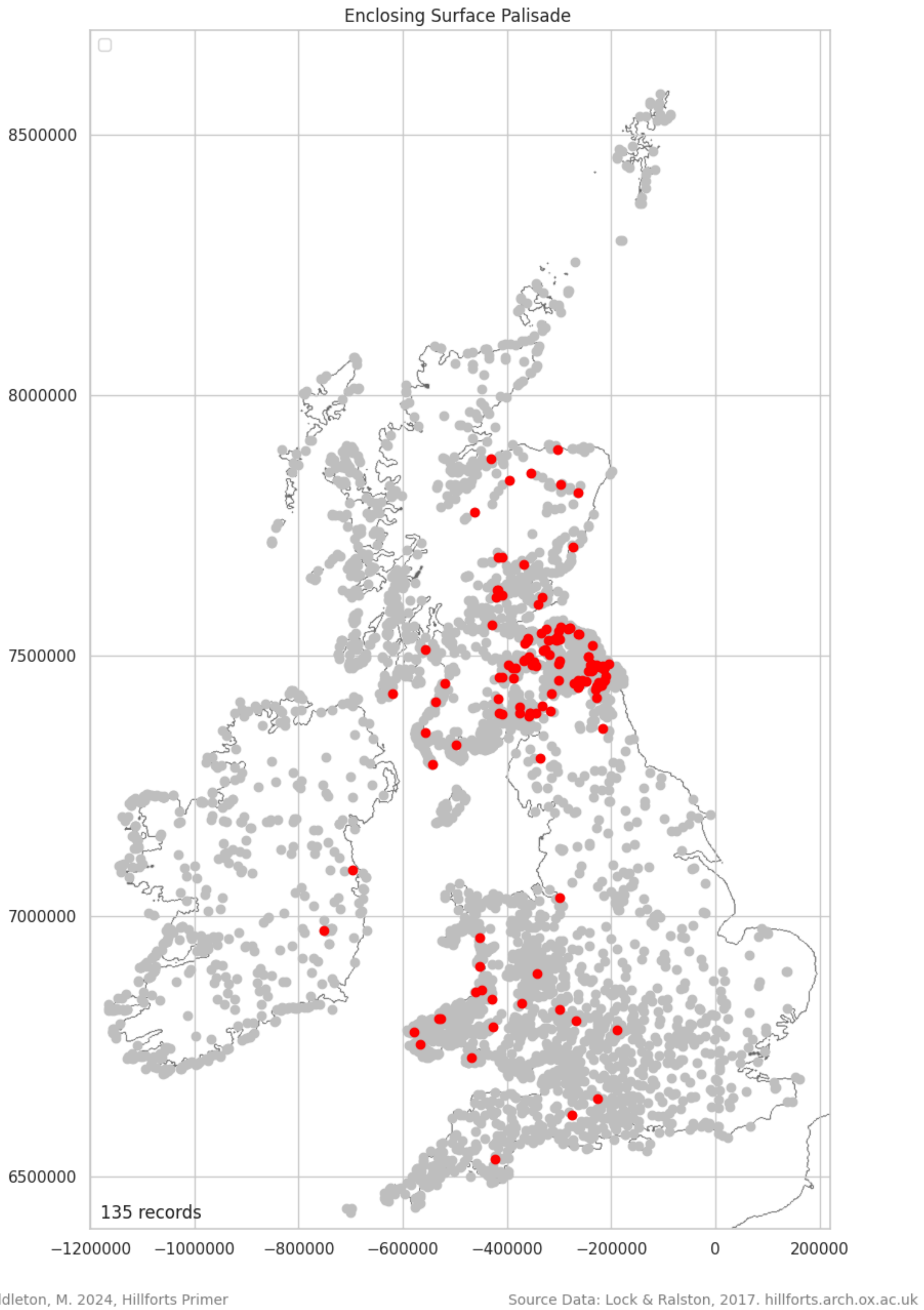
```
In [ ]: surface_palisade_counts = \
        enclosing_encodeable_data['Enclosing_Surface_Palisade'].value_counts()
        surface_palisade_counts
```

```
Out[ ]: No     4012
        Yes     135
        Name: Enclosing_Surface_Palisade, dtype: int64
```

```
In [ ]: print(f'{round(surface_palisade_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
3.26%
```

```
In [ ]: surface_palisade_data_yes = \
        plot_over_grey(location_enclosing_encodeable_data, \
                       'Enclosing_Surface_Palisade', 'Yes', '')
```

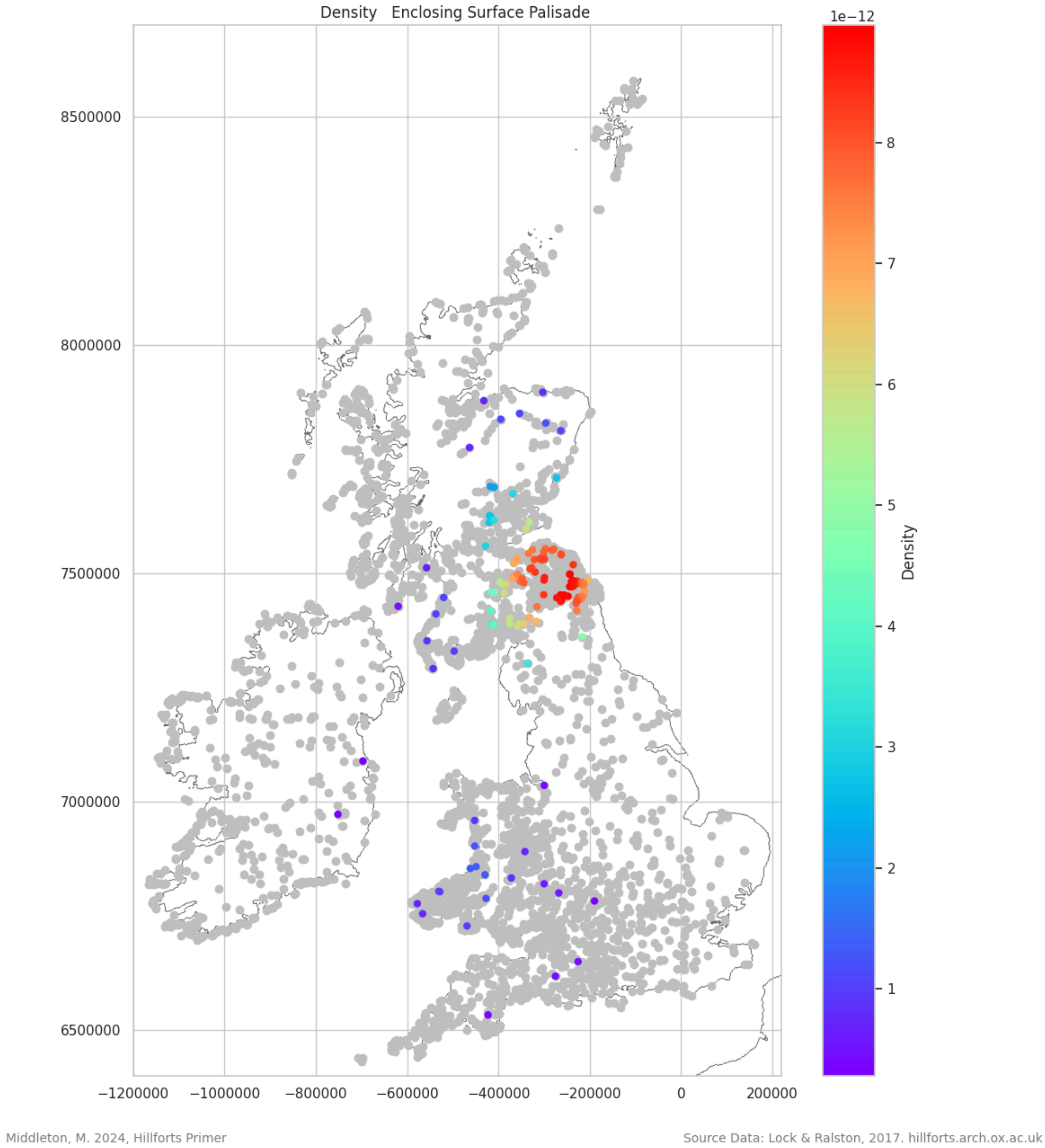



```
3.26%
```

Enclosing Surface Palisade Density Mapped

The main cluster for palisades is in the Northeast. Due to the ephemeral nature of these features this class is likely to have a recording bias toward areas where surveyors have been trained to identify these features.

In [ ]:
```
plot_density_over_grey(surface_palisade_data_yes, 'Enclosing_Surface_Palisade')
```



Enclosing Surface Counter Scarp Mapped

561 (13.53%) of hillforts have a counterscarp. It is assumed that a counterscarp requires the presence of a ditch although there are ten hillforts where this is not the case.

In [ ]:
```
surface_scarp_counts = \
enclosing_encodeable_data['Enclosing_Surface_Counter_Scarp'].value_counts()
surface_scarp_counts
```

Out[ ]:
```
No     3586
Yes     561
Name: Enclosing_Surface_Counter_Scarp, dtype: int64
```

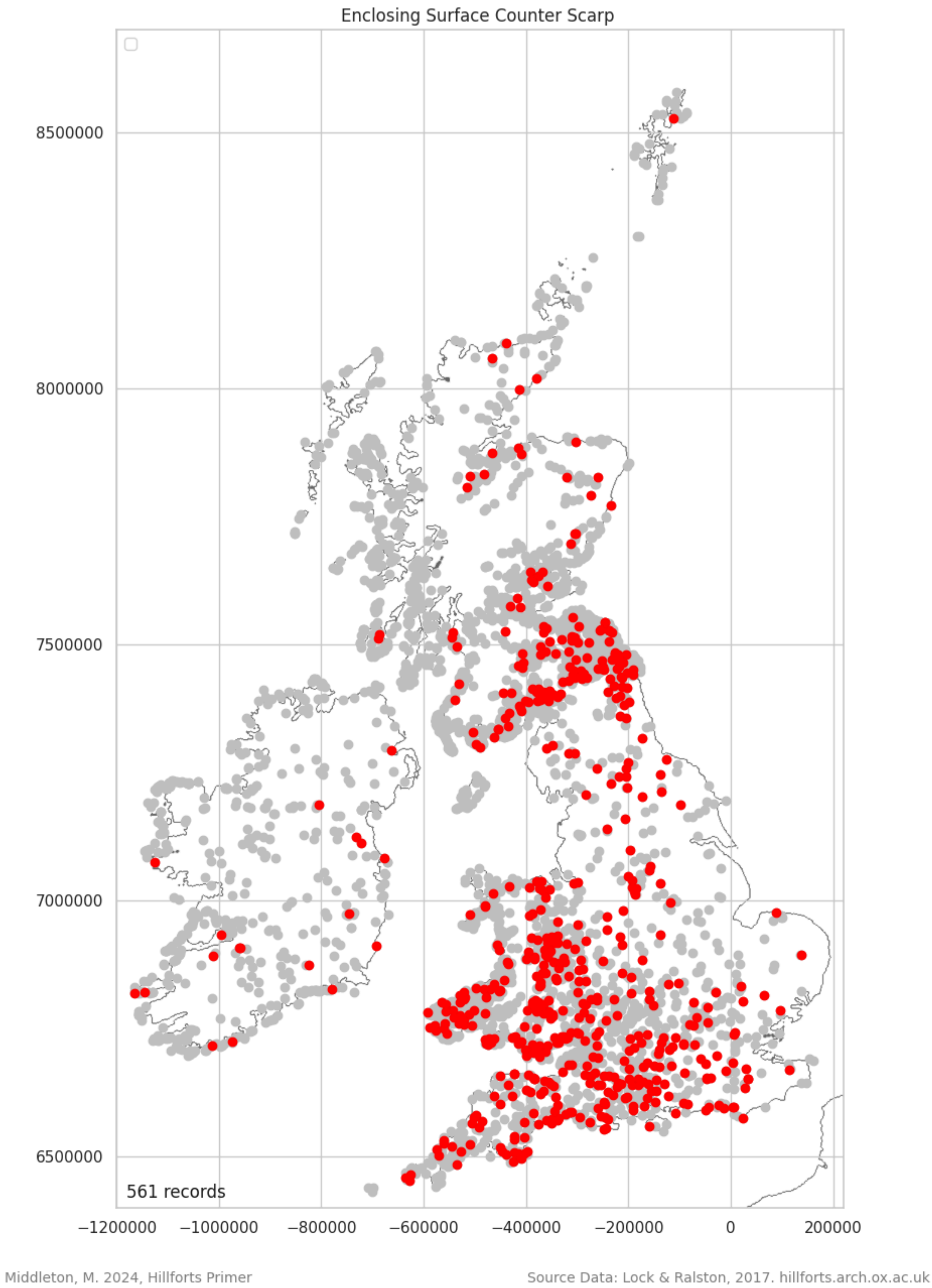
In [ ]:
```
print(f'{round(surface_scarp_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

13.53%

In [ ]:
```
counterscarp_ditch = \
len(enclosing_data[(enclosing_data['Enclosing_Surface_Counter_Scarp'] == "Yes") & \
 (enclosing_data['Enclosing_Ditches_Number'] > 0)])
counterscarp_ditch
```

Out[ ]:
```
551
```

In [ ]:
```
surface_scarp_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Counter_Scarp', 'Yes', '')
```



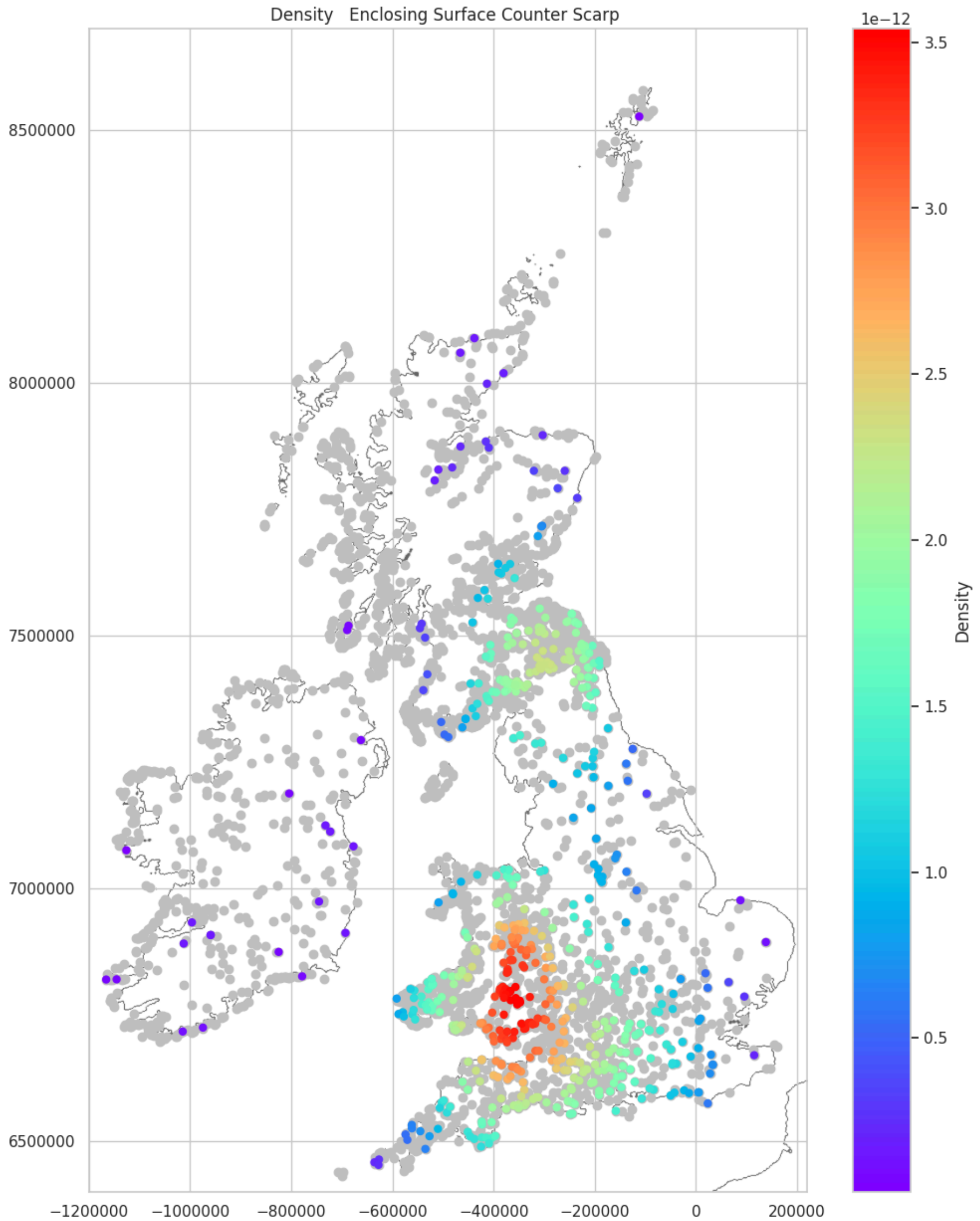

```
13.53%
```

Enclosing Surface Counter Scarp Density Mapped

The main cluster of hillforts with a counterscarp is over the Brecon Beacons and up along the eastern fringe of the Cambrian Mountains.

In [ ]:

```
plot_density_over_grey(surface_scarp_data_yes, \
                       'Enclosing_Surface_Counter_Scarp')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Surface Berm Mapped

There are 136 (3.28%) hillforts where a berm has been recorded. The distribution is unusual and is likely the result of a recording bias across the south of England and up along the Welsh border.

In [ ]:

```
surface_burm_counts = \
enclosing_encodeable_data['Enclosing_Surface_Berm'].value_counts()
surface_burm_counts
```

Out[ ]:

```
No     4011
Yes     136
Name: Enclosing_Surface_Berm, dtype: int64
```

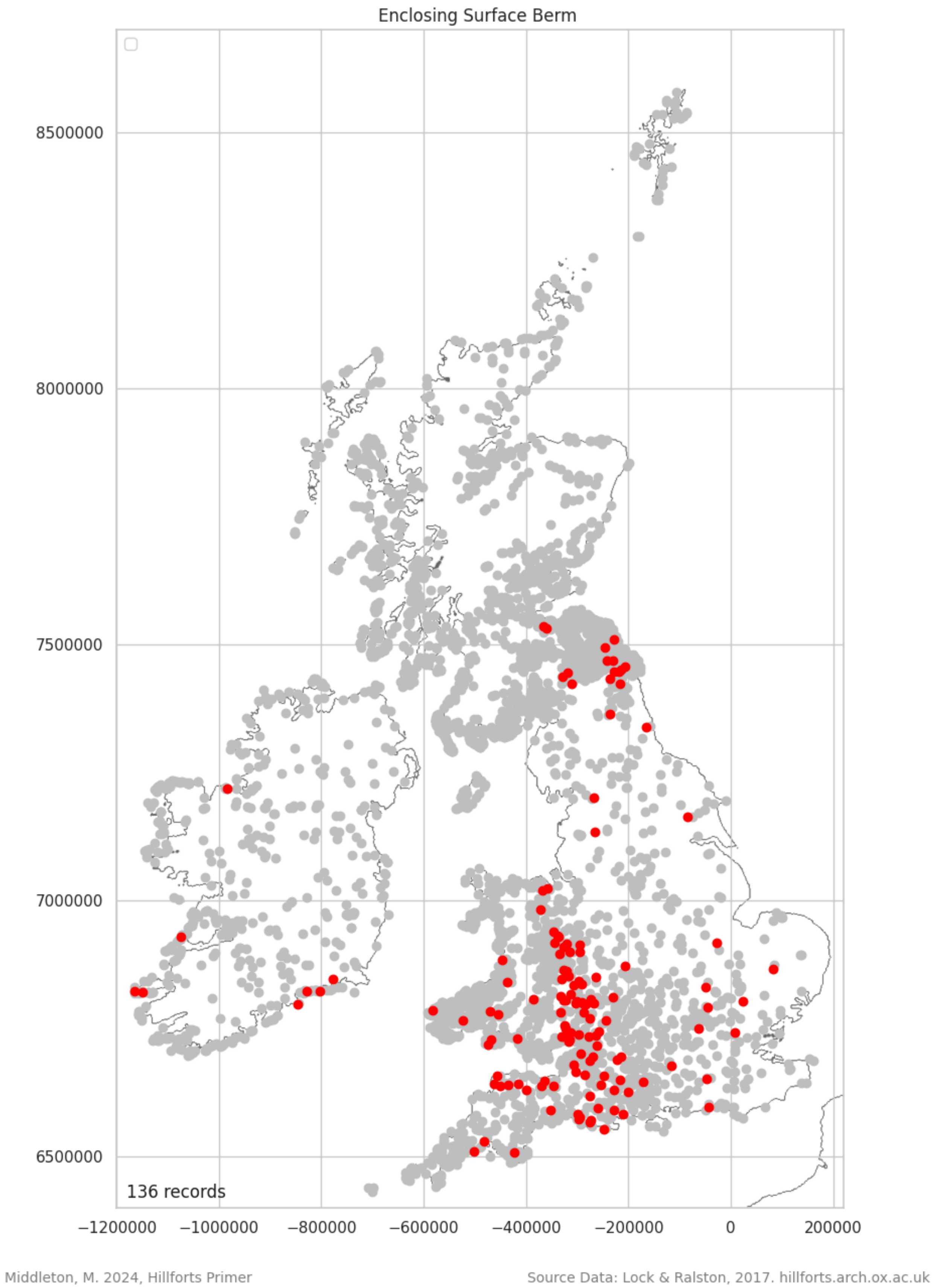
In [ ]:

```
print(f'{round(surface_burm_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
3.28%
```

In [ ]:

```
surface_burm_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Berm', 'Yes', '')
```

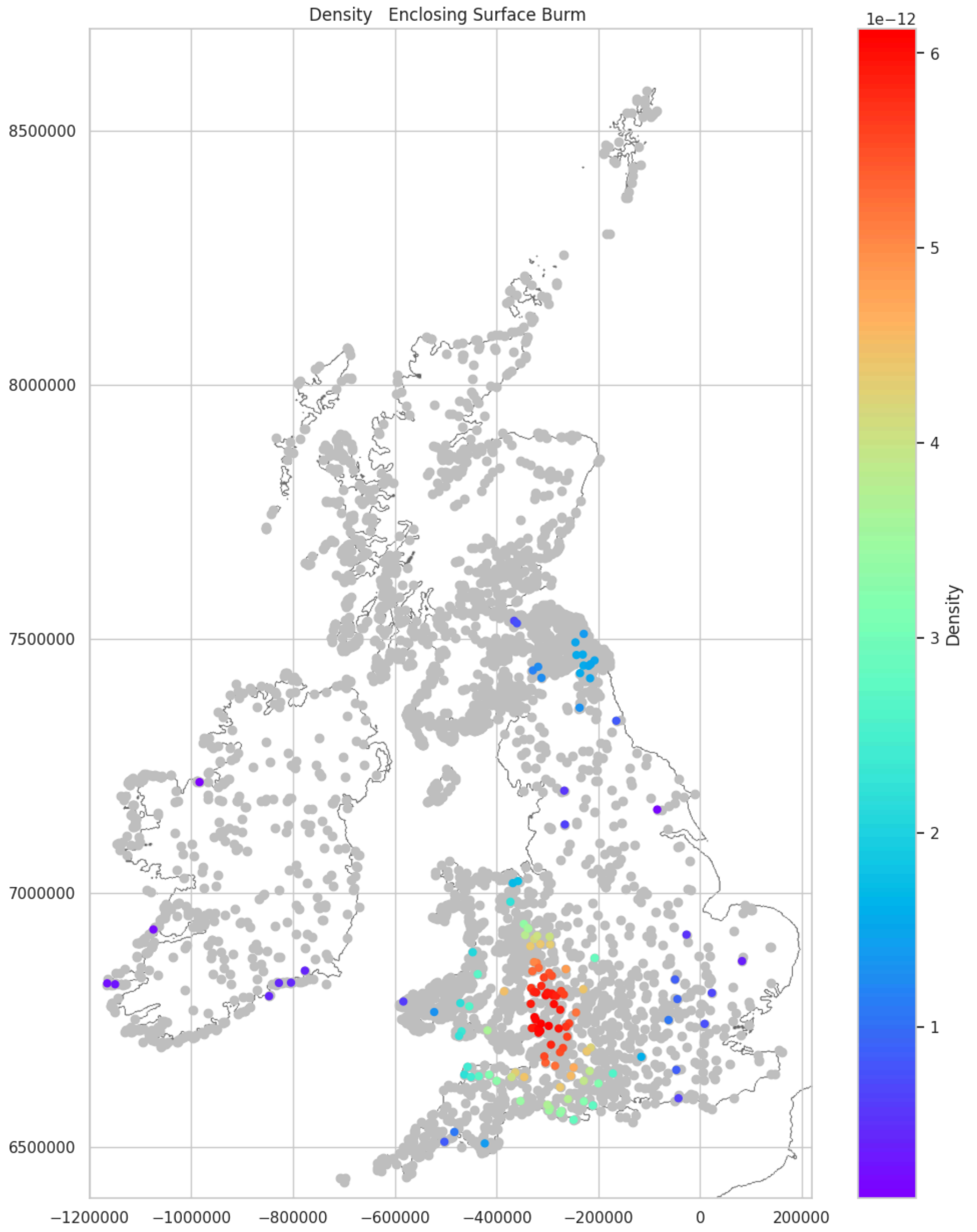



```
3.28%
```

Enclosing Surface Burm Density Mapped

This cluster is likely to be highly biased and should only be used with caution.

In [ ]:
```
plot_density_over_grey(surface_burm_data_yes, 'Enclosing_Surface_Burm')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Surface Unfinished Mapped

175 (4.22%) of hillforts have an enclosing surface that has been recorded as unfinished.

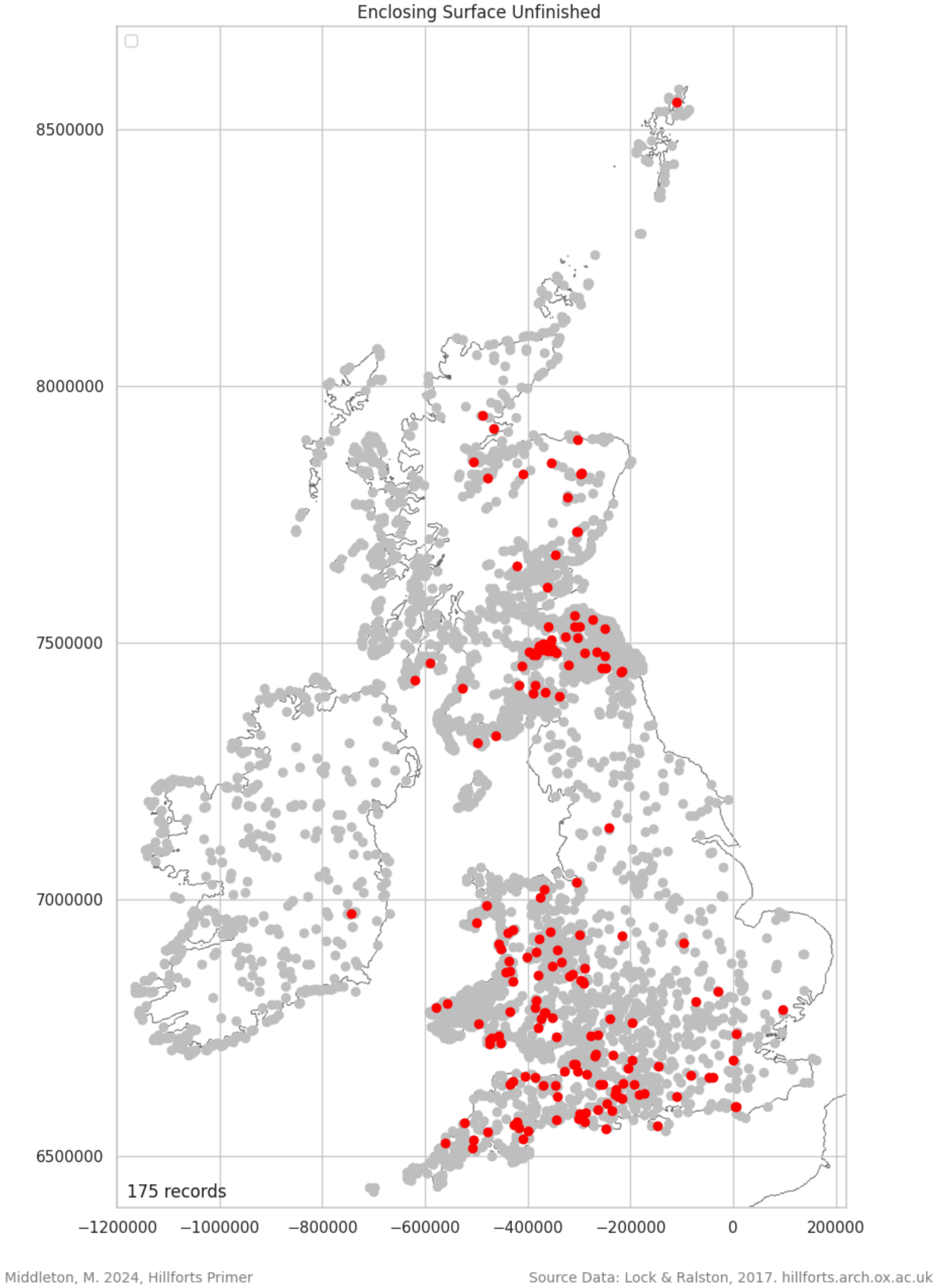
In [ ]:
```
surface_unfinished_counts = \
enclosing_encodeable_data['Enclosing_Surface_Unfinished'].value_counts()
surface_unfinished_counts
```

Out[ ]:
```
No      3972
Yes      175
Name: Enclosing_Surface_Unfinished, dtype: int64
```

In [ ]:
```
print(f'{round(surface_unfinished_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
4.22%
```

In [ ]:
```
surface_unfinished_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Surface_Unfinished', 'Yes', '')
```



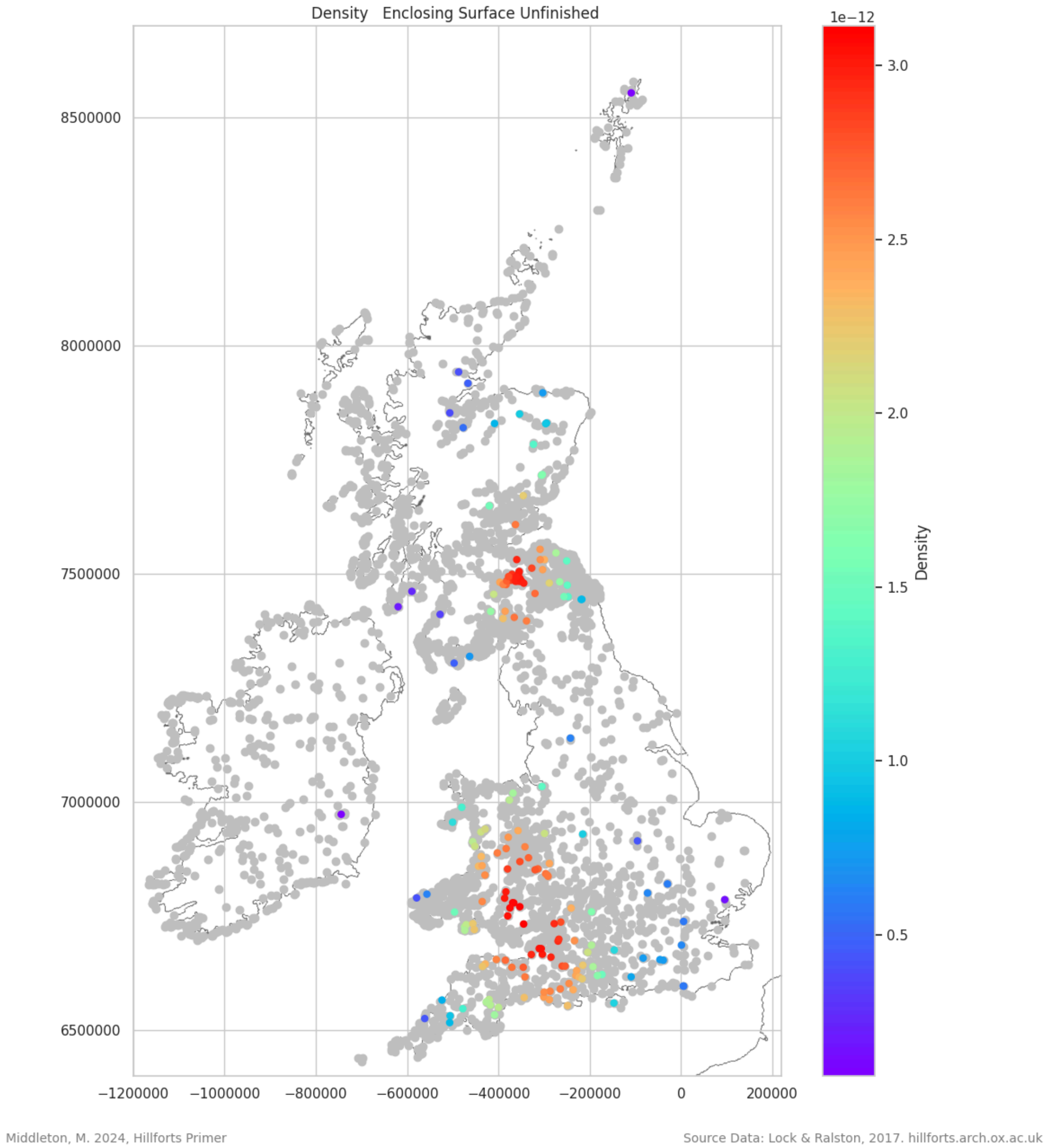

```
4.22%
```

Enclosing Surface Unfinished Density Mapped

There are two clusters. A diffuse cluster in the South and a small cluster in the North. Due to the small number of records used to create these clusters, caution should be taken to not over interpret these results

In [ ]:

```
plot_density_over_grey(surface_unfinished_data_yes, \
                       'Enclosing_Surface_Unfinished')
```



Enclosing Surface Other Mapped

80 (4.22%) of hillforts have an unclassified enclosing surface.

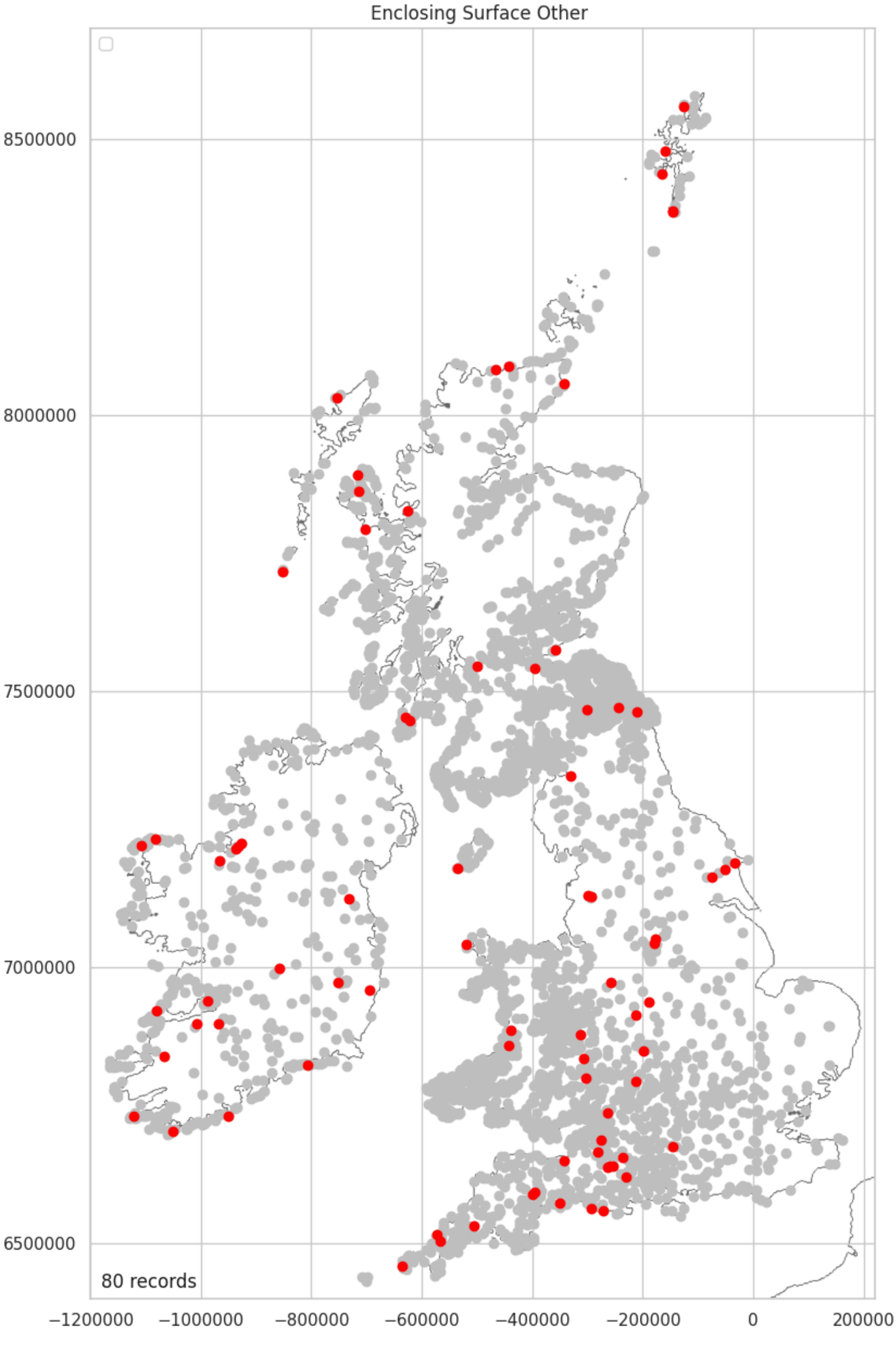
In [ ]:
```
surface_other_counts = \
enclosing_encodeable_data['Enclosing_Surface_Other'].value_counts()
surface_other_counts
```

Out[ ]:
```
No     4067
Yes      80
Name: Enclosing_Surface_Other, dtype: int64
```

In [ ]:
```
print(f'{round(surface_unfinished_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
4.22%
```

In [ ]:
```
surface_other_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
                'Enclosing_Surface_Other', 'Yes', '')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

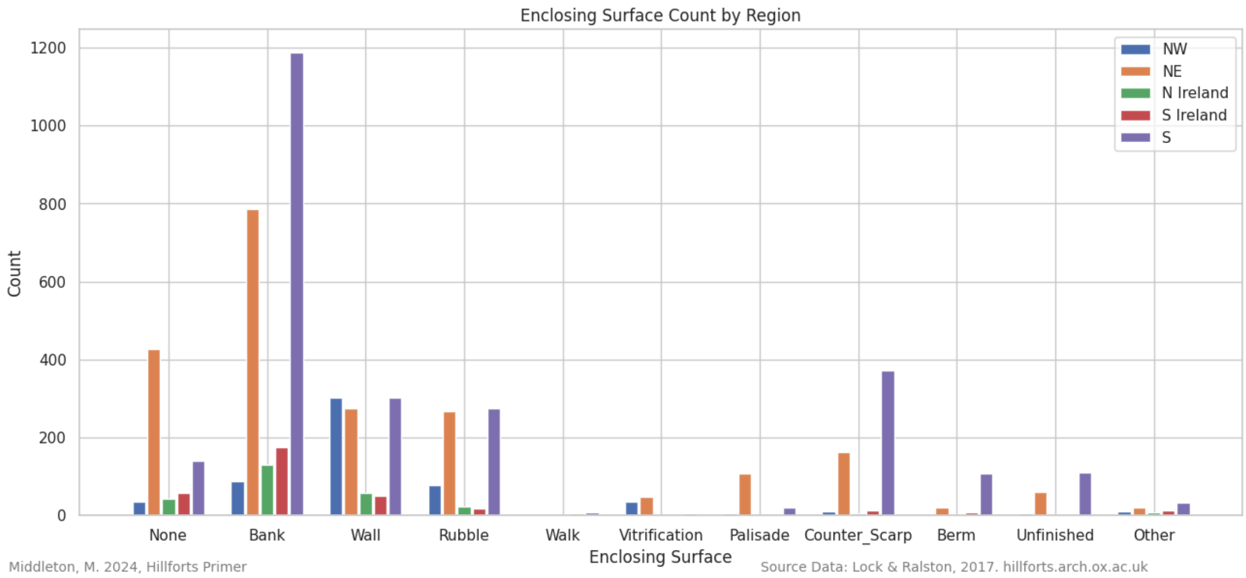
```
1.93%
```

Enclosing Surface by Region (Count)

The counts for both 'Burning' and 'Timber' are in single figures and have not been inculded in the following plots. As was seen earlier, counts can be difficult to interpret. See below for the same data presented by proportion.

In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
             location_enclosing_encodeable_data_ne,
             location_enclosing_encodeable_data_ireland_n,
```

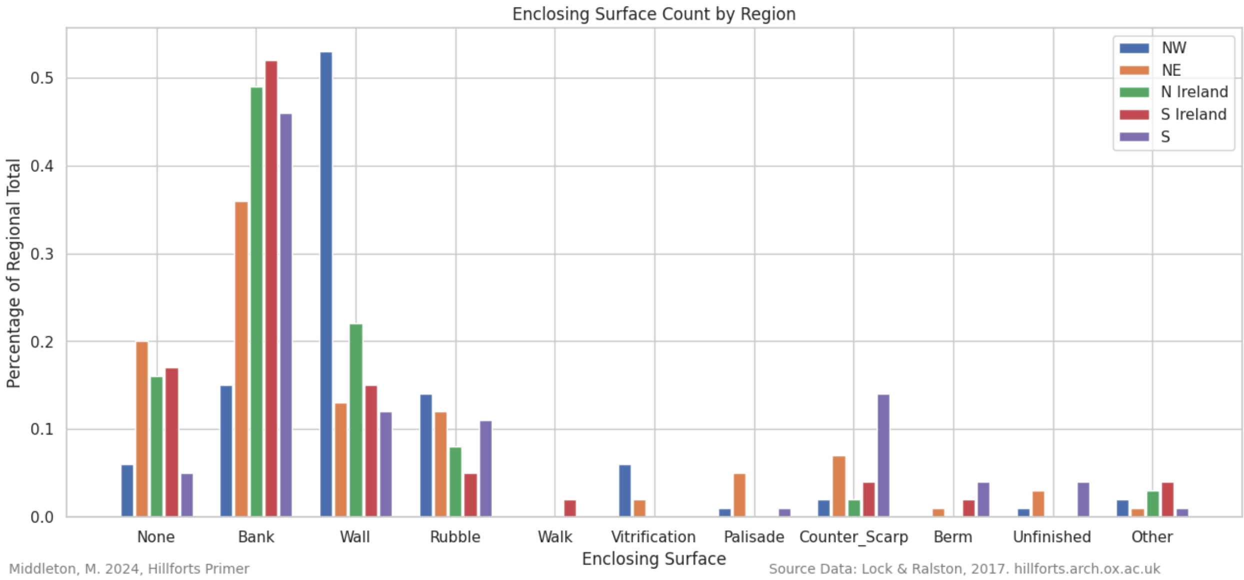
```
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Surface_None',
                 'Enclosing_Surface_Bank',
                 'Enclosing_Surface_Wall',
                 'Enclosing_Surface_Rubble',
                 'Enclosing_Surface_Walk',
                 #'Enclosing_Surface_Timber',
                 'Enclosing_Surface_Vitrification',
                 #'Enclosing_Surface_Burning',
                 'Enclosing_Surface_Palisade',
                 'Enclosing_Surface_Counter_Scarp',
                 'Enclosing_Surface_Berm',
                 'Enclosing_Surface_Unfinished',
                 'Enclosing_Surface_Other'],
                 'Enclosing Surface',
                 'Enclosing_Surface Count by Region', 2, 'Yes')
```



Enclosing Surface by Region (Percentage)

When plotted as a proportion of the data by region, banks dominate in the South, Northeast and across Ireland. In the Northwest walls are dominant. Walls, banks and rubble are the predominant enclosing structural forms.

In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Surface_None',
                 'Enclosing_Surface_Bank',
                 'Enclosing_Surface_Wall',
                 'Enclosing_Surface_Rubble',
                 'Enclosing_Surface_Walk',
                 #'Enclosing_Surface_Timber',
                 'Enclosing_Surface_Vitrification',
                 #'Enclosing_Surface_Burning',
                 'Enclosing_Surface_Palisade',
                 'Enclosing_Surface_Counter_Scarp',
                 'Enclosing_Surface_Berm',
                 'Enclosing_Surface_Unfinished',
                 'Enclosing_Surface_Other'],
                 'Enclosing Surface',
                 'Enclosing_Surface Count by Region', 2, 'Yes', True)
```



Enclosing Excavation Nothing Mapped

194 (4.22%) of hillforts have had an excavation where no enclosing circuit was identified.

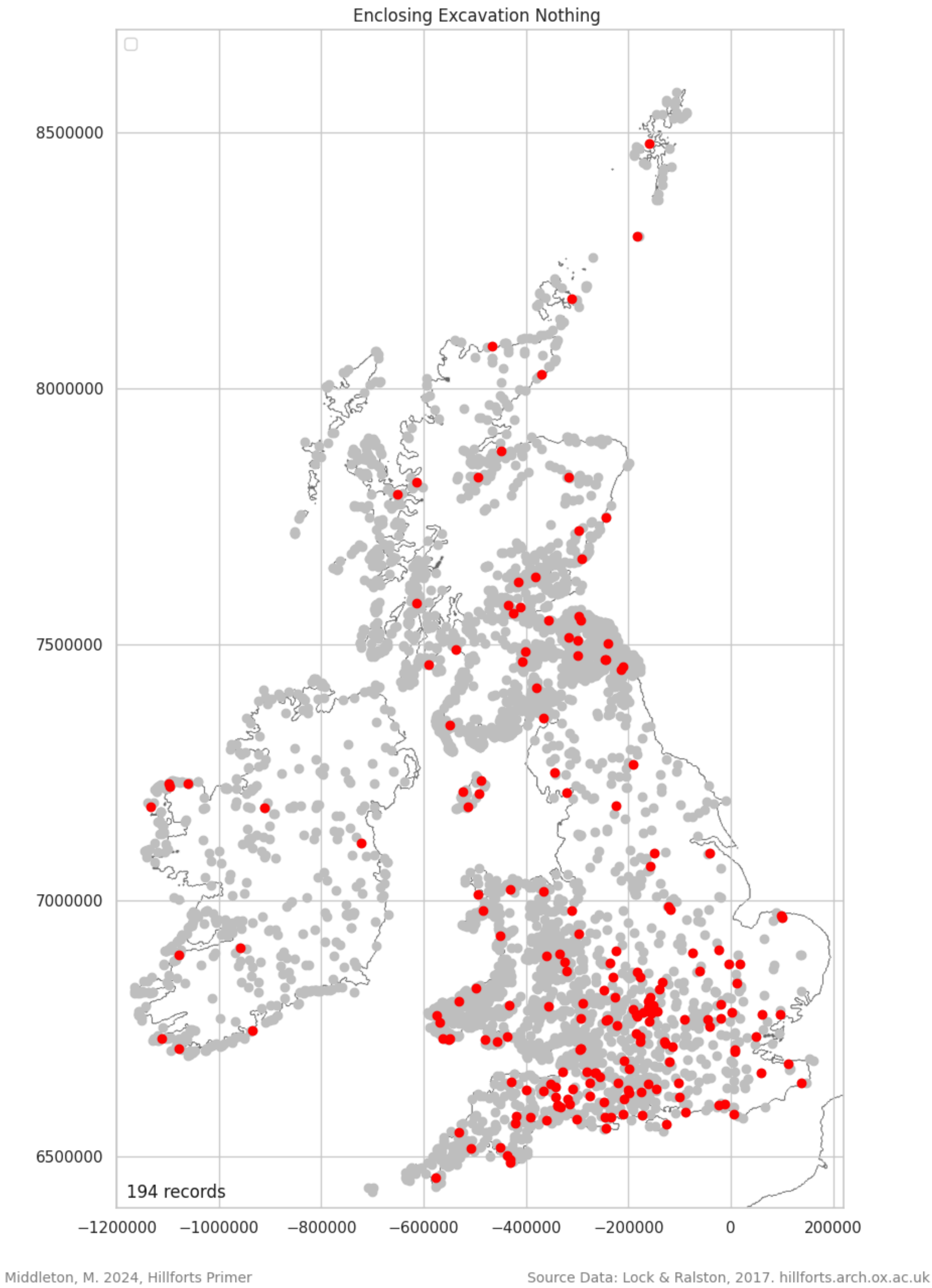
In [ ]:
```
excavation_nothing_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Nothing'].value_counts()
excavation_nothing_counts
```

Out[ ]:
```
No      3953
Yes      194
Name: Enclosing_Excavation_Nothing, dtype: int64
```

In [ ]:
```
print(f'{round(surface_unfinished_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
4.22%
```

In [ ]:
```
excavation_nothing_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Nothing', 'Yes', '', \
               False, False, False, False)
```

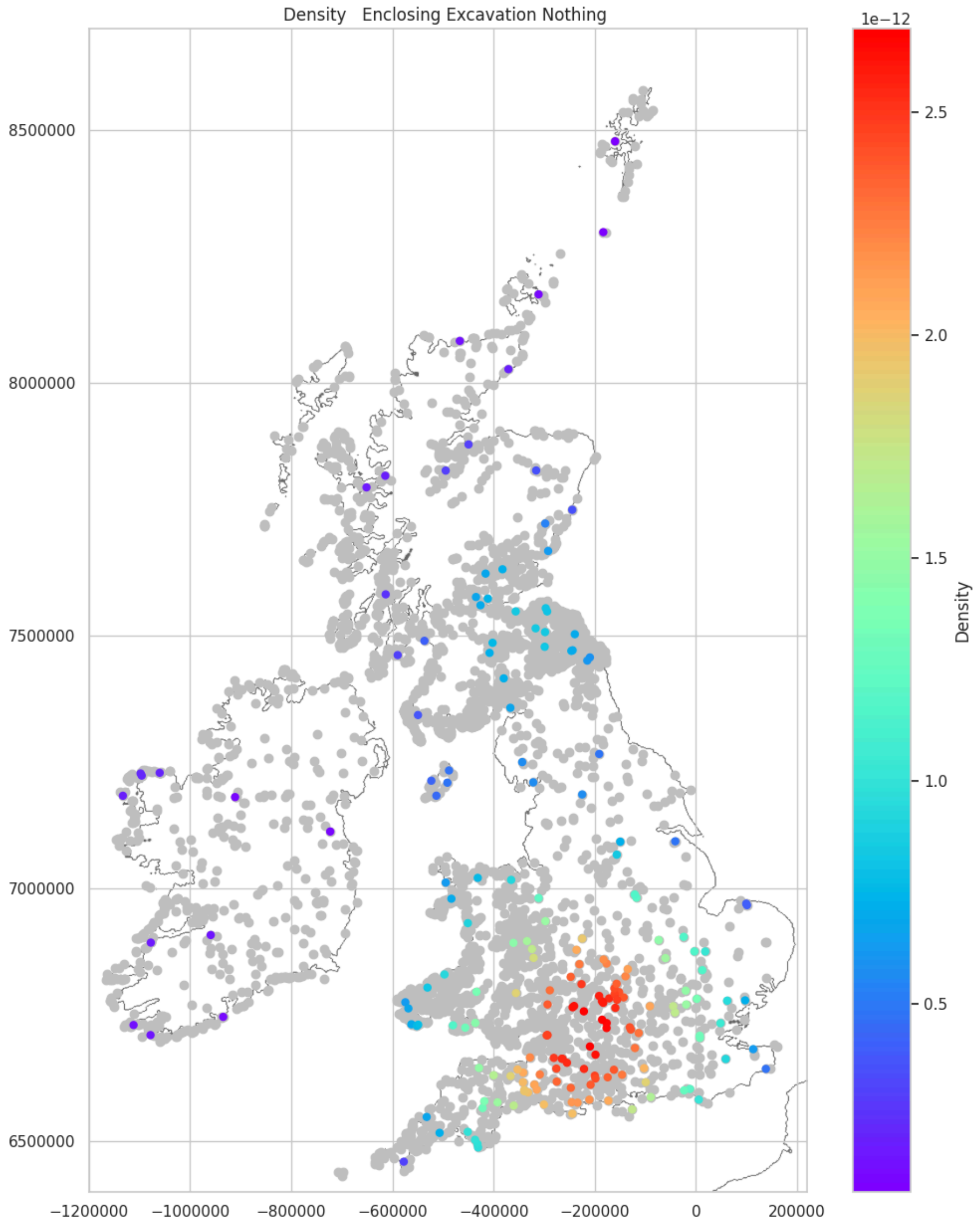



```
4.68%
```

Enclosing Excavation Nothing Density Mapped

This cluster is biased and likely reflects the focus of excavation rather than anything more meaningful.

```
In [ ]: plot_density_over_grey(excavation_nothing_data_yes, \
                               'Enclosing_Excavation_Nothing', '')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Excavation Bank Mapped

351 (8.4%) of hillforts have a bank exposed during excavation.

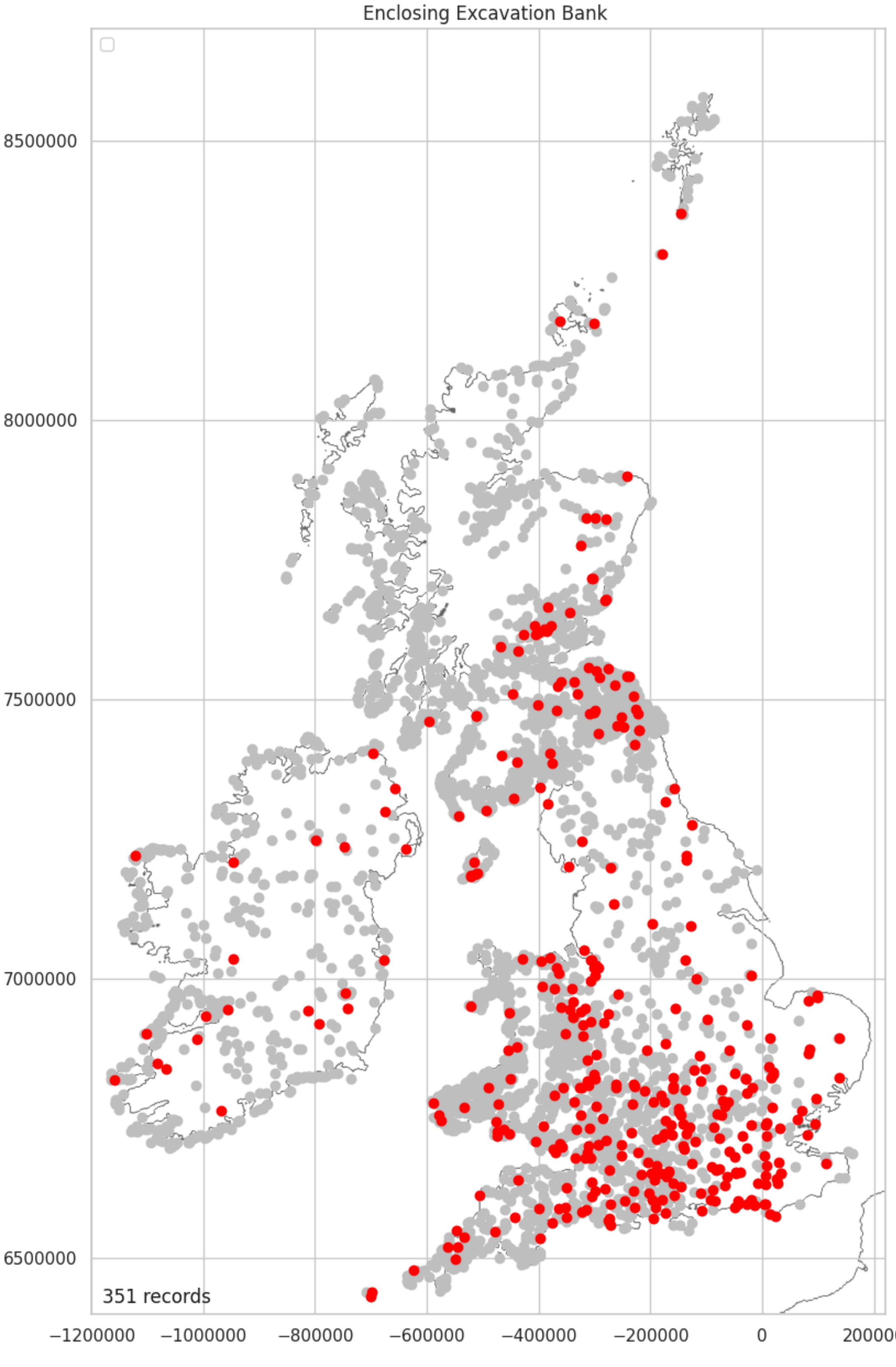
In [ ]:
```
excavation_bank_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Bank'].value_counts()
excavation_bank_counts
```

Out[ ]:
```
No      3796
Yes      351
Name: Enclosing_Excavation_Bank, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_bank_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
8.46%
```

In [ ]:
```
excavation_bank_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Excavation_Bank', \
               'Yes', '', False, False, False)
```



Middleton, M. 2024, Hillforts Primer

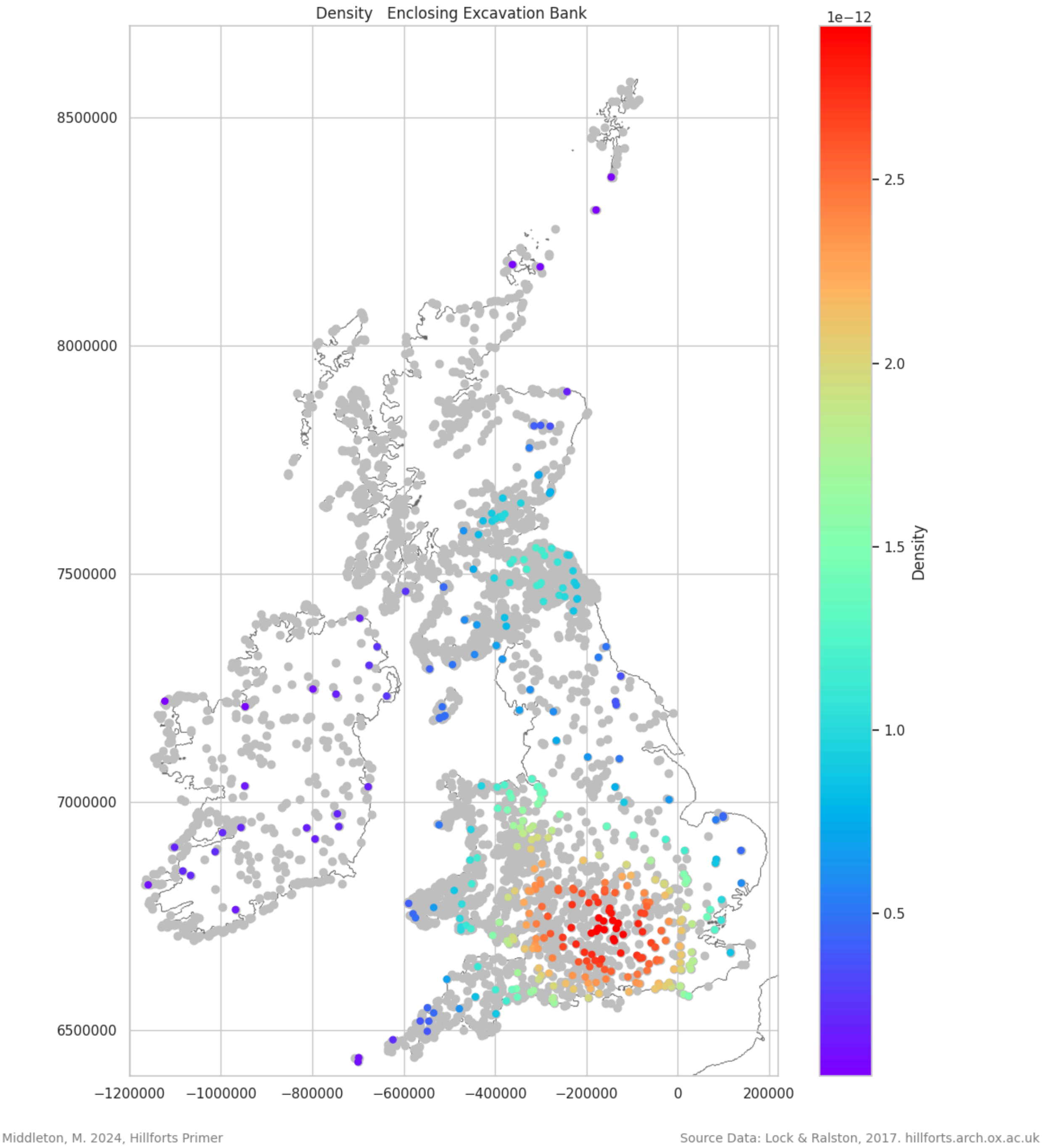
Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
8.46%
```

Enclosing Excavation Bank Density Mapped

The main cluster for this is in the South but, compared with the cluster seen in, Part 1: Southern Data Density Mapped (Transformed), the focus is considerably further east. Compared to the clusters seen in Enclosing Surface Bank Density Mapped, where the main focus of banked enclosing circuits was in Wales and the Northeast, this distribution is misleading. The distribution is likely to reflect a excavation bias rather than being meaningful.

In [ ]:
```
plot_density_over_grey(excavation_bank_data_yes, 'Enclosing_Excavation_Bank')
```



Enclosing Excavation Wall Mapped

304 (7.33%) of hillforts excavated have revealed an enclosing wall.

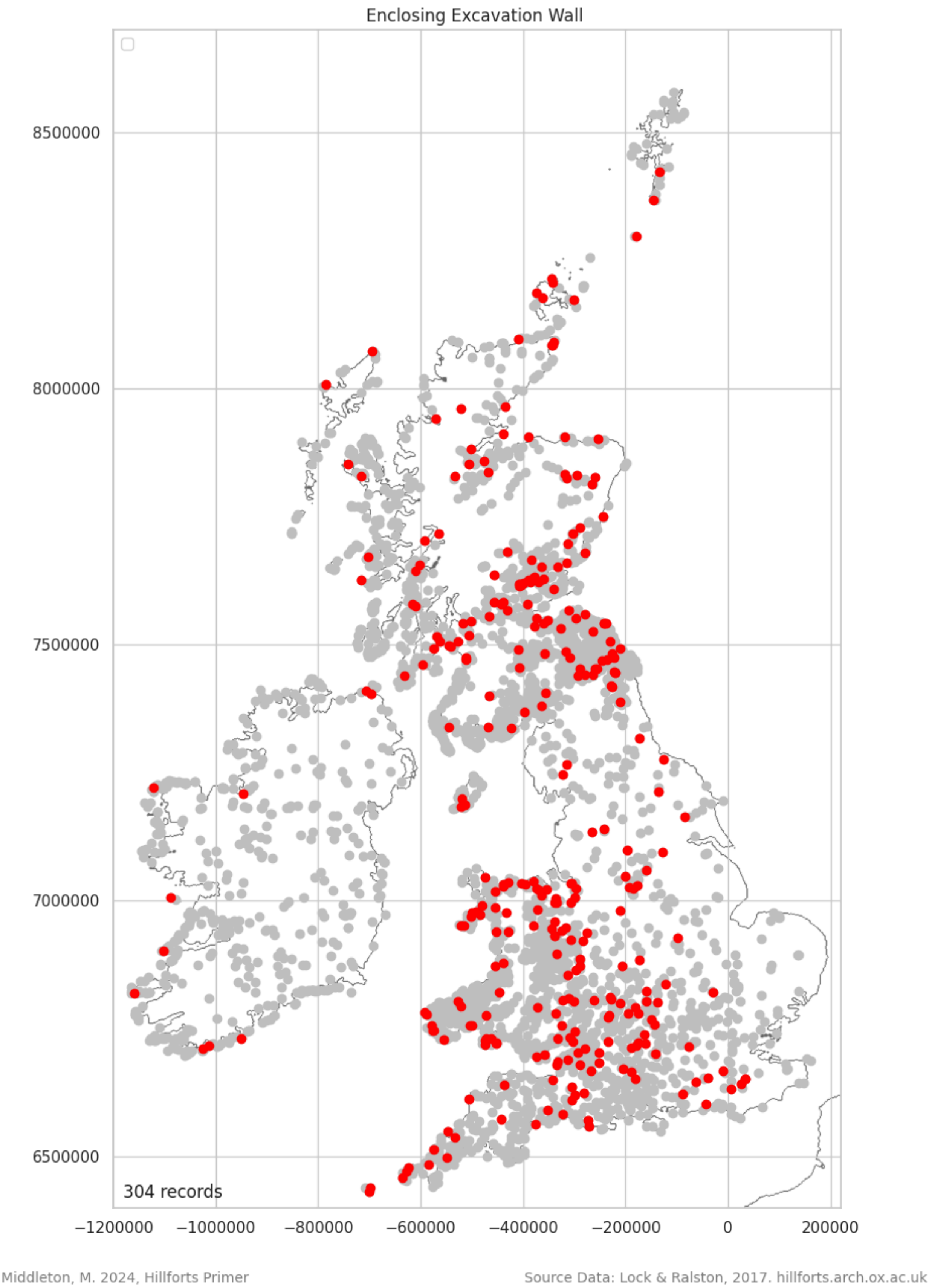
In [ ]:
```
excavation_wall_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Wall'].value_counts()
excavation_wall_counts
```

Out[ ]:
```
No      3843
Yes      304
Name: Enclosing_Excavation_Wall, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_wall_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
7.33%
```

In [ ]:
```
excavation_wall_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Excavation_Wall', \
               'Yes', '')
```



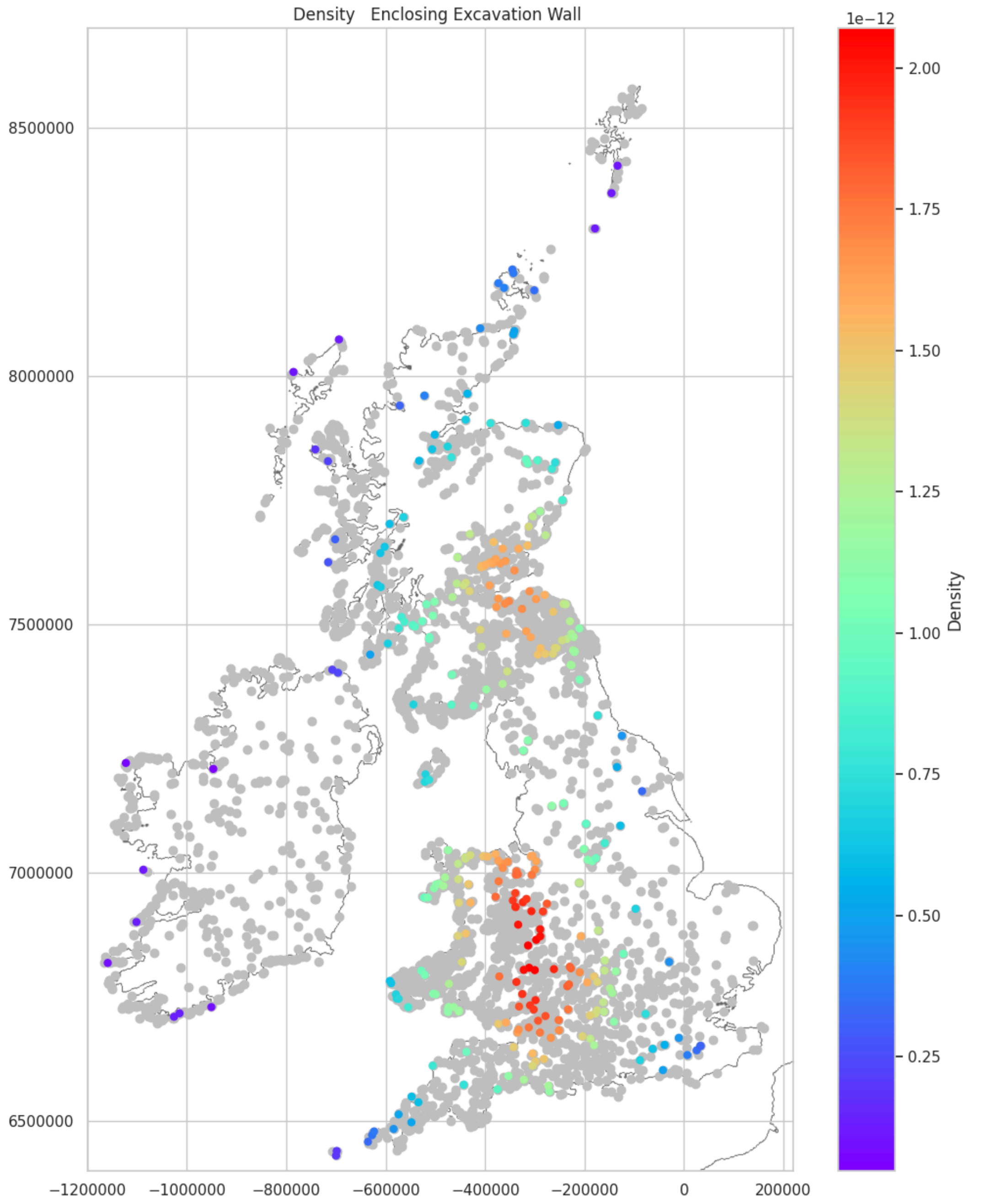

```
7.33%
```

Enclosing Excavation Wall Density

The main clusters of hillforts with walls seen in Enclosing Surface Wall Density Mapped was focussed in the Northwest and in west Wales. The main cluster here is to the east of the Cambrian Mountains and a smaller, secondary cluster can be seen in the Northeast. As with previous classes in the Enclosing Excavation section, this distribution suffers from survey bias.

In [ ]:

```
plot_density_over_grey(excavation_wall_data_yes, \
                       'Enclosing_Excavation_Wall','')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Enclosing Excavation Murus Gallicus Mapped

Just two hillforts have revealed Murus Gallicus recorded in excavation.

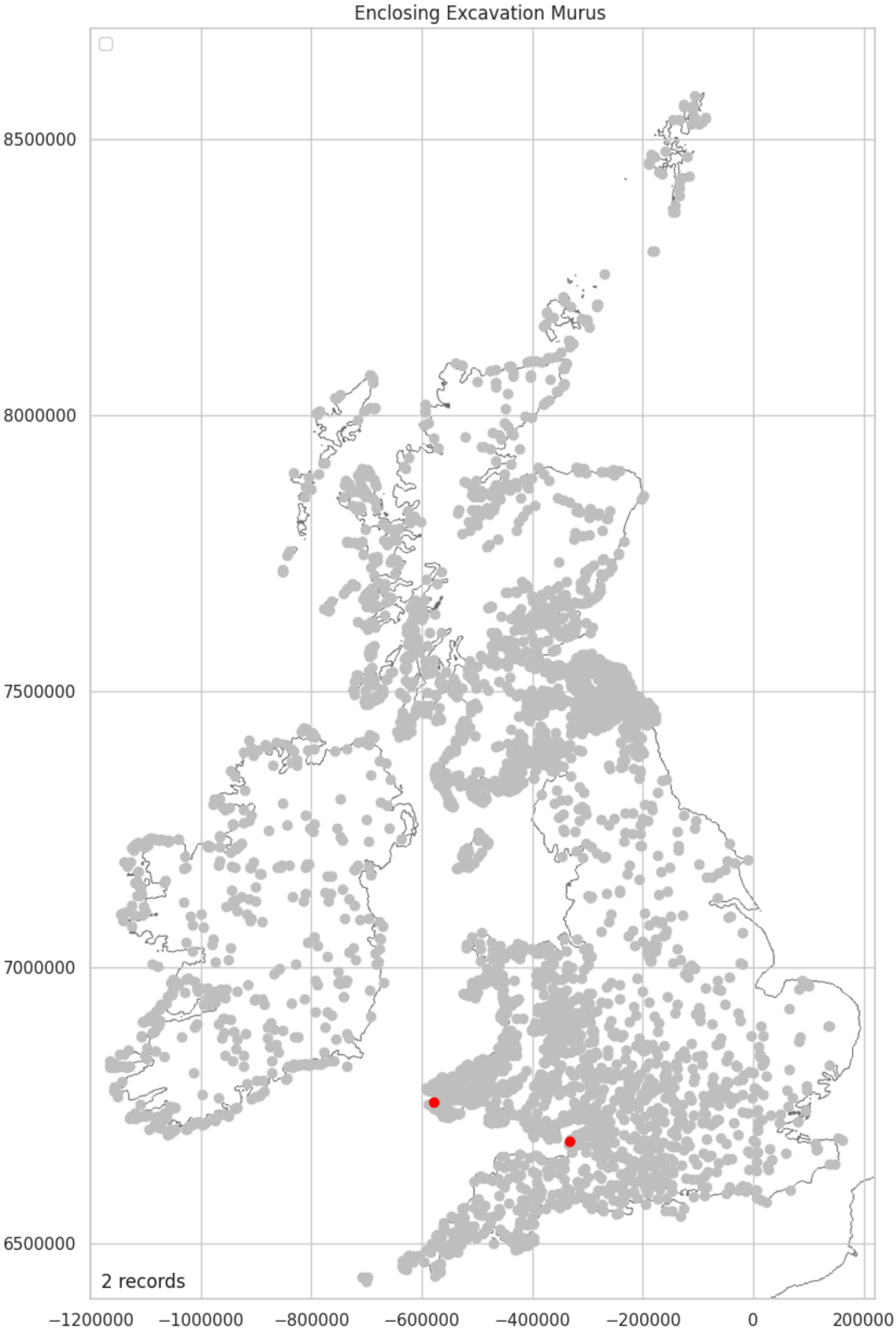
In [ ]:
```
excavation_murus_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Murus'].value_counts()
excavation_murus_counts
```

Out[ ]:
```
No      4145
Yes        2
Name: Enclosing_Excavation_Murus, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_murus_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.05%
```

In [ ]:
```
excavation_murus_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Murus', 'Yes', '')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

```
0.05%
```

Enclosing Excavation Timber Framed Mapped

57 (1.37%) of hillforts have had a Timber Frame revealed during excavation.

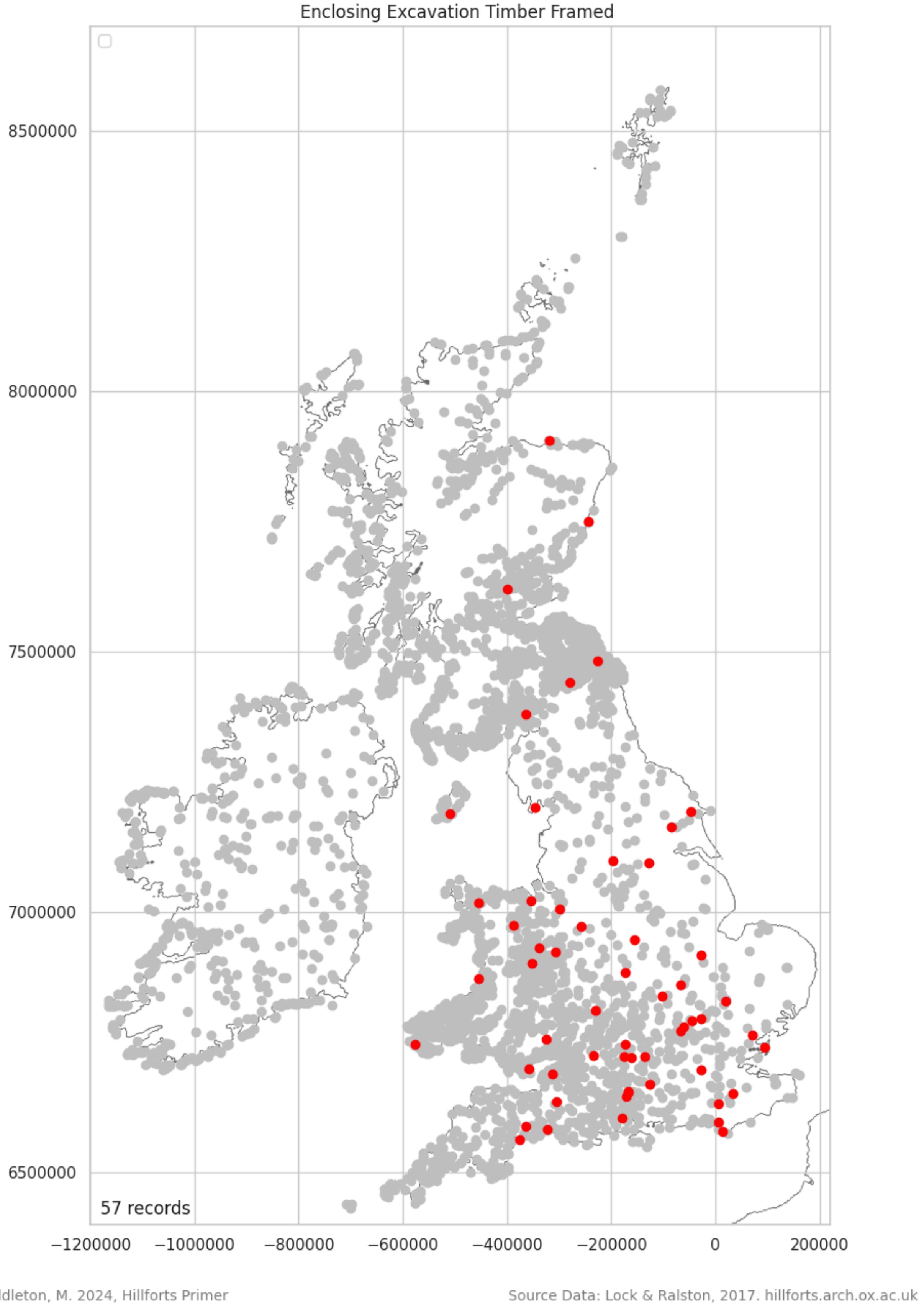
In [ ]:
```
excavation_tf_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Timber_Framed'].value_counts()
excavation_tf_counts
```

Out[ ]:
```
No      4090
Yes       57
Name: Enclosing_Excavation_Timber_Framed, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_tf_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.37%
```

In [ ]:
```
excavation_tf_data_yes = plot_over_grey(location_enclosing_encodeable_data, \
                                        'Enclosing_Excavation_Timber_Framed', \
                                        'Yes', '')
```




```
1.37%
```

Enclosing Excavation Timber Laced Mapped

46 (1.11%) of hillforts have had a Timber Lacing revealed during excavation.

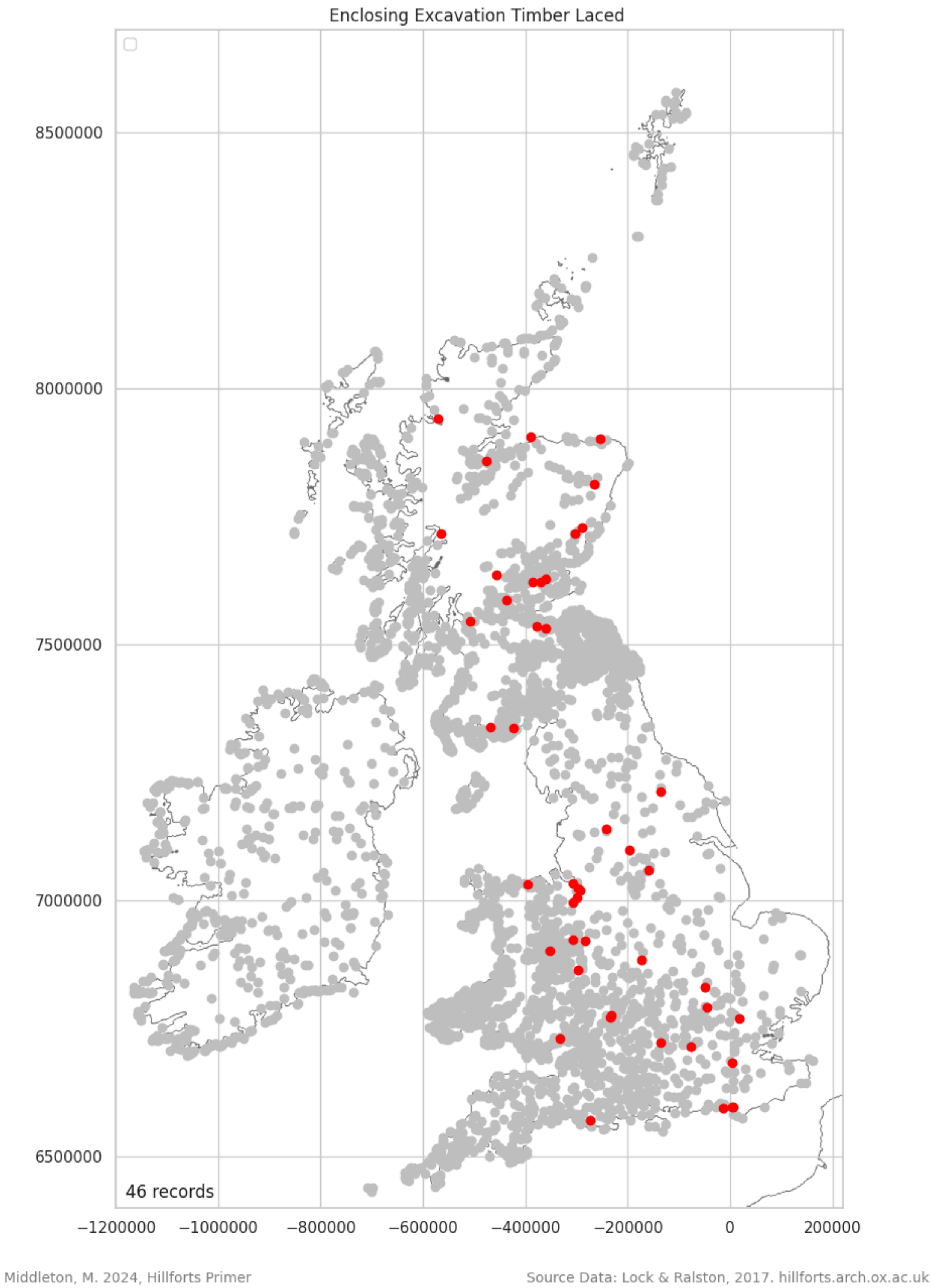
In [ ]:
```
excavation_tl_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Timber_Laced'].value_counts()
excavation_tl_counts
```

Out[ ]:
```
No     4101
Yes      46
Name: Enclosing_Excavation_Timber_Laced, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_tl_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.11%
```

In [ ]:
```
excavation_tl_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Timber_Laced', 'Yes', '')
```




```
1.11%
```

Enclosing Excavation Vitrification Mapped

48 (1.16%) of hillforts have had Vitrification identified during excavation. See: Enclosing Surface Vitrification Mapped

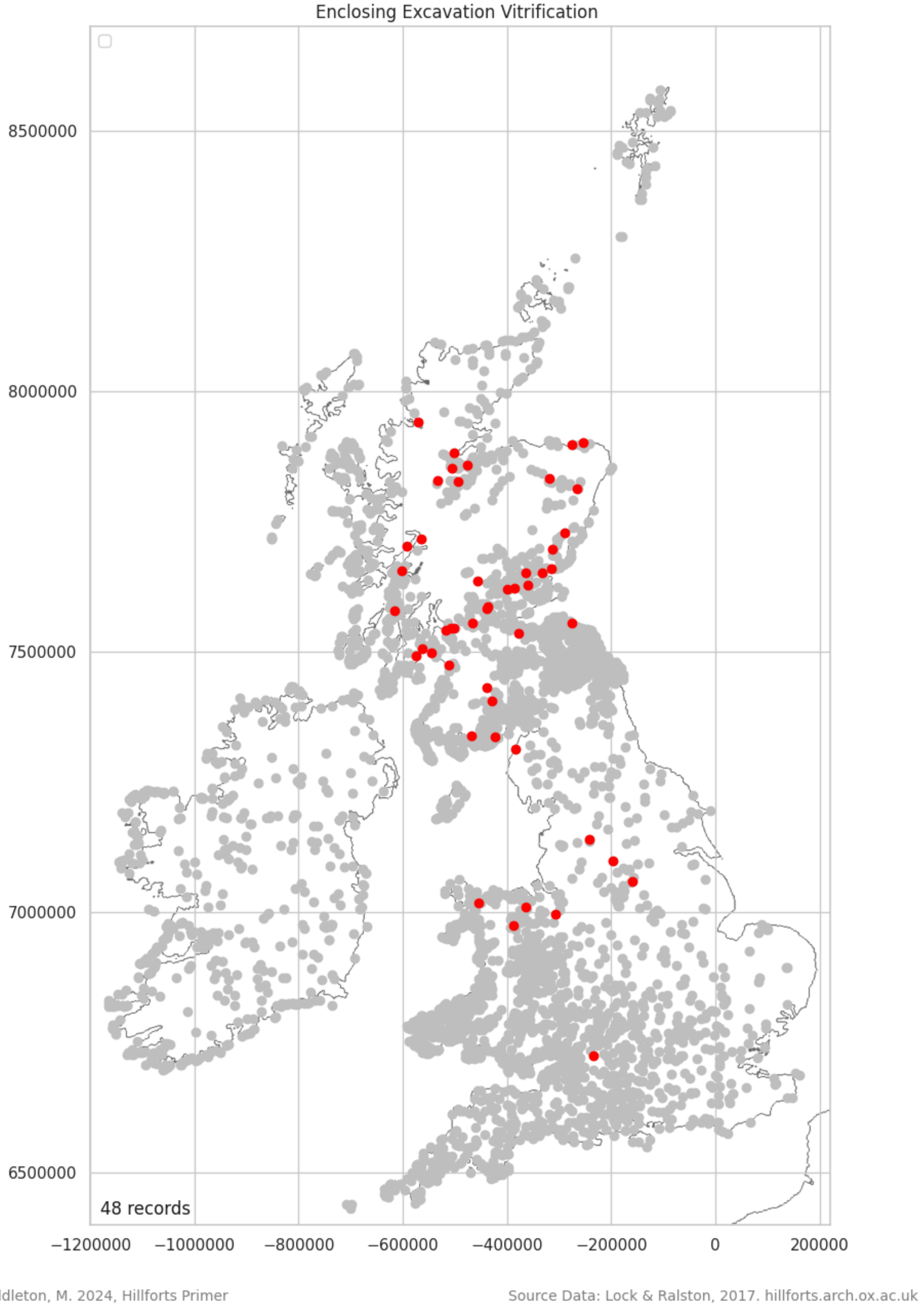
In [ ]:
```
excavation_vitrification_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Vitrification'].value_counts()
excavation_vitrification_counts
```

```
Out[ ]: No      4099
        Yes       48
        Name: Enclosing_Excavation_Vitrification, dtype: int64
```

```
In [ ]: print(f'{round(excavation_vitrification_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.16%
```

```
In [ ]: excavation_vitrification_data_yes = \
        plot_over_grey(location_enclosing_encodeable_data, \
                       'Enclosing_Excavation_Vitrification', 'Yes', '')
```




```
1.16%
```

Enclosing Excavation Burning Mapped

46 (1.11%) of hillforts have had burning, associated with the enclosing structure, identified during excavation.

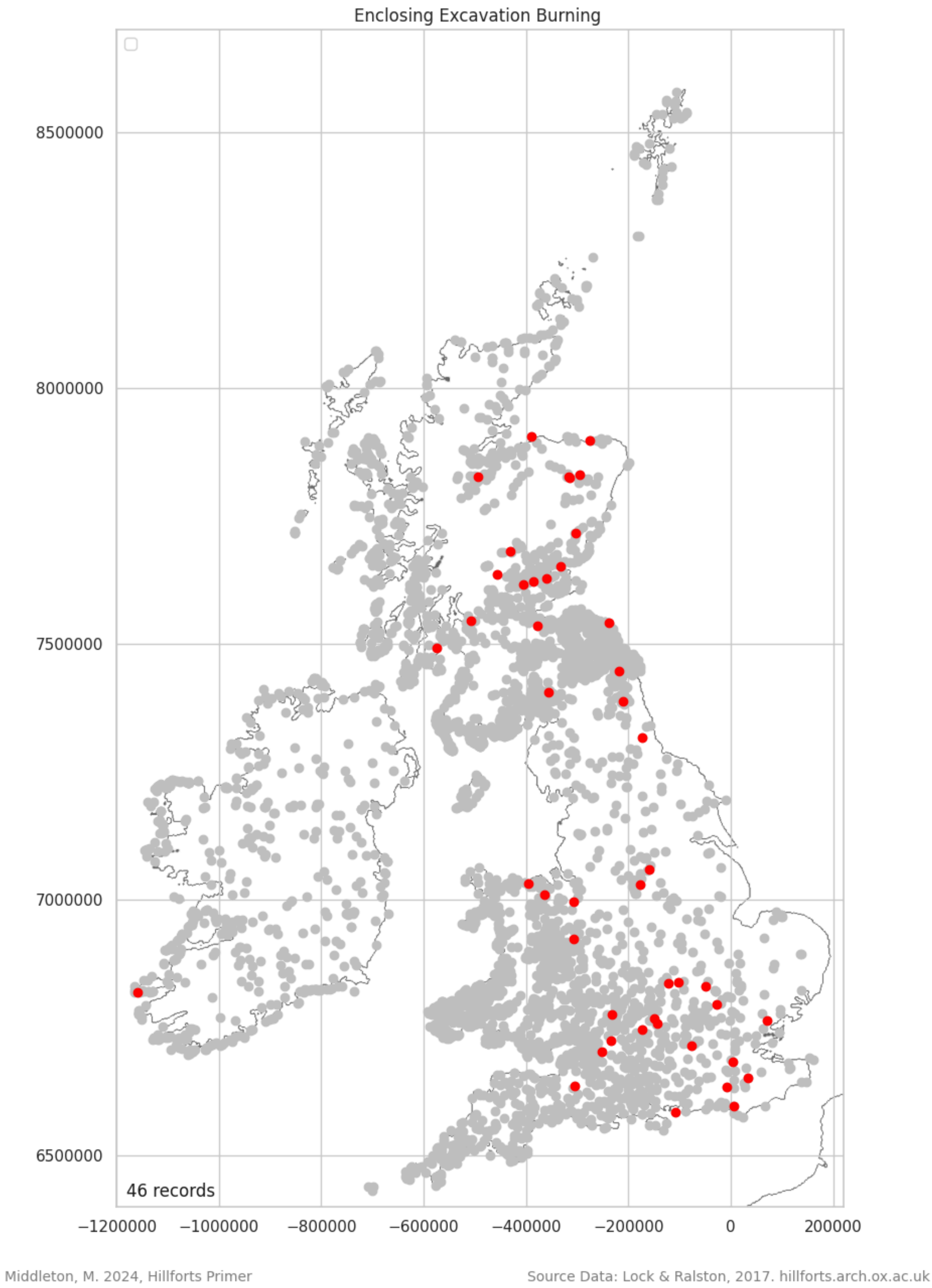
In [ ]:
```
excavation_burning_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Burning'].value_counts()
excavation_burning_counts
```

Out[ ]:
```
No     4101
Yes      46
Name: Enclosing_Excavation_Burning, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_burning_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.11%
```

In [ ]:
```
excavation_burning_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Burning', 'Yes', '')
```




```
1.11%
```

Enclosing Excavation Palisade Mapped

135 (3.26%) of hillforts have had a palisade revealed during excavation.

In [ ]:

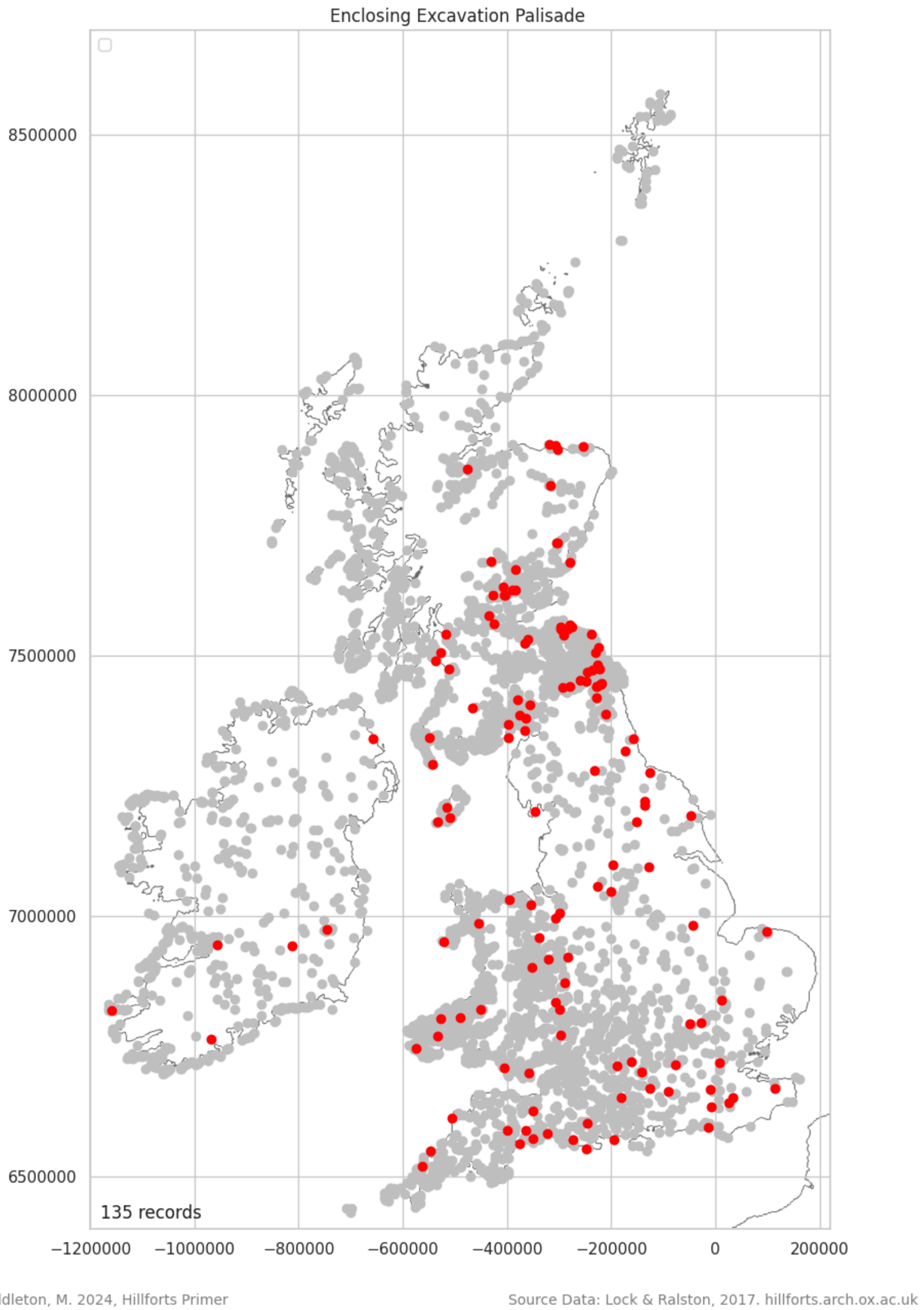
```
excavation_palisade_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Palisade'].value_counts()
excavation_palisade_counts
```

Out[ ]:
```
No     4012
Yes     135
Name: Enclosing_Excavation_Palisade, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_palisade_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
3.26%
```

In [ ]:
```
excavation_palisade_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Palisade', 'Yes', '')
```

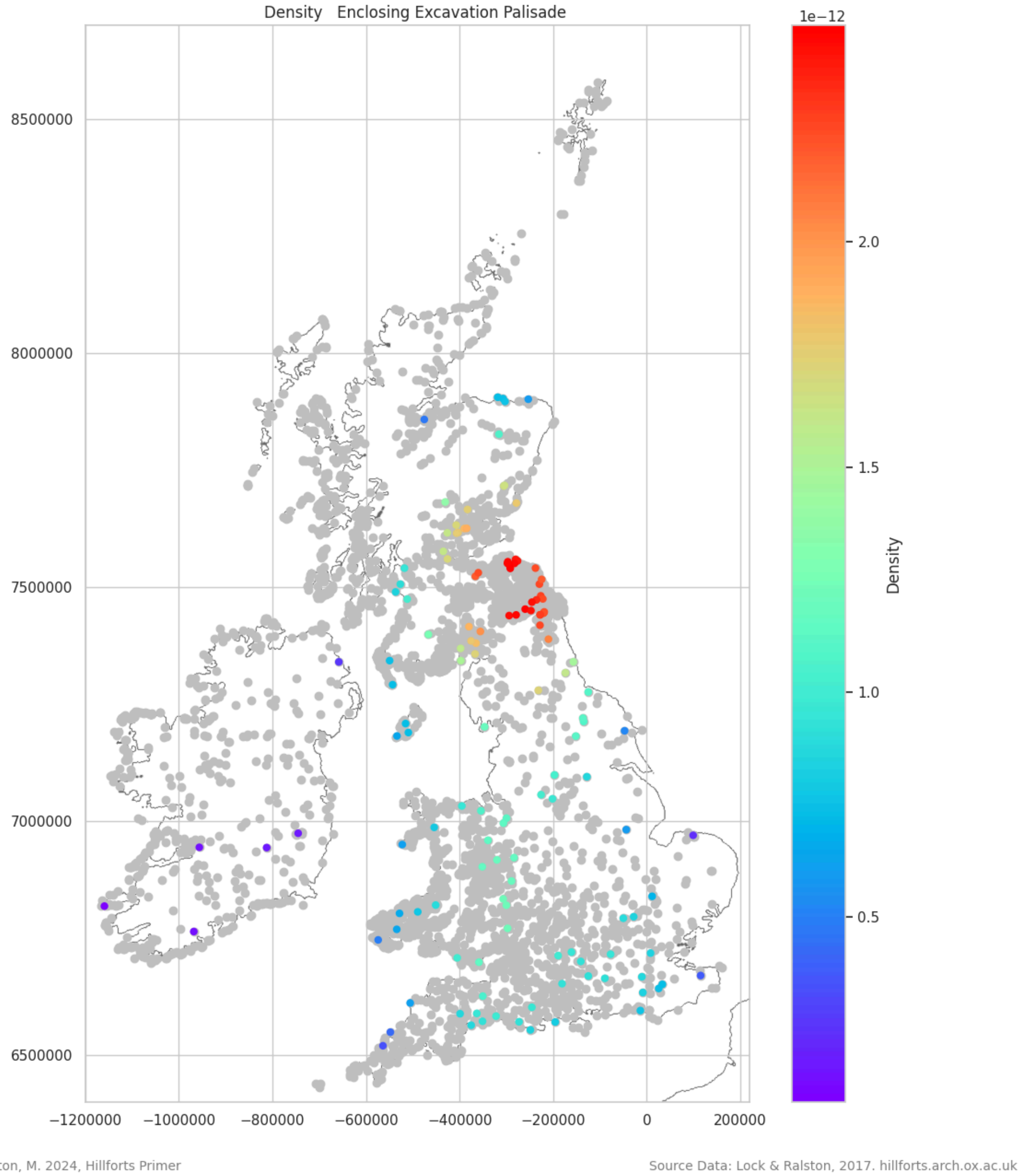



```
3.26%
```

Enclosing Excavation Palisade Density Mapped

The main cluster for excavated palisades is in the Northeast. This distribution mirrors that seen in Enclosing Surface Palisade Density Mapped.

In [ ]:
```
plot_density_over_grey(excavation_palisade_data_yes, \
                        'Enclosing_Excavation_Palisade')
```



Enclosing Excavation Counter Scarp Mapped

64 (1.54%) of hillforts have had a counterscarp exposed during excavation. See: Enclosing Surface Counter Scarp Density Mapped.

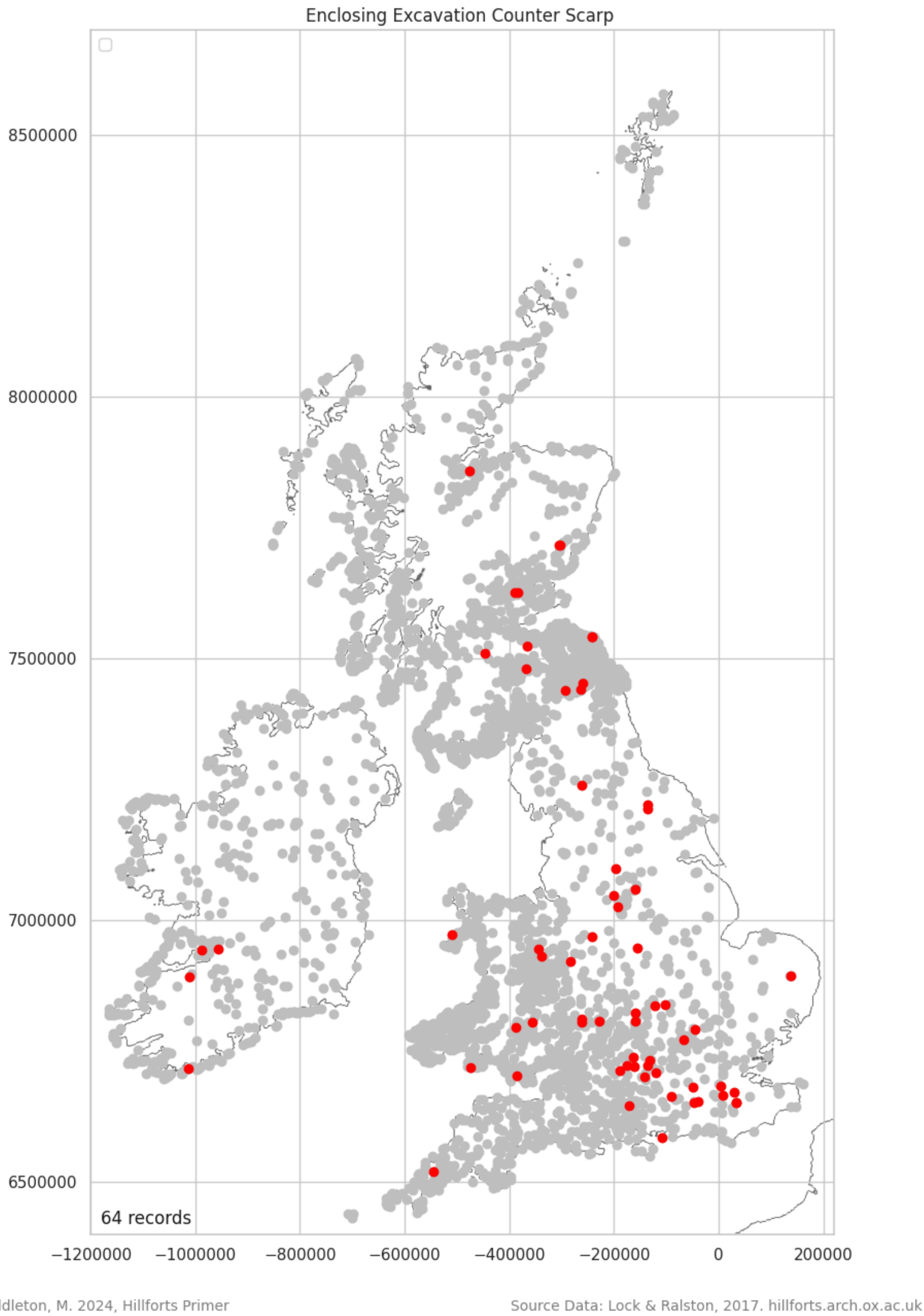
In [ ]:
```
excavation_cs_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Counter_Scarp'].value_counts()
excavation_cs_counts
```

Out[ ]:
```
No     4083
Yes      64
Name: Enclosing_Excavation_Counter_Scarp, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_cs_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.54%
```

In [ ]:
```
excavation_cs_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Counter_Scarp', 'Yes', '')
```




```
1.54%
```

Enclosing Excavation Berm Mapped

24 (0.58%) of hillforts have had a berm revealed during excavation.

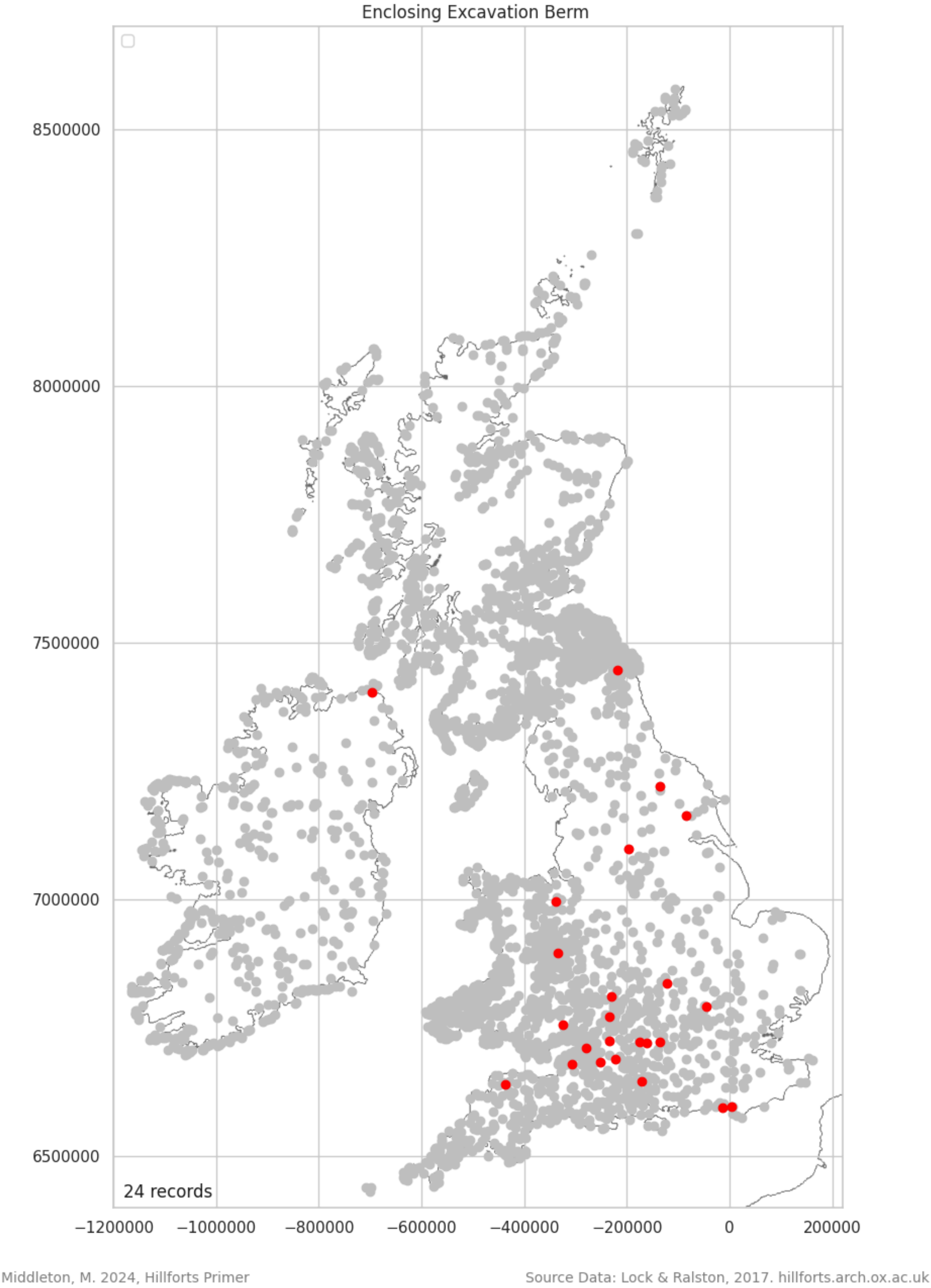
In [ ]:
```
excavation_berm_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Berm'].value_counts()
excavation_berm_counts
```

Out[ ]:
```
No     4123
Yes      24
Name: Enclosing_Excavation_Berm, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_berm_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.58%
```

In [ ]:
```
excavation_berm_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_Berm', 'Yes', '')
```




```
0.58%
```

Enclosing Excavation Unfinished Mapped

18 (0.43%) of hillforts have unfinished enclosing works revealed during excavation.

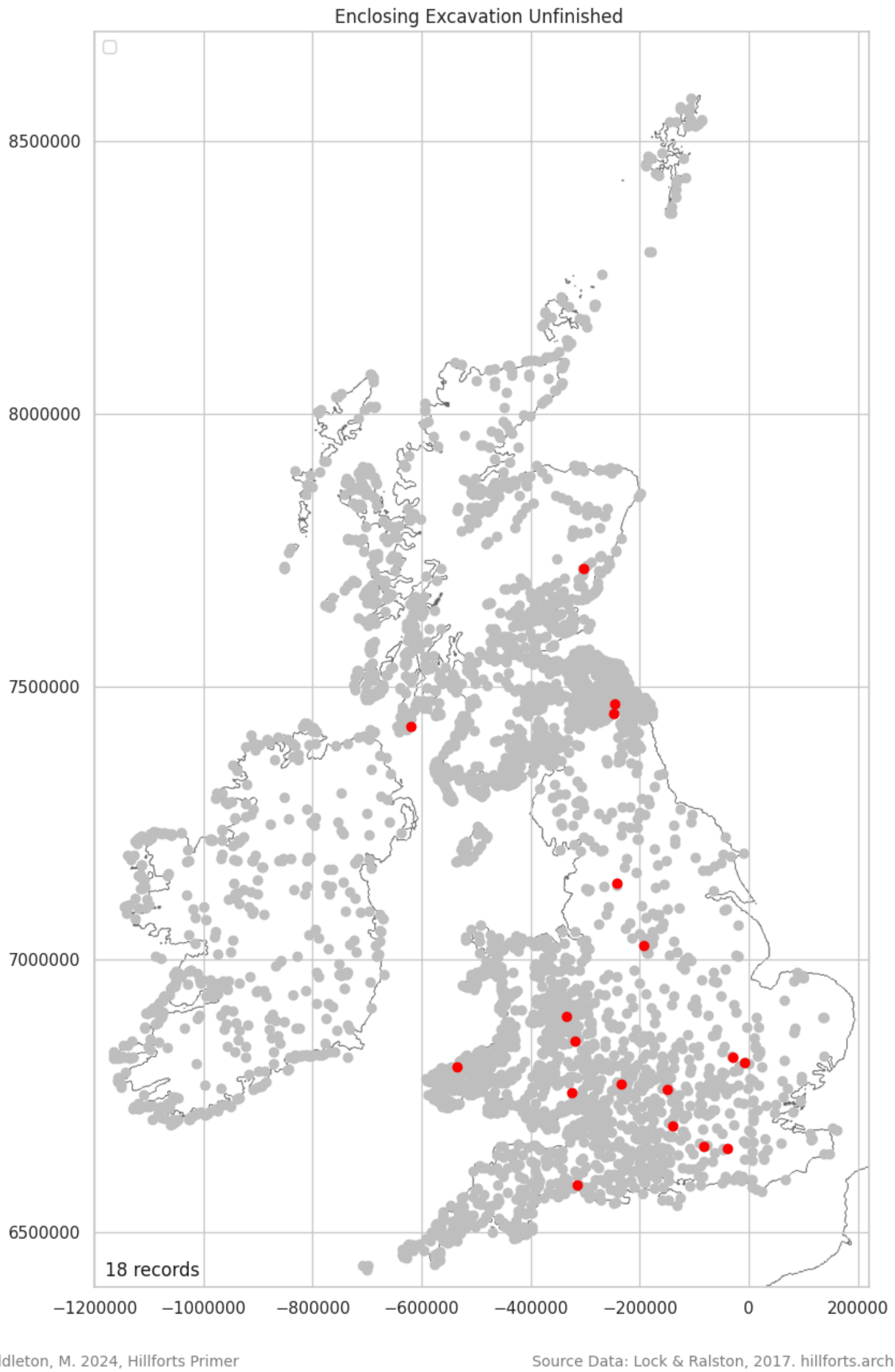
In [ ]:
```
excavation_unfinished_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Unfinished'].value_counts()
excavation_unfinished_counts
```

```
Out[ ]: No      4129
        Yes       18
        Name: Enclosing_Excavation_Unfinished, dtype: int64
```

```
In [ ]: print(f'{round(excavation_unfinished_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
0.43%
```

```
In [ ]: excavation_unfinished_data_yes = \
        plot_over_grey(location_enclosing_encodeable_data, \
                       'Enclosing_Excavation_Unfinished', 'Yes', '')
```




```
0.43%
```

Enclosing Excavation Other Mapped

230 (5.55%) of hillforts have an enclosing circuit class, other than those listed above, recorded during excavation.

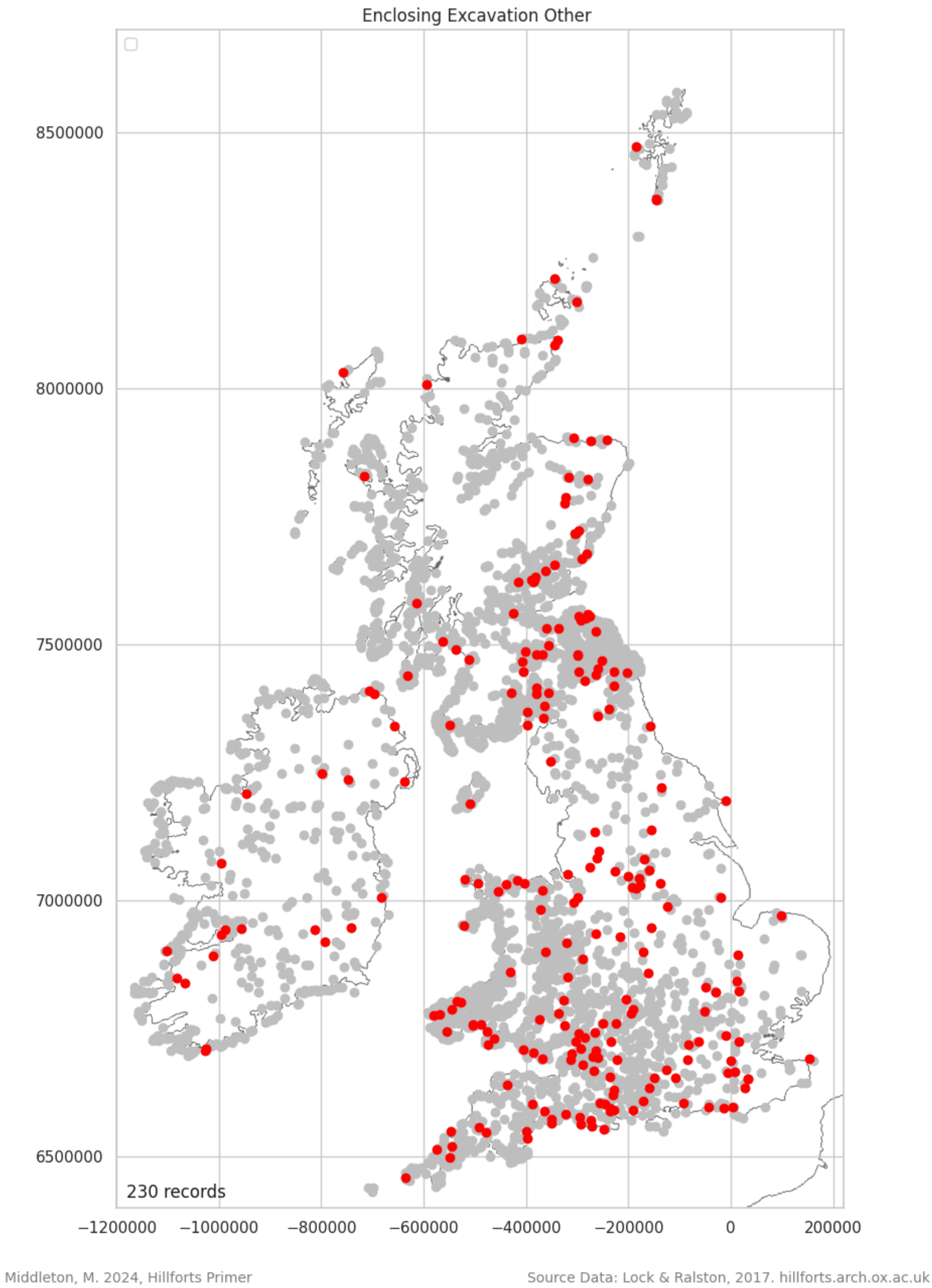
In [ ]:
```
excavation_other_counts = \
enclosing_encodeable_data['Enclosing_Excavation_Other'].value_counts()
excavation_other_counts
```

Out[ ]:
```
No     3917
Yes     230
Name: Enclosing_Excavation_Other, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_other_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
5.55%
```

In [ ]:
```
excavation_other_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Excavation_Other', \
               'Yes', '')
```




```
5.55%
```

Enclosing Excavation No Known Mapped

3329 (80.27%) of hillforts have had no known excavation on their enclosing circuit.

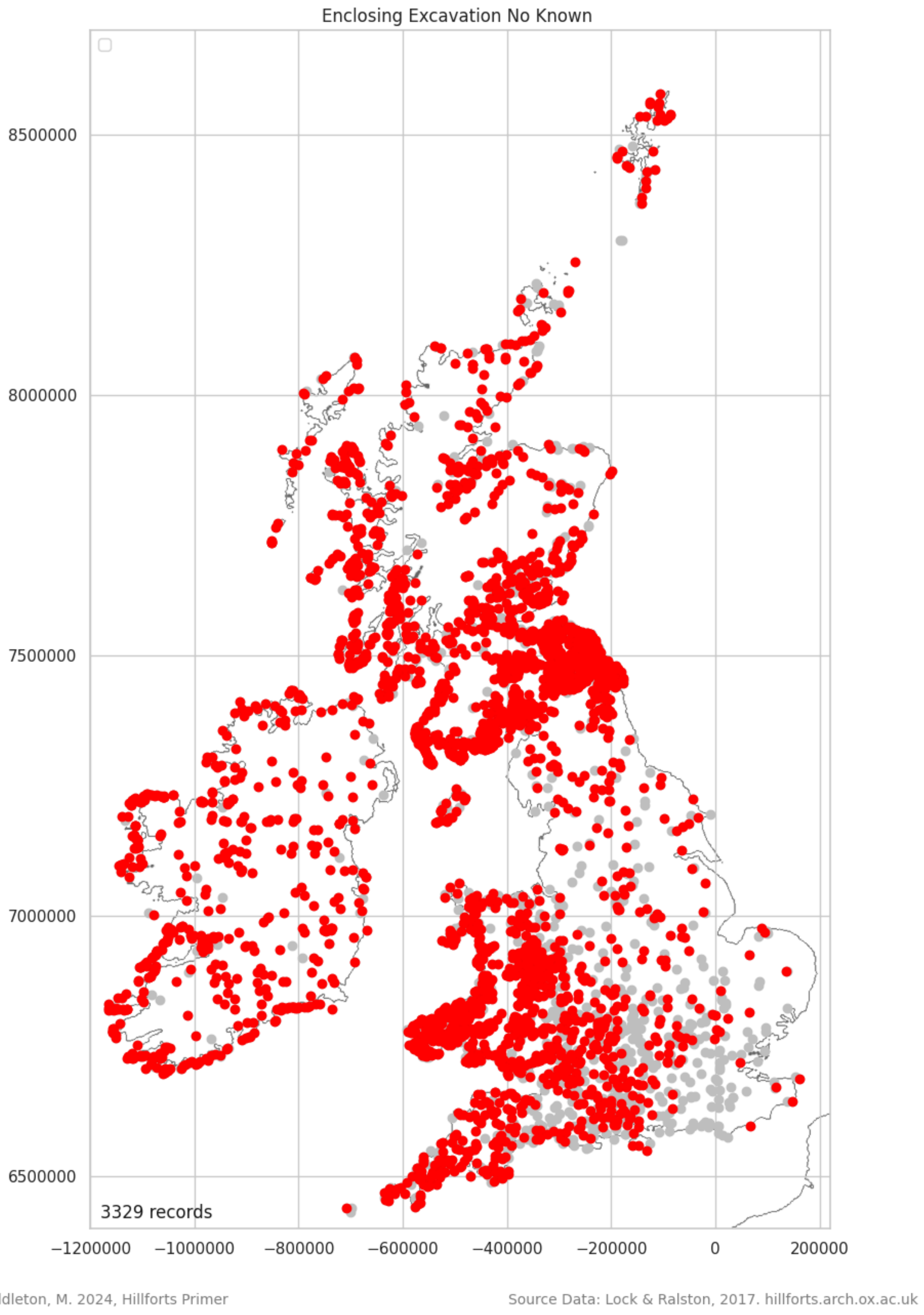
In [ ]:
```
excavation_no_known_counts = \
enclosing_encodeable_data['Enclosing_Excavation_No_Known'].value_counts()
excavation_no_known_counts
```

Out[ ]:
```
Yes    3329
No      818
Name: Enclosing_Excavation_No_Known, dtype: int64
```

In [ ]:
```
print(f'{round(excavation_no_known_counts[0]/len(enclosing_encodeable_data)*100,2)}%')
```

```
80.27%
```

In [ ]:
```
excavation_no_known_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Excavation_No_Known', 'Yes', '')
```




```
80.27%
```

Enclosing Gang Working Mapped

44 (1.06%) of hillforts have signs of gang working recorded.

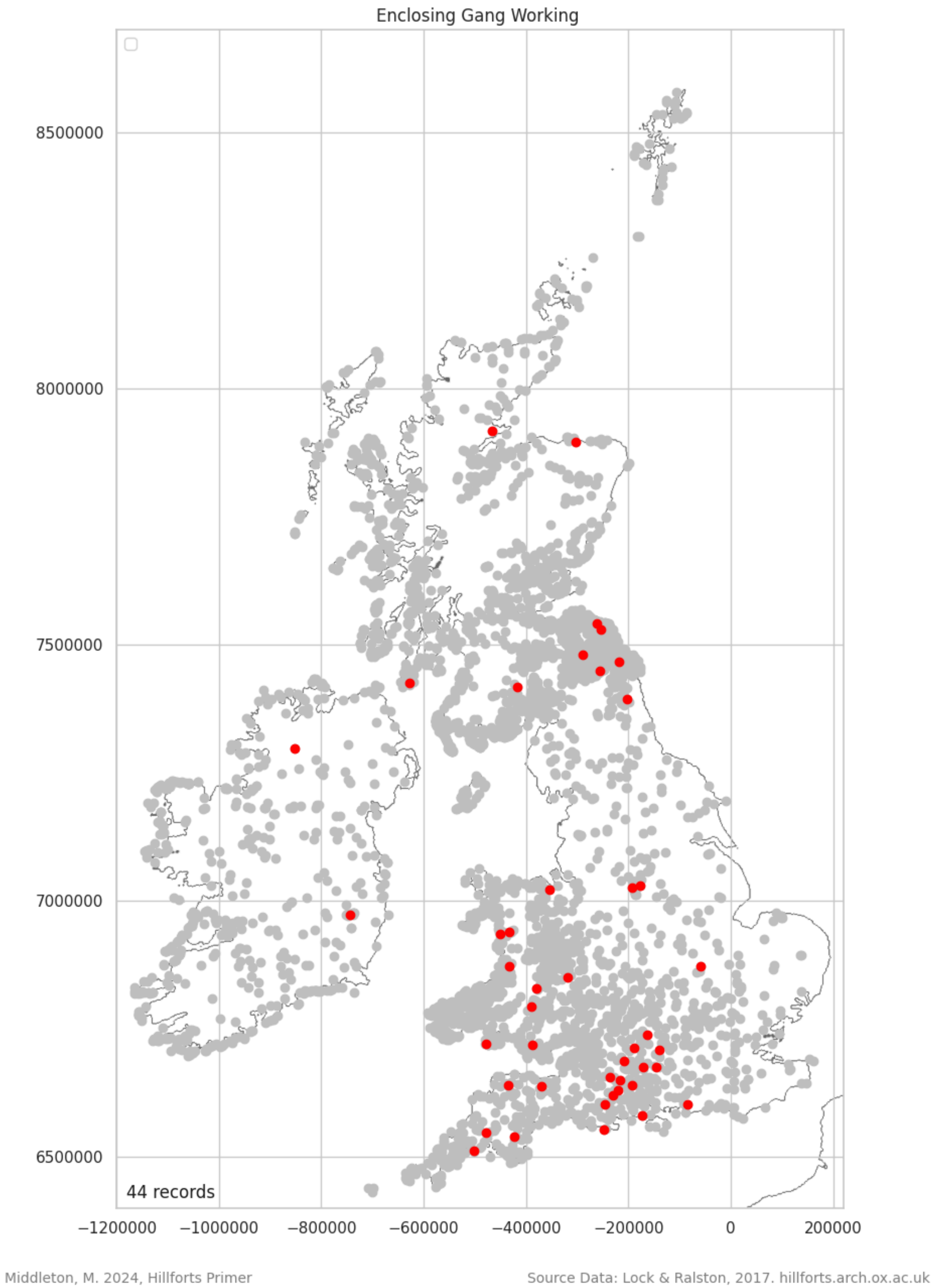
In [ ]:
```
enclosing_gang_counts = \
enclosing_encodeable_data['Enclosing_Gang_Working'].value_counts()
enclosing_gang_counts
```

Out[ ]:
```
No     4103
Yes      44
Name: Enclosing_Gang_Working, dtype: int64
```

In [ ]:
```
print(f'{round(enclosing_gang_counts[1]/len(enclosing_encodeable_data)*100,2)}%')
```

```
1.06%
```

In [ ]:
```
enclosing_gang_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, 'Enclosing_Gang_Working', \
               'Yes', '')
```




```
1.06%
```

Enclosing Ditches Mapped

2864 (69.06%) of hillforts are recorded as having ditches. This is nine less than the 2873 recorded in Ditches Plotted. It is assumed that these nine do not form part of the enclosing circuit. With 91.89% of ditches, recorded in the ditches section above, also found here in the enclosing section, it can be said that ditches are predominantly an enclosing feature.

In [ ]:

```
enclosing_ditches_counts = \
enclosing_encodeable_data['Enclosing_Ditches'].value_counts()
enclosing_ditches_counts
```

Out[ ]:
```
Yes    2864
No     1283
Name: Enclosing_Ditches, dtype: int64
```

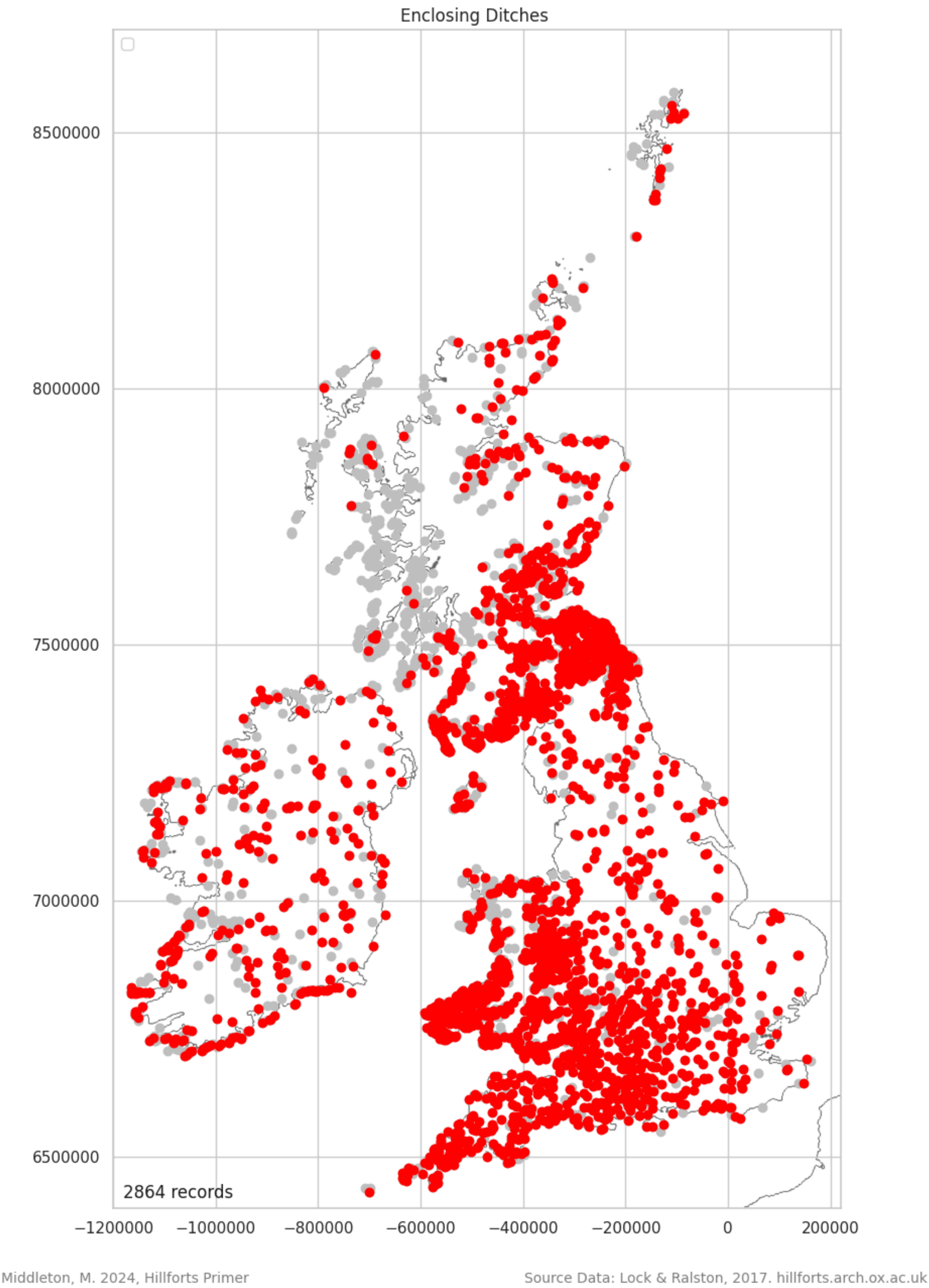
In [ ]:
```
print(f'{round(enclosing_ditches_counts[0]/len(enclosing_encodeable_data)*100,2)}%')
```

```
69.06%
```

In [ ]:
```
enclosing_ditches_number_count = \
len(ditches_location_enc_data[ditches_location_enc_data\
 ['Enclosing_Ditches_Number']>0])
enclosing_ditches_number_count
```

Out[ ]:
```
2873
```

In [ ]:
```
enclosing_ditches_data_yes = \
plot_over_grey(location_enclosing_encodeable_data, \
               'Enclosing_Ditches', 'Yes', '')
```

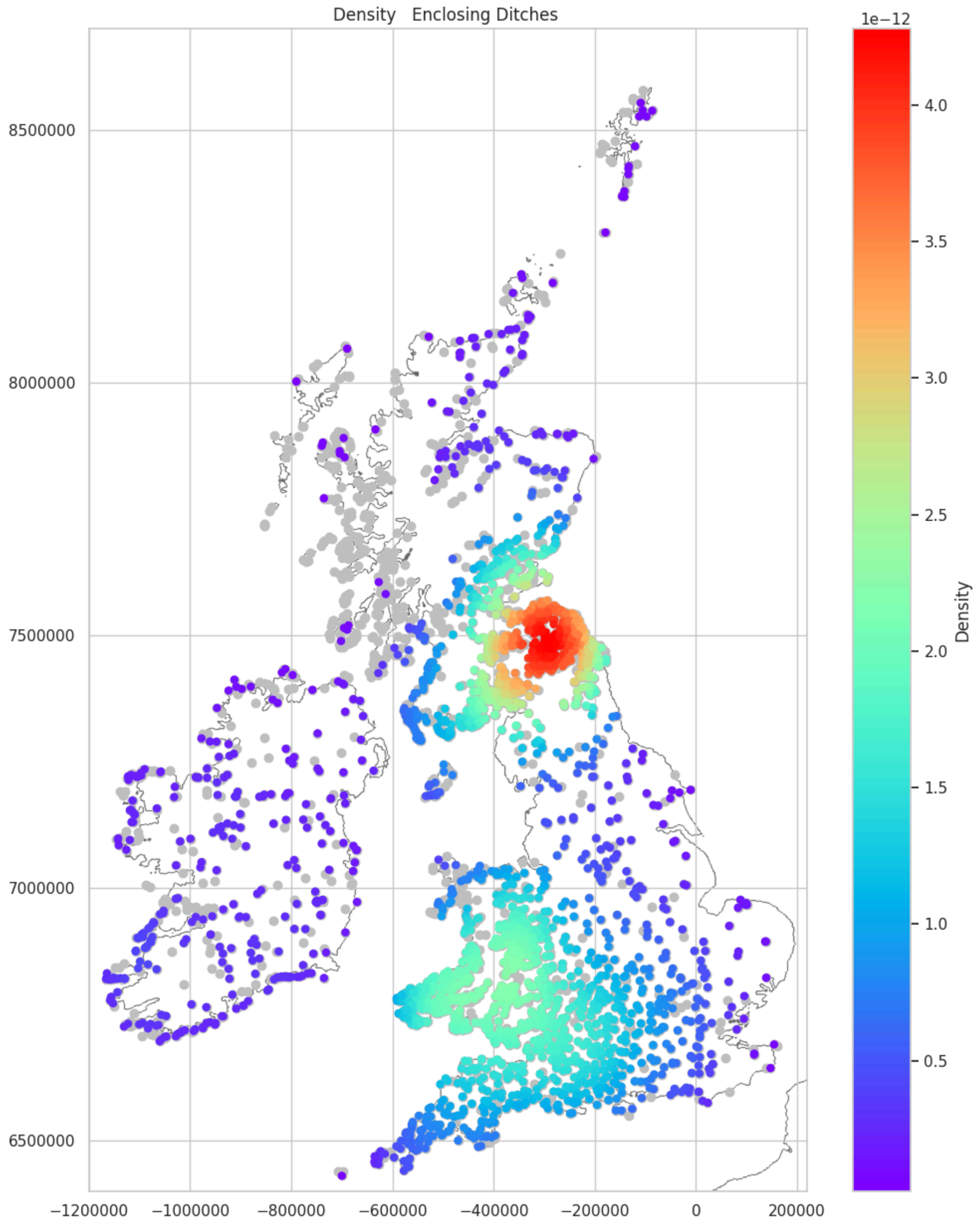



```
69.06%
```

Enclosing Ditches Density Mapped

As there is a 91.89% correlation between the ditches recorded above and the ditches in the enclosing section, it is unsurprising that the distribution of enclosure ditches matches that seen in Ditches Clipped Mapped. The recording bias, specifically over the Northwest, and discussed in Ditches Mapped (Not Recorded), can be seen.

```
In [ ]: plot_density_over_grey(enclosing_ditches_data_yes, 'Enclosing_Ditches')
```



Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

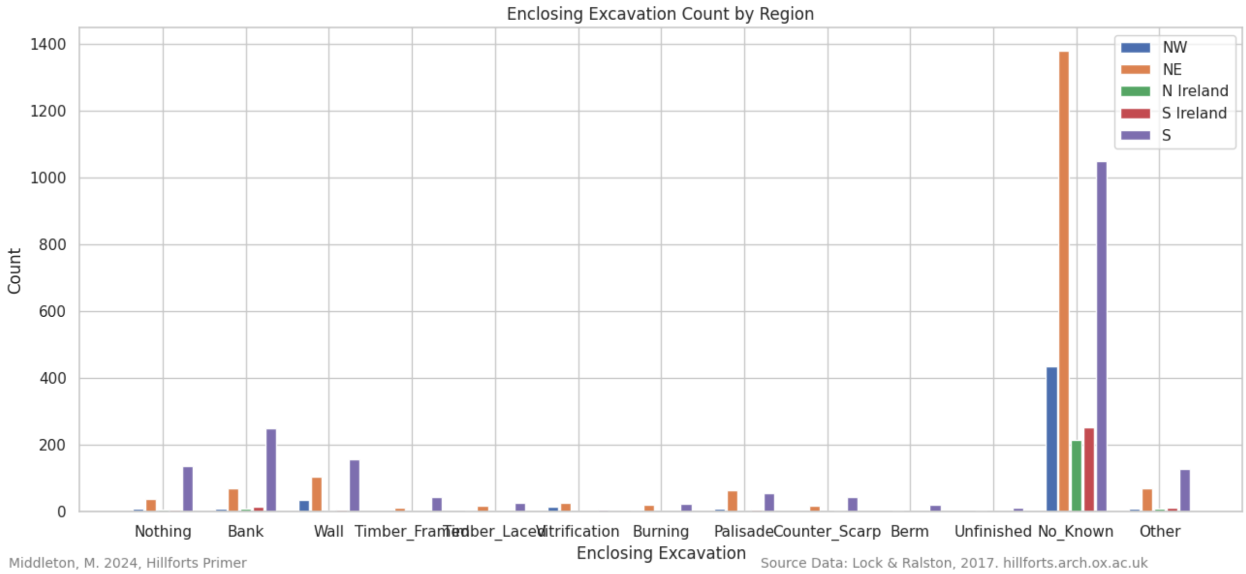
Enclosing Excavation by Region (Count)

No known excavation dominates this plot and will be removed in the following figure to improve the figure's legibility.

In [ ]:

```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Excavation_Nothing',
                  'Enclosing_Excavation_Bank',
                  'Enclosing_Excavation_Wall',
                  #'Enclosing_Excavation_Murus',
                  'Enclosing_Excavation_Timber_Framed',
                  'Enclosing_Excavation_Timber_Laced',
                  'Enclosing_Excavation_Vitrification',
                  'Enclosing_Excavation_Burning',
                  'Enclosing_Excavation_Palisade',
                  'Enclosing_Excavation_Counter_Scarp',
                  'Enclosing_Excavation_Berm',
                  'Enclosing_Excavation_Unfinished',
                  'Enclosing_Excavation_No_Known',
                  'Enclosing_Excavation_Other'],
```

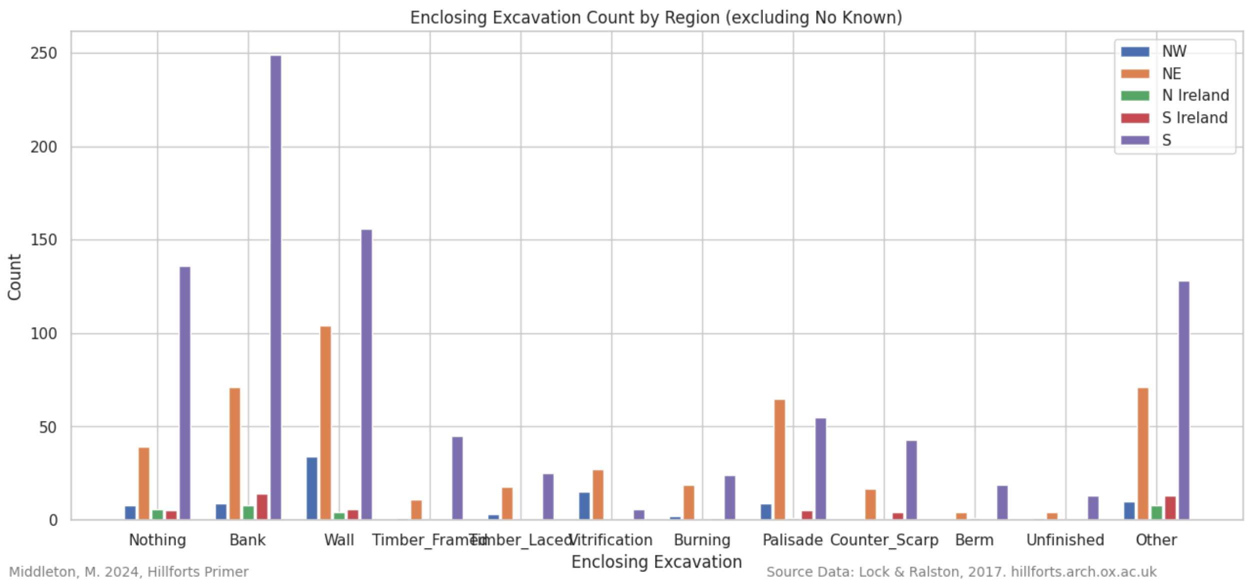
```
                 'Enclosing Excavation',
                 'Enclosing_Excavation Count by Region', 2, 'Yes')
```



Enclosing Excavation by Region (Count) (Excluding No Known)

As was seen in Enclosing Surface by Region (Count), raw counts are difficult to read. See the figures plotted by proportion below.

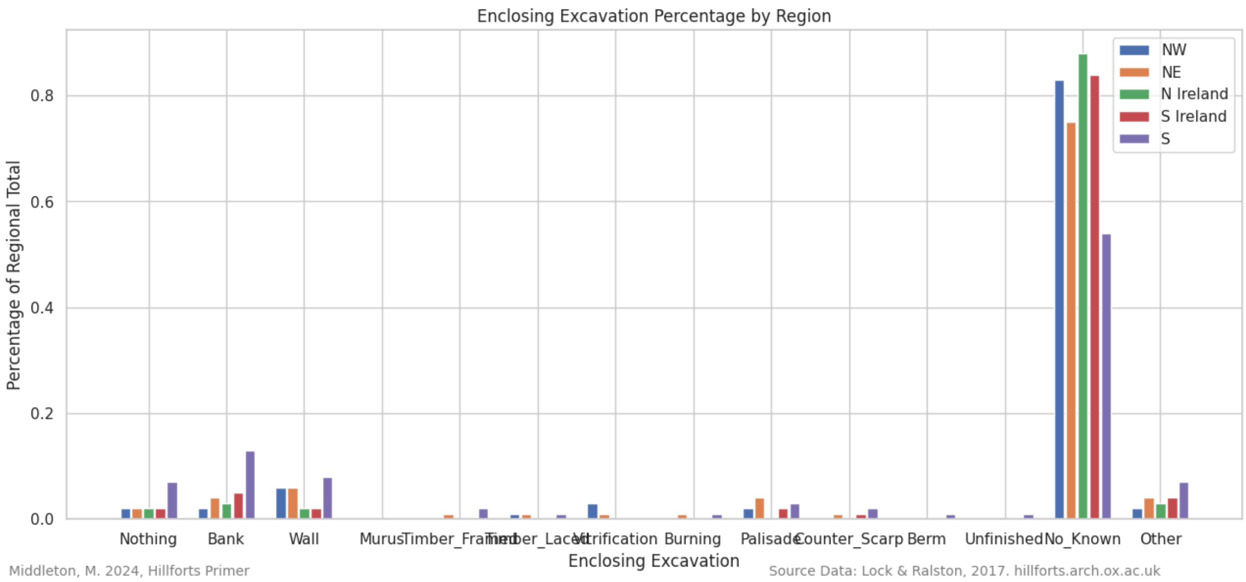
In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Excavation_Nothing',
                  'Enclosing_Excavation_Bank',
                  'Enclosing_Excavation_Wall',
                  #'Enclosing_Excavation_Murus',
                  'Enclosing_Excavation_Timber_Framed',
                  'Enclosing_Excavation_Timber_Laced',
                  'Enclosing_Excavation_Vitrification',
                  'Enclosing_Excavation_Burning',
                  'Enclosing_Excavation_Palisade',
                  'Enclosing_Excavation_Counter_Scarp',
                  'Enclosing_Excavation_Berm',
                  'Enclosing_Excavation_Unfinished',
                  #'Enclosing_Excavation_No_Known',
                  'Enclosing_Excavation_Other'],
                 'Enclosing Excavation',
                 'Enclosing_Excavation Count by Region (excluding No Known)', 2, 'Yes')
```



Enclosing Excavation by Region (Percentage)

This chart shows that most hillforts, in all regions, have not been excavated. The low bar for South, under 'No Known', shows that more hillforts have been excavated in the South than elsewhere. This translates to more features, by class, having been found in the South than across the other regions.

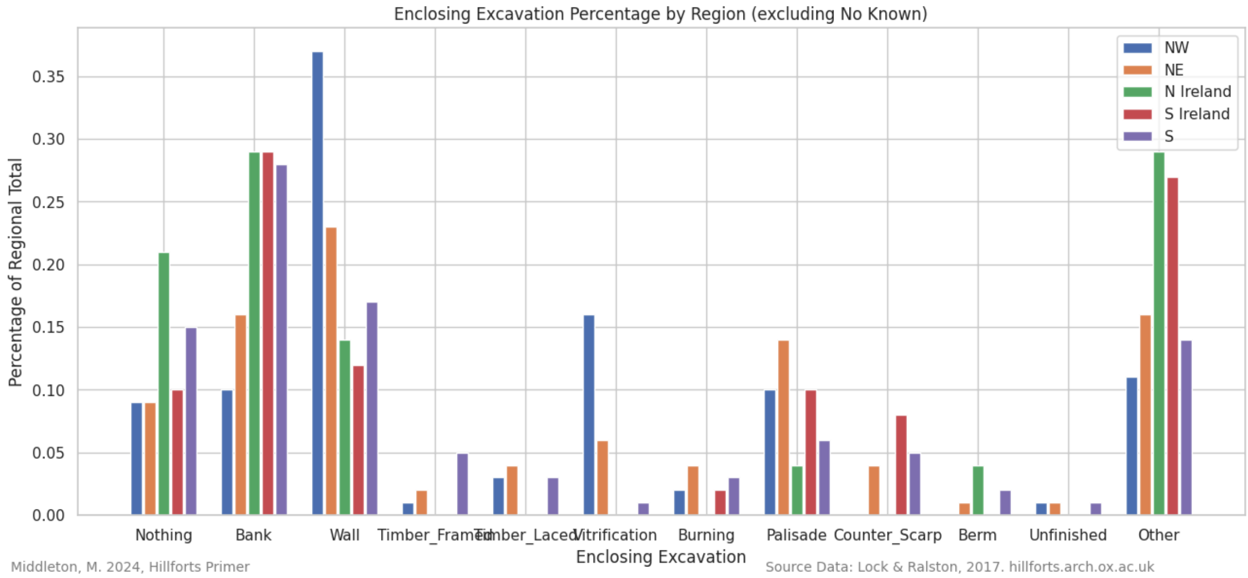
In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Excavation_Nothing',
                  'Enclosing_Excavation_Bank',
                  'Enclosing_Excavation_Wall',
                  'Enclosing_Excavation_Murus',
                  'Enclosing_Excavation_Timber_Framed',
                  'Enclosing_Excavation_Timber_Laced',
                  'Enclosing_Excavation_Vitrification',
                  'Enclosing_Excavation_Burning',
                  'Enclosing_Excavation_Palisade',
                  'Enclosing_Excavation_Counter_Scarp',
                  'Enclosing_Excavation_Berm',
                  'Enclosing_Excavation_Unfinished',
                  'Enclosing_Excavation_No_Known',
                  'Enclosing_Excavation_Other'],
                 'Enclosing Excavation',
                 'Enclosing_Excavation Percentage by Region', 2, 'Yes', True)
```



Enclosing Excavation by Region (Percentage) (Excluding No Known)

By excluding the 'No Known' excavation data, the remaining data can be plotted as a proportion of the total recorded classes by area. This reduces the dominance of the South data and enable the remaining plots to be comparable, proportionally, across the regions. This shows that in excavation, walls dominate the Northwest data while banks dominate in the South and across Ireland. In the Northeast, walls are proportionally the most common, but banks and palisades are also common. Of the remainder, vitrification is most common in the Northwest but is also found in the Northeast.

In [ ]:
```
plot_regions(location_enclosing_encodeable_data_nw,
                 location_enclosing_encodeable_data_ne,
                 location_enclosing_encodeable_data_ireland_n,
                 location_enclosing_encodeable_data_ireland_s,
                 location_enclosing_encodeable_data_south,
                 ['Enclosing_Excavation_Nothing',
                  'Enclosing_Excavation_Bank',
                  'Enclosing_Excavation_Wall',
                  #'Enclosing_Excavation_Murus',
                  'Enclosing_Excavation_Timber_Framed',
                  'Enclosing_Excavation_Timber_Laced',
                  'Enclosing_Excavation_Vitrification',
                  'Enclosing_Excavation_Burning',
                  'Enclosing_Excavation_Palisade',
                  'Enclosing_Excavation_Counter_Scarp',
                  'Enclosing_Excavation_Berm',
                  'Enclosing_Excavation_Unfinished',
                  #'Enclosing_Excavation_No_Known',
                  'Enclosing_Excavation_Other'],
                 'Enclosing Excavation',
                 'Enclosing_Excavation Percentage by Region (excluding No Known)', \
             2, 'Yes', True)
```



Review Enclosing Data Split

In [ ]:
```
review_data_split(enclosing_data, enclosing_numeric_data, \
                  enclosing_text_data, enclosing_encodeable_data)
```

```
Data split good.
```

Enclosing Data Package

In [ ]:
```
enclosing_data_list = \
 [enclosing_numeric_data, enclosing_text_data, enclosing_encodeable_data]
```

Enclosing Data Download Packages

If you do not wish to download the data using this document, all the processed data packages, notebooks and images are available here:
https://github.com/MikeDairsie/Hillforts-Primer.

In [ ]:
```
download(enclosing_data_list, 'enclosing_package')
```

Annex Data

There are just two annex features.

In [ ]:
```
annex_features = [
 'Annex',
 'Annex_Summary']

annex_data = hillforts_data[annex_features]
annex_data.head()
```

Out[ ]:

	Annex	Annex_Summary
0	No	NaN
1	No	NaN
2	No	NaN
3	No	NaN
4	No	NaN

Annex Numeric Data

There is no annex numeric data.

In [ ]:
```
annex_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Annex          4147 non-null   object
 1   Annex_Summary  533 non-null    object
dtypes: object(2)
memory usage: 64.9+ KB
```

In [ ]:
```
annex_numeric_data = pd.DataFrame()
```

Annex Text Data

There is a single annex text feature and it contains null values.

In [ ]:
```
annex_text_data = pd.DataFrame(annex_data['Annex_Summary'].copy())
annex_text_data.head()
```

Out[ ]:

	Annex_Summary
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

Annex Text Data - Resolve Null Values

Test for 'NA'.

In [ ]:
```
test_cat_list_for_NA(annex_text_data, ['Annex_Summary'])
```

```
Annex_Summary 0
```

Fill null values with 'NA'.

In [ ]:
```
annex_text_data = update_cat_list_for_NA(annex_text_data, ['Annex_Summary'])
annex_text_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 1 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Annex_Summary  4147 non-null   object
dtypes: object(1)
memory usage: 32.5+ KB
```

Annex Encodeable Data

There is a single encodable annex feature. It does not contain null values.

In [ ]:
```
annex_encodeable_data = pd.DataFrame(annex_data['Annex'].copy())
annex_encodeable_data.head()
```

Out[ ]:

	Annex
0	No
1	No
2	No
3	No
4	No

In [ ]:
```
location_annex_encodeable_data = \
pd.merge(location_numeric_data_short, annex_encodeable_data, \
         left_index=True, right_index=True)
```

In [ ]:
```
location_annex_encodeable_data_ne = \
pd.merge(north_east.reset_index(), annex_encodeable_data, \
         left_on='uid', right_index=True)
location_annex_encodeable_data_ne = \
```

```
pd.merge(name_and_number, location_annex_encodeable_data_ne, \
        left_index=True, right_on='uid')
```

In [ ]:
```
location_annex_encodeable_data_nw = \
pd.merge(north_west.reset_index(), annex_encodeable_data, \
        left_on='uid', right_index=True)
location_annex_encodeable_data_nw = \
pd.merge(name_and_number, location_annex_encodeable_data_nw, \
        left_index=True, right_on='uid')
```

In [ ]:
```
location_annex_encodeable_data_ireland_n = \
pd.merge(north_ireland.reset_index(), annex_encodeable_data, \
        left_on='uid', right_index=True)
location_annex_encodeable_data_ireland_n = \
pd.merge(name_and_number, location_annex_encodeable_data_ireland_n, \
        left_index=True, right_on='uid')
```

In [ ]:
```
location_annex_encodeable_data_ireland_s = \
pd.merge(south_ireland.reset_index(), annex_encodeable_data, \
        left_on='uid', right_index=True)
location_annex_encodeable_data_ireland_s = \
pd.merge(name_and_number, location_annex_encodeable_data_ireland_s, \
        left_index=True, right_on='uid')
```

In [ ]:
```
location_annex_encodeable_data_south = \
pd.merge(south, annex_encodeable_data, left_on='uid', right_index=True)
location_annex_encodeable_data_south = \
pd.merge(name_and_number, location_annex_encodeable_data_south, \
        left_index=True, right_on='uid')
```

Annex Mapped

271 (6.53%) of hillforts have an annex recorded.

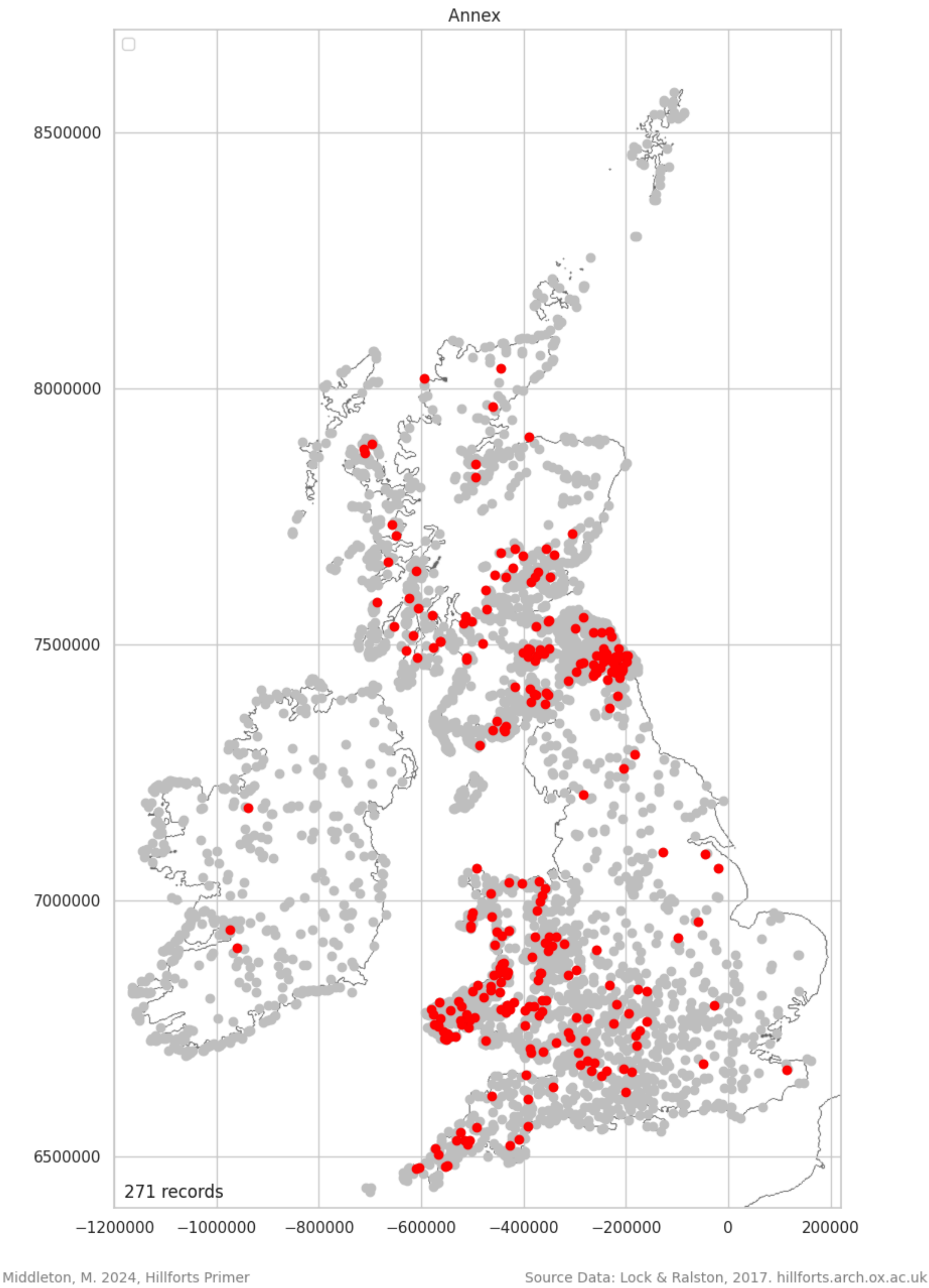
In [ ]:
```
annex_counts = annex_encodeable_data[['Annex']].value_counts()
annex_counts
```

Out[ ]:
```
Annex
No       3876
Yes       271
dtype: int64
```

In [ ]:
```
print(f'{round(annex_counts[1]/len(annex_encodeable_data)*100,2)}%')
```

```
6.53%
```

In [ ]:
```
annex_data_yes = \
plot_over_grey(location_annex_encodeable_data, 'Annex', 'Yes', '')
```

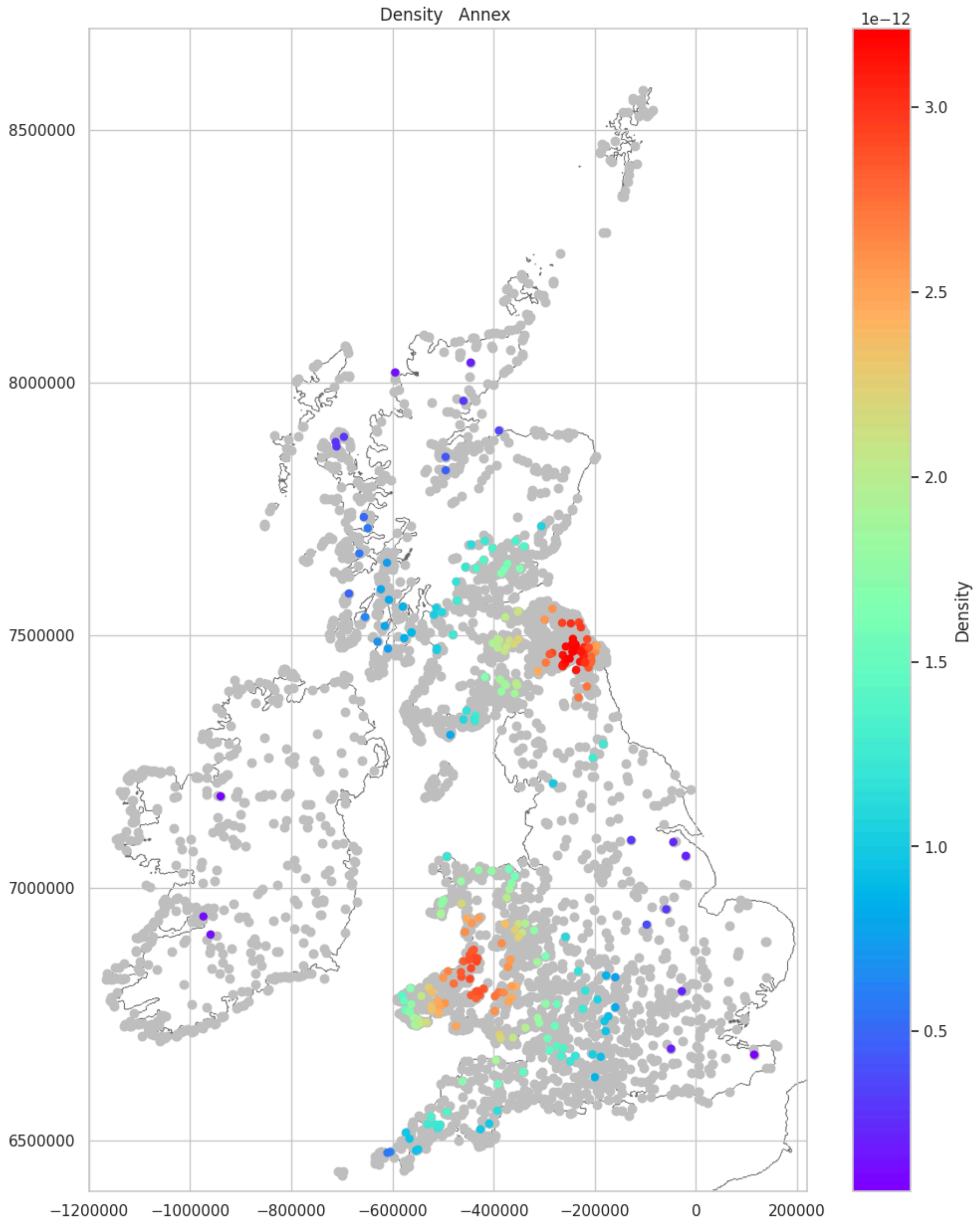



```
6.53%
```

Annex Density Mapped

The two main annex clusters coincide with clusters seen in the general density distribution. See: Part 1: Density Data Transformed Mapped. There is a cluster in the Northeast and another over the southern end of the Cambrian mountains. There are very few annexes out with these areas, and this may indicate there is a recording bias.

In [ ]:
```
plot_density_over_grey(annex_data_yes, 'Annex')
```



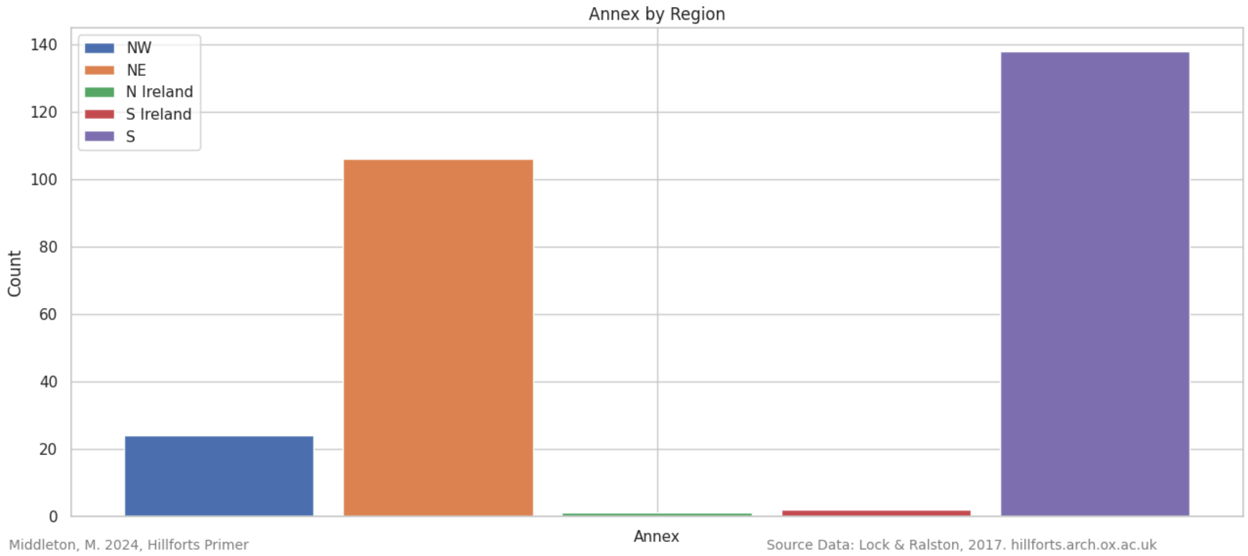
Middleton, M. 2024, Hillforts Primer

Source Data: Lock & Ralston, 2017. hillforts.arch.ox.ac.uk

Annex by Region (Count)

By count, most annexes are in the South and the Northeast. Annexes are rare in Ireland.

In [ ]:
```
plot_regions(location_annex_encodeable_data_nw,
                 location_annex_encodeable_data_ne,
                 location_annex_encodeable_data_ireland_n,
                 location_annex_encodeable_data_ireland_s,
                 location_annex_encodeable_data_south,
                 ['Annex'],
                 '',
                 'Annex by Region', 0, 'Yes')
```



Review Annex Data Split

In [ ]:
```
review_data_split(annex_data, annex_numeric_data, annex_text_data, annex_encodeable_data)
```

```
Data split good.
```

Annex Data Package

In [ ]:
```
annex_data_list = [annex_numeric_data, annex_text_data, annex_encodeable_data]
```

Annex Data Download Packages

If you do not wish to download the data using this document, all the processed data packages, notebooks and images are available here:
https://github.com/MikeDairsie/Hillforts-Primer.

In [ ]:
```
download(annex_data_list, 'annex_package')
```

Reference Data

Additional information relating to references is contained in a References Table. This can be downloaded from the Hillforts Atlas Rest Service API here or this project's data store here. The References Table has not been analysed as part of the Hillforts Primer at this time.

There are eight reference data features. Three have no null values and two contain no data.

In [ ]:
```
reference_features = [
 'References',
 'URL_Atlas',
 'URL_Wiki',
 'URL_NMR_Resource',
 'NMR_URL',
 'URL_HER_Resource',
 'URL_HER',
 'Record_URL']

reference_data = hillforts_data[reference_features]
reference_data.head()
```

Out[ ]:

	References	URL_Atlas	URL_Wiki	URL_NMR_Resource	NMR_URL	URL_HER_Resource
0	Dorling, P. and Wigley, A. 2012. Assessment of...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31113987	NaN	NaN	NaN
1	Dorling, P. and Wigley, A. 2012. Assessment of...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31113996	NaN	NaN	NaN
2	Cooke, W.H. 1882. Collections towards the hist...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31114017	NaN	NaN	NaN
3	Dorling, P. and Wigley, A. 2012. Assessment of...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31114037	NaN	NaN	NaN
4	Bowden, M. 2000. British Camp or Herefordshire...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31114060	NaN	NaN	NaN

In [ ]:

```
reference_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   References        3643 non-null   object 
 1   URL_Atlas         4147 non-null   object 
 2   URL_Wiki          4147 non-null   object 
 3   URL_NMR_Resource  1695 non-null   object 
 4   NMR_URL           1690 non-null   object 
 5   URL_HER_Resource  0 non-null      float64
 6   URL_HER           0 non-null      float64
 7   Record_URL        4147 non-null   object 
dtypes: float64(2), object(6)
memory usage: 259.3+ KB
```

URL_HER_Resource and URL_HER contain no data. All other features in this class are object features. The names of these two features suggest they hold urls, thus object features. It looks like this data has been lost from the online Atlas because of a feature type mismatch. Certainly, urls cannot be stored as numeric float64. As they contain no data they will be dropped.

Reference Numeric Data

Because of the issue with the URL_HER_Resource and URL_HER feartures, mentioned above, reference numeric data contains no infromation.

In [ ]:

```
reference_numeric_data = pd.DataFrame()
```

Reference Text Data

Six of the reference features are text

In [ ]:

```
reference_text_features = [
 'References',
 'URL_Atlas',
 'URL_Wiki',
 'URL_NMR_Resource',
 'NMR_URL',
 'Record_URL']

reference_text_data = pd.DataFrame(reference_data[reference_text_features].copy())
reference_text_data.head()
```

Out[ ]:

	References	URL_Atlas	URL_Wiki	URL_NMR_Resource	NMR_URL	
0	Dorling, P. and Wigley, A. 2012. Assessment of...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31113987	NaN	NaN	http://hillforts.arch.ox.
1	Dorling, P. and Wigley, A. 2012. Assessment of...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31113996	NaN	NaN	http://hillforts.arch.ox.
2	Cooke, W.H. 1882. Collections towards the hist...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31114017	NaN	NaN	http://hillforts.arch.ox.
3	Dorling, P. and Wigley, A. 2012. Assessment of...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31114037	NaN	NaN	http://hillforts.arch.ox.
4	Bowden, M. 2000. British Camp or Herefordshire...	https://hillforts.arch.ox.ac.uk/?query=Atlas_o...	http://www.wikidata.org/entity/Q31114060	NaN	NaN	http://hillforts.arch.ox.

Reference Text Data - Resolve Null Values

Test for 'NA'.

In [ ]:
```
test_cat_list_for_NA(reference_text_data, reference_text_features)
```

```
References 0
URL_Atlas 0
URL_Wiki 0
URL_NMR_Resource 0
NMR_URL 0
Record_URL 0
```

Fill null values with 'NA'.

In [ ]:
```
reference_text_data = update_cat_list_for_NA(reference_text_data, reference_text_features)
reference_text_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4147 entries, 0 to 4146
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   References        4147 non-null   object
 1   URL_Atlas         4147 non-null   object
 2   URL_Wiki          4147 non-null   object
 3   URL_NMR_Resource  4147 non-null   object
 4   NMR_URL           4147 non-null   object
 5   Record_URL        4147 non-null   object
dtypes: object(6)
memory usage: 194.5+ KB
```

Reference Encodeable Data

There is no reference encodable data.

In [ ]:
```
reference_encodeable_data = pd.DataFrame()
```

Review Reference Data Split

In [ ]:
```
review_data_split(reference_data, reference_numeric_data, \
                  reference_text_data, reference_encodeable_data)
```

```
There are missing features: ['URL_HER', 'URL_HER_Resource']
```

Reference Data Package

In [ ]:
```
reference_data_list = \
 [reference_numeric_data, reference_text_data, reference_encodeable_data]
```

Reference Data Download Packages

If you do not wish to download the data using this document, all the processed data packages, notebooks and images are available here:
https://github.com/MikeDairsie/Hillforts-Primer.

In [ ]:
```
download(reference_data_list, 'reference_package')
```

Save Figure List

In [ ]:
```
if save_images:
    path = os.path.join(IMAGES_PATH, f"fig_list_{part.lower()}.csv")
    fig_list.to_csv(path, index=False)
```

Acknowledgements

I would like to thank Emily Middleton for editing; Dr Dave Cowley for his encouragement, support and regular chats throughout this project; Strat Halliday for access to his expert knowledge and his thoughts on the data collection phase of the Hillforts Atlas, and to Professor Jeremy Huggett for his advice on how to summarise this work into abstracts for forthcoming publications.

Postscript

The work in the Hillforts Primer is the first phase in analysing the Hillforts Atlas data. This has been the data review. In reading these documents it is hoped that reader will have a solid grounding and understanding of the data's scope, limitations, areas of opportunity and where new research can complement what is already known. Throughout this document the data has been split into three groups; numeric, encodeable and text. The encoding has not been done in this phase as there is more to do. The data packages output, from this project, remain human readable. The next phase will look at correlations in the data as well as finally encoding and scaling the data. The next phase will render the data difficult to read for a human but it will make it much more likely to be useful for machine learning. Links to the next phase documents will be added once they become available. Thanks for reading.