



UCL

A research project submitted in partial fulfilment of the requirements for the
degree of Master of Science (MSc) in

ECOLOGY AND DATA SCIENCE

UNIVERSITY COLLEGE LONDON

THE EFFECTS OF URBAN GROWTH ON BATS IN KENT

By

FZWM5

Primary Supervisor: Dr Jim Labisko
Secondary Supervisor: Dr Ella Browning

BIOS0034

People and Nature Lab
Department of Biosciences

UNIVERSITY COLLEGE LONDON

Date of submission: Friday 23rd August 2024

Word count: 5,115

Word count (inc. title page, figure legends and references): 8,393

Journal writing style: Journal of Animal Ecology

Conflicts of Interest

The author has no conflicts of interest to declare.

Statement on Inclusion

This study was undertaken using data collected in the United Kingdom only. Local data collection had previously been carried out by citizen scientists resident in the study area, and conversations with these stakeholders were included during the conceptualisation of the research question. The author is a resident of the United Kingdom, but not the study region. Further discussion of the outcomes of the work is planned with local stakeholders.

Data availability statement

An anonymised version of the original data will be achieved. The original dataset is not available due to data protection restrictions, but the majority of the anonymising code can be found within the main repository. All code and supporting data used to generate the results of this paper can be found at https://github.com/daisy-jowers/kent_bat_project.

ABSTRACT

1. Urban area is expected to grow by 30-60% globally by 2050 as governments work to deliver adequate housing for growing human populations in line with their commitments under the Sustainable Developments Goals, but this expansion poses a significant threat to global biodiversity.
2. The aim of this study was to quantify the effect of historical expansion on populations of *Chiroptera* in order to inform future development policy to limit harm to these species.
3. In this study, I investigated the effects of urban and suburban area on odds of occurrence of 6 genera and 11 species of bats in Kent, UK, using a dataset of c.28,000 individual observations of bats collected by citizen scientists over 43 years (1980-2023). I also made preliminary assessments of the differential effects of growth pattern (infilling, edge expansion, outlying development) on a subset of this data (2017-2023).
4. I found that increasing urban land cover significantly increased the odds of presence of *Myotis* and *Eptesicus serotinus*, whilst decreasing odds of presence of *Plecotus auritus*. Where the inclusion of urban growth improves model fit, this improvement is

always greatest when the dominant pattern of urban growth is included within the model, however there is no one pattern of growth discernible as a clear 'winner for biodiversity', with responses varying within and across genera.

5. The findings from this study suggest that where patches of the most dense urban areas are small, species otherwise thought to be urban avoiders (such as *Myotis*) are able to capitalise on the resources provided by anthropogenic features of their environment. The power of the analysis was limited by data availability, and future work would benefit from broadening the spatial and temporal scope of the study.

Keywords: urban growth; bats; landscape composition; urban growth pattern; landscape expansion index; biodiversity

ACKNOWLEDGEMENTS

I would like to thank the Kent Bat Group for their ongoing work to support the understanding and conservation of bats in the UK, and particularly to John Puckett and Peter Scrimshaw, for taking the time to introduce me to key aspects of the Bat Group's work and ongoing enthusiasm. I am also grateful to my supervisors for providing very constructive criticism and sustained support throughout the project.

34 CONTENTS

35	Abstract	3
36	Acknowledgements	5
37	1 Introduction	8
38	2 Materials and Methods	13
39	2.1 Materials	13
40	2.2 Methods	20
41	3 Results	22
42	3.1 Effects of landscape composition	24
43	3.2 Reduced Time Period	26
44	3.3 Model Comparison	27
45	3.4 Effects of urban growth	28
46	4 Discussion	31
47	4.1 Landscape Composition	31
48	4.2 Urban Growth and Growth Pattern	34
49	4.3 Limitations and Recommendations	34

50	5 Conclusion	36
51	References	37
52	Supplementary Materials	i
53	A - Species Detection Inclusions	ii
54	B - Pre-scaling means and standard deviations of land cover classes	iii
55	C - Full Model Results	iv
56	D - Code	v
57	E - Risk Assessment	lxxxix

1. INTRODUCTION

Biodiversity is globally threatened by the expansion of urban areas (Li et al., 2022; Simkin et al., 2022; Urban et al., 2024). Through the Sustainable Development Goals (SDGs) and the 2030 Agenda for Sustainable Development (United Nations General Assembly, 2015), state actors have committed to protecting the natural environment (SDGs 14&15), but Kroll et al. (2019) show that these commitments can stand in conflict with other responsibilities, such as providing adequate quality housing (SDG 11) for growing populations (United Nations Department of Economic and Social Affairs, Population Division, 2024). McDonald et al. (2020) estimate that providing adequate housing at urban densities, as opposed to rural ones, will save 50 million km² of land from settlement between 1990 and 2030. Land use change is consistently found to be the primary driver of global biodiversity declines (IPBES, 2019; Isbell et al., 2023; Sala et al., 2000). Given this, minimisation of land use change by developing land already urban in character or surrounded by urban land (densification) presents an opportunity to minimise negative impacts of population growth on global biodiversity.

However, evidence on the effects of urban area density on species abundances and diversity is mixed. The Intermediate Disturbance Hypothesis (IDH) (Connell, 1978) suggests that species diversity shows a 'hump'-shaped relationship with urban density, as

at low-to-medium densities non-natives and urban adapters are able to colonise, but there is sufficient habitat remaining to support (though potentially reduce in size) populations of native species. A global meta-analysis by McKinney (2008) did not find consistent support for this hypothesis outside of plant and avian taxa, but more recent studies have found evidence of the effect in both invertebrates (Tajthi et al., 2017) and non-flying mammals (Battisti et al., 2017), as well as confounding evidence in birds (Battisti & Fanelli, 2016; Callaghan et al., 2019). In contrast with the land-use effect, on its surface the increased diversity supported in mid-density urban areas under the IDH seems to promote expansion of the suburban border ("urban sprawl") over densification. However, this hides important differences in response across species, with Concepción et al. (2016) finding a significant increase in the ratio of generalist to specialist birds breeding along increasing urban gradients, and similar effects seen in bats (Starik et al., 2024). This homogenising effect of urban density may be mitigated by inclusion of large urban parks and freshwater designed so as to create 'wildlife' corridors through densely urban areas (Zellmer and Goto, 2022).

Urban growth is directly connected to urban form, with growth being classified into three basic patterns: infilling (Fig. 1B), where unused sites within the boundary of an urban area are developed (often though not always brownfield); edge expansion (Fig. 1C), where existing urban areas grow in geographic area and subsume surrounding land (often agricultural, suburban, or previously conserved for nature); and outlying development (Fig. 1D), where new urban hubs are built away from existing urban centres (Liu et al., 2010). Infilling tends to lead to densification; whilst edge expansion represents a degree of urban sprawl. Outlying development sits outside of this direct relationship between urban growth and urban form, as its effect on the overall landscape depends on both its size, density and its isolation from existing urban patches.

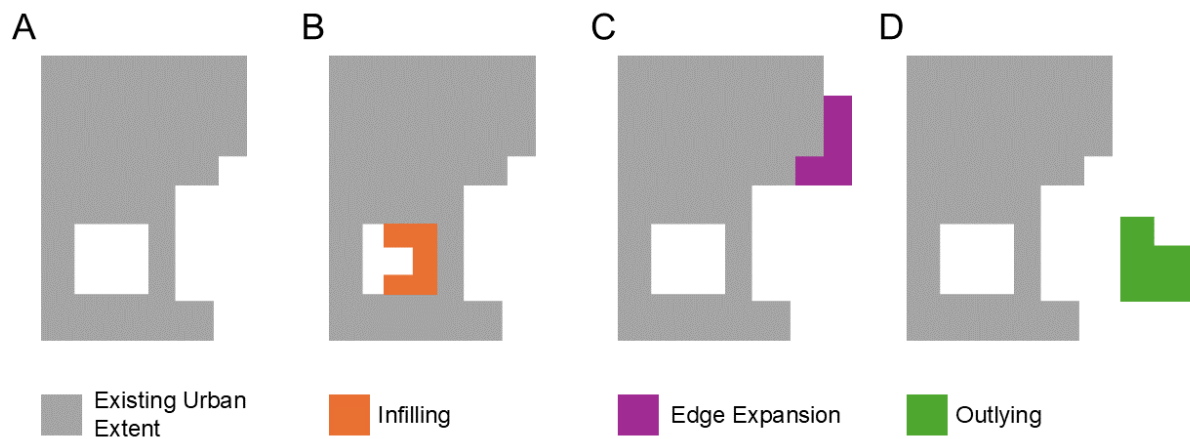


Figure 1: Patterns of urban growth.

In the face of rapid change, the ability to monitor populations and respond efficiently is critical. To simplify the biodiversity monitoring process, indicator species are often used as a proxy for ecosystem health and broad species trends (though this practice has been criticised, see Tälle et al., 2023). In order to be a useful proxy, indicator species must reliably track the measure they are purported to represent. Bats (*Chiroptera*) are proposed as good candidates for bioindicators for riverine habitats (De Conno et al., 2018), bioaccumulation (Zukal et al., 2015) and various other ecological processes (Jones et al., 2009). The benefits of using bats as indicator species increase as the cost and difficulty of monitoring their populations is reduced by improvements in passive acoustic monitoring (PAM) and automated call detection (Gibb et al., 2019; Sugai et al., 2019), but greater understanding of their responses to urban pressures is required before their use as an indicator can be justified (Russo et al., 2021).

There are multiple mechanisms through which urban growth may affect bat populations. Increased direct human conflict, roost disturbance, and the building of new roads have universally negative effects, whilst increased artificial night at light (ALAN) creates winners and losers across species (Browning et al., 2021). Flexibility in roost preference, high

aspect ratios (low wing span to wing area), and availability of preferred diet are all indicators of increased adaptability to urban environments (Jung & Threlfall, 2018; Piano et al., 2020). Despite increasing levels of research, evidence of species-specific effects is still mixed, particularly in urban adapters, where success seems to vary across study sites (Border et al., 2017; Fleming et al., 2024; Gili et al., 2020; Hale et al., 2012; Lintott et al., 2016; Starik et al., 2024; Stidsholt et al., 2024; Vlaschenko et al., 2023).

Bat populations in the United Kingdom (UK) are some of the best studied, with elements of the National Bat Monitoring Programme (NBMP) having been in place since 1997 (Barlow et al., 2015) and more than 80 regional bat groups, many of which were collecting data long before nation-wide monitoring began. Whilst the UK has been actively supporting citizen science for 50 years through the Biological Records Centre, this organisation only employs the equivalent of 7 full-time employees to process the contributions of an estimated 70,000 ‘amateur’ recorders (many of whom have significant knowledge of their fields) (Pocock et al., 2015). In this study, I will use a hitherto unexplored dataset of observation records collected by citizen scientists in Kent, UK, over a period of 43 years (1980 - 2023) to test three hypotheses about the effect of urban area, urban growth, and urban growth pattern on populations of bats:

H1: The proportion of urban and suburban area within the core sustenance zone (CSZ) will have an effect on presence of bats.

H2: The proportion of *new* urban/suburban area (urban growth) within the CSZ will have an effect on presence of bats (over and above the size effect of the urban area).

H3: The effect of urban growth on the presence of bats will be conditional on the

pattern of urban/suburban growth.

I expect to find divergent responses between species, with increased urban and suburban area benefiting generalist species over specialist ones(Jung and Threlfall, 2018). Amongst species benefiting from a moderate amount of urban density, edge expansion is expected to be the preferred pattern of expansion, whilst both urban exploiters and urban avoiders are expected to benefit more from infilling, the former because of the reduction in competition within the dense urban niche and increased resources, and the latter because infilling reduces the overall expanse of developed area (Lato et al., 2023; Martin & Bonier, 2018). The expected effect of outlying development is unclear, and is likely to depend on factors such as size of development and distance from existing urban patches, neither of which are within the scope of this study.

2. MATERIALS AND METHODS

2.1 Materials

All data manipulation and analysis was performed in R version 4.4.1 (R Core Team, [2024](#)).

2.1.1 Study Area

The county of Kent¹ lies at the southeast of the UK, and is predominately composed of a mosaic of arable and improved grassland, interspersed with stands of deciduous woodland, small towns, and villages. There are also a number of dense urban areas scattered throughout the county, most of which are surrounded by a suburban border of equal or greater size.

Between 1990 and 2023, urban area in Kent increased from 96km² to 137km² (42.3%), with suburban area increasing from 284km² to 340km² (19.8%) (Fig. 2). This represents a 97km² increase in absolute developed area, alongside the an increase in built density causing land to move from the suburban to the urban classification.

¹The ceremonial county of Kent is composed of both the *administrative* county of Kent and the Unitary Authority of Medway. In this paper, 'Kent' refers to the ceremonial county (i.e. Kent and Medway).

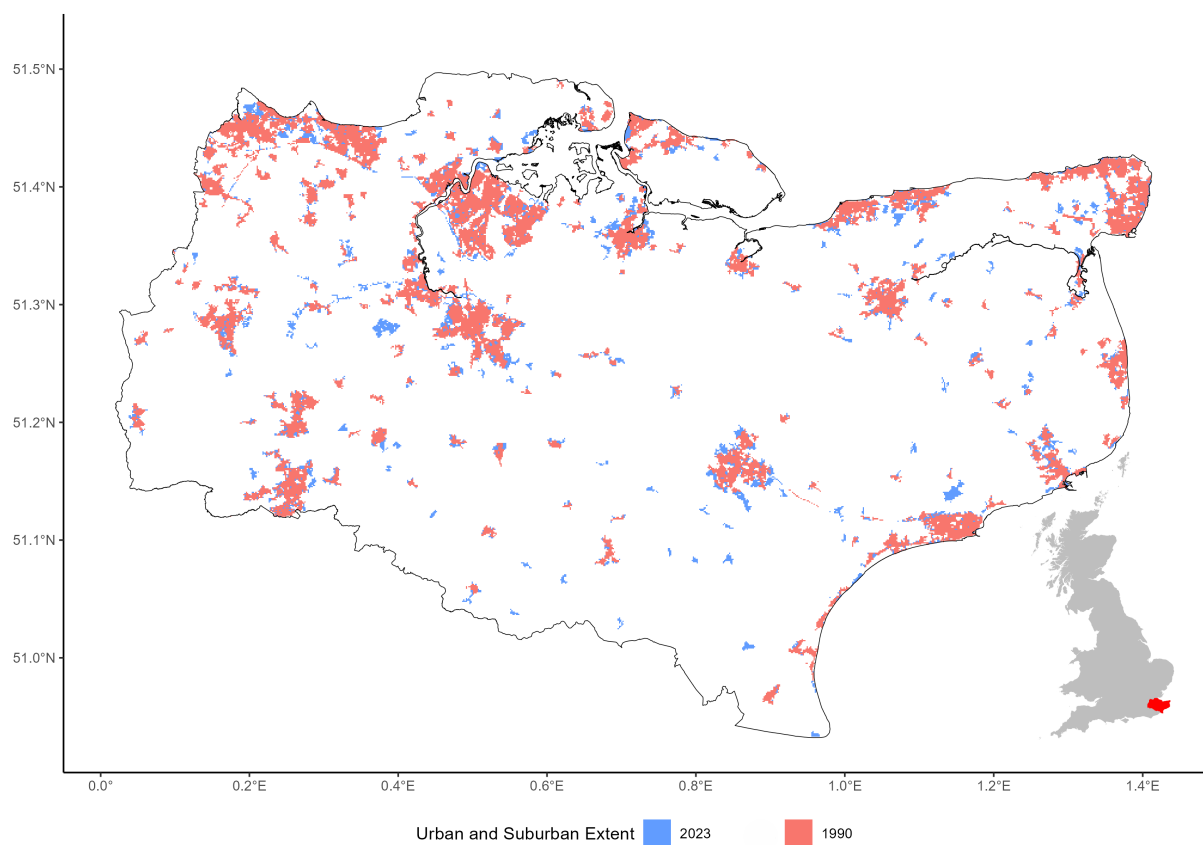


Figure 2: Overview of urban growth in Kent, UK, with 1990 extent (pink) and 2023 extent (blue). Only built-up areas over 20 ha in size are shown. Inset shows the location of Kent (red) within the United Kingdom.

2.1.2 Bat Observation Data

Two sets of observation data were provided by Kent Bat Group (KBG), which contained 28,307 unique records. Records range from incidental sightings reported by the public to the results of standardised surveys run under the NBMP (Barlow et al., 2015; Bat Conservation Trust, 2023). Observations were removed if they did not record:

- a) a location to within an accuracy of 1km² (n = 3);
- b) a location within Kent (n = 1,391);
- c) the identity of the bat to at least genus-level (n = 2,128).

As detection rates are expected to vary across seasons (Barlow et al., 2015; Baumgardt et al., 2022), a categorical variable was added splitting observations into the 'active' (April - October) and 'hibernation' (November - March) seasons. This was chosen as a generalisation of species activity patterns, with consistent classification required for the generation of species detection matrices (outlined later), however as with all generalisation, this obscures some variation between species (Newson et al., 2015). Two observations did not contain date information sufficient to determine their season, and these were removed. In total, 24,783 records were included in the final analysis.

To control for potential double counting of individuals across sites, all observation locations were generalised to their respective 1 km x 1 km British National Grid Square (Ordnance Survey (GB), 2012). Whilst this will reduce the effect of variation in sampling effort and detection probability per grid square, double counting may still occur when individual bats travel across boundaries, especially notable when points lie close to grid boundaries.

Observation data was then collated by year, season, grid square, and genus/species identification, resulting a singular presence entry (or not) for each unique combination of these variables.

As the dataset contained very few true absence records ($n = 142$), so pseudo-absences were generated by inferring absence when an observation had been recorded, but had failed to detect the focal genus/species. no adjustment was made for varying detection rates of species across survey type. due to a lack of consistent information. Two detection matrices were created - one at the genera level and one at the species level. In the genera matrix, all records of the focal genus, including genus-only, species-group, and species-level identifications were used as presence data, with absences generated from all non-focal records. In the species matrix, all species-level observations were treated as presence records, but when generating absence records, only detections which could not be the focal species were included (i.e. species-group and genus-only identifications were not used when the focal species was part of the group)(Supplementary Information, Table S1). Due to difficulties distinguishing *M. mystacinus* and *M. brandtii* both acoustically and visually (without handling), they were treated as a species pair.

2.1.3 Land cover variables

For each grid square and year 1980-2023, land cover variables were extracted from UK Centre for Ecology and Hydrology (UKCEH) Land Cover 25 m Raster Maps (Fuller et al., 2002; R. Morton et al., 2014; R. D. Morton et al., 2020a, 2020b, 2020c, 2021, 2022, 2024a, 2024b; Rowland et al., 2017, 2020). These datasets provide the dominant land cover class at a 25 m resolution, derived from satellite imagery. For each year, variables were extracted from the land cover raster temporally closest. The geometry of each grid

square was combined with that of its eight neighbours prior to extracting landscape variables to better reflect the scale of the average size of the Core Sustainance Zone of UK bats, which measures the area around a roost where habitat strongly influences the colony (Bat Conservation Trust, 2016). For each grid square, the percentage cover of each land cover class in the broader area surrounding the grid square was calculated using R package `terra` (Hijmans, 2024). Due to high correlation within the study area, arable and improved grassland categories were combined to avoid co-linearity of predictors. Both classes represent non-natural areas with low diversity of flora, and have high confusion rates with the land cover classification method (R. D. Morton et al., 2024a). To account for variation in the magnitude of coverage of different land cover classes, all variables were standardised to Z-scores where $\mu = 0$ and $\sigma = 1$. Pre-scaling means and standard deviations can be found in the Supplementary Materials (Table S2).

2.1.4 Urban growth variables

For each land cover map from 2015 - 2023, all patches of contiguous urban and suburban (combined) land cover were identified using `terra` (Hijmans, 2024). The two land cover classes were combined for analysis of urban growth to maintain simplicity in the model, but future studies should address the potential divergence in urban growth, suburban growth, and the transformation of suburban into urban land. To ensure focus on 'built-up' areas, as opposed to isolated buildings/small groups of buildings, all patches under 20 ha were removed. This aligns with the UK-wide definition of areas 'irreversibly urban in character' (Office for National Statistics, 2023). New urban extent ('urban growth') was identified as the raster cells classified as urban/suburban in year n but not in year $n - 1$ ($n - 2$ in the case of 2017 only) (Fig. 3).

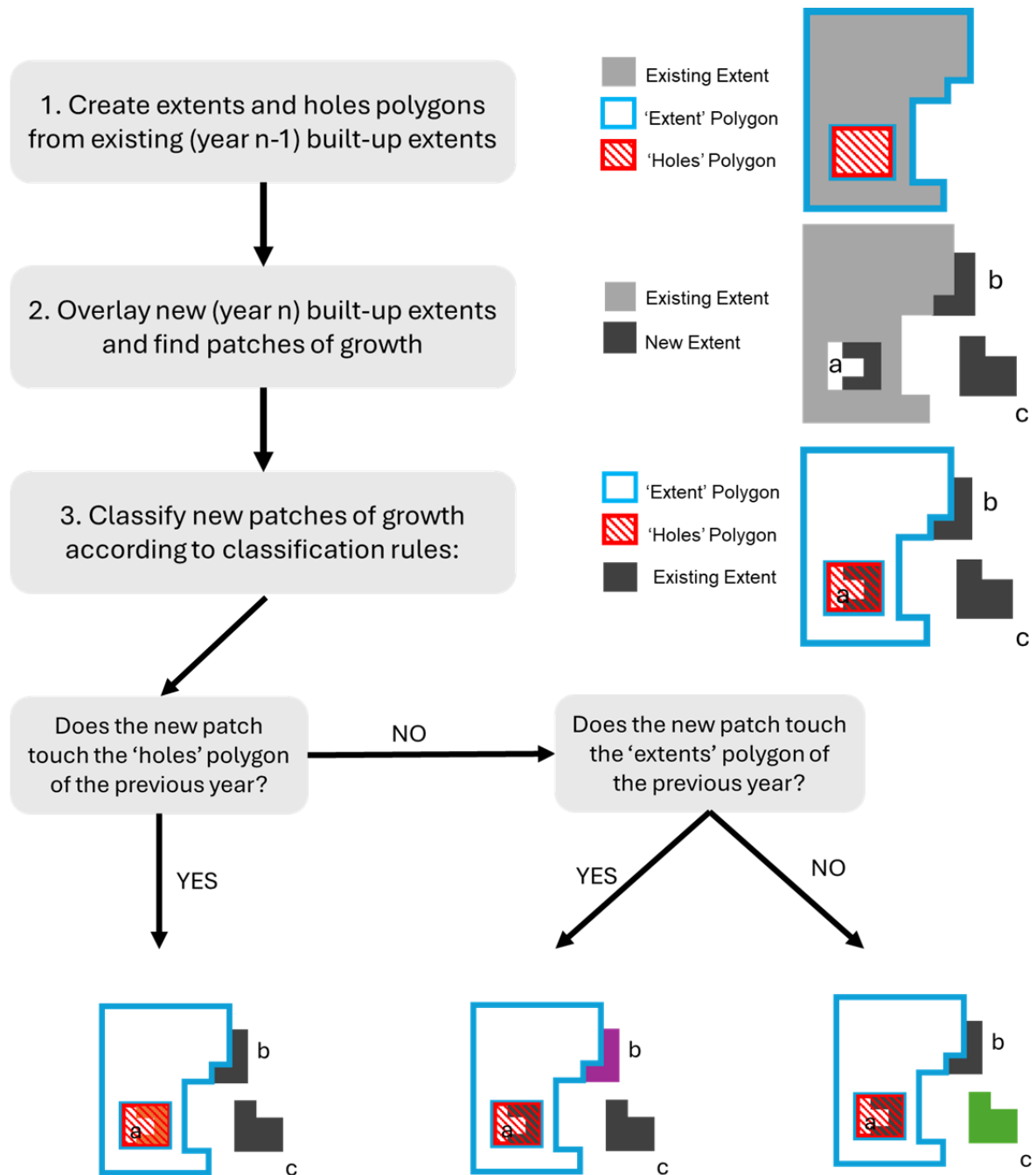


Figure 3: Classification process of from identifying patches of urban growth to assigning a pattern category to these patches.

The Landscape Expansion Index (LEI) as proposed by Tian et al. (2022) was used to topologically classify each patch of urban growth into one of three patterns (Fig. 1). First, two sets of polygons were created to spatially represent the urban extent of the year prior to the focal year (Fig. 3):

- A set of 'extent' polygons, the combination of all individual urban patch polygons for the year.
- A set of 'holes' polygons, representing all areas fully enclosed by the 'extent' polygons, but not themselves classified as urban or suburban.

. For each new urban patch, if any part of the patch shared any point with the 'holes' polygons of the previous year, it was classified as the 'infilling' type. For the remaining patches, if the new patch shared any point with the previous year's 'extent' polygons it was classified as 'edge expansion'. Otherwise, it was classified as 'outlying' (Fig. 3).

Once new urban extents had been identified and classified, the percentage cover of new urban extent and of each urban growth type was extracted as for the landscape composition variables. A final urban growth variable, 'dominant growth pattern' was calculated, based on the growth pattern with the largest (pre-scaling) area in each 3 km area.

2.2 Methods

Statistical Analysis

Generalised linear mixed effects models (GLMMs) with a binomial logit link were used to determine the effects of all variables on the odds of presence of each genera and species. In total, five models were considered, all implemented using the R package `lme4` (Bates et al., 2015). Land cover classes included in the models were: deciduous woodland, arable/improved grassland, semi-natural grassland, coastal, freshwater, urban and suburban. These land cover classes were chosen to represent those most relevant to habitat selection of bats, and to reflect variation in the study area. Coniferous woodland and mountain, heath, and bog land cover classes were excluded from the analysis due to their low presence within Kent. Model 1 included all land cover variables as fixed effects, but no urban growth variables, and included the full temporal range of the observation dataset (Table 1). As the urban growth variables were only extracted for years 2017-2023, all further models were run on a truncated set of observations (2017-2023). To allow for comparisons with later models and to the results of the full time period, Model 2 was identical to Model 1 in formula. Model 3 introduced the area of urban growth variable only, whilst Models 4 and 5 included both the area of urban growth variable and the dominant pattern variable (Table 1). Model 5 included the interaction between these two terms, as it was suspected the effect of dominant growth pattern might be conditional on the size of the new development. Season and year were included as random effects in all models, to account for variations across seasons and over time (potentially caused by changes in technology, or common survey types)(Barlow et al., 2015; Baumgardt et al., 2022).

Table 1: Overview of temporal range of dataset and included covariates for Models 1-5. ✓ indicates a covariate was included in the model, whilst ✕ indicates it was not.

	Model 1	Model 2	Model 3	Model 4	Model 5
Dataset	1980-2023	2017-2023	2017-2023	2017-2023	2017-2023
Random Effects					
Year	✓	✓	✓	✓	✓
Season	✓	✓	✓	✓	✓
Fixed Effects					
Coastal Land Cover	✓	✓	✓	✓	✓
Deciduous Woodland Cover	✓	✓	✓	✓	✓
Arable Cover	✓	✓	✓	✓	✓
Improved Grassland Cover	✓	✓	✓	✓	✓
Semi-Natural Grassland Cover	✓	✓	✓	✓	✓
Freshwater Cover	✓	✓	✓	✓	✓
Urban Cover	✓	✓	✓	✓	✓
Suburban Cover	✓	✓	✓	✓	✓
Urban Growth	✕	✕	✓	✓	✓
Urban Growth Pattern	✕	✕	✕	✓	✓
Urban Growth*Urban Growth Pattern	✕	✕	✕	✕	✓

Aikaike Information Criterion (AIC) values were used to determine the best-fitting models for each genera and species. As AIC values are a measure of the relative fit of models to a single dataset, the results of Model 1 were not included within this comparison, nor should the values given be interpreted across species/genera. To quantify the difference in effect between the three categories of dominant growth, post-hoc pairwise contrast tests were performed using R package `emmeans` (Lenth, 2024).

3. RESULTS

Excluding *Rhinolophus hipposideros* (the lesser horseshoe bat), all 18 species of British bat were recorded within the boundary of Kent. The most frequently recorded genera was *Pipistrellus* ($n = 12,289$), with the most common species detection *P. pipistrellus* ($n = 6,395$). Seven species were recorded fewer than 10 times (Table 2). Geographic coverage was fair, 59.6% of 1 km² grid squares had at least one observation within the full time period (Fig. 4A). However, there was a significant variation in the number of observations per year (2 in 1980 - 470 in 2006), Fig. 4B).

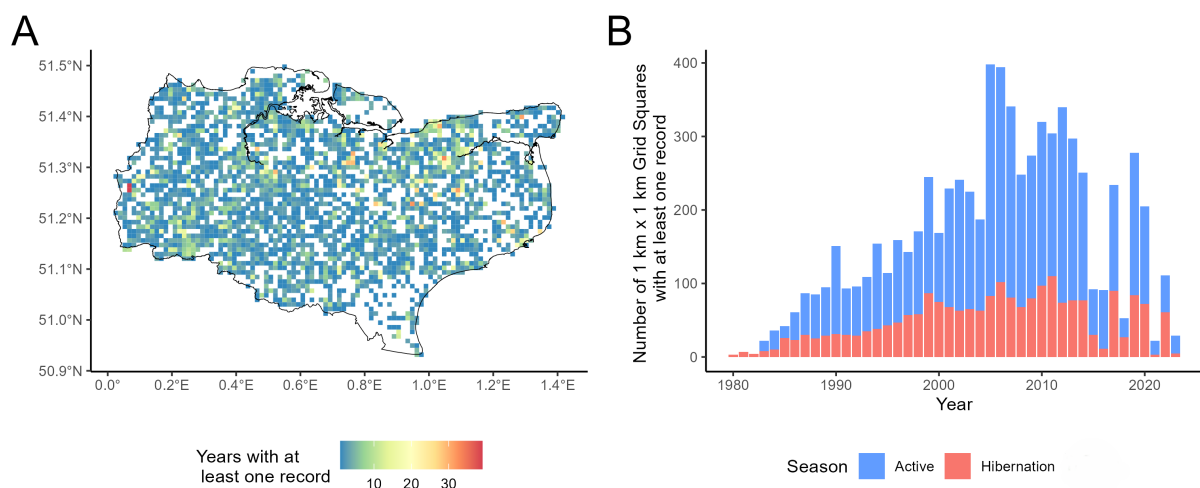
















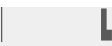








Figure 4: Spatial and temporal distribution of records within the dataset. Individuals records are aggregated to the grid square and year. Observations of 'bat', or missing identification information, are excluded.

Table 2: Temporal distribution of genera and species observations used in the final models. Note that observations of species will not necessarily sum to those of their genera because of the inclusion of genera-only identifications and the spatial aggregation of records. Histograms of observations are binned at 2-year intervals, and minimum and maximum lines represent 1980 and 2023, respectively.

Common Name	Scientific Name	Number of Observations (1980-2023)	First Observation	Last Observation	Yearly Distribution	Number of Observations (2017-2023)
Barbastelle	<i>Barbastella barbastella</i>	1	2009	2009		0
Serotine	<i>Eptesicus serotinus</i>	1569	1983	2023		201
Mouse-eared bats	<i>Myotis gen.</i>	5685	1980	2023		879
Alcathoe	<i>Myotis alcathoe</i>	3	2010	2017		2
Bechsteins	<i>Myotis bechsteinii</i>	37	2000	2019		2
Brandt's/Whiskered	<i>Myotis brandtii/mystacinus</i>	574	1980	2022		34
Daubentons	<i>Myotis daubentonii</i>	2692	1980	2023		356
Grey mouse-eared	<i>Myotis myotis</i>	2	1985	1985		0
Natterers	<i>Myotis nattereri</i>	1738	1980	2023		267
Noctule bats	<i>Nyctalus gen.</i>	1271	1983	2023		211
Lesser noctule/Leislars	<i>Nyctalus leisleri</i>	259	1985	2023		32
(Common) Noctule	<i>Nyctalus noctula</i>	997	1983	2022		173
Pipistrelles	<i>Pipistrellus gen.</i>	12289	1981	2023		2038
Kuhls pipistrelle	<i>Pipistrellus kuhlii</i>	4	2013	2020		3
Nathusius pipistrelle	<i>Pipistrellus nathusii</i>	299	2000	2022		127
Common pipistrelle	<i>Pipistrellus pipistrellus</i>	6395	1994	2023		1222
Soprano pip	<i>Pipistrellus pygmaeus</i>	2640	1984	2023		587
Long-eared bats	<i>Plecotus gen.</i>	3813	1980	2023		598
Brown long-eared	<i>Plecotus auritus</i>	2931	1980	2023		464
Grey long-eared	<i>Plecotus austriacus</i>	2	1985	1995		0
Horseshoe bats	<i>Rhinolophus gen.</i>	6	2018	2020		6
Greater horseshoe	<i>Rhinolophus ferrumequinum</i>	6	2018	2020		6
Lesser horseshoe	<i>Rhinolophus hipposideros</i>	0	-	-		0

3.1 Effects of landscape composition

Odds ratios were calculated from the log-odds estimates returned by the GLMMs. The odds ratio (OR) represents the relative odds of presence given a 1 s.d. increase in land cover from the mean (0 in all cases, due to scaling), and is calculated as

Odds Ratio = $e^{\text{log-odds}}$. OR > 1 represents a relative increase in odds, whilst OR < 1 represents a decrease in odds (Norton et al., 2018).

Models were not fitted for *B. barbastellus*, *M. alcathoe*, *M. myotis*, *P. kuhlii*, *P. austriacus*, or *R. ferrumequinum* due to lack of records.

3.1.1 Genera

Within the full dataset (1980-2023, Model 1), higher proportions of deciduous woodland and freshwater significantly affected the occurrence of most genera (Fig. 5). Deciduous woodland cover had significant effects on four of five genera: positive on *Myotis* (OR=1.51, 95%CI [1.33, 1.70]), *Eptesicus* (OR=1.67, 95%CI [1.31, 2.12]), and *Nyctalus* (OR=1.32, 95%CI [1.11, 1.56]); and negative on *Pipistrellus* (OR=0.75, 95%CI [0.67, 0.84]). Higher freshwater cover had significant positive effects in *Myotis* (OR=1.45, 95%CI [1.35, 1.56]) and *Nyctalus* (OR=1.31, 95%CI [1.21, 1.41]), but it negatively affected *Plecotus* (OR=0.73, 95%CI [0.66, 0.81]). All other non-built up land cover variables (arable/improved grassland, coastal, and semi-natural grassland) were significant for 1-2 genera, but did not show any generalisable effects (Fig. 5; Supplementary Material, Table S3).

In all genera except *Plecotus* increasing urban area had a positive effect on the odds of species presence, though this effect was significant in *Myotis* (OR=1.15, 95%CI [1.05,

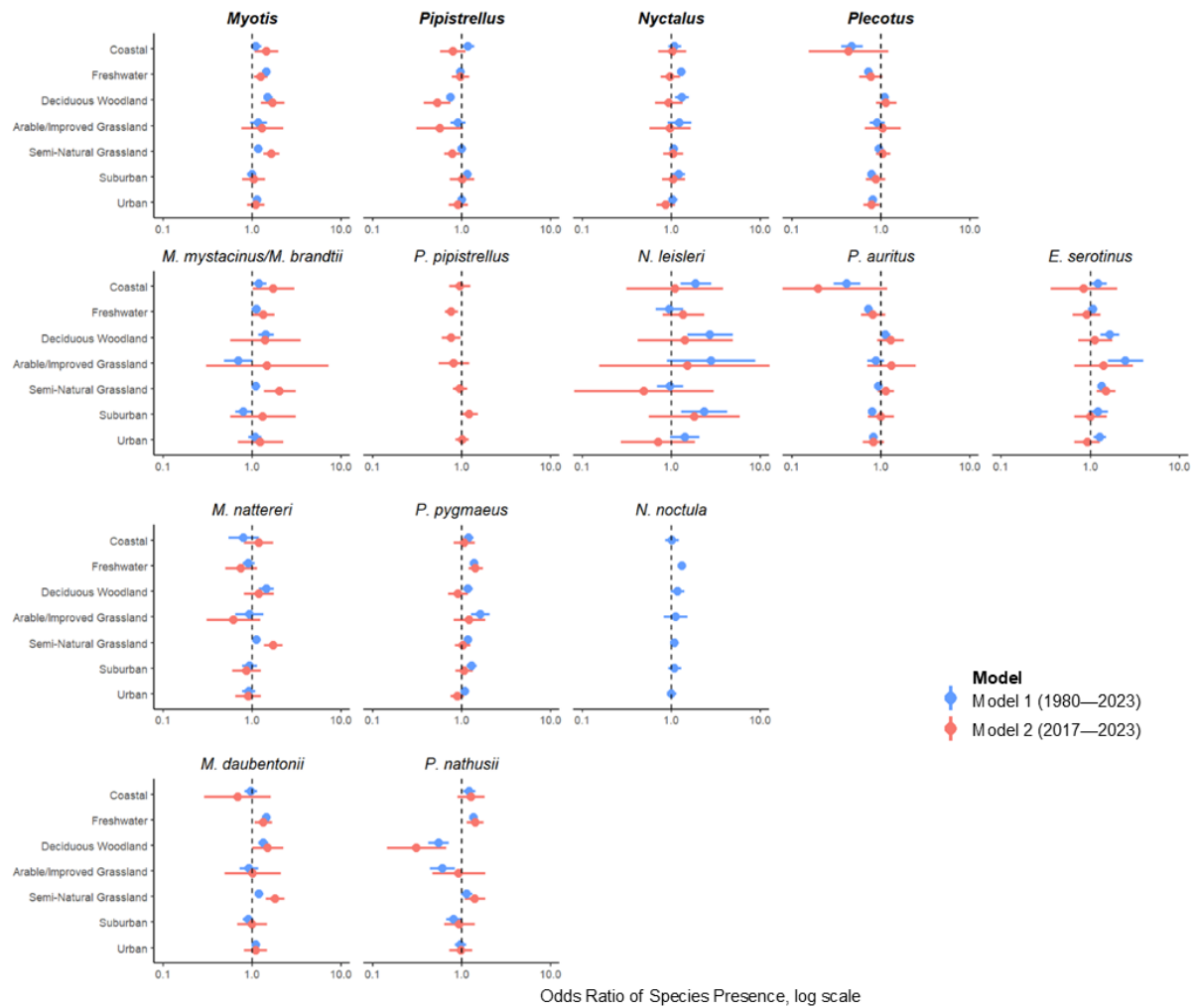


Figure 5: Effects of landscape composition variables on the odds ratio of genera/species presence. Line gives Wald 95% confidence interval (where large, these may extend beyond the axes shown). Effects are significant when CIs do not cross 1.

1.25]) and *Eptesicus* (OR=1.27, 95%CI [1.08, 1.50]) only (Fig. 5; Supplementary Material, Table S3). Greater suburban area had significant positive effects in *Pipistrellus* (OR=1.16, 95%CI [1.05, 1.29]) and *Nyctalus* (OR=1.21, 95%CI [1.03, 1.42]). It had a significant negative effect in *Plecotus* (OR=0.79, 95% CI [0.71, 0.88]).

3.1.2 Species

Mirroring the genera findings, deciduous woodland was significant in six of eight species. *Myotis* spp. consistently showed significant positive effects, but there was a difference in

effect direction in *Pipistrellus spp.* between *P. pygmaeus* (OR=1.18, 95%CI [1.03, 1.35]) and *P. nathusii* (OR=0.55, 95%CI [0.42, 0.72]). Freshwater had a significant negative effect in *P. auritus* (OR=0.73, 95%CI [0.65, 0.82]), and significant positive effects in *N. noctule*, *P. nathusii*, *P. pygmaeus* and *M. daubentonii* (Fig. 5; Supplementary Material, Table S3). Coastal, semi-natural, and arable/improved grassland were again significant in fewer than half of the species.

In contrast with the genera results, urban area was only significant in *P. auritus*, where it had a negative effect (OR=0.82, 95%CI [0.75, 0.91]). Suburban area also had negative effects in *P. auritus* (OR=0.80, 95%CI [0.71, 0.90]), as well as *P. nathusii* (OR=0.81, 95%CI [0.67, 0.98]) (Fig. 5; Supplementary Material, Table S3). Once again, *Myotis spp.* all showed the same effect direction, whilst *Pipistrellus spp.* showed divergent responses, as *P. pygmaeus* was positively affected by suburban area (OR=1.30, 95%CI [1.14, 1.48]).

3.2 Reduced Time Period

Compared to the full dataset, the range of CIs increased for all genera/species and all variables in Model 2. For *N. noctula* and *P. nathusii*, the reduction in data meant the model had insufficient information to converge, however in *P. pipistrellus* model convergence was achieved with the subset.

Effect directions were the same or, where they shifted, CIs overlapped both 1 and the estimate of Model 1 (e.g. the effect of freshwater on odds of presence of *Eptesicus*) (Fig. 5; Supplementary Material, Table S3). Confidence intervals of Models 1 and 2 overlapped in all cases except for the effect of semi-natural grassland on *Myotis* and the three *Myotis*

Table 3: Akaike Information Criterion (AIC) values across models 2-5 for all genera and species. Lowest AIC value is highlighted in light green, second lowest in orange. Change in AIC given is between lowest value and second-lowest value.

Genus/species	AIC				ΔAIC
	Model 2	Model 3	Model 4	Model 5	
<i>Pipistrellus</i>	1066.18	1066.74	1069.90	1073.17	-0.56
<i>P. pipistrellus</i>	1354.68	-	1357.90	1361.69	-3.22
<i>P. pygmaeus</i>	1219.86	1228.67	1219.36	1219.21	-0.15
<i>Myotis</i>	1063.77	1065.34	1062.36	1064.89	-1.42
<i>M. mysticanus/brandtii</i>	219.03	220.88	218.13	220.54	-0.90
<i>M. nattereri</i>	575.89	578.47	579.45	583.02	-2.57
<i>M. daubentonii</i>	751.76	751.87	754.07	754.51	-0.11
<i>Plecotus</i>	1390.48	1401.48	1404.14	1404.96	-11.00
<i>P. auritus</i>	1277.39	1284.82	1287.36	1288.03	-7.43
<i>Nyctalus</i>	738.01	738.59	738.76	740.73	-0.58
<i>N. leisler</i>	231.45	231.89	233.26	237.26	-0.44
<i>Eptesicus</i>	683.92	684.29	685.49	687.26	-0.37

species, where all three species and the genus showed an increased positive effect in Model 2 over Model 1 (Fig. 5; Supplementary Material, Table S3).

3.3 Model Comparison

In four of five genera and six of eight species, AIC was lowest in Model 2, which included no urban growth variables (Table 3). For *Myotis* and *M. mysticanus/brandtii*, Model 4 (including both urban growth and growth pattern) had the best fit, whilst for *P. pygmaeus* inclusion of the interaction between urban growth and growth pattern (Model 5) led to a better fit. Model 3 (urban growth included but no growth pattern) was not the best fitting model for any genera or species.

3.4 Effects of urban growth

The urban growth variable was significant in only one case, for *Myotis* in Model 5, which was not the best fitting model for this genus and the upper CI was only marginally different from 1 (OR:0.78, 95%CI [0.61, 0.99(7)]) (Supplementary Materials, Table S3). In all cases where a model with urban growth variables was the best-fitting, the infilling pattern had a negative effect on odds of presence compared to the reference case of edge expansion (*Myotis*: OR=0.56, 95%CI [0.37, 0.87]; *M.mystacinus/brandtii*: OR=0.29, 95%CI [0.09, 0.89]; *P. pygmaeus*: OR=0.58, 95%CI [0.41, 0.83]) (Table 4).

Table 4: Odds ratios, 2.5% CI (LCI) and 97.5% CI (UCI) for the urban-associated variables of the best-fitting model for each genera and species. Missing values indicate where a variable was not included in the best-fitting model. Model 3 was not the best-fitting model for any genera or species.

Genus/Species	Urban Extent						Suburban Extent						Urban Growth						Dominant Growth Pattern						Urban Growth x Dominant Growth Pattern					
	Urban Extent			Suburban Extent			Urban Growth			Infilling			Outlying			Urban Growth			Infilling			Outlying								
	OR	LCI	UCI	OR	LCI	UCI	OR	LCI	UCI	OR	LCI	UCI	OR	LCI	UCI	OR	LCI	UCI	OR	LCI	UCI	OR	LCI	UCI						
Model 2																														
<i>E. serotinus</i>	0.92	0.67	1.28	1.01	0.66	1.54	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>M. nattereri</i>	0.90	0.65	1.26	0.87	0.60	1.26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>M. daubentonii</i>	1.10	0.81	1.48	1.00	0.68	1.47	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>Plecotus</i>	0.79	0.64	0.98	0.88	0.68	1.13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>P. auritus</i>	0.83	0.63	1.08	1.00	0.72	1.40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>Pipistrellus</i>	0.92	0.72	1.18	1.01	0.73	1.39	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>P. pipistrellus</i>	1.02	0.86	1.21	1.22	0.98	1.52	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>P. nathusi</i>	0.98	0.73	1.32	0.95	0.64	1.41	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>Nyctalus</i>	0.87	0.68	1.11	1.06	0.79	1.42	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
<i>N. leisler</i>	0.71	0.27	1.86	1.83	0.57	5.92	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						
Model 4																														
<i>Myotis</i>	1.12	0.89	1.41	1.03	0.76	1.40	0.85	0.70	1.03	0.57	0.37	0.87	1.26	0.40	4.02	-	-	-	-	-	-	-	-	-						
<i>M.mysticanus/brandtii</i>	1.25	0.68	2.29	1.37	0.59	3.20	0.44	0.17	1.12	0.29	0.10	0.89	4.81	0.49	47.31	-	-	-	-	-	-	-	-	-						
Model 5																														
<i>P. pygmaeus</i>	0.87	0.72	1.04	1.00	0.79	1.27	0.96	0.81	1.13	0.58	0.41	0.83	0.29	0.05	1.62	1.49	0.99	2.24	1.02	0.66	1.56									

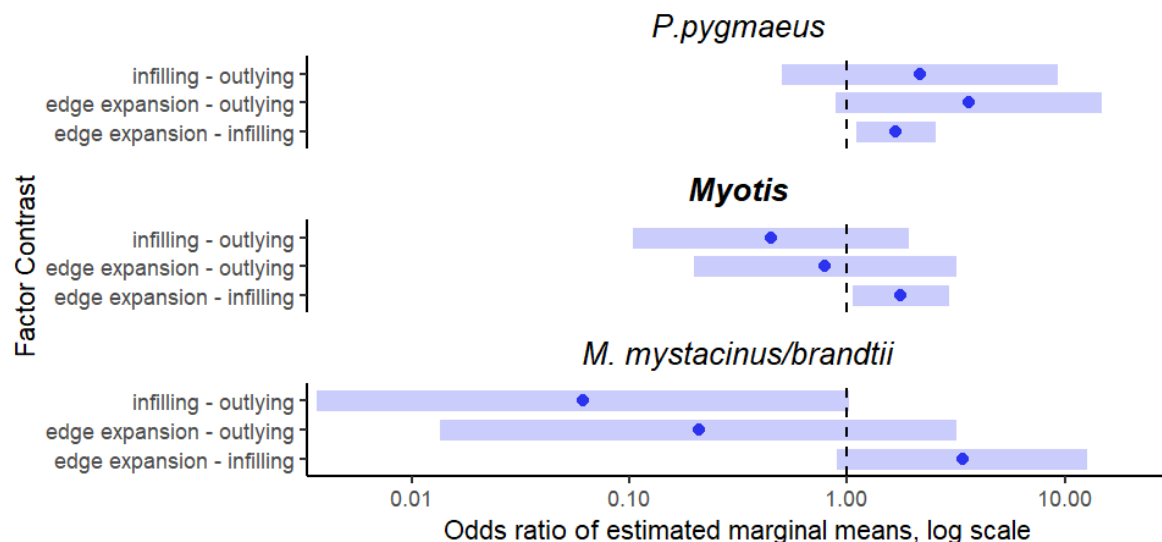


Figure 6: Pairwise comparisons of estimated marginal means for 'Dominant Growth Pattern' in Model 4. Odds ratios are displayed with the second factor as the reference case, so values > 1 indicate that there is a positive effect of the first over the second. Note that this contrasts with Table 4

Post-hoc pairwise comparison of the estimated marginal means of the dominant growth pattern variable using the R package `emmeans` (Lenth, 2024) confirmed significance of the difference in effect between infilling and edge expansion in *P. pygmaeus* and *Myotis*, but in *M. mystacinus/brandtii* the effect was no longer significant (Fig. 6). This reduction in significance is due to the adjustment made to significance values during post-hoc tests to counteract the increased likelihood of finding a statistically significant relationship when multiple comparisons are made (the multiple comparisons problem, Jeffries, 2007). The adjustment made here is a variation of Tukey's Honest Significant Difference (Keselman and Rogan, 1977). All pairwise comparison was made without interactive effects due to the size of the data available. For *P. pygmaeus*, this means post-hoc comparison was not made on the best-fitting model (Model 5). This was considered to still be informative as the ΔAIC between Model 4 and 5 for this species was only -0.15, meaning Model 5 provides only a marginally better fit.

4. DISCUSSION

A long-term citizen science dataset was used for the first time to investigate the historic impacts of urban expansion and densification on bats on a regional scale in Kent, UK. Although 17 of 18 British bat species had been recorded within the region, low observation rates meant it was only possible to investigate the effects of land cover in ten species (and four genera), and of urban growth in nine (four genera). Where the effect of urban growth was significant on odds of presence of bat species, densification (through infilling growth) had a negative effect in comparison with urban sprawl (through edge expansion).

4.1 Landscape Composition

4.1.1 Non-urban land cover

Patterns of association with non-urban landscape composition variables in Models 1 and 2 were largely as expected, with deciduous woodland emerging as a positive landscape characteristic for all except *Pipistrellus*, *P. pipistrellus* and *P. nathusii*. This is consistent with the finding of Border et al. (2017) that across 11 species, presence and activity was positively associated with broadleaved (deciduous) woodland for all except *P. pipistrellus* and *P. nathusius*. However, both studies use the same land cover dataset, which has a

lower detection success as deciduous woodland becomes more sparse (tree cover < 50%) (Marston et al., 2023). Given Fuentes-Montemayor et al.'s (2013) finding that aerial hawkers such as *Pipistrellus* are highly active in low density woodland, whilst gleaning species (e.g. *Myotis*) tend to prefer high densities of cover, misclassification of low density deciduous woodland as other land cover classes may be obscuring the importance of deciduous woodland in *Pipistrellus* spp.. Effects of freshwater are also in line with expected patterns given preferred diets (Vaughan, 1997), and with findings of other studies (Border et al., 2017; Gili et al., 2020). Whilst previous evidence gives no reason to doubt the freshwater results of this study, caution should be taken over their interpretation due to the method of absence generation. The method of absence generation used in this study did not account for survey type, but detection rates of species are not evenly distributed across these. Notably, the Bat Conservation Trust's Daubenton's Waterway Survey (Barlow et al., 2015; Bat Conservation Trust, 2023; Langton et al., 2010) specifically targets *M. daubentonii* at rivers. Generating absences for other species from these data is likely to bias against finding positive effects of freshwater. Preliminary analysis of the 'comments' column of the observations dataset suggests a large number of observations (n > 3028) include survey type information not otherwise recorded. Extraction of this information would allow for survey-specific detection probabilities to be modelled and achieve a more accurate occupancy matrix, in a variation of the methodology presented in Outhwaite et al. (2018).

4.1.2 Urban and Suburban Landcover

The findings of Model 1 support the first hypothesis (that urban and suburban area will have an effect on presence) in three of five genera and five of eight species for suburban

area; and three of five genera and one of eight species for urban area. The positive effect of increasing urban area on presence of *E. serotinus* is consistent with existing understanding of availability of its prey, roost preferences, and urban tolerance (Kervyn & Libois, 2008; Lesiński et al., 2000; Starik et al., 2024; Vlaschenko et al., 2023). Similarly, the negative effect of urban area on presence of *Plecotus* (expected to be predominantly *P. auritus*) is aligned with expectations given its avoidance of artificial lighting and its low amplitude echolocation, which is more susceptible to interference from elevated background noise levels commonly found in urban environments (de Framond et al., 2023; Reusch et al., 2024; Schaub et al., 2008). The positive effect of increased urban coverage on *Myotis* is in contrast with prior research finding negative associations between presence and human population density (Border et al., 2017), impervious surface (Gili et al., 2020), and urban density (Starik et al., 2024). It is possible that these findings are attributable to spatial bias of observations towards urban areas, though absences being generated from the same distribution should reduce this effect (Dickinson et al., 2010). Alternatively, this difference may be driven by compositional differences between Kent and other study locations. Border et al. (Border et al., 2017) use a 1 km² radius for their measure of human population density, whilst Starik et al. (Starik et al., 2024) focuses on the single rural-urban gradient moving outwards from central Berlin. Kent's largest settlement (Maidstone) is significantly smaller than Berlin (human population: 184,200 vs 3.6 million), suggesting either that a 3 km radius is insufficient to detect avoidance of *Myotis* from urban areas, or that the bats are able to utilise these spaces so long as they form part of a mosaic of habitat. The latter is supported by Lintott et al.'s (2015) finding that *M. daubentonii* is able to exploit channelised urban waterways provided the surrounding urban landscape is sufficiently fragmented.

4.2 Urban Growth and Growth Pattern

Whilst no evidence was found to support hypothesis 2 (that urban growth has an effect on presence), findings for *P. pygmaeus* and *Myotis* did support hypothesis 3 (that the effect of urban growth is conditional on the pattern of that growth). In both cases, a dominant growth pattern of infilling was more negatively associated with presence than edge expansion, though different mechanisms are potentially at play. In *P. pygmaeus* this effect supports the theory that species positively associated with suburban cover will be best supported by an avoiding densification of the landscape by reducing the homogenisation caused by infilling; whilst in *Myotis* it gives further support to the importance of total urban area size, as individuals may be utilising areas of less developed land within urban patches, which are then removed with infilling growth. The effect of outlying growth remains unclear following this study. This may be due to insufficient classification of types and/or size of outlying developments. Using growth classifications which quantify this, such as the Multi-Order Landscape Expansion Index (Jiao et al., 2015) or the Proximity Expansion Index (Jiao et al., 2018) may improve understanding of the effects of outlying development.

4.3 Limitations and Recommendations

Modelling of *P. pipistrellus* through presence data was limited by its ubiquity within the county, an issue also encountered in Border et al. (2017). Activity or count data would be better placed to model this species, which is available for some of the observations within the dataset. The temporal spread of records with this information is unknown, though the increased use and affordability of both heterodyne detectors and passive acoustic

monitoring over time leads to expectations that this information increases. As well as count data, there is a significant amount of standardisable information recorded within 'comment' columns, notably: a) records with survey type identified to a greater specificity than recorded in the main dataframe ($n > 3,028$), b) observation-specific environmental variables, such as temperature, cloud cover, and wind levels ($n > 2,707$). It is likely this pattern of record holding is broadly replicated across regional- and taxon-specific datasets held by small conservation groups, particularly those with long histories of recording. This presents a clear opportunity for increasing data availability, particularly of historic information.

Temporal coverage of the study was limited by the availability of suitable landcover data. The UKCEH UK-SCAPE programme now produce a land cover map annually, and have applied the same methodology to back date comparable land cover maps (the 1990 data used in this study)(UKCEH, [2021](#)). An extension of this work to increase the temporal coverage of the period 1990 - 2015 would be a beneficial resource to extend the work of this study and those similar. Such data would also allow for better change detection and mitigation of the effects of year-to-year misclassifications. Spatial broadening of this study would also increase power of the sample sizes, as well as providing more variation in urban-rural gradient.

5. CONCLUSION

By analysing the effect of urban and suburban area on bats in Kent over a period of 43 years, this study showed that observed trends of habitat preference across species are on the whole reflected in Kent. A notable exception to this is in the positive effect of the most dense urban areas on *Myotis*, suggesting that this genus's avoidance of towns and cities is dependent on the overall size of this area. The findings of the study support the protection of green spaces, tree cover and riparian habitats throughout urban development, as opposed to absolute minimisation of land cover change through extreme densification (infilling). However, this study was not able to draw conclusions about all British bat species, which may have diverging needs and would benefit from further investigation. The wealth of information uncovered during this study's investigation of a previously unexamined dataset points to the benefit of exploiting the rich resources held by regional groups of citizen scientists and conservation workers, which have historically been overlooked.

REFERENCES

- Barlow, K. E., Briggs, P. A., Haysom, K. A., Hutson, A. M., Lechiara, N. L., Racey, P. A., Walsh, A. L., & Langton, S. D. (2015). Citizen science reveals trends in bat populations: The National Bat Monitoring Programme in Great Britain. *Biological Conservation*, 182, 14–26. <https://doi.org/10.1016/j.biocon.2014.11.022>
- Bat Conservation Trust. (2016). Core Sustainance Zones Explained. Retrieved August 22, 2024, from https://cdn.bats.org.uk/uploads/pdf/Resources/Core_Sustainance_Zones_Explained_04.02.16.pdf?v=1550597495&_gl=1*1f5ujyd*_ga*MTAwMDI1NjY3Ny4xNzI0MzQ0MTEx*_ga_G28378TB9V*MTcyNDM0NDExMC4xLjAuMTcyNDM0NDExMC4wLjAuMA..
- Bat Conservation Trust. (2023). *The National Bat Monitoring Programme Annual Report 2023* (tech. rep.). Bat Conservation Trust. London. Retrieved August 21, 2024, from www.bats.org.uk/our-work/national-bat-monitoring-programme/reports/nbmp-annual-report
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67, 1–48. <https://doi.org/10.18637/jss.v067.i01>

- Battisti, C., & Fanelli, G. (2016). Applying indicators of disturbance from plant ecology to vertebrates: The hemeroby of bird species [Num Pages: 7 Patent Number: 2 Place: Amsterdam Publisher: Elsevier Science Bv Web of Science ID: WOS:000367411200061]. *ECOLOGICAL INDICATORS*, 61, 799–805. <https://doi.org/10.1016/j.ecolind.2015.10.032>
- Battisti, C., Fanelli, G., Mariani, L., & Capizzi, D. (2017). Assessing disturbance-sensitivity and generalism in mammals: Corroborating a hump-shaped relationship using a hemerobiotic approach. *Ecological Indicators*, 76, 178–183. <https://doi.org/10.1016/j.ecolind.2017.01.014>
- Baumgardt, J. A., Morrison, M. L., Brennan, L. A., Davis, H. T., Fern, R. R., Szewczak, J. M., & Campbell, T. A. (2022). Monitoring occupancy of bats with acoustic data: Power and sample size recommendations [Num Pages: 14 Place: Provo Publisher: Brigham Young Univ Web of Science ID: WOS:000895811400004]. *WESTERN NORTH AMERICAN NATURALIST*, 82(1), 36–49. Retrieved August 20, 2024, from <https://www.webofscience.com/wos/woscc/full-record/WOS:000895811400004>
- Border, J. A., Newson, S. E., White, D. C. J., & Gillings, S. (2017). Predicting the likely impact of urbanisation on bat populations using citizen science data, a case study for Norfolk, UK [Num Pages: 12 Place: Amsterdam Publisher: Elsevier Web of Science ID: WOS:000399627800005]. *LANDSCAPE AND URBAN PLANNING*, 162, 44–55. <https://doi.org/10.1016/j.landurbplan.2017.02.005>
- Browning, E., Barlow, K. E., Burns, F., Hawkins, C., & Boughey, K. (2021). Drivers of European bat population change: A review reveals evidence gaps [Num Pages: 16

Place: Hoboken Publisher: Wiley Web of Science ID: WOS:000619808000001].

MAMMAL REVIEW, 51(3), 353–368. <https://doi.org/10.1111/mam.12239>

Callaghan, C. T., Bino, G., Major, R. E., Martin, J. M., Lyons, M. B., & Kingsford, R. T.

(2019). Heterogeneous urban green areas are bird diversity hotspots: Insights

using continental-scale citizen science data [Num Pages: 16 Place: Dordrecht

Publisher: Springer Web of Science ID: WOS:000475643000003]. *LANDSCAPE*

ECOLOGY, 34(6), 1231–1246. <https://doi.org/10.1007/s10980-019-00851-6>

Concepción, E. D., Obrist, M. K., Moretti, M., Altermatt, F., Baur, B., & Nobis, M. P. (2016).

Impacts of urban sprawl on species richness of plants, butterflies, gastropods and

birds: Not only built-up area matters. *Urban Ecosystems*, 19(1), 225–242.

<https://doi.org/10.1007/s11252-015-0474-4>

Connell, J. H. (1978). Diversity in Tropical Rain Forests and Coral Reefs [Publisher:

American Association for the Advancement of Science]. *Science*, 199(4335),

1302–1310. <https://doi.org/10.1126/science.199.4335.1302>

De Conno, C., Nardone, V., Ancillotto, L., De Bonis, S., Guida, M., Jorge, I., Scarpa, U., &

Russo, D. (2018). Testing the performance of bats as indicators of riverine

ecosystem quality. *Ecological Indicators*, 95, 741–750.

<https://doi.org/10.1016/j.ecolind.2018.08.018>

de Framond, L., Beleyur, T., Lewanzik, D., & Goerlitz, H. R. (2023). Calibrated microphone

array recordings reveal that a gleaning bat emits low-intensity echolocation calls

even in open-space habitat [Num Pages: 6 Place: Cambridge Publisher: Company

Biologists Ltd Web of Science ID: WOS:001149098100006]. *JOURNAL OF*

EXPERIMENTAL BIOLOGY, 226(18), jeb245801.

<https://doi.org/10.1242/jeb.245801>

- Dickinson, J. L., Zuckerberg, B., & Bonter, D. N. (2010). Citizen Science as an Ecological Research Tool: Challenges and Benefits [Publisher: Annual Reviews]. *Annual Review of Ecology, Evolution, and Systematics*, 41(Volume 41, 2010), 149–172. <https://doi.org/10.1146/annurev-ecolsys-102209-144636>
- Fleming, G. M., Ramsay, E. E., Trias Jaya Putra Nurdin, M. R., Duffy, G. A., Faber, P. A., & Chown, S. L. (2024). Sensitivity of insectivorous bat foraging guilds to urbanization and implications for sustainable development. *Global Ecology and Conservation*, 51, e02929. <https://doi.org/10.1016/j.gecco.2024.e02929>
- Fuentes-Montemayor, E., Goulson, D., Cavin, L., Wallace, J. M., & Park, K. J. (2013). Fragmented woodlands in agricultural landscapes: The influence of woodland character and landscape context on bats and their insect prey. *Agriculture, Ecosystems & Environment*, 172, 6–15. <https://doi.org/10.1016/j.agee.2013.03.019>
- Fuller, R., Smith, G., Sanderson, J., Hill, R., Thomson, A., Cox, R., Brown, N., Clarke, R., Rothery, P., & Gerard, F. (2002). Land Cover Map 2000 (25m raster, GB). <https://doi.org/10.5285/F802EDFC-86B7-4AB9-B8FA-87E9135237C9>
- Gibb, R., Browning, E., Glover-Kapfer, P., & Jones, K. E. (2019). Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring [Num Pages: 17 Place: Hoboken Publisher: Wiley Web of Science ID: WOS:000459020800002]. *METHODS IN ECOLOGY AND EVOLUTION*, 10(2), 169–185. <https://doi.org/10.1111/2041-210X.13101>
- Gili, F., Newson, S. E., Gillings, S., Chamberlain, D. E., & Border, J. A. (2020). Bats in urbanising landscapes: Habitat selection and recommendations for a sustainable future [Num Pages: 10 Place: Oxford Publisher: Elsevier Sci Ltd Web of Science

ID: WOS:000518695100070]. *BIOLOGICAL CONSERVATION*, 241, 108343.

<https://doi.org/10.1016/j.biocon.2019.108343>

Hale, J. D., Fairbrass, A. J., Matthews, T. J., & Sadler, J. P. (2012). Habitat Composition and Connectivity Predicts Bat Presence and Activity at Foraging Sites in a Large UK Conurbation [Publisher: Public Library of Science]. *PLOS ONE*, 7(3), e33300.

<https://doi.org/10.1371/journal.pone.0033300>

Hijmans, R. (2024). Terra: Spatial Data Analysis.

<https://CRAN.R-project.org/package=terra>

IPBES. (2019, November). *Summary for policymakers of the global assessment report on biodiversity and ecosystem services* (tech. rep.). Zenodo.

<https://doi.org/10.5281/zenodo.3553579>

Isbell, F., Balvanera, P., Mori, A. S., He, J.-S., Bullock, J. M., Regmi, G. R., Seabloom, E. W., Ferrier, S., Sala, O. E., Guerrero-Ramírez, N. R., Tavella, J., Larkin, D. J., Schmid, B., Outhwaite, C. L., Pramual, P., Borer, E. T., Loreau, M., Omotoriogun, T. C., Obura, D. O., . . . Palmer, M. S. (2023). Expert perspectives on global biodiversity loss and its drivers and impacts on people [eprint:

<https://onlinelibrary.wiley.com/doi/pdf/10.1002/fee.2536>]. *Frontiers in Ecology and the Environment*, 21(2), 94–103. <https://doi.org/10.1002/fee.2536>

Jeffries, N. O. (2007). Multiple comparisons distortions of parameter estimates.

Biostatistics, 8(2), 500–504. <https://doi.org/10.1093/biostatistics/kxl025>

Jiao, L., Liu, J., Xu, G., Dong, T., Gu, Y., Zhang, B., Liu, Y., & Liu, X. (2018). Proximity Expansion Index: An improved approach to characterize evolution process of urban expansion. *Computers, Environment and Urban Systems*, 70, 102–112.

<https://doi.org/10.1016/j.compenvurbsys.2018.02.005>

- Jiao, L., Mao, L., & Liu, Y. (2015). Multi-order Landscape Expansion Index: Characterizing urban expansion dynamics [Num Pages: 10 Place: Amsterdam Publisher: Elsevier Science Bv Web of Science ID: WOS:000351647700004]. *LANDSCAPE AND URBAN PLANNING*, 137, 30–39. <https://doi.org/10.1016/j.landurbplan.2014.10.023>
- Jones, G., Jacobs, D., Kunz, T., Willig, M., & Racey, P. (2009). Carpe noctem: The importance of bats as bioindicators. *Endangered Species Research*, 8, 93–115. <https://doi.org/10.3354/esr00182>
- Jung, K., & Threlfall, C. G. (2018). Trait-dependent tolerance of bats to urbanization: A global meta-analysis [Num Pages: 9 Place: London Publisher: Royal Soc Web of Science ID: WOS:000443163500022]. *PROCEEDINGS OF THE ROYAL SOCIETY B-BIOLOGICAL SCIENCES*, 285(1885), 20181222. <https://doi.org/10.1098/rspb.2018.1222>
- Kervyn, T., & Libois, R. (2008). The Diet of the serotine bat A Comparison between rural and urban environments. *Belgian Journal of Zoology*, 138.
- Keselman, H. J., & Rogan, J. C. (1977). The Tukey multiple comparison test: 1953–1976 [Place: US Publisher: American Psychological Association]. *Psychological Bulletin*, 84(5), 1050–1056. <https://doi.org/10.1037/0033-2909.84.5.1050>
- Kroll, C., Warchold, A., & Pradhan, P. (2019). Sustainable Development Goals (SDGs): Are we successful in turning trade-offs into synergies? [Publisher: Palgrave]. *Palgrave Communications*, 5(1), 1–11. <https://doi.org/10.1057/s41599-019-0335-5>
- Langton, S. D., Briggs, P. A., & Haysom, K. A. (2010). Daubenton's bat distribution along rivers - developing and testing a predictive model [Num Pages: 10 Place: Hoboken Publisher: Wiley Web of Science ID: WOS:000277779500005]. *AQUATIC*

CONSERVATION-MARINE AND FRESHWATER ECOSYSTEMS, 20, S45–S54.

<https://doi.org/10.1002/aqc.1077>

Lato, K. A., Furst, M., Veit, R. R., & Thorne, L. H. (2023). Peri-urban systems alter trophic niche size and overlap in sympatric coastal bird species [Num Pages: 16 Place: Hoboken Publisher: Wiley Web of Science ID: WOS:001046095100001].

ECOSPHERE, 14(8), e4643. <https://doi.org/10.1002/ecs2.4643>

Lenth, R. (2024). Emmeans: Estimated Marginal Means, aka Least-Squares Means.

<https://CRAN.R-project.org/package=emmeans>

Lesiński, G., Fuszara, E., & Kowalski, M. (2000). Foraging areas and relative density of bats (Chiroptera) in differently human transformed landscapes. *Zeitschrift fur Saugetierkunde*, 65, 129–137.

Li, G., Fang, C., Li, Y., Wang, Z., Sun, S., He, S., Qi, W., Bao, C., Ma, H., Fan, Y., Feng, Y., & Liu, X. (2022). Global impacts of future urban expansion on terrestrial vertebrate diversity [Publisher: Nature Publishing Group]. *Nature Communications*, 13(1), 1628. <https://doi.org/10.1038/s41467-022-29324-2>

Lintott, P. R., Barlow, K., Bunnefeld, N., Briggs, P., Gajas Roig, C., & Park, K. J. (2016).

Differential responses of cryptic bat species to the urban landscape. *Ecology and Evolution*, 6(7), 2044–2052. <https://doi.org/10.1002/ece3.1996>

Lintott, P. R., Bunnefeld, N., & Park, K. J. (2015). Opportunities for improving the foraging potential of urban waterways for bats. *Biological Conservation*, 191, 224–233.

<https://doi.org/10.1016/j.biocon.2015.06.036>

Liu, X., Li, X., Chen, Y., Tan, Z., Li, S., & Ai, B. (2010). A new landscape index for quantifying urban expansion using multi-temporal remotely sensed data [Num Pages: 12 Place: Dordrecht Publisher: Springer Web of Science ID:

WOS:000276609800002]. *LANDSCAPE ECOLOGY*, 25(5), 671–682.

<https://doi.org/10.1007/s10980-010-9454-5>

Marston, C. G., O'Neil, A. W., Morton, R. D., Wood, C. M., & Rowland, C. S. (2023).

LCM2021 – the UK Land Cover Map 2021 [Publisher: Copernicus GmbH]. *Earth System Science Data*, 15(10), 4631–4649.

<https://doi.org/10.5194/essd-15-4631-2023>

Martin, P. R., & Bonier, F. (2018). Species interactions limit the occurrence of

urban-adapted birds in cities [Num Pages: 10 Place: Washington Publisher: Natl

Acad Sciences Web of Science ID: WOS:000452124700011]. *PROCEEDINGS OF*

THE NATIONAL ACADEMY OF SCIENCES OF THE UNITED STATES OF

AMERICA, 115(49), E11495–E11504. <https://doi.org/10.1073/pnas.1809317115>

McDonald, R. I., Mansur, A. V., Ascensão, F., Colbert, M., Crossman, K., Elmqvist, T.,

Gonzalez, A., Güneralp, B., Haase, D., Hamann, M., Hillel, O., Huang, K.,

Kahnt, B., Maddox, D., Pacheco, A., Pereira, H. M., Seto, K. C., Simkin, R.,

Walsh, B., ... Ziter, C. (2020). Research gaps in knowledge of the impact of urban

growth on biodiversity [Publisher: Nature Publishing Group]. *Nature Sustainability*,

3(1), 16–24. <https://doi.org/10.1038/s41893-019-0436-6>

McKinney, M. L. (2008). Effects of urbanization on species richness: A review of plants

and animals. *Urban Ecosystems*, 11(2), 161–176.

<https://doi.org/10.1007/s11252-007-0045-4>

Morton, R., Rowland, C. S., Wood, C. M., Meek, L., Marston, C. G., & Smith, G. (2014).

Land Cover Map 2007 (25m raster, GB) v1.2. <https://doi.org/https://doi.org/10.5285/A1F88807-4826-44BC-994D-A902DA5119C2>

[//doi.org/10.5285/A1F88807-4826-44BC-994D-A902DA5119C2](https://doi.org/10.5285/A1F88807-4826-44BC-994D-A902DA5119C2)

Morton, R. D., Marston, C. G., O'Neil, A. W., & Rowland, C. S. (2020a). Land Cover Map
2017 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/499212cd-d64a-43ba-b801-95402e4d4098>

Morton, R. D., Marston, C. G., O'Neil, A. W., & Rowland, C. S. (2020b). Land Cover Map
2018 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/25c6451b-5c88-40da-9a63-c3ec473e4874>

Morton, R. D., Marston, C. G., O'Neil, A. W., & Rowland, C. S. (2020c). Land Cover Map
2019 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/f15289da-6424-4a5e-bd92-48c4d9c830cc>

Morton, R. D., Marston, C. G., O'Neil, A. W., & Rowland, C. S. (2021). Land Cover Map
2020 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/6c22cf6e-b224-414e-aa85-900325baedbd>

Morton, R. D., Marston, C. G., O'Neil, A. W., & Rowland, C. S. (2024a). Land Cover Map
2022 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/c9449bf5-b8f6-4a1c-b3eb-0d70575cba39>

Morton, R. D., Marston, C. G., O'Neil, A. W., & Rowland, C. S. (2024b). Land Cover Map
2023 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/ab10ea4a-1788-4d25-a6df-f1aff829dfff>

Morton, R. D., O'Neil, A. W., Rowland, C. S., & Marston, C. G. (2022). Land Cover Map
2021 (25m rasterised land parcels, GB).

<https://doi.org/https://doi.org/10.5285/a1f85307-cad7-4e32-a445-84410efdfa70>

Newson, S. E., Evans, H. E., & Gillings, S. (2015). A novel citizen science approach for
large-scale standardised monitoring of bat activity and distribution, evaluated in

682 eastern England. *Biological Conservation*, 191, 38–49.

683 <https://doi.org/10.1016/j.biocon.2015.06.009>

684 Norton, E. C., Dowd, B. E., & Maciejewski, M. L. (2018). Odds Ratios—Current Best

685 Practice and Use. *JAMA*, 320(1), 84. <https://doi.org/10.1001/jama.2018.6971>

686 Office for National Statistics. (2023). *Towns and cities, characteristics of built-up areas,*

687 *England and Wales* (tech. rep.). Retrieved August 22, 2024, from

688 <https://www.ons.gov.uk/peoplepopulationandcommunity/housing/articles/>

689 [townsandcitiescharacteristicsofbuiltupareasenglandandwales/census2021](https://www.ons.gov.uk/peoplepopulationandcommunity/housing/articles/townsandcitiescharacteristicsofbuiltupareasenglandandwales/census2021)

690 Ordnance Survey (GB). (2012, January). GB National Grids Squares [SHAPE geospatial

691 data], Scale 1:250000, Tiles: GB. Retrieved June 29, 2024, from

692 <https://digimap.edina.ac.uk>

693 Outhwaite, C. L., Chandler, R. E., Powney, G. D., Collen, B., Gregory, R. D., &

694 Isaac, N. J. B. (2018). Prior specification in Bayesian occupancy modelling

695 improves analysis of species occurrence data [Num Pages: 11 Place: Amsterdam

696 Publisher: Elsevier Science Bv Web of Science ID: WOS:000452692600034].

697 *ECOLOGICAL INDICATORS*, 93, 333–343.

698 <https://doi.org/10.1016/j.ecolind.2018.05.010>

699 Piano, E., Souffreau, C., Merckx, T., Baardsen, L. F., Backeljau, T., Bonte, D., Brans, K. I.,

700 Cours, M., Dahirel, M., Debortoli, N., Decaestecker, E., De Wolf, K.,

701 Engelen, J. M. T., Fontaneto, D., Gianuca, A. T., Govaert, L., Hanashiro, F. T. T.,

702 Higuti, J., Lens, L., . . . Hendrickx, F. (2020). Urbanization drives cross-taxon

703 declines in abundance and diversity at multiple spatial scales [eprint:

704 <https://onlinelibrary.wiley.com/doi/pdf/10.1111/gcb.14934>]. *Global Change Biology*,

705 26(3), 1196–1211. <https://doi.org/10.1111/gcb.14934>

- Pocock, M. J. O., Roy, H. E., Preston, C. D., & Roy, D. B. (2015). The Biological Records Centre: A pioneer of citizen science. *Biological Journal of the Linnean Society*, 115(3), 475–493. <https://doi.org/10.1111/bij.12548>
- R Core Team. (2024). R: A Language and Environment for Statistical Computing. <https://www.R-project.org/>
- Reusch, C., Vindas-Picado, J. B., Scholz, C., Hoffmeister, U., & Voigt, C. C. (2024). Light-averse behaviour of attic-dwelling bats when commuting through urban areas [Num Pages: 11 Place: Amsterdam Publisher: Elsevier Web of Science ID: WOS:001286705200001]. *GLOBAL ECOLOGY AND CONSERVATION*, 54, e03112. <https://doi.org/10.1016/j.gecco.2024.e03112>
- Rowland, C. S., Marston, C. G., Morton, R. D., & O'Neil, A. W. (2020). Land Cover Map 1990 (25m raster, GB) v2. <https://doi.org/https://doi.org/10.5285/1be1912a-916e-42c0-98cc-16460fac00e8>
- Rowland, C. S., Morton, R. D., Carrasco, L., McShane, G., O'Neil, A. W., & Wood, C. M. (2017). Land Cover Map 2015 (25m raster, GB). <https://doi.org/https://doi.org/10.5285/BB15E200-9349-403C-BDA9-B430093807C7>
- Russo, D., Salinas-Ramos, V. B., Cistrone, L., Smeraldo, S., Bosso, L., & Ancillotto, L. (2021). Do We Need to Use Bats as Bioindicators? *Biology*, 10(8), 693. <https://doi.org/10.3390/biology10080693>
- Sala, O. E., Stuart Chapin, F., III, Armesto, J. J., Berlow, E., Bloomfield, J., Dirzo, R., Huber-Sanwald, E., Huenneke, L. F., Jackson, R. B., Kinzig, A., Leemans, R., Lodge, D. M., Mooney, H. A., Oesterheld, M., Poff, N. L., Sykes, M. T., Walker, B. H., Walker, M., & Wall, D. H. (2000). Global Biodiversity Scenarios for

the Year 2100 [Publisher: American Association for the Advancement of Science].

Science, 287(5459), 1770–1774. <https://doi.org/10.1126/science.287.5459.1770>

Schaub, A., Ostwald, J., & Siemers, B. M. (2008). Foraging bats avoid noise. *Journal of*

Experimental Biology, 211(19), 3174–3180. <https://doi.org/10.1242/jeb.022863>

Simkin, R. D., Seto, K. C., McDonald, R. I., & Jetz, W. (2022). Biodiversity impacts and

conservation implications of urban land expansion projected to 2050 [Publisher:

Proceedings of the National Academy of Sciences]. *Proceedings of the National*

Academy of Sciences, 119(12), e2117297119.

<https://doi.org/10.1073/pnas.2117297119>

Starik, N., Gygas, L., & Göttert, T. (2024). Unexpected bat community changes along an

urban–rural gradient in the Berlin–Brandenburg metropolitan area [Publisher:

Nature Publishing Group]. *Scientific Reports*, 14(1), 10552.

<https://doi.org/10.1038/s41598-024-61317-7>

Stidsholt, L., Scholz, C., Hermanns, U., Teige, T., Post, M., Stapelfeldt, B., Reusch, C., &

Voigt, C. C. (2024). Low foraging rates drive large insectivorous bats away from

urban areas [eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/gcb.17063>].

Global Change Biology, 30(1), e17063. <https://doi.org/10.1111/gcb.17063>

Sugai, L. S. M., Silva, T. S. F., Ribeiro, J. W., Jr, & Llusia, D. (2019). Terrestrial Passive

Acoustic Monitoring: Review and Perspectives. *BioScience*, 69(1), 15–25.

<https://doi.org/10.1093/biosci/biy147>

Tajthi, B., Horvath, R., Mizser, S., Nagy, D. D., & Tothmeresz, B. (2017). Spider

assemblages in floodplain forests along an urbanization gradient [Num Pages: 8

Place: Heidelberg Publisher: Springer Heidelberg Web of Science ID:

WOS:000424882900010]. *COMMUNITY ECOLOGY*, 18(3), 311–318.

<https://doi.org/10.1556/168.2017.18.3.10>

Tälle, M., Ranius, T., & Öckinger, E. (2023). The usefulness of surrogates in biodiversity conservation: A synthesis. *Biological Conservation*, 288, 110384.

<https://doi.org/10.1016/j.biocon.2023.110384>

Tian, Y., Shuai, Y., Ma, X., Shao, C., Liu, T., & Tuerhanjiang, L. (2022). Improved Landscape Expansion Index and Its Application to Urban Growth in Urumqi [Number: 20 Publisher: Multidisciplinary Digital Publishing Institute]. *Remote Sensing*, 14(20), 5255. <https://doi.org/10.3390/rs14205255>

UKCEH. (2021, November). Latest Land Cover Map provides greater detail about British landscape. Retrieved August 23, 2024, from <https://www.ceh.ac.uk/latest-land-cover-map-provides-greater-detail-about-british-landscape>

United Nations Department of Economic and Social Affairs, Population Division. (2024). *World Population Prospects 2024: Summary of Results* (tech. rep. No. UN DESA/POP/2024/TR/NO. 9). Retrieved July 16, 2024, from https://www.un.org/development/desa/pd/sites/www.un.org.development.desa/pd/files/undesa_pd_2024_wpp_2024_advance_unedited_0.pdf

United Nations General Assembly. (2015, October). Transforming our world: The 2030 Agenda for Sustainable Development.

Urban, M. C., Alberti, M., De Meester, L., Zhou, Y., Verrelli, B. C., Szulkin, M., Schmidt, C., Savage, A. M., Roberts, P., Rivkin, L. R., Palkovacs, E. P., Munshi-South, J., Malesis, A. N., Harris, N. C., Gotanda, K. M., Garroway, C. J., Diamond, S. E., Roches, S. D., Charmantier, A., & Brans, K. I. (2024). Interactions between climate change and urbanization will shape the future of biodiversity [Publisher: Nature

Publishing Group]. *Nature Climate Change*, 14(5), 436–447.

<https://doi.org/10.1038/s41558-024-01996-2>

Vaughan, N. (1997). The diets of British bats (Chiroptera) [Num Pages: 18 Place: Oxford

Publisher: Blackwell Science Ltd Web of Science ID: WOS:A1997XE77000001].

MAMMAL REVIEW, 27(2), 77–94.

<https://doi.org/10.1111/j.1365-2907.1997.tb00373.x>

Vlaschenko, A., Rodenko, O., Hukov, V., Kovalov, V., Prylutska, A., & Kravchenko, K.

(2023). Do Bats Avoid the Urban Core in the Breeding Season? A Case Study from

Temperate Latitudes [Number: 9 Publisher: Multidisciplinary Digital Publishing

Institute]. *Diversity*, 15(9), 967. <https://doi.org/10.3390/d15090967>

Zellmer, A. J., & Goto, B. S. (2022). Urban wildlife corridors: Building bridges for wildlife

and people [Num Pages: 14 Place: Lausanne Publisher: Frontiers Media Sa Web

of Science ID: WOS:000913032800001]. *FRONTIERS IN SUSTAINABLE CITIES*,

4, 954089. <https://doi.org/10.3389/frsc.2022.954089>

Zukal, J., Pikula, J., & Bandouchova, H. (2015). Bats as bioindicators of heavy metal

pollution: History and prospect. *Mammalian Biology*, 80(3), 220–227.

<https://doi.org/10.1016/j.mambio.2015.01.001>

793 SUPPLEMENTARY MATERIALS

794 All code can also be found in the Supplementary Files as .R files. **Contains:**

795 A - Species Detection Inclusions

796 B - Pre-scaling means and standard deviations of land cover classes

797 C - Full Model Results

798 D - Code

799 00_run_all_scripts.R

800 01_install_packages.R

801 02_anonymise_data.R

802 1_

803 2_prepare_landcover_variables.R

804 3_standardise_obs_data.R

805 4_attach_environmental_variables.R

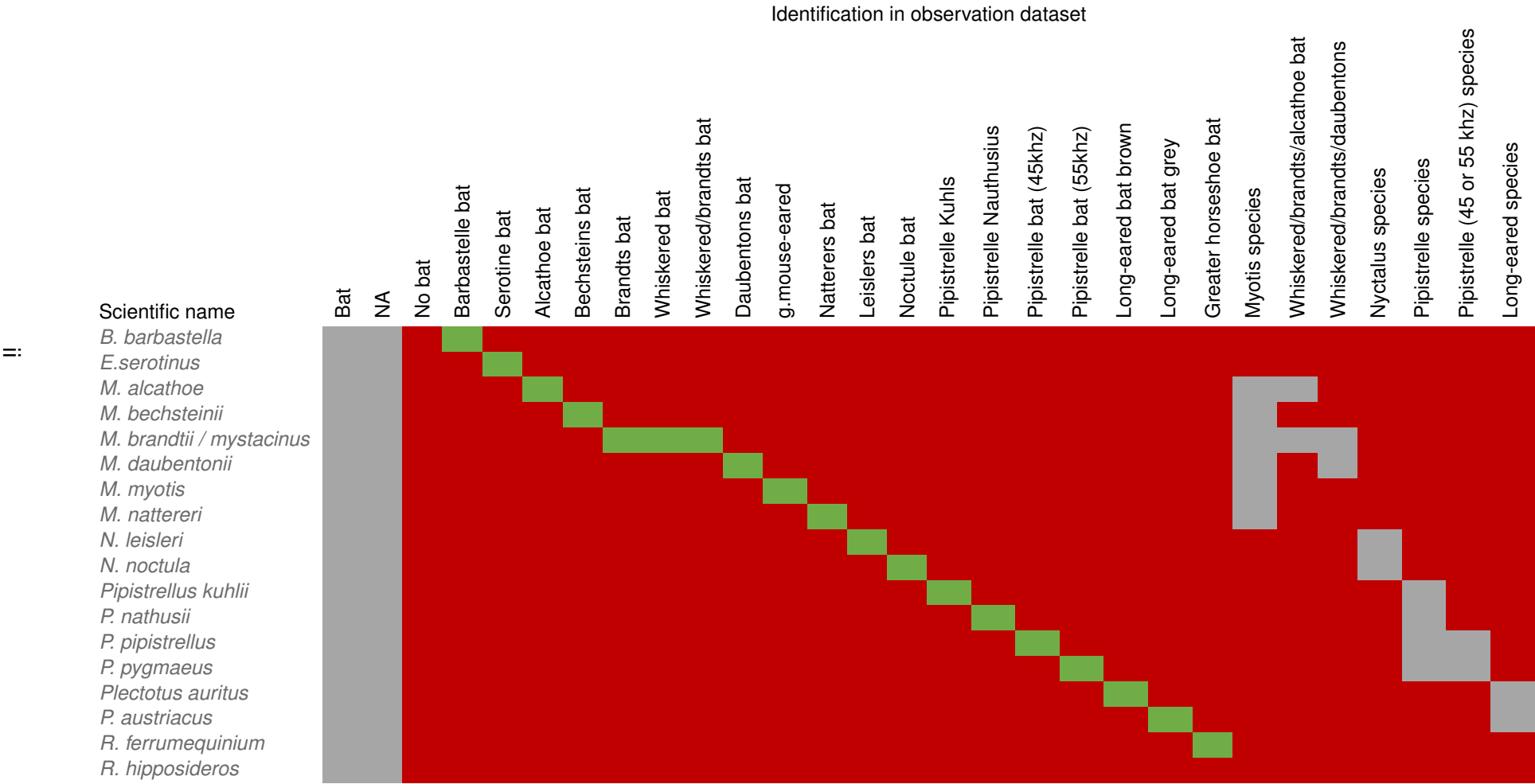
806 5_run_final_models.R

807 A_create_images.R

808 E - Risk Assessment

A - Species Detection Inclusions

Table S1: Presence/absence generation for all species. Red indicates a identification category was used to generate an absence, whilst green indicates it was used as a presence record. Grey indicates that the identification category was not used to generate presence or absence data for a particular species.



B - Pre-scaling means and standard deviations of land cover classes

Table S2: Population means and standard deviations for each land cover class across all datasets 1990-2023

Land Cover Class	Mean	s.d.
Arable	0.0814	0.0650
Coastal	0.0048	0.0206
Coniferous woodland	0.0021	0.0100
Deciduous woodland	0.0273	0.0337
Freshwater	0.0021	0.0096
Improved grassland	0.0753	0.0535
Arable/improved grassland	0.1567	0.0700
Marsh, heath, bog	0.0004	0.0022
Saltwater	0.0084	0.0352
Semi-natural grassland	0.0065	0.0149
Suburban	0.0225	0.0356
Urban	0.0072	0.0201
Urban/Suburban	0.0297	0.0460

812

C - Full Model Results

Table S3

dummy table to

remove later

add here

813 **D - Code**814 **00_run_all_scripts.R**

```
#####
#####
#####          RUN ALL SCRIPTS          #####
#####          #####
#####
#####
# This script runs all code files in the correct order, with the
# exception of 0_anonymise_KBG_data (as this file will not run without access
# to the non-anonymised dataset and the string of names to remove). It instead
# reads in the results of this script. The anonymisation script used (less the
# string of individual names to remove) is included in the 'scripts' folder.

# install package to set working directory
## UNCOMMENT TO INSTALL
if ("rstudioapi" %in% rownames(installed.packages()) == FALSE){
  install.packages("rstudioapi")
}
library(rstudioapi)

# find script directory, find parent directory, set working directory to
# parent directory
script_dir <- dirname(getActiveDocumentContext())$path
parent_dir <- dirname(script_dir)
setwd(parent_dir)
print(getwd())

# remove directory vectors from working environment for tidiness
rm(script_dir, parent_dir)

#####          PREPROCESSING          #####

# UNCOMMENT TO INSTALL REQUIRED PACKAGES
source("./scripts/0_install_packages.R")

# run anonymising script. This line will NOT run unless the non-anonymised
# data is present.
#source("./scripts/0_anonymise_KBG_data.R")

# run cleaning script on anonymised data
source("./scripts/1_clean_KBG_data.R")
```

```

# clear working directory from data cleaning
rm(list = ls())

# prepare land cover variables from CEH data
source("./scripts/2_prepare_landcover_variables.R")

# clear working directory from extracting variables
rm(list = ls())

# standardise observation data and save genus and then species-level
# observations tables
source("./scripts/3_standardise_obs_data.R")
rm(list = ls())

#####          MODEL SELECTION AND COMPARATIVE STATICS          #####

# this script runs one variable only environmental models at 1km and 3km spatial
# resolutions across genera and species. The AIC values, direction, and
# significance of these models can be found in the supplementary materials.

source("./scripts/4_analyse_environmental_variables.R")
rm(list = ls())

#####          RUN FINAL MODELS          #####

# this script runs the final selected models for each genera and species
# grouping, and produces graphics included in the paper.
source("./scripts/5_run_final_models.R")

```

815 01_install_packages.R

```

  if ("pacman" %in% rownames(installed.packages()) == FALSE){
    install.packages("pacman")
  }

# p_load function from pacman package checks if packages are installed,

```



```

# installs if not, and loads

pacman::p_load(tokenizers, tidyverse)

require(magrittr)

pacman::p_load(ggplot2, sf, spatialEco, rnrfa, lme4, raster, terra, tidyterra,
               patchwork, emmeans)

```

816 02_anonymise_data.R

```

#####
#####
#####          KBG DATA ANONYMISATION          #####
#####
#####
#####

# this script imports the non-anonymised datasets from a non-public folder,
# runs scripts to anonymise as much of the comments as possible, and outputs
# the anonymised dataset to the main data folder of the public repository.

dat <- read_csv("../non_anonymised/20240629_KBG_data_primary_non_anonymised.csv")

#### REMOVE PHONE NUMBERS ####
dat %<>%
  mutate(...16 = str_replace(...16, "(0|\\+\\d{2})\\d{4}(\\s|-)?\\d{6}",
                             "phone number"),
         # next three line removes specific unusually formatted phone numbers
         ...16 = str_replace(...16, "020\\d{1}\\s\\d{7}",
                             "phone number"),
         ...16 = str_replace(...16, "\\d{4}-\\d{6}",
                             "phone number"),
         ...16 = str_replace(...16, "\\d{6}\\s\\d{6}",
                             "phone number"))

#### REMOVE EMAIL ADDRESSES ####
# (?i) makes search case-insensitive
dat %<>%
  mutate(...16 = str_replace(...16, "(?i)\\b.{1,}\\s\\.co(m|\\.uk)",
                             "email address"))

#### REMOVE HONORIFICS AND FIRST WORD FOLLOWING ####

```

```

# only one doctor in dataset, removed in direct names remove below
dat %<>%
  mutate(...16 = str_replace(...16, "(?i)\\bmr(s)?\\s\\w{1,}\\b",
                              "name"))

#### REMOVE COMMON NAMES ####
# data is taken from Office for National Statistics 'Top 100 baby names in
# England and Wales: historical data, 1904-1994 edition'

common_boys_names <- read_csv("./data/raw/ons_common_boys_names_1904-1994.csv") %>%
  dplyr::select(2:11) %>%
  slice(2:101) %>%
  unlist(use.names = FALSE) %>%
  base::unique()

common_girls_names <- read_csv("./data/raw/ons_common_girls_names_1904-1994.csv") %>%
  dplyr::select(2:11) %>%
  slice(2:101) %>%
  unlist(use.names = FALSE) %>%
  base::unique()

# remove 'JUNE', 'ROSE', 'DAWN', 'MAY', 'IVY' - all are included in comments but not
# as names
common_girls_names <- common_girls_names[!common_girls_names %in%
                                           c("JUNE", "ROSE", "DAWN", "MAY", "IVY")]

dat %<>%
  mutate(...16 = str_replace(...16, paste0("(?i)\\b(",
                                             paste(common_boys_names, collapse = "|"),
                                             ")\\b"),
                              "first_name"),
    ...16 = str_replace(...16, paste0("(?i)\\b(",
                                             paste(common_girls_names, collapse = "|"),
                                             ")\\b"),
                              "first_name"))

#### REMOVE COMMON SURNAMES ####
# data is taken from forebears.io (in absence of available census data)

common_surnames = c("Smith", "Jones", "Taylor", "Brown", "Williams",
                    "Wilson", "Johnson", "Davies", "Patel", "Robinson",
                    "Wright", "Thompson", "Evans", "Walker", "White",
                    "Roberts", "Green", "Hall", "Thomas", "Clarke", "Jackson",
                    "Wood", "Harris", "Edwards", "Turner", "Martin", "Cooper",
                    "Hill", "Ward", "Hughes", "Moore", "Clark", "King",
                    "Harrison", "Lewis", "Baker", "Lee", "Allen", "Morris",
                    "Khan", "Scott", "Watson", "Davis", "Parker", "James",
                    "Bennett", "Young", "Phillips", "Richardson", "Mitchell",

```

```

      "Bailey", "Carter", "Cook", "Singh", "Shaw", "Bell",
      "Collins", "Morgan", "Kelly", "Begum", "Miller", "Cox",
      "Hussain", "Marshall", "Simpson", "Price", "Anderson",
      "Adams", "Wilkinson", "Ali", "Ahmed", "Foster", "Ellis",
      "Murphy", "Chapman", "Mason", "Gray", "Richards", "Webb",
      "Griffiths", "Hunt", "Palmer", "Campbell", "Holmes",
      "Mills", "Rogers", "Barnes", "Knight", "Matthews", "Barker",
      "Powell", "Stevens", "Kaur", "Fisher", "Butler", "Dixon",
      "Russell", "Harvey", "Pearson", "Graham")

# remove 'Young', 'Wood', 'Hall', 'Brown', 'White'
common_surnames <- common_surnames[!common_surnames %in%
                                   c("Young", "Wood", "Hall", "Brown",
                                     "White")]

dat %<>%
  mutate(...16 = str_replace(...16, paste0("(?i)\\b(",
                                             paste(common_surnames, collapse = "|"),
                                             ")\\b"),
                                             "surname"))

#### REMOVE VECTOR OF OTHER NAMES IDENTIFIED ####
# vector not supplied for self-evident reasons
source("../non_anonymised/remove_names.R")

dat %<>%
  mutate(...16 = str_replace(...16, paste(remove_names, collapse = ""),
                                             "name"))

#### WRITE DATA TO NEW CSV ####
write_csv(dat, "../data/raw/KBG_data_primary_anonymised.csv")

```

817 1_clean_obs_data.R

```

#####
#####
##### CLEAN OBS DATA #####
#####
#####

# This script loads the two datasets (primary and update), cleans their date columns
# and column names individually to allow the two sets to be combined, combines
# the two datasets, and then cleans all other columns of the combined dataset,
# including extracting some information from comments columns, though this
# information is not used in further analysis.

```

```

# load in both datasets. encoding is specified to avoid reading-in issues
# with non-standard characters present in dataset.
primary_data <- read_csv("./data/raw/KBG_data_primary_anonymised.csv",
                          locale = locale(encoding = "latin1"))
update_data <- read_csv("./data/raw/KBG_data_update_anonymised.csv",
                         locale = locale(encoding = "latin1"))

### PRIMARY DATASET ###
# move column names from row 2 to colnames
colnames(primary_data) <- primary_data[2,]
primary_data <- primary_data[-c(1,2),]

# prepare column names for merging (add comment column names, make lower case, remove sp
primary_data %<>% rename("comment1" = 16,
                        "comment2" = 17,
                        "comment3" = 18) %>%
  rename_with(~tolower(gsub(" ", "_", .x)))

# clean date column where possible and convert output to date format
# loop cycles through all elements of the raw_date column and trys to parse
# according to formats found in the data. If not possible, NA is returned.

primary_data %<>%
  rename("raw_date" = "date") %>%
  mutate("clean_date" = NA, .before = "raw_date")

for (i in 1:nrow(primary_data)) {
  date_value <- tryCatch(as.Date(primary_data$raw_date[i],
                                tryFormats = c("%d/%m/%Y", "%d-%m-%Y",
                                                "%Y-%m-%d", "%d.%m.%Y",
                                                "%d/%m %Y")),
                        error = function(e) return(NA))
  primary_data$clean_date[i] <- date_value
}

# converts from numeric dates to human-parsable Date class
primary_data$clean_date <- as.Date(primary_data$clean_date)

sum(is.na(primary_data$clean_date))

# 112 rows have only a date range or year

### UPDATE DATASET ###
# first two rows are removed as contain metadata only
update_data <- update_data[-c(1,2),]

# columns after 16 are removed as they are empty for all rows

```

```

update_data <- update_data[,-c(17:251)]

# clean date column is added (these dates are already cleaner than the other set)
update_data %<>%
  rename("raw_date" = "DATE") %>%
  mutate("clean_date" = NA, .before = "raw_date")

update_data$clean_date <- as.Date(update_data$raw_date, "%d/%m/%Y")

# columns are renamed to match original dataset
update_data %<>%
  rename("comment1" = "COMMENTS",
        "osgridref" = "OS GRID REF") %>%
  rename_with(~tolower(gsub(" ", "_", .x)))

### COMBINE AND CHECK OVERLAPPING DATA
# primary and update datasets are combined, with source column to identify
# which dataset an entry is from
# all data is reindexed and an id number added in the id column so that filtered
# values can be traced back to entries in main dataset
# all entries with dates between the end of the first dataset and the start
# of the second dataset are manually checked for duplicate records (none are found)

combined_data <- bind_rows(primary_data, update_data, .id="source_data")
combined_data <- combined_data %>%
  mutate("id" = 1:nrow(combined_data), .before = "source_data")

overlap_max <- range(primary_data$clean_date, na.rm = TRUE)[2]
overlap_min <- range(update_data$clean_date, na.rm = TRUE)[1]

date_overlap_check <- combined_data %>%
  filter(clean_date > overlap_min & clean_date < overlap_max) %>%
  arrange(clean_date)

rm(date_overlap_check)

##### COMBINED DATASET #####

str(combined_data)
# many columns include non-Unix entries

# there are large numbers of grammar/formatting differences across the dataset
# in all columns, convert all values to lowercase
# and convert osgridref to uppercase and address to sentence case

combined_data %<>%
  mutate(across(is.character, ~ gsub("'", "''", .))) %>%

```

```

mutate(across(c(where(is.character), -c("osgridref", "address", "raw_date")),
              tolower)) %>%
mutate(osgridref = toupper(osgridref)) %>%
mutate(address = str_to_title(address)) %>%
# remove trailing whitespace, remove all whitespace in osgridref
mutate(across(is.character, ~ gsub(" $", "", .))) %>%
# clean categories in bat species
rename("common_name" = "bat_species_(common_name)") %>%
mutate(across(common_name, ~ gsub("\u0092", "", .))) %>%
mutate(across(common_name, ~ gsub("'", "", .))) %>%
mutate(across(common_name, ~ gsub("no bats", "no bat", .))) %>%
mutate(across(common_name, ~ gsub("pipistrellus", "pipistrelle", .))) %>%
mutate(across(common_name, ~ gsub("vespertilionidae", "bat", .))) %>%
mutate(across(common_name, ~ gsub("nathusius' pipistrelle bat",
                                "pipistrelle nathusius'", .))) %>%
mutate(across(common_name, ~ gsub("nathusius pipistrelle bat",
                                "pipistrelle nathusius", .))) %>%
mutate(across(common_name, ~ gsub("brown long-eared", "long-eared bat brown", .))) %>%
mutate(common_name = ifelse(common_name=="", NA, common_name)) %>%
# where 'count' gives a number of passes, append 'passes' to 'count_type'
# column, and remove 'passes' from count column
mutate(count_type = ifelse(str_detect(count, "pass"), "passes", count_type)) %>%
mutate(across(count, ~gsub(" pass| passes", "", .))) %>%
# where 'count' gives a number and sex, append sex to sex column
mutate(sex = case_when(count == "1m" ~ "male",
                      count == "2f" ~ "female",
                      .default = sex)) %>%
mutate(count = case_when(count == "1m" ~ "1",
                      count == "2f" ~ "2",
                      .default = count)) %>%
# remove ambiguous count values (1 to 5, up to 10, etc), and "n/a"
mutate(count = case_when(str_detect(count, "\\D") ~ NA,
                      count == "" ~ NA,
                      .default = count))

# manually check entries where 'record_type' gives species
check_record_type <- combined_data %>%
  filter(record_type %in% c("pipistrelle bat", "no bats", "natterers",
                          "whiskered bat"))

# all have data recorded in species column. remove values in record type
combined_data %<>%
  mutate(record_type = ifelse(record_type %in% c("pipistrelle bat",
                                                "no bats", "natterers",
                                                "whiskered bat", "",
                                                "unknown"),
                             NA, record_type))
# combine two variations of 'feeding and droppings' in identification column

```

```

# add NA, including to record of '20:27' (unclear information)
combined_data %<>%
  mutate(identification = case_when(identification %in% c("feeding & droppings",
                                                         "feeding remains and droppings",
                                                         ~ "feeding and droppings",
                                                         identification %in% c("", "20:27", "unknown")
                                                         ~ NA,
                                                         .default = identification))

# correct grammar error in habitat type, add NA
combined_data %<>%
  mutate(habitat = case_when(habitat == "woodland - ixed" ~ "woodland - mixed",
                             habitat == "" ~ NA,
                             .default = habitat),
         # remove ! in structure, add NA
         structure = case_when(structure == "roundabout!" ~ "roundabout",
                                structure == "" ~ NA,
                                .default = structure),
         # fix spelling in count_type, add NA
         count_type = case_when(count_type == "estiate" ~ "estimate",
                                count_type %in% c("", "unknown") ~ NA,
                                .default = count_type),
         # fix spelling in sex, add NA
         sex = case_when(sex == "ale" ~ "male",
                         sex == "feale" ~ "female",
                         sex %in% c("", "unknown", "?", "-") ~ NA,
                         .default = sex))

# manually check entries where 'sex' column has other data
check_sex <- combined_data %>%
  filter(!(sex %in% c("male", "female", NA)))

# where 'sex' gives range of numbers or 'estimate', give 'count_type' as estimate
# all other unusual values recorded elsewhere, remove
combined_data %<>%
  mutate(count_type = ifelse(sex %in% c("2-3", "estimate"), "estimate", count_type),
         sex = ifelse(sex %in% c("male", "female"), sex, NA),
         # add NAs to age column
         age = ifelse(age %in% c("", "unknown"), NA, age))

# check unusual entries in 'age' column
check_age <- combined_data %>%
  filter(!(age %in% c("adult", "juvenile", NA)))

combined_data %<>%
  # line with 'nbp survey' in age has 'licenced bat worker' in event. this is
  # moved to experience to allow for nbp survey to be moved into event
  mutate(experience = ifelse(age == "nbp survey",

```

```

      "licenced bat worker", experience),
  # survey types included within the age column moved
  event = case_when(age == "consultant survey" ~ "consultant_survey",
    age == "nbmp survey" ~ "nbmp_survey",
    .default = event),
  # all cases of 'licenced bat worker' in age column also have information in
  # remove all values except 'adult' and 'juvenile' from age column
  # NOTE: there are three values of 'general public' where the experience column
  # gives 'bat group member'. In these cases, the experience column is taken to
  # be authoritative
  age = ifelse(age %in% c("adult", "juvenile"), age, NA))

# fix spelling, grammar, categorisation issues in 'event', add NA
# add NA to 'experience'
combined_data %<>%
  mutate(event = case_when(event == "consultant_survey" ~ "consultant survey",
    event == "nbmp" ~ "nbmp_survey",
    event %in% c("", "-") ~ NA,
    .default = event),
    experience = ifelse(experience %in% c("", "unknown"), NA, experience))

# 'comment3' seems to give an old referencing system. It is removed as it does not
# provide any informative information and has no associated metadata
# blanks in 'comment1' and 'comment2' are made NA to allow for a true picture
# of the number of comments to be made
combined_data %<>%
  mutate(comment3 = NULL,
    comment1 = ifelse(comment1 == "", NA, comment1),
    comment2 = ifelse(comment2 == "", NA, comment2))

comment1_no_na <- combined_data %>%
  filter(!is.na(comment1))

comment1_vector <- comment1_no_na$comment1
comment1_string <- str_c(comment1_vector, collapse = " ")

frequency <- tokenize_ngrams(comment1_string, n = 3L, n_min = 2L,
  simplify = TRUE) %>%
  as_tibble() %>%
  count(value, sort = TRUE)

# working through from top of frequency list for items that appear to give
# information on survey type
filtered_hibernation_count <- combined_data %>%
  filter(str_detect(comment1, "hibernation count"))
# all confirm record is from hibernation count/survey. Many also give the series
unique(filtered_hibernation_count$comment1)

```



```

# number of which mine series in Westerham was being checked.
# check how many have 'hibernating bat' in record type column
sum(filtered_hibernation_count$record_type == "hibernating bat", na.rm = TRUE)
# all but two have this information, check these. both are records of no bat presence
check_set <- filtered_hibernation_count %>%
  filter(is.na(record_type))
# append information to event_from_comment column and add checked column
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "hibernation count"),
                                     "hibernation count", NA),
         checked = case_when(str_detect(comment1, "hibernation count") ~ TRUE,
                             comment1 == NA ~ TRUE,
                             .default = FALSE))

## 'emergence survey'
filtered_emergence_survey <- combined_data %>%
  filter(str_detect(comment1, "emergence survey"))

unique(filtered_emergence_survey$comment1)
# all confirm emergence survey. most reference consultant work
sum(!is.na(filtered_emergence_survey$count))
# one record with no count data does not give count data in comment
filtered_emergence_survey %>% filter(is.na(count))
# append 'emergence survey' to event_from_comment
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "emergence survey"),
                                     "emergence survey", event_from_comment),
         checked = ifelse(str_detect(comment1, "emergence survey"),
                           TRUE, checked))

# rerun frequency analysis on non-checked rows
comment1_check2 <- combined_data %>%
  filter(checked == FALSE)

comment1_vector2 <- comment1_check2$comment1
comment1_string2 <- str_c(comment1_vector2, collapse = " ")

frequency2 <- tokenize_ngrams(comment1_string2, n = 3L, n_min = 2L,
                              simplify = TRUE) %>%
  as_tibble() %>%
  count(value, sort = TRUE)

# 'nbmp'
filtered_nbmp <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp"))

unique(filtered_nbmp$comment1)

```

```

sum(str_detect(filtered_nbmp$comment1, "nbmp temp"))
sum(str_detect(filtered_nbmp$comment1, "^nbmp$"))
sum(str_detect(filtered_nbmp$comment1, "nbmp daubenton's"))
sum(str_detect(filtered_nbmp$comment1, "nbmp serotine"))
sum(str_detect(filtered_nbmp$comment1, "nbmp sunrise"))
sum(str_detect(filtered_nbmp$comment1, "nbmp nsp"))
sum(str_detect(filtered_nbmp$comment1, "nbmp cloud"))
sum(str_detect(filtered_nbmp$comment1, "nbmp ext"))
sum(str_detect(filtered_nbmp$comment1, "nbmp field"))
sum(str_detect(filtered_nbmp$comment1, "nbmp waterway"))

# filter over each of the above, manually check, then add info to event_from_comment,
# change status of checked column

#### NBMP TEMP
filtered_nbmp_temp <- combined_data %>%
  filter(changed == FALSE) %>%
  filter(str_detect(comment1, "nbmp temp"))
# none of these have information more specific than nbmp - though they do contain
# environmental variable info which might be interesting in other studies.
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbmp temp") &
                                     checked == FALSE,
                                     "nbmp", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbmp temp"),
                           TRUE, checked))

#### NBMP
filtered_nbmp_only <- combined_data %>%
  filter(changed == FALSE) %>%
  filter(str_detect(comment1, "^nbmp$"))
# none of these have information more specific than nbmp
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "^nbmp$") &
                                     checked == FALSE,
                                     "nbmp", event_from_comment),
         checked = ifelse(str_detect(comment1, "^nbmp$"),
                           TRUE, checked))

#### NBMP DAUBENTONS
filtered_nbmp_daubentons <- combined_data %>%
  filter(changed == FALSE) %>%
  filter(str_detect(comment1, "nbmp daubenton's"))
# all are nbmp daubenton's surveys (even when recording other species)
# these could be used to derive absence data for daubs
# some also contain environmental info
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbmp daubenton's") &
                                     checked == FALSE,
                                     "nbmp daubenton's", event_from_comment),

```

```

checked = ifelse(str_detect(comment1, "nbp daubenton's"),
  TRUE, checked))

#### NBMP SEROTINE
filtered_nbp_serotine <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp serotine"))
# ID 20947 notes no bats, but does not specify nbp serotine *survey* (only
# serotine roost). all others specify serotine survey
combined_data %<%
  mutate(count = ifelse(id == 20947, 0, count),
    checked = ifelse(id == 20947, TRUE, checked)) %>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp serotine") &
    checked == FALSE,
      "nbp serotine", event_from_comment),
    checked = ifelse(str_detect(comment1, "nbp serotine"),
      TRUE, checked))

#### NBMP SUNRISE
filtered_nbp_sunrise <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp sunrise"))
# all sunrise
combined_data %<%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp sunrise") &
    checked == FALSE,
      "nbp sunrise", event_from_comment),
    checked = ifelse(str_detect(comment1, "nbp sunrise"),
      TRUE, checked))

#### NBMP NSP
filtered_nbp_nsp <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp nsp"))
# all are nsp (noctule/serotine/pip)/field surveys (these are the same survey)
# with two different names. a few records contain environmental information
combined_data %<%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp nsp") &
    checked == FALSE,
      "nbp nsp/field", event_from_comment),
    checked = ifelse(str_detect(comment1, "nbp nsp"),
      TRUE, checked))

#### NBMP CLOUD
filtered_nbp_cloud <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp cloud"))
# none contain survey information greater than 'nbp'. most have environmental data.

```

```

# one long-eared brown bat record notes incorrect count of pipistrelles, but
# unclear where this record is
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp cloud") &
                                     checked == FALSE,
                                     "nbp", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbp cloud"),
                          TRUE, checked))

#### NBMP EXT
filtered_nbp_ext <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp ext"))
# none have information greater than nbmp, though highly likely all are hibernation
# counts. all but three (one no info, two flying bat) specify they are observations
# of hibernating bats. two flying bat observations specify bats flying in shafts in
# comment. some entries have more specific location data inside comment
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp ext") &
                                     checked == FALSE,
                                     "nbp probable hibernation count",
                                     event_from_comment),
         checked = ifelse(str_detect(comment1, "nbp ext"),
                          TRUE, checked))

#### NBMP FIELD
filtered_nbp_field <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp field"))
# all specify field survey, none give additional information. field survey is
# same as nsp
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp field") &
                                     checked == FALSE,
                                     "nbp nsp/field", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbp field"),
                          TRUE, checked))

#### NBMP WATERWAY
filtered_nbp_waterway <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp waterway"))
# all specify waterway survey (daubs). many contain the passes information in
# the comment, though here count estimate is retained.
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp waterway") &
                                     checked == FALSE,
                                     "nbp waterway", event_from_comment),

```

```

        checked = ifelse(str_detect(comment1, "nbmp waterway"),
                          TRUE, checked))

### RECHECK remaining nbmp entries
filtered_nbmp2 <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp"))

unique(filtered_nbmp2$comment1)
sum(str_detect(filtered_nbmp2$comment1, "nbmp nathusius"))
sum(str_detect(filtered_nbmp2$comment1, "nbmp roost count"))
sum(str_detect(filtered_nbmp2$comment1, "nbmp wind"))

# filter over each of the above, manually check, then add info to event_from_comment,
# change status of checked column

#### NBMP NATHUSIUS
filtered_nbmp_nathusius <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp nathusius"))
# these are all records from the 2009-2014 version of the Nauthusius' Pipistrelle
# Survey (distinct from the 2014-2023 National Nauthusius' Pipistrelle Project).
# most records contain more specific location data in comment
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbmp nathusius") &
                                     checked == FALSE,
                                     "nbmp nathusius 2009-2014", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbmp nathusius"),
                           TRUE, checked))

#### NBMP ROOST COUNT
filtered_nbmp_roost_count <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp roost count"))
# all from nbmp roost counts, many contain additional location information,
# no other additional information
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbmp roost count") &
                                     checked == FALSE,
                                     "nbmp roost count", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbmp roost count"),
                           TRUE, checked))

#### NBMP WIND
filtered_nbmp_wind <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp wind"))
# none of these have information more specific than nbmp - though they do contain

```

```

# environmental variable info which might be interesting in other studies.
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp wind") &
                                     checked == FALSE,
                                     "nbp wind", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbp wind"),
                          TRUE, checked))

### RECHECK remaining nbmp entries
filtered_nbmp3 <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp"))

unique(filtered_nbmp2$comment1)
sum(str_detect(filtered_nbmp2$comment1, "nsp"))
sum(str_detect(filtered_nbmp2$comment1, "noctule serotine pipistrelle"))
sum(str_detect(filtered_nbmp2$comment1, "daub"))

# filter over each of the above, manually check, then add info to event_from_comment,
# change status of checked column

#### NSP
filtered_nsp <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp")) %>%
  filter(str_detect(comment1, "nsp"))
# all nbmp nsp/field surveys
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp") &
                                     str_detect(comment1, "nsp") &
                                     checked == FALSE,
                                     "nbp nsp/field", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbp") &
                          str_detect(comment1, "nsp"),
                          TRUE, checked))

#### NOCTULE SEROTINE PIPISTRELLE
filtered_noctule_serotine_pipistrelle <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbp")) %>%
  filter(str_detect(comment1, "noctule serotine pipistrelle"))
# all contain only survey type data
combined_data %<>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbp") &
                                     str_detect(comment1,
                                     "noctule serotine pipistrelle") &
                                     checked == FALSE,
                                     "nbp nsp/field", event_from_comment),

```

```

checked = ifelse(str_detect(comment1, "nbmp") &
                  str_detect(comment1, "noctule serotine pipistrelle"),
                  TRUE, checked))

#### NBMP WIND
filtered_daub <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp")) %>%
  filter(str_detect(comment1, "daub"))
# ID 26083 includes comment on daubenton's numbers but does not specify survey
# type beyond nbmp. all others confirm daubenton's survey
combined_data %<>%
  mutate(event_from_comment = ifelse(id == 26083, "nbmp", event_from_comment),
         checked = ifelse(id == 26083, TRUE, checked)) %>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbmp") &
                                      str_detect(comment1, "daub") &
                                      checked == FALSE,
                                      "nbmp daubenton's", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbmp") &
                           str_detect(comment1, "daub"),
                           TRUE, checked))

### RECHECK remaining nbmp entries
filtered_nbmp4 <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nbmp"))

## of remaining 79 observations:
## IDs 13762, 17552 and 25645 are not nbmp surveys (and do not contain other
# survey information)
## ID 10717 is a sunrise survey.
## IDs 8872 and 8806 are brown long-eared bat surveys.
## ID 6005 is an nsp/field survey.
# all others are nbmp surveys with no further detail on survey type
combined_data %<>%
  mutate(event_from_comment = case_when(id == 10717 ~ "nbmp sunrise",
                                         id %in% c(8872, 8806) ~ "nbmp ble",
                                         id == 6005 ~ "nbmp nsp/field",
                                         .default = event_from_comment),
         checked = ifelse(id %in% c(13762, 17552, 25645, 10717, 8872,
                                         8806, 6005), TRUE, checked)) %>%
  mutate(event_from_comment = ifelse(str_detect(comment1, "nbmp") &
                                      checked == FALSE,
                                      "nbmp", event_from_comment),
         checked = ifelse(str_detect(comment1, "nbmp"), TRUE, checked))
# remove nbmp subsets to clear working directory
files <- ls()
nbmp_remove <- str_subset(files, "filtered")

```



```

rm(list = nbmp_remove)

# rerun frequency analysis on non-checked rows
comment1_check3 <- combined_data %>%
  filter(checked == FALSE)

comment1_vector3 <- comment1_check3$comment1
comment1_string3 <- str_c(comment1_vector3, collapse = " ")

frequency3 <- tokenize_ngrams(comment1_string3, n = 3L, n_min = 2L,
                              simplify = TRUE) %>%
  as_tibble() %>%
  count(value, sort = TRUE)

## 'ext temp'
filtered_ext_temp <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "ext temp"))

# all are recordings of various environmental factors. These could be interesting
# to look at, and are reasonably well-formatted, but not here.
# none contain information about survey type or species id further than given
# some give more specific location data
combined_data %<>%
  mutate(checked = ifelse(str_detect(comment1, "ext temp"), TRUE, checked))

## 'car survey'
filtered_car_survey <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "car survey"))

unique(filtered_car_survey$comment1)

# all contain brm car survey information and nothing else. none have correct
# information recorded in event
combined_data %<>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                     str_detect(comment1, "car survey"),
                                     "brm car survey", event_from_comment),
         checked = ifelse(str_detect(comment1, "car survey"), TRUE, checked))

## 'activity survey'
filtered_activity_survey <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "activity survey"))

# all confirm activity survey, many give consultancy survey was undertaken for,
# none give environmental variables

```



```

combined_data %<>%
  mutate(event_from_comment = ifelse(checkered == FALSE &
                                     str_detect(comment1, "activity survey"),
                                     "activity survey", event_from_comment),
         checked = ifelse(str_detect(comment1, "activity survey"), TRUE, checked))

## 'transect survey'
filtered_transect_survey <- combined_data %>%
  filter(checkered == FALSE) %>%
  filter(str_detect(comment1, "transect survey"))

unique(filtered_transect_survey$comment1)

# most have just type of survey (/and consultant) - however large batch include
# passes number not otherwise included (all in same format).
# appended and 'passes' added to count_type for these and another batch where
# count is recorded but count_type is not. other NAs here likely to also be
# passes, but left as NA as not explicit
combined_data %<>%
  mutate(count = ifelse(checkered == FALSE &
                        str_detect(comment1, "transect survey") &
                        str_detect(comment1, "recording"),
                        # this extracts 1 to 3 digits which are followed by ' recording
                        as.numeric(str_extract(comment1, "\\d{1,3}(?= recording)")),
                        count),
         count_type = case_when(checkered == FALSE &
                                str_detect(comment1, "transect_survey") &
                                str_detect(comment1, "recording") ~ "passes",
                                checked == FALSE &
                                str_detect(comment1, "transect_survey") &
                                str_detect(comment1, "passes") ~ "passes",
                                .default = count_type),
         event_from_comment = ifelse(checkered == FALSE &
                                     str_detect(comment1, "transect survey"),
                                     "transect survey", event_from_comment),
         checked = ifelse(str_detect(comment1, "transect survey"),
                          TRUE, checked))

## 'harp trapping'
filtered_harp_trapping <- combined_data %>%
  filter(checkered == FALSE) %>%
  filter(str_detect(comment1, "harp trapping"))

unique(filtered_harp_trapping$comment1)
# large group that are harp trapping under nathusius project (2014-2023)
# rest are confirmed harp trapping. more information about number of traps,
# individual bat conditions, etc. is given in comments.
combined_data %<>%

```

```

mutate(event_from_comment = ifelse(checked == FALSE &
                                   str_detect(comment1,
                                               "nathusius|nathuius survey. harp trapping" |
                                               "nathusius 2014-2023, harp trapping",
                                               event_from_comment),
                                   checked = ifelse(checked == FALSE &
                                                  str_detect(comment1,
                                                              "nathusius|nathuius survey. harp trapping"),
                                                  TRUE, checked)) %>%
mutate(event_from_comment = ifelse(checked == FALSE &
                                   str_detect(comment1,
                                               "harp trapping"),
                                   "harp trapping", event_from_comment),
                                   checked = ifelse(checked == FALSE &
                                                  str_detect(comment1,
                                                              "harp trapping"),
                                                  TRUE, checked))

## 'bechstein's survey'
filtered_bechsteins_survey <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "bechstein's survey"))

# all confirm bechsteins survey. some specify harp trapping but as this was part
# of the standard survey design it is not added. some give additional information
# on bats trapped in comment
combined_data %<>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                   str_detect(comment1, "bechstein's survey"),
                                   "bechstein's survey", event_from_comment),
                                   checked = ifelse(str_detect(comment1, "bechstein's survey"),
                                   TRUE, checked))

# rerun frequency analysis on non-checked rows
comment1_check4 <- combined_data %>%
  filter(checked == FALSE)

comment1_vector4 <- comment1_check4$comment1
comment1_string4 <- str_c(comment1_vector4, collapse = " ")

frequency4 <- tokenize_ngrams(comment1_string4, n = 3L, n_min = 2L,
                              simplify = TRUE) %>%
  as_tibble() %>%
  count(value, sort = TRUE)

## 'colony survey'
filtered_colony_survey <- combined_data %>%
  filter(checked == FALSE) %>%

```

```

  filter(str_detect(comment1, "colony survey"))

# all are 'ite colony survey'
combined_data %<>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                     str_detect(comment1, "colony survey"),
                                     "ite colony survey", event_from_comment),
         checked = ifelse(str_detect(comment1, "colony survey"),
                          TRUE, checked))

## 'nsp survey'
filtered_nsp_survey <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nsp survey"))

# all confirm nsp/field survey, no other information given
combined_data %<>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                     str_detect(comment1, "nsp survey"),
                                     "nbp nsp/field", event_from_comment),
         checked = ifelse(str_detect(comment1, "nsp survey"), TRUE, checked))

## 'group survey'
filtered_group_survey <- combined_data %>%
  #filter(checked == FALSE) %>%
  filter(str_detect(comment1, "group survey"))

# all include information on the group doing the survey
combined_data %<>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                     str_detect(comment1, "group survey"),
                                     str_extract(comment1, "^.{1,} group survey"),
                                     event_from_comment),
         checked = ifelse(str_detect(comment1, "group survey"),
                          TRUE, checked))

# remove filtered subsets and previous vector lists to clear working directory
files <- ls()
filter_remove <- str_subset(files, "filtered")
rm(list = filter_remove)

# rerun frequency analysis on non-checked rows
comment1_check5 <- combined_data %>%
  filter(checked == FALSE)

comment1_vector5 <- comment1_check5$comment1
comment1_string5 <- str_c(comment1_vector5, collapse = " ")

```

```

frequency5 <- tokenize_ngrams(comment1_string5, n = 3L, n_min = 2L,
                               simplify = TRUE) %>%
  as_tibble() %>%
  count(value, sort = TRUE)

## 'nathuius'
filtered_nathuius_pip <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "nathuius"))

# all are nathuius pip project 2014-2023
combined_data %<>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                     str_detect(comment1, "nathuius"),
                                     "nbmp nathuius 2014-2023",
                                     event_from_comment),
         checked = ifelse(str_detect(comment1, "nathuius"), TRUE, checked))

## 'maternity roost'
filtered_maternity_roost <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "maternity roost"))

# most give records of probable maternity roosts. those confirmed have this
# information included already. no count or survey data not already included
combined_data %<>%
  mutate(checked = ifelse(str_detect(comment1, "maternity roost"), TRUE,
                          checked))

## 'rain =' - to exclude
filtered_rain_dry <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "rain ="))

# ID 20082 is nsp/field survey. all others give no more survey/count information
# not already recorded
combined_data %<>%
  mutate(event_from_comment = ifelse(id == 20082, "nbmp nsp/field",
                                     event_from_comment),
         checked = ifelse(id == 20082, TRUE, checked)) %>%
  mutate(checked = ifelse(str_detect(comment1, "rain ="), TRUE, checked))

# rerun frequency analysis on non-checked rows
comment1_check6 <- combined_data %>%
  filter(checked == FALSE)

comment1_vector6 <- comment1_check6$comment1
comment1_string6 <- str_c(comment1_vector6, collapse = " ")

```

```

frequency6 <- tokenize_ngrams(comment1_string6, n = 3L, n_min = 2L,
                               simplify = TRUE) %>%
  as_tibble() %>%
  count(value, sort = TRUE)

## 'entry survey'
filtered_entry_survey <- combined_data %>%
  filter(checked == FALSE) %>%
  filter(str_detect(comment1, "entry survey"))

# IDs 27798, 27799, and 27800 are dawn re-entry surveys
# all others are emergence surveys
combined_data %<%
  mutate(event_from_comment = ifelse(id %in% c(27798, 27799, 27800),
                                     "dawn re-entry", event_from_comment),
         checked = ifelse(id %in% c(27798, 27799, 27800), TRUE, checked)) %>%
  mutate(event_from_comment = ifelse(checked == FALSE &
                                     str_detect(comment1, "entry survey"),
                                     "emergence/re-entry", event_from_comment),
         checked = ifelse(str_detect(comment1, "entry survey"), TRUE, checked))

### STOPPED HERE, COMMENT CHECK NOT EXHAUSTIVE

with_event_data <- combined_data %>%
  filter(!(is.na(event_from_comment)) | !(is.na(event)))

# vast majority of comment2 is either 'chiroptera', (which are removed)
# or includes reference number for bms car survey (which are moved to seperate
# column
check_comment2 <- combined_data %>%
  filter(!is.na(comment2))

combined_data %<%
  mutate(bms_survey_number = ifelse(str_detect(comment2, "\\d{2}/\\d{2}/3\\d{3}"),
                                     comment2, NA),
         comment2 = ifelse(str_detect(comment2, "\\d{2}/\\d{2}/3\\d{3}"),
                            NA, comment2),
         comment2 = ifelse(comment2 %in% c("", "chiroptera"), NA, comment2))

  # stopped here - comment 2 not checked exhaustively

write_csv(combined_data, "./data/clean/KBG_data_clean.csv")

```

818 2_prepare_landcover_variables.R

```
#####
```

```
#####
#####          PREPARE LANDCOVER VARIABLES          #####
#####          #####
#####          #####
#####          #####
# this script reads in combines South East and London rasters (Centre for
# Ecology and Hydrology) and extracts landcover variables for each grid square
# within a 10km buffer of Kent in each year of data. It also extracts the same
# landcover variables for each grid square AND its neighbouring grid squares (*_3km_*).
# At the end of this script, the land cover statistics reported in the methods
# section for the whole of Kent are calculated.

# load kent county boundary and create 10km buffer to allow for cropping and
# masking of land cover rasters
kent_boundary <- sf::st_read("./data/county_boundaries_2023_bfe/CTYUA_MAY_2023_UK_BFE.shp")
  filter(CTYUA23NM %in% c("Kent", "Medway")) %>%
  st_combine() %>%
  st_as_sf()

kent_buffer <- kent_boundary %>%
  st_buffer(10000)

target_crs <- crs(kent_boundary)

# load in national grid squares within 10km buffer of Kent to enable calculation
# of neighbours and base dataframe

grid_squares <- sf::st_read("./data/grid_squares/1km_grid_region.shp") %>%
  sf::st_transform(crs = target_crs) %>%
  mutate(intersects = sf::st_intersects(., kent_buffer, sparse = FALSE)) %>%
  filter(intersects == TRUE) %>%
  dplyr::select(-"intersects")

# add geometry for 9km2 (up to eight neighbouring squares, dependent on
# coastline). 250m buffer is arbitrary, this would work with any value from 1 to
# 999 (1000 would extend to 25km2). THIS MUST BE DONE BEFORE FILTERING TO GRID
# SQUARES ONLY WITHIN KENT, otherwise the indexing won't work

grid_squares_buffered <- st_buffer(grid_squares, dist = 250)

# Find neighbours for each buffered grid square
neighbours <- st_intersects(grid_squares_buffered, grid_squares, sparse = TRUE)

# combine geometries of each grid square with its neighbours
grid_squares$geometry_neighbours <- lapply(neighbours, function(ids) {
  st_union(grid_squares$geometry[ids])
})
```

```

# convert list of geometries to an sfc object
grid_squares$geometry_neighbours <- st_sfc(do.call(c, grid_squares$geometry_neighbours),

# export dataframe of neighbouring squares (can only write one geometry
# column to shapefile, individual grid geometry is much easier to append later)

grid_squares_buffer_only <- grid_squares %>%
  st_drop_geometry()

st_write(grid_squares_buffer_only, "../data/grid_squares/processed_grid_squares.shp")

#####          COMBINE ACROSS LONDON AND SOUTH EAST          #####
# first the Kent buffer boundary is saved as a SpatVector (to allow for cropping
# and masking the land cover rasters)
kent_spat <- vect(kent_buffer)
target_crs <- crs(kent_spat)

# dataframe filtered to only Kent grid squares is created, and then two
# SpatVectors (1km and 3km), and a PLAN_NO list are created to allow for
# variable extraction with the terra package.
kent_grid_squares <- grid_squares %>%
  filter(st_intersects(.$geometry, kent_boundary, sparse = TRUE) %>% lengths > 0)
kent_grid_squares_buffered <- grid_squares_buffered %>%
  filter(PLAN_NO %in% unique(kent_grid_squares$PLAN_NO))

grids_vect_1km <- vect(kent_grid_squares$geometry)
grids_vect_3km <- vect(kent_grid_squares_buffered$geometry)
grid_names <- kent_grid_squares$PLAN_NO

#####          1990          #####

se_1990 <- rast("../data/CEH_landcover_rasters/south_east/LCM1990.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_1990 <-rast("../data/CEH_landcover_rasters/london/LCM1990.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_1990 <-rast("../data/CEH_landcover_rasters/eastern/LCM1990.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_1990 <- mosaic(se_1990, lon_1990, fun = "first")
lc_1990 <- mosaic(lc_1990, east_1990, fun = "first")

clr <- read_delim("../data/CEH_landcover_rasters/1990_and_2015_onwards_colours.txt",

```



```

        col_names = c("value", "red", "green", "blue"))
clr <- as.matrix(clr)
coltab(lc_1990) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                 col_names = c("1990"))
levels(lc_1990) <- lvl
# take only land cover layer
lc_1990 <- lc_1990[[1]]
plot(lc_1990)

writeRaster(lc_1990, "./data/CEH_landcover_rasters/LCM1990.tif")

# extract variables for each landcover type
lc_vars_1990_1km <- zonal(lc_1990, grids_vect_1km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_1990_3km <- zonal(lc_1990, grids_vect_3km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_1990 <- left_join(lc_vars_1990_1km, lc_vars_1990_3km) %>%
  mutate(year = 1990)

# write to folder
write_csv(lc_vars_1990, "./data/landscape_variables/1990.csv")

#####      2000      #####
se_2000 <- rast("./data/CEH_landcover_rasters/south_east/LCM2000.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2000 <-rast("./data/CEH_landcover_rasters/london/LCM2000.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2000 <-rast("./data/CEH_landcover_rasters/eastern/LCM2000.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2000 <- mosaic(se_2000, lon_2000, fun = "first")
lc_2000 <- mosaic(lc_2000, east_2000, fun = "first")

# colours and levels added
clr <- read_delim("./data/CEH_landcover_rasters/2000_colours.txt",
                 col_names = c("value", "red", "green", "blue"))

```



```

clr <- as.matrix(clr)
coltab(lc_2000) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/2000_levels.txt",
                  col_names = c("2000"))
levels(lc_2000) <- lvl
# take only land cover layer
lc_2000 <- lc_2000[[1]]
plot(lc_2000)

writeRaster(lc_2000, "./data/CEH_landcover_rasters/LCM2000.tif")

# extract variables for each landcover type
lc_vars_2000_1km <- zonal(lc_2000, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2000_3km <- zonal(lc_2000, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2000 <- left_join(lc_vars_2000_1km, lc_vars_2000_3km) %>%
  mutate(year = 2000)

# write to folder
write_csv(lc_vars_2000, "./data/landscape_variables/2000.csv")

#####      2007      #####

se_2007 <- rast("./data/CEH_landcover_rasters/south_east/LCM2007.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
se_2007[se_2007 == 0] <- NA
lon_2007 <-rast("./data/CEH_landcover_rasters/london/LCM2007.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2007[lon_2007 == 0] <- NA
east_2007 <-rast("./data/CEH_landcover_rasters/eastern/LCM2007.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2007[east_2007 == 0] <- NA

lc_2007 <- mosaic(se_2007, lon_2007, fun = "first")
lc_2007 <- mosaic(lc_2007, east_2007, fun = "first")

```

```

# colours and levels added
clr <- read_delim("./data/CEH_landcover_rasters/2007_colours.txt",
                  col_names = c("value", "red", "green", "blue"))
clr <- as.matrix(clr)
coltab(lc_2007) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/2007_levels.txt",
                  col_names = c("2007"))
levels(lc_2007) <- lvl
# take only land cover layer
lc_2007 <- lc_2007[[1]]
plot(lc_2007)

writeRaster(lc_2007, "./data/CEH_landcover_rasters/LCM2007.tif")

# extract variables for each landcover type
lc_vars_2007_1km <- zonal(lc_2007, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2007_3km <- zonal(lc_2007, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2007 <- left_join(lc_vars_2007_1km, lc_vars_2007_3km) %>%
  mutate(year = 2007)

# write to folder
write_csv(lc_vars_2007, "./data/landscape_variables/2007.csv")

#####      2015      #####

se_2015 <- rast("./data/CEH_landcover_rasters/south_east/LCM2015.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
# 2015 raster includes 0 values after mask for unknown reason. Removed here.
se_2015[se_2015 == 0] <- NA
lon_2015 <-rast("./data/CEH_landcover_rasters/london/LCM2015.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2015[lon_2015 == 0] <- NA
east_2015 <-rast("./data/CEH_landcover_rasters/eastern/LCM2015.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2015[east_2015 == 0] <- NA

```

```

lc_2015 <- mosaic(se_2015, lon_2015, fun = "first")
lc_2015 <- mosaic(lc_2015, east_2015, fun = "first")

# colours and levels added
clr <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_colours.txt",
                  col_names = c("value", "red", "green", "blue"))
clr <- as.matrix(clr)
coltab(lc_2015) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                  col_names = c("2015"))
levels(lc_2015) <- lvl
# take only land cover layer
lc_2015 <- lc_2015[[1]]
plot(lc_2015)

writeRaster(lc_2015, "./data/CEH_landcover_rasters/LCM2015.tif")

# extract variables for each landcover type
lc_vars_2015_1km <- zonal(lc_2015, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2015_3km <- zonal(lc_2015, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2015 <- left_join(lc_vars_2015_1km, lc_vars_2015_3km) %>%
  mutate(year = 2015)

# write to folder
write_csv(lc_vars_2015, "./data/landscape_variables/2015.csv")

#####      2017      #####

se_2017 <- rast("./data/CEH_landcover_rasters/south_east/LCM2017.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2017 <-rast("./data/CEH_landcover_rasters/london/LCM2017.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2017 <-rast("./data/CEH_landcover_rasters/eastern/LCM2017.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

```

```

lc_2017 <- mosaic(se_2017, lon_2017, fun = "first")
lc_2017 <- mosaic(lc_2017, east_2017, fun = "first")

# colours and levels added
coltab(lc_2017) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                  col_names = c("2017"))
levels(lc_2017) <- lvl
# take only land cover layer
lc_2017 <- lc_2017[[1]]
plot(lc_2017)

writeRaster(lc_2017, "./data/CEH_landcover_rasters/LCM2017.tif")

# extract variables for each landcover type
lc_vars_2017_1km <- zonal(lc_2017, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2017_3km <- zonal(lc_2017, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2017 <- left_join(lc_vars_2017_1km, lc_vars_2017_3km) %>%
  mutate(year = 2017)

# write to folder
write_csv(lc_vars_2017, "./data/landscape_variables/2017.csv")

#####      2018      #####

se_2018 <- rast("./data/CEH_landcover_rasters/south_east/LCM2018.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2018 <-rast("./data/CEH_landcover_rasters/london/LCM2018.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2018 <-rast("./data/CEH_landcover_rasters/eastern/LCM2018.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2018 <- mosaic(se_2018, lon_2018, fun = "first")
lc_2018 <- mosaic(lc_2018, east_2018, fun = "first")

```

```

# colours and levels added

coltab(lc_2018) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                  col_names = c("2018"))
levels(lc_2018) <- lvl
# take only land cover layer
lc_2018 <- lc_2018[[1]]
plot(lc_2018)

writeRaster(lc_2018, "./data/CEH_landcover_rasters/LCM2018.tif")

# extract variables for each landcover type
lc_vars_2018_1km <- zonal(lc_2018, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2018_3km <- zonal(lc_2018, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2018 <- left_join(lc_vars_2018_1km, lc_vars_2018_3km) %>%
  mutate(year = 2018)

# write to folder
write_csv(lc_vars_2018, "./data/landscape_variables/2018.csv")

#####      2019      #####

se_2019 <- rast("./data/CEH_landcover_rasters/south_east/LCM2019.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2019 <-rast("./data/CEH_landcover_rasters/london/LCM2019.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2019 <-rast("./data/CEH_landcover_rasters/eastern/LCM2019.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2019 <- mosaic(se_2019, lon_2019, fun = "first")
lc_2019 <- mosaic(lc_2019, east_2019, fun = "first")
# colours and levels added
coltab(lc_2019) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",

```

```

        col_names = c("2019"))
levels(lc_2019) <- lvl
# take only land cover layer
lc_2019 <- lc_2019[[1]]
plot(lc_2019)

writeRaster(lc_2019, "../data/CEH_landcover_rasters/LCM2019.tif")

# extract variables for each landcover type
lc_vars_2019_1km <- zonal(lc_2019, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2019_3km <- zonal(lc_2019, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2019 <- left_join(lc_vars_2019_1km, lc_vars_2019_3km) %>%
  mutate(year = 2019)

# write to folder
write_csv(lc_vars_2019, "../data/landscape_variables/2019.csv")

#####      2020      #####

se_2020 <- rast("../data/CEH_landcover_rasters/south_east/LCM2020.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2020 <-rast("../data/CEH_landcover_rasters/london/LCM2020.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2020 <-rast("../data/CEH_landcover_rasters/eastern/LCM2020.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2020 <- mosaic(se_2020, lon_2020, fun = "first")
lc_2020 <- mosaic(lc_2020, east_2020, fun = "first")
# colours and levels added
coltab(lc_2020) <- clr
lvl <- read_delim("../data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                  col_names = c("2020"))
levels(lc_2020) <- lvl
# take only land cover layer
lc_2020 <- lc_2020[[1]]

```

```

plot(lc_2020)

writeRaster(lc_2020, "./data/CEH_landcover_rasters/LCM2020.tif")

# extract variables for each landcover type
lc_vars_2020_1km <- zonal(lc_2020, grids_vect_1km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2020_3km <- zonal(lc_2020, grids_vect_3km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2020 <- left_join(lc_vars_2020_1km, lc_vars_2020_3km) %>%
  mutate(year = 2020)

# write to folder
write_csv(lc_vars_2020, "./data/landscape_variables/2020.csv")

#####      2021      #####

se_2021 <- rast("./data/CEH_landcover_rasters/south_east/LCM2021.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2021 <-rast("./data/CEH_landcover_rasters/london/LCM2021.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2021 <-rast("./data/CEH_landcover_rasters/eastern/LCM2021.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2021 <- mosaic(se_2021, lon_2021, fun = "first")
lc_2021 <- mosaic(lc_2021, east_2021, fun = "first")
# colours and levels added
coltab(lc_2021) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                 col_names = c("2021"))
levels(lc_2021) <- lvl
# take only land cover layer
lc_2021 <- lc_2021[[1]]
plot(lc_2021)

writeRaster(lc_2021, "./data/CEH_landcover_rasters/LCM2021.tif")

# extract variables for each landcover type

```

```

lc_vars_2021_1km <- zonal(lc_2021, grids_vect_1km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2021_3km <- zonal(lc_2021, grids_vect_3km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2021 <- left_join(lc_vars_2021_1km, lc_vars_2021_3km) %>%
  mutate(year = 2021)

# write to folder
write_csv(lc_vars_2021, "./data/landscape_variables/2021.csv")

#####      2022      #####

se_2022 <- rast("./data/CEH_landcover_rasters/south_east/LCM2022.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2022 <- rast("./data/CEH_landcover_rasters/london/LCM2022.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2022 <- rast("./data/CEH_landcover_rasters/eastern/LCM2022.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2022 <- mosaic(se_2022, lon_2022, fun = "first")
lc_2022 <- mosaic(lc_2022, east_2022, fun = "first")
# colours and levels added
coltab(lc_2022) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                 col_names = c("2022"))
levels(lc_2022) <- lvl
# take only land cover layer
lc_2022 <- lc_2022[[1]]
plot(lc_2022)

writeRaster(lc_2022, "./data/CEH_landcover_rasters/LCM2022.tif")

# extract variables for each landcover type
lc_vars_2022_1km <- zonal(lc_2022, grids_vect_1km, fun = "table",
                        wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

```



```

lc_vars_2022_3km <- zonal(lc_2022, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

# combine datasets
lc_vars_2022 <- left_join(lc_vars_2022_1km, lc_vars_2022_3km) %>%
  mutate(year = 2022)

# write to folder
write_csv(lc_vars_2022, "./data/landscape_variables/2022.csv")

#####      2023      #####

se_2023 <- rast("./data/CEH_landcover_rasters/south_east/LCM2023.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
lon_2023 <- rast("./data/CEH_landcover_rasters/london/LCM2023.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
east_2023 <- rast("./data/CEH_landcover_rasters/eastern/LCM2023.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

lc_2023 <- mosaic(se_2023, lon_2023, fun = "first")
lc_2023 <- mosaic(lc_2023, east_2023, fun = "first")
# colours and levels added
coltab(lc_2023) <- clr
lvl <- read_delim("./data/CEH_landcover_rasters/1990_and_2015_onwards_levels.txt",
                  col_names = c("2023"))
levels(lc_2023) <- lvl
# take only land cover layer
lc_2023 <- lc_2023[[1]]
plot(lc_2023)

writeRaster(lc_2023, "./data/CEH_landcover_rasters/LCM2023.tif")

# extract variables for each landcover type
lc_vars_2023_1km <- zonal(lc_2023, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), ~"zone") %>%
  mutate(zone = grid_names)

lc_vars_2023_3km <- zonal(lc_2023, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), ~"zone") %>%
  mutate(zone = grid_names)

```

```

# combine datasets
lc_vars_2023 <- left_join(lc_vars_2023_1km, lc_vars_2023_3km) %>%
  mutate(year = 2023)

# write to folder
write_csv(lc_vars_2023, "./data/landscape_variables/2023.csv")

# save all as raster stack - though note that these need individual colour
# tables re-adding when they are read back in
lc_all_rasters <- c(lc_1990, lc_2000, lc_2007, lc_2015, lc_2017, lc_2018,
  lc_2019, lc_2020, lc_2021, lc_2022, lc_2023)

names(lc_all_rasters) <- c("1990", "2000", "2007", "2015", "2017", "2018",
  "2019", "2020", "2021", "2022", "2023")

writeRaster(lc_all_rasters, "./data/CEH_landcover_rasters/all_years.tif",
  overwrite = TRUE)

lc_all_years <- bind_rows(lc_vars_1990, lc_vars_2000, lc_vars_2007, lc_vars_2015,
  lc_vars_2017, lc_vars_2018, lc_vars_2019, lc_vars_2020,
  lc_vars_2021, lc_vars_2022, lc_vars_2023)

write_csv(lc_all_years, "./data/landscape_variables/all_years.csv")
# standardise land cover variables
lc_standardised <- lc_all_years %>%
  rowwise %>%
  mutate(total_arable_1km = sum(arable_1km, arable_horticulture_1km,
    arable_cereal_1km, arable_non_rotational_1km,
    na.rm=TRUE),
    total_arable_3km = sum(arable_3km, arable_horticulture_3km,
    arable_cereal_3km, arable_non_rotational_3km,
    na.rm = TRUE),
    total_sn_grassland_1km = sum(rough_grassland_1km, neutral_grassland_1km,
    bracken_grassland_1km,
    acid_grassland_1km, calcareous_grassland_1km,
    fen_marsh_swamp_1km, na.rm = TRUE),
    total_sn_grassland_3km = sum(rough_grassland_3km, neutral_grassland_3km,
    bracken_grassland_3km,
    acid_grassland_3km, calcareous_grassland_3km,
    fen_marsh_swamp_3km, na.rm = TRUE),
    total_mbh_1km = sum(heather_1km, heather_grassland_1km, inland_rock_1km,
    na.rm = TRUE),
    total_mbh_3km = sum(heather_3km, heather_grassland_3km, inland_rock_3km,
    na.rm = TRUE),

```

```

total_coastal_1km = sum(supralittoral_rock_1km, supralittoral_sediment_1km,
                        littoral_rock_1km, littoral_sediment_1km,
                        saltmarsh_1km, na.rm= TRUE),
total_coastal_3km = sum(supralittoral_rock_3km, supralittoral_sediment_3km,
                        littoral_rock_3km, littoral_sediment_3km,
                        saltmarsh_3km, na.rm = TRUE),
total_urban_suburban_1km = sum(urban_1km, suburban_1km,
                                na.rm = TRUE),
total_urban_suburban_3km = sum(urban_3km, suburban_3km,
                                na.rm = TRUE),
total_deciduous_1km = deciduous_woodland_1km,
total_deciduous_3km = deciduous_woodland_3km,
total_coniferous_1km = coniferous_woodland_1km,
total_coniferous_3km = coniferous_woodland_3km,
total_im_grassland_1km = sum(improved_grassland_1km, setaside_grassland_1km,
                              na.rm = TRUE),
total_im_grassland_3km = sum(improved_grassland_3km, setaside_grassland_3km,
                              na.rm = TRUE),
total_saltwater_1km = saltwater_1km,
total_saltwater_3km = saltwater_3km,
total_freshwater_1km = freshwater_1km,
total_freshwater_3km = freshwater_3km,
total_urban_1km = urban_1km,
total_urban_3km = urban_3km,
total_suburban_1km = suburban_1km,
total_suburban_3km = suburban_3km,
total_imgrass_arable_1km = sum(total_arable_1km, total_im_grassland_1km),
total_imgrass_arable_3km = sum(total_arable_3km, total_im_grassland_3km)) %>%
relocate(year, .after = zone) %>%
dplyr::select(-c(3:56))

# add percentage cover variables (1600 squares per 1km grid, 14400 squares per
# 3km grid)

vars_1km <- str_subset(colnames(lc_standardised), "1km")
vars_3km <- str_subset(colnames(lc_standardised), "3km")

lc_standardised %<>%
  mutate(across(all_of(vars_1km), ~ (./1600), .names = "{col}_perc"),
         across(all_of(vars_3km), ~ (./14400), .names = "{col}_perc"))

perc_vars_1km <- str_subset(colnames(lc_standardised), "1km_perc")
perc_vars_3km <- str_subset(colnames(lc_standardised), "3km_perc")

for (col in colnames(lc_standardised)[-c(1:28)]) {
  var_attributes <- str_extract(col, "(?<=total_).+(?=_perc)")
  new_col_name <- paste0(var_attributes, "_scale")
  lc_standardised[[new_col_name]] <- scale(lc_standardised[col])[,1]
}

```

```
}
```

```
write_csv(lc_standardised, "./data/landscape_variables/standardised.csv")
```

```
#####          EXTRACT URBAN GROWTH VARIABLES          #####
```

```
# urban growth variables are extracted for 2017 - 2023, with 2017 change  
# based on 2015, and all others based on year before. All others have too large  
# a time gap to make judging expansion reasonable.
```

```
#####          2015          #####
```

```
# new raster is created to include only urban and suburban classes, then  
# raster is transformed into patches
```

```
ug_2015 <- max(seggregate(lc_2015, classes = c(20,21)))  
ug_2015 <- patches(ug_2015, zeroAsNA = TRUE, directions = 8)
```

```
# patches under 20ha are removed, in line with UK built up area classifications
```

```
ug_2015 <- zonal(cellSize(ug_2015, unit = "ha"),  
                 ug_2015, sum, as.raster = TRUE)
```

```
ug_2015 <- ifel(ug_2015 < 20, NA, 1)  
ug_2015_polygons <- as.polygons(ug_2015)
```

```
# base and 'holes' polygons created to allow for classification of growth in  
# 2017
```

```
base_2015 <- st_as_sf(ug_2015_polygons)  
holes_2015 <- st_as_sf(fillHoles(ug_2015_polygons, inverse = TRUE))
```

```
#####          2017          #####
```

```
# new raster is created to include only urban and suburban classes, then  
# raster is transformed into patches
```

```
ug_2017 <- max(seggregate(lc_2017, classes = c(20,21)))  
ug_2017 <- patches(ug_2017, zeroAsNA = TRUE, directions = 8)
```

```
# patches under 20ha are removed, in line with UK built up area classifications
```

```
ug_2017 <- zonal(cellSize(ug_2017, unit = "ha"),  
                 ug_2017, sum, as.raster = TRUE)
```

```
ug_2017 <- ifel(ug_2017 < 20, NA, 1)  
ug_2017_polygons <- as.polygons(ug_2017)
```

```
# base and 'holes' polygons created to allow for classification of growth in  
# next year
```

```
base_2017 <- st_as_sf(ug_2017_polygons)
```

```

holes_2017 <- st_as_sf(fillHoles(ug_2017_polygons, inverse = TRUE))

# create set of new patches
new_patches_2017 <- erase(ug_2017_polygons, ug_2015_polygons)
# classify pattern type of new parts of patches

new_patches_2017 <- st_cast(st_as_sf(new_patches_2017), "POLYGON") %>%
  rename(ref = area) %>%
  mutate(ref = row_number(),
         edge = st_intersects(.$geometry, base_2015$geometry,
                              sparse = TRUE, dist = 1) %>% lengths>0,
         hole = st_intersects(.$geometry, holes_2015$geometry, sparse = TRUE) %>% lengths>0,
         pattern = case_when(hole ~ "infilling",
                              edge ~ "edge expansion",
                              .default = "outlying"))

# convert back to raster
ugp_2017_rast <- rasterize(new_patches_2017, lc_2017, field = "pattern")

#####      2018      #####

# new raster is created to include only urban and suburban classes, then
# raster is transformed into patches
ug_2018 <- max(seggregate(lc_2018, classes = c(20,21)))
ug_2018 <- patches(ug_2018, zeroAsNA = TRUE, directions = 8)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2018 <- zonal(cellSize(ug_2018, unit = "ha"),
                 ug_2018, sum, as.raster = TRUE)
ug_2018 <- ifel(ug_2018 < 20, NA, 1)
ug_2018_polygons <- as.polygons(ug_2018)

# base and 'holes' polygons created to allow for classification of growth in
# next year
base_2018 <- st_as_sf(ug_2018_polygons)
holes_2018 <- st_as_sf(fillHoles(ug_2018_polygons, inverse = TRUE))

# create set of new patches
new_patches_2018 <- erase(ug_2018_polygons, ug_2017_polygons)
# classify pattern type of new parts of patches
new_patches_2018 <- st_cast(st_as_sf(new_patches_2018), "POLYGON") %>%
  rename(ref = area) %>%
  mutate(ref = row_number(),
         edge = st_intersects(.$geometry, base_2017$geometry, sparse = TRUE) %>% lengths>0,
         hole = st_intersects(.$geometry, holes_2017$geometry, sparse = TRUE) %>% lengths>0,
         pattern = case_when(hole ~ "infilling",
                              edge ~ "edge expansion",

```

```

                                .default = "outlying"))
# convert back to raster
ugp_2018_rast <- rasterize(new_patches_2018, lc_2018, field = "pattern")

#####      2019      #####

# new raster is created to include only urban and suburban classes, then
# raster is transformed into patches
ug_2019 <- max(segregate(lc_2019, classes = c(20,21)))
ug_2019 <- patches(ug_2019, zeroAsNA = TRUE, directions = 8)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2019 <- zonal(cellSize(ug_2019, unit = "ha"),
                 ug_2019, sum, as.raster = TRUE)
ug_2019 <- ifel(ug_2019 < 20, NA, 1)
ug_2019_polygons <- as.polygons(ug_2019)

# base and 'holes' polygons created to allow for classification of growth in
# next year
base_2019 <- st_as_sf(ug_2019_polygons)
holes_2019 <- st_as_sf(fillHoles(ug_2019_polygons, inverse = TRUE))

# create set of new patches
new_patches_2019 <- erase(ug_2019_polygons, ug_2018_polygons)
# classify pattern type of new parts of patches
new_patches_2019 <- st_cast(st_as_sf(new_patches_2019), "POLYGON") %>%
  rename(ref = area) %>%
  mutate(ref = row_number(),
         edge = st_intersects(.$geometry, base_2018$geometry, sparse = TRUE) %>% lengths
         hole = st_intersects(.$geometry, holes_2018$geometry, sparse = TRUE) %>% lengths
         pattern = case_when(hole ~ "infilling",
                             edge ~ "edge expansion",
                             .default = "outlying"))
# convert back to raster
ugp_2019_rast <- rasterize(new_patches_2019, lc_2019, field = "pattern")

#####      2020      #####

# new raster is created to include only urban and suburban classes, then
# raster is transformed into patches
ug_2020 <- max(segregate(lc_2020, classes = c(20,21)))
ug_2020 <- patches(ug_2020, zeroAsNA = TRUE, directions = 8)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2020 <- zonal(cellSize(ug_2020, unit = "ha"),

```

```

      ug_2020, sum, as.raster = TRUE)
ug_2020 <- ifel(ug_2020 < 20, NA, 1)
ug_2020_polygons <- as.polygons(ug_2020)

# base and 'holes' polygons created to allow for classification of growth in
# next year
base_2020 <- st_as_sf(ug_2020_polygons)
holes_2020 <- st_as_sf(fillHoles(ug_2020_polygons, inverse = TRUE))

# create set of new patches
new_patches_2020 <- erase(ug_2020_polygons, ug_2019_polygons)
# classify pattern type of new parts of patches
new_patches_2020 <- st_cast(st_as_sf(new_patches_2020), "POLYGON") %>%
  rename(ref = area) %>%
  mutate(ref = row_number(),
         edge = st_intersects(.$geometry, base_2019$geometry, sparse = TRUE) %>% length,
         hole = st_intersects(.$geometry, holes_2019$geometry, sparse = TRUE) %>% length,
         pattern = case_when(hole ~ "infilling",
                             edge ~ "edge expansion",
                             .default = "outlying"))

# convert back to raster
ugp_2020_rast <- rasterize(new_patches_2020, lc_2020, field = "pattern")

#####      2021      #####

# new raster is created to include only urban and suburban classes, then
# raster is transformed into patches
ug_2021 <- max(segregate(lc_2021, classes = c(20,21)))
ug_2021 <- patches(ug_2021, zeroAsNA = TRUE, directions = 8)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2021 <- zonal(cellSize(ug_2021, unit = "ha"),
                 ug_2021, sum, as.raster = TRUE)
ug_2021 <- ifel(ug_2021 < 20, NA, 1)
ug_2021_polygons <- as.polygons(ug_2021)

# base and 'holes' polygons created to allow for classification of growth in
# next year
base_2021 <- st_as_sf(ug_2021_polygons)
holes_2021 <- st_as_sf(fillHoles(ug_2021_polygons, inverse = TRUE))

# create set of new patches
new_patches_2021 <- erase(ug_2021_polygons, ug_2020_polygons)
# classify pattern type of new parts of patches
new_patches_2021 <- st_cast(st_as_sf(new_patches_2021), "POLYGON") %>%
  rename(ref = area) %>%

```



```

mutate(ref = row_number(),
       edge = st_intersects(.$geometry, base_2020$geometry, sparse = TRUE) %>% lengths
       hole = st_intersects(.$geometry, holes_2020$geometry, sparse = TRUE) %>% lengths
       pattern = case_when(hole ~ "infilling",
                           edge ~ "edge expansion",
                           .default = "outlying"))

# convert back to raster
ugp_2021_rast <- rasterize(new_patches_2021, lc_2021, field = "pattern")

#####      2022      #####

# new raster is created to include only urban and suburban classes, then
# raster is transformed into patches
ug_2022 <- max(seggregate(lc_2022, classes = c(20,21)))
ug_2022 <- patches(ug_2022, zeroAsNA = TRUE, directions = 8)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2022 <- zonal(cellSize(ug_2022, unit = "ha"),
                 ug_2022, sum, as.raster = TRUE)
ug_2022 <- ifel(ug_2022 < 20, NA, 1)
ug_2022_polygons <- as.polygons(ug_2022)

# base and 'holes' polygons created to allow for classification of growth in
# next year
base_2022 <- st_as_sf(ug_2022_polygons)
holes_2022 <- st_as_sf(fillHoles(ug_2022_polygons, inverse = TRUE))

# create set of new patches
new_patches_2022 <- erase(ug_2022_polygons, ug_2021_polygons)
# classify pattern type of new parts of patches
new_patches_2022 <- st_cast(st_as_sf(new_patches_2022), "POLYGON") %>%
  rename(ref = area) %>%
  mutate(ref = row_number(),
         edge = st_intersects(.$geometry, base_2021$geometry, sparse = TRUE) %>% lengths
         hole = st_intersects(.$geometry, holes_2021$geometry, sparse = TRUE) %>% lengths
         pattern = case_when(hole ~ "infilling",
                             edge ~ "edge expansion",
                             .default = "outlying"))

# convert back to raster
ugp_2022_rast <- rasterize(new_patches_2022, lc_2022, field = "pattern")

#####      2023      #####

# new raster is created to include only urban and suburban classes, then
# raster is transformed into patches

```



```

ug_2023 <- max(segregate(lc_2023, classes = c(20,21)))
ug_2023 <- patches(ug_2023, zeroAsNA = TRUE, directions = 8)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2023 <- zonal(cellSize(ug_2023, unit = "ha"),
                 ug_2023, sum, as.raster = TRUE)
ug_2023 <- ifel(ug_2023 < 20, NA, 1)
ug_2023_polygons <- as.polygons(ug_2023)

# base and 'holes' polygons created to allow for classification of growth in
# next year
base_2023 <- st_as_sf(ug_2023_polygons)
holes_2023 <- st_as_sf(fillHoles(ug_2023_polygons, inverse = TRUE))

# create set of new patches
new_patches_2023 <- erase(ug_2023_polygons, ug_2022_polygons)
# classify pattern type of new parts of patches
new_patches_2023 <- st_cast(st_as_sf(new_patches_2023), "POLYGON") %>%
  rename(ref = area) %>%
  mutate(ref = row_number(),
         edge = st_intersects(.$geometry, base_2022$geometry, sparse = TRUE) %>% lengths
         hole = st_intersects(.$geometry, holes_2022$geometry, sparse = TRUE) %>% lengths
         pattern = case_when(hole ~ "infilling",
                             edge ~ "edge expansion",
                             .default = "outlying"))

# convert back to raster, combine, and save raster of all urban growth pattern
# years
ugp_2023_rast <- rasterize(new_patches_2023, lc_2023, field = "pattern")

ugp_all_rasters <- c(ugp_2017_rast, ugp_2018_rast, ugp_2019_rast,
                    ugp_2020_rast, ugp_2021_rast, ugp_2022_rast,
                    ugp_2023_rast)
names(ugp_all_rasters) <- c("2017", "2018", "2019", "2020", "2021", "2022",
                           "2023")

writeRaster(ugp_all_rasters, "./data/urban_growth_variables/all_years.tif",
            overwrite = TRUE)

# create urban cover rasters for 1990, 2000, and 2007, then combine all and
# save
ug_1990 <- max(segregate(lc_1990, classes = c(20,21)))
ug_1990 <- patches(ug_1990, zeroAsNA = TRUE)

# patches under 20ha are removed, in line with UK built up area classifications
ug_1990 <- zonal(cellSize(ug_1990, unit = "ha"),
                 ug_1990, sum, as.raster = TRUE)
ug_1990 <- ifel(ug_1990 < 20, NA, 1)

```

```

ug_2000 <- max(segregate(lc_2000, classes = c(171,172)))
ug_2000 <- patches(ug_2000, zeroAsNA = TRUE)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2000 <- zonal(cellSize(ug_2000, unit = "ha"),
                 ug_2000, sum, as.raster = TRUE)
ug_2000 <- ifel(ug_2000 < 20, NA, 1)

ug_2007 <- max(segregate(lc_2007, classes = c(22,23)))
ug_2007 <- patches(ug_2007, zeroAsNA = TRUE)

# patches under 20ha are removed, in line with UK built up area classifications
ug_2007 <- zonal(cellSize(ug_2007, unit = "ha"),
                 ug_2007, sum, as.raster = TRUE)
ug_2007 <- ifel(ug_2007 < 20, NA, 1)

# combine all and save to file
urban_cover_rast <- c(ug_1990, ug_2000, ug_2007, ug_2015, ug_2017, ug_2018,
                     ug_2019, ug_2020, ug_2021, ug_2022, ug_2023)

names(urban_cover_rast) <- c("1990", "2000", "2007", "2015", "2017", "2018",
                             "2019", "2020", "2021", "2022", "2023")

writeRaster(urban_cover_rast, "./data/urban_growth_variables/cover_all_years.tif",
            overwrite = TRUE)

ug_vars_2017_1km <- zonal(ugp_2017_rast, grids_vect_1km, fun = "table",
                         wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

ug_vars_2017_3km <- zonal(ugp_2017_rast, grids_vect_3km, fun = "table",
                         wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2017 <- left_join(ug_vars_2017_1km, ug_vars_2017_3km) %>%
  mutate(year = 2017)

write_csv(ugp_vars_2017, "./data/urban_growth_variables/2017.csv")

ug_vars_2018_1km <- zonal(ugp_2018_rast, grids_vect_1km, fun = "table",
                         wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

ug_vars_2018_3km <- zonal(ugp_2018_rast, grids_vect_3km, fun = "table",

```

```

                                wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2018 <- left_join(ug_vars_2018_1km, ug_vars_2018_3km) %>%
  mutate(year = 2018)

write_csv(ugp_vars_2018, "./data/urban_growth_variables/2018.csv")

ug_vars_2019_1km <- zonal(ugp_2019_rast, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

ug_vars_2019_3km <- zonal(ugp_2019_rast, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2019 <- left_join(ug_vars_2019_1km, ug_vars_2019_3km) %>%
  mutate(year = 2019)

write_csv(ugp_vars_2019, "./data/urban_growth_variables/2019.csv")

ug_vars_2020_1km <- zonal(ugp_2020_rast, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

ug_vars_2020_3km <- zonal(ugp_2020_rast, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2020 <- left_join(ug_vars_2020_1km, ug_vars_2020_3km) %>%
  mutate(year = 2020)

write_csv(ugp_vars_2020, "./data/urban_growth_variables/2020.csv")

ug_vars_2021_1km <- zonal(ugp_2021_rast, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

```

```
ug_vars_2021_3km <- zonal(ugp_2021_rast, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2021 <- left_join(ug_vars_2021_1km, ug_vars_2021_3km) %>%
  mutate(year = 2021)

write_csv(ugp_vars_2021, "./data/urban_growth_variables/2021.csv")

ug_vars_2022_1km <- zonal(ugp_2022_rast, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

ug_vars_2022_3km <- zonal(ugp_2022_rast, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2022 <- left_join(ug_vars_2022_1km, ug_vars_2022_3km) %>%
  mutate(year = 2022)

write_csv(ugp_vars_2022, "./data/urban_growth_variables/2022.csv")

ug_vars_2023_1km <- zonal(ugp_2023_rast, grids_vect_1km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_1km"), -"zone") %>%
  mutate(zone = grid_names)

ug_vars_2023_3km <- zonal(ugp_2023_rast, grids_vect_3km, fun = "table",
                          wide = TRUE) %>%
  rename_with(~ paste0(., "_3km"), -"zone") %>%
  mutate(zone = grid_names)

ugp_vars_2023 <- left_join(ug_vars_2023_1km, ug_vars_2023_3km) %>%
  mutate(year = 2023)

write_csv(ugp_vars_2023, "./data/urban_growth_variables/2023.csv")

# combine all years

ugp_all_years <- bind_rows(ugp_vars_2017, ug_vars_2018, ug_vars_2019,
```

```

    ugp_vars_2020, ugp_vars_2021, ugp_vars_2022,
    ugp_vars_2023)

# standardise urban growth variables

ugp_standardised <- ugp_all_years %>%
  rowwise %>%
  mutate(total_urban_growth_1km = sum(infilling_1km, `edge expansion_1km`,
                                     outlying_1km, na.rm = TRUE),
         total_urban_growth_3km = sum(infilling_3km, `edge expansion_3km`,
                                     outlying_3km, na.rm = TRUE)) %>%
  relocate(year, .after = zone)

# add percentage cover variables (1600 squares per 1km grid, 14400 squares per
# 3km grid)

vars_1km <- str_subset(colnames(ugp_standardised), "1km")
vars_3km <- str_subset(colnames(ugp_standardised), "3km")

ugp_standardised %<>%
  mutate(across(vars_1km, ~ (./1600), .names = "{col}_perc"),
         across(vars_3km, ~ (./14400), .names = "{col}_perc"))

ugp_standardised$growth_1km_scale <- scale(ugp_standardised$total_urban_growth_1km_perc)
ugp_standardised$growth_3km_scale <- scale(ugp_standardised$total_urban_growth_3km_perc)

growth_1km <- unlist(ugp_standardised[,c("infilling_1km_perc", "edge expansion_1km_perc",
                                         "outlying_1km_perc")])
mean_growth_1km <- mean(growth_1km)
sd_growth_1km <- sd(growth_1km)

ugp_standardised$infilling_1km_scale <- (ugp_standardised$infilling_1km_perc - mean_growth_1km)
ugp_standardised$edge_expansion_1km_scale <- (ugp_standardised$edge_expansion_1km_perc - mean_growth_1km)
ugp_standardised$outlying_1km_scale <- (ugp_standardised$outlying_1km_perc - mean_growth_1km)

growth_3km <- unlist(ugp_standardised[,c("infilling_3km_perc", "edge expansion_3km_perc",
                                         "outlying_3km_perc")])
mean_growth_3km <- mean(growth_3km)
sd_growth_3km <- sd(growth_3km)

ugp_standardised$infilling_3km_scale <- (ugp_standardised$infilling_3km_perc - mean_growth_3km)
ugp_standardised$edge_expansion_3km_scale <- (ugp_standardised$edge_expansion_3km_perc - mean_growth_3km)
ugp_standardised$outlying_3km_scale <- (ugp_standardised$outlying_3km_perc - mean_growth_3km)

write_csv(ugp_standardised, "../data/urban_growth_variables/standardised.csv")

```

```
#####          CALCULATE LAND COVER FOR WHOLE OF KENT          #####
```

```
kent_boundary_spat <- vect(kent_boundary)
LCMall <- all_years %>%
  project(y=target_crs) %>%
  crop(y=kent_boundary_spat, mask = TRUE)
all_lc <- freq(LCMall, wide = TRUE) %>%
  mutate(total = rowSums(.),
    across(c(2:29), ~.x/1600))
```

819 3_standardise_obs_data.R

```
#####
#####          STANDARDISE OBSERVATION DATA          #####
#####          #####
#####          #####
#####
```

```
# this script identifies the 1km grid square an observation was made in, and
# collates records that have the same grid reference, year, season, and
# species/genus. It then produces a detection matrix based on the observation
# of other species/genera. It also removes observations outside of Kent, and
# appends some 'year' and 'season' attributes where this was not possible during
# data cleaning because of missing full dates.
```

```
# load in Kent boundary and 10km buffer
```

```
kent_boundary <- sf::st_read("./data/county_boundaries_2023_bfe/CTYUA_MAY_2023_UK_BFE.shp",
  crs = 27700) %>%
  filter(CTYUA23NM %in% c("Kent", "Medway")) %>%
  st_combine()
```

```
# load in observation data, add geometry based on grid reference, and filter to within
# Kent
```

```
obs_data <- read_csv("./data/clean/kbg_data_clean.csv") %>%
  mutate(osgridref = str_replace_all(osgridref, " ", ""))
```

```
east_north <- osg_parse(obs_data$osgridref,
  coord_system = "BNG")
```

```
obs_data$easting <- east_north$easting
obs_data$northing <- east_north$northing
```

```
obs_data %<>%
  filter(!is.na(easting)) %>%
  sf::st_as_sf(coords = c("easting", "northing"),
```

```

      crs = 27700) %>%
  filter(sf::st_intersects(geometry, kent_boundary, sparse = FALSE))

# load in 1km grid squares data
grid_1km <- sf::st_read("../data/grid_squares/1km_grid_region.shp") %>%
  st_transform(crs = st_crs(obs_data)) %>%
  filter(sf::st_intersects(geometry, kent_boundary, sparse = FALSE))

# add grid square to observation data
grids <- sf::st_intersects(obs_data, grid_1km)
obs_data$PLAN_NO <- grid_1km$PLAN_NO[unlist(grids)]
# remove 'grids' to save space
rm(grids)

# check no year data
date_check <- obs_data %>%
  filter(is.na(clean_date))

# add year and season to observation data, appending values where appropriate
obs_data %<>%
  mutate(clean_date = as.Date(clean_date),
    year = case_when(raw_date == "July/August 2018" ~ 2018,
      raw_date == "April to June 2016" ~ 2016,
      raw_date == "April to October 2016" ~ 2016,
      raw_date == "April, September & October 2016" ~ 2016,
      raw_date == "February 2016" ~ 2016,
      raw_date == "June, August & October 2016" ~ 2016,
      raw_date == "May to October 2016" ~ 2016,
      raw_date == "September to October 2016" ~ 2016,
      raw_date == "Summer 2015" ~ 2015,
      raw_date == "Winter 2015" ~ 2015,
      raw_date == "July-Sept 2019" ~ 2019,
      raw_date == "may-June 2019" ~ 2019,
      raw_date == "July/August 2018" ~ 2018,
      raw_date == "Sep-19" ~ 2019,
      raw_date == "July/August 2018" ~ 2018,
      raw_date == "Dec 17 - Jan 18" ~ 2017,
      .default = year(clean_date)),
    month = month(clean_date),
    season = case_when(month %in% c(1,2,3,11,12) ~ "hibernation",
      month %in% c(4,5,6,7,8,9,10) ~ "active",
      .default = NA),
    season = case_when(raw_date == "July/August 2018" ~ "active",
      raw_date == "April to June 2016" ~ "active",
      raw_date == "April to October 2016" ~ "active",
      raw_date == "April, September & October 2016" ~ "active",

```

```

raw_date == "Febuary 2016" ~ "hibernation",
raw_date == "June, August & October 2016" ~ "active",
raw_date == "May to October 2016" ~ "active",
raw_date == "September to October 2016" ~ "active",
raw_date == "Summer 2015" ~ "active",
raw_date == "Winter 2015" ~ "hibernation",
raw_date == "July-Sept 2019" ~ "active",
raw_date == "may-June 2019" ~ "active",
raw_date == "July/August 2018" ~ "active",
raw_date == "Sep-19" ~ "active",
raw_date == "July/August 2018" ~ "active",
raw_date == "Dec 17 - Jan 18" ~ "hibernation",
.default = season))

# add presence/absence column to whole dataframe
obs_data %<>%
  mutate(presence = case_when(common_name == "no bat" ~ 0,
                              count == 0 ~ 0,
                              .default = 1))

# convert grid square, season, and presence to factor columns, and drop geometry
# (as points data won't be used, 'PLAN_NO' gives grid reference)
obs_data %<>%
  mutate(PLAN_NO = as.factor(PLAN_NO),
         season = as.factor(season),
         presence = as.factor(presence)) %>%
  st_drop_geometry()

#####      CREATE GENERA AND SPECIES DETECTION MATRICES      #####

# 1st Tier - genus level. Group to genus level, then use observations of
# other genera to generate absences
obs_data %<>%
  mutate(genus = case_when(str_detect(common_name, "pipistrelle") ~
                           "pipistrelle",
                           str_detect(common_name, "daub|whisker|natterer|brandt|myotis|
                           "myotis",
                           common_name == "barbastelle bat" ~ "barbastella",
                           str_detect(common_name, "horseshoe") ~ "rhinolophus",
                           common_name == "serotine bat" ~ "eptesicus",
                           str_detect(common_name, "long-eared") ~ "plecotus",
                           str_detect(common_name, "leisler|noctule|nyctalus") ~
                           "nyctalus",
                           .default = NA),
         genus = as.factor(genus))

```



```

# filter to remove NAs or observations of 'bat' (which could be any genera),
# group by 1km grid, period, season, genus, and presence, then pivot wider
genus_presabs <- obs_data %>%
  filter(!is.na(genus)) %>%
  group_by(PLAN_NO, year, season, genus, presence) %>%
  summarize() %>%
  pivot_wider(names_from = genus, values_from = presence,
              values_fill = "0")

# recombine the genus-level dataset with grid square geometry, and save to csv
genus_presabs %<>%
  left_join(grid_1km) %>%
  st_as_sf()

st_write(genus_presabs, "../data/species_detection_matrices/det_genera.shp")

### create genera observations table for methods
table_obs_gen <- obs_data %>%
  filter(!is.na(genus)) %>%
  group_by(genus, presence) %>%
  summarize(num_obs = n(),
            first_obs = min(year, na.rm = TRUE),
            last_obs = max(year, na.rm = TRUE),
            dist_years = list(year),
            numb_obs_trunc = sum(year %in% c("2017", "2018", "2019", "2020",
                                             "2021", "2022", "2023")) %>%
  filter(presence == "1")

tab_hists <- obs_data %>%
  filter(!is.na(genus)) %>%
  group_by(genus, presence, PLAN_NO, year) %>%
  summarize(num_obs = n()) %>%
  filter(presence == "1")

for (gen in unique(tab_hists$genus)){
  dat_filt <- tab_hists %>%
    filter(genus == gen)
  plot_filt <- ggplot() +
    geom_histogram(data=dat_filt, aes(x = year), binwidth = 2) +
    xlim(1980,2023) +
    geom_vline(xintercept = 1980, linewidth = 2) +
    geom_vline(xintercept = 2023, linewidth = 2) +
    theme_void()
  ggsave(paste0("../images/dists/", gen, ".png"))
}

```

SPECIES GROUPS

*# species-wise absences are generated based on all observations made that
 # COULD NOT be the species of interest (e.g. records of 'pipistrelle species'
 # are not used as presence OR absence points of the Common Pipistrelle
 # specifically, they are removed from the detection matrix for that species).*

*# records of 'bat' and NA are removed at the beginning, as they are not used in
 # any absence generation*

```
sp_det <- obs_data %>%
  filter(!(common_name %in% c("bat", NA))) %>%
  group_by(PLAN_NO, year, season, common_name, presence) %>%
  summarize() %>%
  pivot_wider(names_from = common_name, values_from = presence,
              values_fill = "0")

sp_det %<>%
  ungroup() %>%
  mutate(
    # Common pipistrelle
    p_pip = case_when(`pipistrelle bat (45khz)` == "1" ~ "1",
                      `pipistrelle species` == "1" ~ NA,
                      `pipistrelle (45 or 55khz) species` == "1" ~ NA,
                      .default = "0"),
    p_pip = as.factor(p_pip),
    # Soprano pipistrelle
    p_pyg = case_when(`pipistrelle bat (55khz)` == "1" ~ "1",
                      `pipistrelle species` == "1" ~ NA,
                      `pipistrelle (45 or 55khz) species` == "1" ~ NA,
                      .default = "0"),
    p_pyg = as.factor(p_pyg),
    # Nathusius pipistrelle
    p_nat = case_when(`pipistrelle nathusius` == "1" ~ "1",
                      `pipistrelle species` == "1" ~ NA,
                      .default = "0"),
    p_nat = as.factor(p_nat),
    # Kuhls pipistrelle
    p_khl = case_when(`pipistrelle kuhls` == "1" ~ "1",
                      `pipistrelle species` == "1" ~ NA,
                      .default = "0"),
    p_khl = as.factor(p_khl),
    # Alcahthoe
    m_alc = case_when(`alcathoe bat` == "1" ~ "1",
                      `myotis species` == "1" ~ NA,
                      `whiskered / brandts / alcathoe bat` == "1" ~ NA,
```

```

        .default = "0"),
m_alc = as.factor(m_alc),
# Brandts / Whiskered (treated as species pair)
m_br_wh = case_when(`whiskered bat` == "1" ~ "1",
                    `brandts bat` == "1" ~ "1",
                    `whiskered / brandts bat` == "1" ~ "1",
                    `myotis species` == "1" ~ NA,
                    `whiskered / brandts / alcathoe bat` == "1" ~ NA,
                    `whiskered / brandts / daubentons bat` == "1" ~ NA,
                    .default = "0"),
m_br_wh = as.factor(m_br_wh),
# Bechsteins
m_bch = case_when(`bechsteins bat` == "1" ~ "1",
                  `myotis species` == "1" ~ NA,
                  .default = "0"),
m_bch = as.factor(m_bch),
# Natterers
m_nat = case_when(`natterers bat` == "1" ~ "1",
                  `myotis species` == "1" ~ NA,
                  .default = "0"),
m_nat = as.factor(m_nat),
# Grey mouse-eared
m_myo = case_when(`g.mouse-eared` == "1" ~ "1",
                  `myotis species` == "1" ~ NA,
                  .default = "0"),
m_myo = as.factor(m_myo),
# Daubentons
m_dau = case_when(`daubentons bat` == "1" ~ "1",
                  `myotis species` == "1" ~ NA,
                  `whiskered / brandts / daubentons bat` == "1" ~ NA,
                  .default = "0"),
m_dau = as.factor(m_dau),
# Brown long-eared
p_aur = case_when(`long-eared bat brown` == "1" ~ "1",
                  `long-eared species` == "1" ~ NA,
                  .default = "0"),
p_aur = as.factor(p_aur),
# Grey long-eared
p_aus = case_when(`long-eared bat grey` == "1" ~ "1",
                  `long-eared species` == "1" ~ NA,
                  .default = "0"),
p_aus = as.factor(p_aus),
# Greater horseshoe
r_fer = case_when(`greater horseshoe bat` == "1" ~ "1",
                  .default = "0"),
r_fer = as.factor(r_fer),
# no lesser horseshoe records
# Leislars

```

```

n_lsr = case_when(`leislars bat` == "1" ~ "1",
                  `nyctalus species` == "1" ~ NA,
                  .default = "0"),
n_lsr = as.factor(n_lsr),
# Noctule
n_noc = case_when(`noctule bat` == "1" ~ "1",
                  `nyctalus species` == "1" ~ NA,
                  .default = "0"),
n_noc = as.factor(n_noc),
# Barbastelle
b_bar = case_when(`barbastelle bat` == "1" ~ "1",
                  .default = "0"),
b_bar = as.factor(b_bar),
# Serotine
e_ser = case_when(`serotine bat` == "1" ~ "1",
                  .default = "0"),
e_ser = as.factor(e_ser)) %>%
# remove non-standardised columns
dplyr::select(c(PLAN_NO, year, season,
                p_pip, p_pyg, p_nat, p_khl,
                m_alc, m_br_wh, m_bch, m_nat, m_myo, m_dau,
                p_aur, p_aus,
                r_fer,
                n_lsr, n_noc,
                b_bar,
                e_ser))

# recombine the species-level dataset with grid square geometry, and save to csv
sp_det %<>%
  left_join(grid_1km) %>%
  st_as_sf()

st_write(sp_det, "./data/species_detection_matrices/det_species.shp")

# create species-level observations table for methods
table_obs_sp <- obs_data %>%
  mutate(common_name = ifelse(common_name %in% c("whiskered bat", "brandts bat",
                                                "whiskered / brandts bat"),
                              "whiskered / brandts bat", common_name)) %>%
  group_by(common_name, presence, PLAN_NO, year) %>%
  summarize(num_obs = n()) %>%
  filter(presence == "1",
         common_name %in% c("alcathe bat", "bechsteins bat", "daubentons bat",
                           "g.mouse-eared", "greater horseshoe bat", "leislars bat",
                           "long-eared bat brown", "long-eared bat grey",
                           "natterers bat", "noctule bat", "pipistrelle bat (45khz)",

```

```

        "pipistrelle bat (55khz)", "pipistrelle kuhls",
        "pipistrelle nathusius", "serotine bat",
        "whiskered / brandts bat"))

for (sp in unique(table_obs_sp$common_name)){
  dat_filt <- table_obs_sp %>%
    filter(common_name == sp)
  plot_filt <- ggplot() +
    geom_histogram(data=dat_filt, aes(x = year), binwidth = 2) +
    xlim(1980,2023) +
    geom_vline(xintercept = 1980, linewidth = 2) +
    geom_vline(xintercept = 2023, linewidth = 2) +
    theme_void()
  ggsave(paste0("./images/dists/", sp, ".png"))
}

# join two datasets together

table_obs_gen %<>%
  rename(common_name = genus) %>%
  bind_rows(table_obs_sp) %>%
  select(-presence) %>%
  ungroup() %>%
  gt() %>%
  gt_plt_dist(dist_years, type = "histogram", line_color = "grey30",
    fill_color = "grey30", bw = 2)

gtsave(table_obs_gen, "./images/observations.png", vheight = 2000)

```

820 4_attach_environmental_variables.R

```

#####
#####
##### CHECK ENVIRONMENTAL VARIABLES #####
#####
#####
#####

# this script finds appropriate range sizes for each genera and species

# read in genera and species detections and land cover variables and join
det_genera <- st_read("./data/species_detection_matrices/det_genera.shp")
det_species <- st_read("./data/species_detection_matrices/det_species.shp")

```

```

lc_vars <- read_csv("../data/landscape_variables/standardised.csv") %>%
  dplyr::select(c(zone, year, contains("scale")))

ugp_vars <- read_csv("../data/urban_growth_variables/standardised.csv") %>%
  dplyr::select(c(zone, year, contains("scale")))

# add lc_year and ugp_year column to det_genera to join from
det_genera %<>%
  mutate(lc_year = case_when(between(year, 1980, 1995) ~ 1990,
                             between(year, 1996, 2003) ~ 2000,
                             between(year, 2004, 2011) ~ 2007,
                             between(year, 2012, 2016) ~ 2015,
                             .default = year))

det_genera <- left_join(det_genera, lc_vars, by = c("PLAN_NO" = "zone",
                                                  "lc_year" = "year"))

det_genera %<>%
  mutate(across(c("eptescs", "ppstrll", "myotis", "rhnlpsh", "nyctals",
                  "brbstll", "plectts", "season"), ~ as.factor(.)))

# create 2017-2023 filtered dataset and add urban growth variables
det_genera_ug <- det_genera %>%
  filter(year > 2016) %>%
  left_join(ugp_vars, by = c("PLAN_NO" = "zone",
                            "year" = "year"))

# do same for species detections
det_species %<>%
  mutate(lc_year = case_when(between(year, 1980, 1995) ~ 1990,
                             between(year, 1996, 2003) ~ 2000,
                             between(year, 2004, 2011) ~ 2007,
                             between(year, 2012, 2016) ~ 2015,
                             .default = year))

det_species <- left_join(det_species, lc_vars, by = c("PLAN_NO" = "zone",
                                                  "lc_year" = "year"))

det_species %<>%
  mutate(across(c("p_pip", "p_pyg", "p_nat", "p_khl", "m_alc", "m_br_wh",
                  "m_bch", "m_nat", "m_myo", "m_dau", "p_aur", "p_aus", "r_fer",
                  "n_lsr", "n_noc", "b_bar", "e_ser", "season"), ~ as.factor(.)))

# create 2017-2023 filtered dataset and add urban growth variables
det_species_ug <- det_species %>%
  filter(year > 2016) %>%
  left_join(ugp_vars, by = c("PLAN_NO" = "zone",

```

```

    "year" = "year"))

det_genera_ug %<>%
  mutate(dominant_growth_3km = case_when(
    infilling_3km_scale == pmax(infilling_3km_scale, edge_expansion_3km_scale,
                                outlying_3km_scale) ~ "infilling",
    edge_expansion_3km_scale == pmax(infilling_3km_scale, edge_expansion_3km_scale,
                                      outlying_3km_scale) ~ "edge expansion",
    outlying_3km_scale == pmax(infilling_3km_scale, edge_expansion_3km_scale,
                                outlying_3km_scale) ~ "outlying",
    .default = "none"),
  dominant_growth_3km = as.factor(dominant_growth_3km),
  dominant_growth_1km = case_when(
    infilling_1km_scale == pmax(infilling_1km_scale, edge_expansion_1km_scale,
                                outlying_1km_scale) ~ "infilling",
    edge_expansion_1km_scale == pmax(infilling_1km_scale, edge_expansion_1km_scale,
                                      outlying_1km_scale) ~ "edge expansion",
    outlying_1km_scale == pmax(infilling_1km_scale, edge_expansion_1km_scale,
                                outlying_1km_scale) ~ "outlying",
    .default = "none"),
  dominant_growth_1km = as.factor(dominant_growth_1km))

det_species_ug %<>%
  mutate(dominant_growth_3km = case_when(
    infilling_3km_scale == pmax(infilling_3km_scale, edge_expansion_3km_scale,
                                outlying_3km_scale) ~ "infilling",
    edge_expansion_3km_scale == pmax(infilling_3km_scale, edge_expansion_3km_scale,
                                      outlying_3km_scale) ~ "edge expansion",
    outlying_3km_scale == pmax(infilling_3km_scale, edge_expansion_3km_scale,
                                outlying_3km_scale) ~ "outlying",
    .default = "none"),
  dominant_growth_3km = as.factor(dominant_growth_3km),
  dominant_growth_1km = case_when(
    infilling_1km_scale == pmax(infilling_1km_scale, edge_expansion_1km_scale,
                                outlying_1km_scale) ~ "infilling",
    edge_expansion_1km_scale == pmax(infilling_1km_scale, edge_expansion_1km_scale,
                                      outlying_1km_scale) ~ "edge expansion",
    outlying_1km_scale == pmax(infilling_1km_scale, edge_expansion_1km_scale,
                                outlying_1km_scale) ~ "outlying",
    .default = "none"),
  dominant_growth_1km = as.factor(dominant_growth_1km))

# save as final completed datasets
write_csv(det_genera, "./data/final_data/genera_lc_only.csv")
write_csv(det_genera_ug, "./data/final_data/genera_lc_ug.csv")
write_csv(det_species, "./data/final_data/species_lc_only.csv")
write_csv(det_species_ug, "./data/final_data/species_lc_ug.csv")

```

```
#####                                URBAN/SUBURBAN EFFECTS                                #####

# for all genera and species, two models are run - one which combines urban
# and suburban variables, and one that does not. The AIC values are then compared

seperate_list_gen <- list()

for (genera in genera_vars) {
  formula <- paste(genera, "~ total_urban_3km_perc +
                    total_suburban_3km_perc + (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
                           family = binomial(link="logit"),
                           data = det_genera,
                           na.action = na.exclude)
}, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
  seperate_list_gen[[genera]] <- model
}

combined_list_gen <- list()

for (genera in genera_vars) {
  formula <- paste(genera, "~ total_urban_suburban_3km_perc +
                    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
                           family = binomial(link="logit"),
                           data = det_genera,
                           na.action = na.exclude)
}, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
  combined_list_gen[[genera]] <- model
}

print(AIC(seperate_list_gen$plectts))
print(AIC(combined_list_gen$plectts))

print(AIC(seperate_list_gen$ppstrll))
print(AIC(combined_list_gen$ppstrll))

print(AIC(seperate_list_gen$myotis))
print(AIC(combined_list_gen$myotis))

print(AIC(seperate_list_gen$septescs))
```



```

print(AIC(combined_list_gen$septescs))

print(AIC(seperate_list_gen$nyctals))
print(AIC(combined_list_gen$nyctals))

#### SPECIES ####
seperate_list_sp <- list()

for (species in species_vars) {
  formula <- paste(species, "~ total_urban_3km_perc +
                    total_suburban_3km_perc + (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
                           family = binomial(link="logit"),
                           data = det_species,
                           na.action = na.exclude)
}, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
  seperate_list_sp[[species]] <- model
}

combined_list_sp <- list()

for (species in species_vars) {
  formula <- paste(species, "~ total_urban_suburban_3km_perc +
                    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
                           family = binomial(link="logit"),
                           data = det_species,
                           na.action = na.exclude)
}, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
  combined_list_sp[[species]] <- model
}

```

821 5_run_final_models.R

```

#####
#####
##### RUN FINAL MODELS #####
#####
#####
#####
# this script runs Generalised Linear Mixed Models for each genus and species

```

```
# grouping, with final set of environmental variables.
```

```
# create response variable set
```

```
genera_vars <- colnames(det_genera)[4:10]
species_vars <- colnames(det_species)[4:19]
```

```
# look at number of presence values for each genera and species
```

```
for (genera in genera_vars) {
  print(paste(genera, ", no. obs:", sum(det_genera[[genera]] == "1", na.rm = TRUE)))
  print(paste(genera, ", no. abs:", sum(det_genera[[genera]] == "0", na.rm = TRUE)))
}
```

```
for (species in species_vars) {
  print(paste(species, ", no. obs:", sum(det_species[[species]] == "1", na.rm = TRUE)))
  print(paste(species, ", no. abs:", sum(det_species[[species]] == "0", na.rm = TRUE)))
  print(paste(species, ", no. NA:", sum(is.na(det_species[[species]])))
}
```

```
##### MODEL 1 #####
```

```
# model 1 includes land cover variables only, on the whole dataset (1980 - 2023)
```

```
m1_list_gen <- list()
```

```
for (genera in genera_vars) {
  print(genera)
  formula <- paste(genera, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
    sn_grassland_3km_scale +
    urban_3km_scale + suburban_3km_scale +
    freshwater_3km_scale + coastal_3km_scale +
    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
    family = binomial(link="logit"),
    data = det_genera,
    control = glmerControl(optCtrl = list(maxfun = 20000)))
  }, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
  }, error = function (e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
  m1_list_gen[[genera]] <- model
}
```

```
m1_list_sp <- list()
```

```
for (species in species_vars) {
  print(species)
```

```

formula <- paste(species, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
                  sn_grassland_3km_scale +
                  urban_3km_scale + suburban_3km_scale +
                  freshwater_3km_scale + coastal_3km_scale +
                  (1|year) + (1|season)")
model <- tryCatch({glmer(formula,
                        family = binomial(link="logit"),
                        data = det_species)
}, warning = function(w) { message(paste("Model for:", species, "warning:", w$message))
  }, error = function(e) { message(paste("Model for", species, "failed:", e$message))
  return(NULL)})
m1_list_sp[[species]] <- model
}

# extract coefficients table

m1_genera_vars <- names(m1_list_gen)
m1_tables_gen <- list()

for (genera in m1_genera_vars) {
  fixedef <- coef(summary(m1_list_gen[[genera]]))
  conf <- confint(m1_list_gen[[genera]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
                  slope = fixedef[,1],
                  std_err = fixedef[,2],
                  p_value = fixedef[,4],
                  lci = conf[,1],
                  uci = conf[,2],
                  gen_sp = genera)
  m1_tables_gen[[genera]] <- table
}

m1_tables_gen <- bind_rows(m1_tables_gen) %>%
  mutate(model = "Model 1")

m1_species_vars <- names(m1_list_sp)
m1_tables_sp <- list()

for (species in m1_species_vars) {
  fixedef <- coef(summary(m1_list_sp[[species]]))
  conf <- confint(m1_list_sp[[species]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
                  slope = fixedef[,1],
                  std_err = fixedef[,2],
                  p_value = fixedef[,4],
                  lci = conf[,1],
                  uci = conf[,2],

```

```

        gen_sp = species)
    m1_tables_sp[[species]] <- table
  }
m1_tables_sp <- bind_rows(m1_tables_sp) %>%
  mutate(model = "Model 1")

#####                                MODEL 2                                #####

# model 2 reruns the analyses of Model 1, but on the truncated dataset when urban
# growth variables are available (2017 - 2023)

m2_list_gen <- list()

for (genera in genera_vars) {
  print(genera)
  formula <- paste(genera, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
    sn_grassland_3km_scale +
    urban_3km_scale + suburban_3km_scale +
    freshwater_3km_scale + coastal_3km_scale +
    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
    family = binomial(link="logit"),
    data = det_genera_ug)
  }, warning = function(w) { message(paste("Model for:", genera, "warning:", w$message))
  }, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
    return(NULL)})
  m2_list_gen[[genera]] <- model
}

m2_list_sp <- list()

for (species in species_vars) {
  print(species)
  formula <- paste(species, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
    sn_grassland_3km_scale +
    urban_3km_scale + suburban_3km_scale +
    freshwater_3km_scale + coastal_3km_scale +
    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
    family = binomial(link="logit"),
    data = det_species_ug)
  }, warning = function(w) { message(paste("Model for:", species, "warning:", w$message))
  }, error = function(e) { message(paste("Model for", species, "failed:", e$message))
    return(NULL)})

```

```

  m2_list_sp[[species]] <- model
}

m2_genera_vars <- names(m2_list_gen)
m2_tables_gen <- list()

for (genera in m2_genera_vars) {
  fixedef <- coef(summary(m2_list_gen[[genera]]))
  conf <- confint(m2_list_gen[[genera]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
                  slope = fixedef[,1],
                  std_err = fixedef[,2],
                  p_value = fixedef[,4],
                  lci = conf[,1],
                  uci = conf[,2],
                  gen_sp = genera)
  m2_tables_gen[[genera]] <- table
}

m2_tables_gen <- bind_rows(m2_tables_gen) %>%
  mutate(model = "Model 2")

m2_species_vars <- names(m2_list_sp)
m2_tables_sp <- list()

for (species in m2_species_vars) {
  fixedef <- coef(summary(m2_list_sp[[species]]))
  conf <- confint(m2_list_sp[[species]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
                  slope = fixedef[,1],
                  std_err = fixedef[,2],
                  p_value = fixedef[,4],
                  lci = conf[,1],
                  uci = conf[,2],
                  gen_sp = species)
  m2_tables_sp[[species]] <- table
}

m2_tables_sp <- bind_rows(m2_tables_sp) %>%
  mutate(model = "Model 2")

##### MODEL 3 #####

# model 3 includes the urban growth variable

m3_list_gen <- list()

for (genera in genera_vars) {

```

```

print(genera)
formula <- paste(genera, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
  sn_grassland_3km_scale +
  urban_3km_scale + suburban_3km_scale +
  freshwater_3km_scale + coastal_3km_scale +
  growth_3km_scale + (1|year) + (1|season)")
model <- tryCatch({glmer(formula,
  family = binomial(link="logit"),
  data = det_genera_ug)
}, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
m3_list_gen[[genera]] <- model
}

m3_list_sp <- list()

for (species in species_vars) {
  print(species)
  formula <- paste(species, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
    sn_grassland_3km_scale +
    urban_3km_scale + suburban_3km_scale +
    freshwater_3km_scale + coastal_3km_scale +
    growth_3km_scale + (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
    family = binomial(link="logit"),
    data = det_species_ug)
  }, warning = function (w) { message(paste("Model for:", species, "warning:", w$message))
  }, error = function(e) { message(paste("Model for", species, "failed:", e$message))
    return(NULL)})
  m3_list_sp[[species]] <- model
}

m3_genera_vars <- names(m3_list_gen)
m3_tables_gen <- list()

for (genera in m3_genera_vars) {
  fixedef <- coef(summary(m3_list_gen[[genera]]))
  conf <- confint(m3_list_gen[[genera]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
    slope = fixedef[,1],
    std_err = fixedef[,2],
    p_value = fixedef[,4],
    lci = conf[,1],
    uci = conf[,2],

```

```

        gen_sp = genera)
    m3_tables_gen[[genera]] <- table
  }

m3_tables_gen <- bind_rows(m3_tables_gen) %>%
  mutate(model = "Model 3")

m3_species_vars <- names(m3_list_sp)
m3_tables_sp <- list()

for (species in m3_species_vars) {
  fixedef <- coef(summary(m3_list_sp[[species]]))
  conf <- confint(m3_list_sp[[species]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
                  slope = fixedef[,1],
                  std_err = fixedef[,2],
                  p_value = fixedef[,4],
                  lci = conf[,1],
                  uci = conf[,2],
                  gen_sp = species)
  m3_tables_sp[[species]] <- table
}
m3_tables_sp <- bind_rows(m3_tables_sp) %>%
  mutate(model = "Model 3")

#####                MODEL 4                #####

# model 4 looks at the effect of urban growth pattern, by including dominant
# urban cover pattern

m4_list_gen <- list()

for (genera in genera_vars) {
  print(genera)
  formula <- paste(genera, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
                    sn_grassland_3km_scale +
                    urban_3km_scale + suburban_3km_scale +
                    freshwater_3km_scale + coastal_3km_scale +
                    growth_3km_scale + dominant_growth_3km +
                    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
                          family = binomial(link="logit"),
                          data = det_genera_ug,
                          na.action = na.exclude)
}, warning = function(w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})

```

```

m4_list_gen[[genera]] <- model

}

m4_list_sp <- list()

for (species in species_vars) {
  print(species)
  formula <- paste(species, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
    sn_grassland_3km_scale +
    urban_3km_scale + suburban_3km_scale +
    freshwater_3km_scale + coastal_3km_scale +
    growth_3km_scale + dominant_growth_3km +
    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
    family = binomial(link="logit"),
    data = det_species_ug,
    na.action = na.exclude)
  }, warning = function (w) { message(paste("Model for:", species, "warning:", w$message))
  }, error = function(e) { message(paste("Model for", species, "failed:", e$message))
    return(NULL)})
  m4_list_sp[[species]] <- model
}

m4_genera_vars <- names(m4_list_gen)
m4_tables_gen <- list()

for (genera in m4_genera_vars) {
  fixedef <- coef(summary(m4_list_gen[[genera]]))
  conf <- confint(m4_list_gen[[genera]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
    slope = fixedef[,1],
    std_err = fixedef[,2],
    p_value = fixedef[,4],
    lci = conf[,1],
    uci = conf[,2],
    gen_sp = genera)
  m4_tables_gen[[genera]] <- table
}

m4_tables_gen <- bind_rows(m4_tables_gen) %>%
  mutate(model = "Model 4")

m4_species_vars <- names(m4_list_sp)
m4_tables_sp <- list()

```



```

for (species in m4_species_vars) {
  fixedef <- coef(summary(m4_list_sp[[species]]))
  conf <- confint(m4_list_sp[[species]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
                  slope = fixedef[,1],
                  std_err = fixedef[,2],
                  p_value = fixedef[,4],
                  lci = conf[,1],
                  uci = conf[,2],
                  gen_sp = species)
  m4_tables_sp[[species]] <- table
}
m4_tables_sp <- bind_rows(m4_tables_sp) %>%
  mutate(model = "Model 4")

```

```

#####                                MODEL 5                                #####

```

```

# model 5 looks at the effect of urban growth pattern, by including dominant
# urban cover pattern as an interactive variable with urban growth pattern

```

```

m5_list_gen <- list()

```

```

for (genera in genera_vars) {
  print(genera)
  formula <- paste(genera, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
                    sn_grassland_3km_scale +
                    urban_3km_scale + suburban_3km_scale +
                    freshwater_3km_scale + coastal_3km_scale +
                    growth_3km_scale*dominant_growth_3km +
                    (1|year) + (1|season)")
  model <- tryCatch({glmer(formula,
                          family = binomial(link="logit"),
                          data = det_genera_ug,
                          na.action = na.exclude)
}, warning = function (w) { message(paste("Model for:", genera, "warning:", w$message))
}, error = function(e) { message(paste("Model for", genera, "failed:", e$message))
  return(NULL)})
  m5_list_gen[[genera]] <- model
}

```

```

m5_list_sp <- list()

```

```

for (species in species_vars) {
  print(species)

```

```

formula <- paste(species, "~ deciduous_3km_scale + imgrass_arable_3km_scale +
  sn_grassland_3km_scale +
  urban_3km_scale + suburban_3km_scale +
  freshwater_3km_scale + coastal_3km_scale +
  growth_3km_scale*dominant_growth_3km +
  (1|year) + (1|season)")
model <- tryCatch({glmer(formula,
  family = binomial(link="logit"),
  data = det_species_ug,
  na.action = na.exclude)
}, warning = function(w) { message(paste("Model for:", species, "warning:", w$message))
}, error = function(e) { message(paste("Model for", species, "failed:", e$message))
  return(NULL)})
m5_list_sp[[species]] <- model

}

m5_genera_vars <- names(m5_list_gen)
m5_tables_gen <- list()

for (genera in m5_genera_vars) {
  fixedef <- coef(summary(m5_list_gen[[genera]]))
  conf <- confint(m5_list_gen[[genera]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
    slope = fixedef[,1],
    std_err = fixedef[,2],
    p_value = fixedef[,4],
    lci = conf[,1],
    uci = conf[,2],
    gen_sp = genera)
  m5_tables_gen[[genera]] <- table
}

m5_tables_gen <- bind_rows(m5_tables_gen) %>%
  mutate(model = "Model 5")

m5_species_vars <- names(m5_list_sp)
m5_tables_sp <- list()

for (species in m5_species_vars) {
  fixedef <- coef(summary(m5_list_sp[[species]]))
  conf <- confint(m5_list_sp[[species]], parm = "beta_", method = "Wald")
  table <- tibble(variable = rownames(fixedef),
    slope = fixedef[,1],
    std_err = fixedef[,2],
    p_value = fixedef[,4],
    lci = conf[,1],
    uci = conf[,2],

```

```

      gen_sp = species)
    m5_tables_sp[[species]] <- table
  }
m5_tables_sp <- bind_rows(m5_tables_sp) %>%
  mutate(model = "Model 5")

##### AIC Comparisons

m2_AIC <- list()
m3_AIC <- list()
m4_AIC <- list()
m5_AIC <- list()

for (genera in m2_genera_vars) {
  m2_AIC[[genera]] <- AIC(m2_list_gen[[genera]])
  m3_AIC[[genera]] <- tryCatch({AIC(m3_list_gen[[genera]])},
    error = function(e) {
      return(NA)
    })
  m4_AIC[[genera]] <- tryCatch({AIC(m4_list_gen[[genera]])},
    error = function(e) {
      return(NA)
    })
  m5_AIC[[genera]] <- tryCatch({AIC(m5_list_gen[[genera]])},
    error = function(e) {
      return(NA)
    })
}

for (species in m2_species_vars) {
  m2_AIC[[species]] <- AIC(m2_list_sp[[species]])
  m3_AIC[[species]] <- tryCatch({AIC(m3_list_sp[[species]])},
    error = function(e) {
      return(NA)
    })
  m4_AIC[[species]] <- tryCatch({AIC(m4_list_sp[[species]])},
    error = function(e) {
      return(NA)
    })
  m5_AIC[[species]] <- tryCatch({AIC(m5_list_sp[[species]])},
    error = function(e) {
      return(NA)
    })
}

aics <- tibble("Genus_species" = names(m2_AIC),

```

```

      "Model 2" = m2_AIC,
      "Model 3" = m3_AIC,
      "Model 4" = m4_AIC,
      "Model 5" = m5_AIC) %>%
  arrange(factor(Genus_species, levels = c("ppstrll", "p_pip", "p_pyg", "p_nat",
      "p_khl", "myotis", "m_br_wh", "m_bch",
      "m_nat", "m_dau", "plectts", "p_aur",
      "nyctals", "n_lsr", "n_noc", "eptescs",
      "r_fer")))) %>%

  mutate(across(c(2:5), ~ as.numeric(.x)))

write_csv(aics, "./images/aics.csv")

##### MODEL PLOTS

model_tables <- bind_rows(m1_tables_gen, m1_tables_sp, m2_tables_gen,
  m2_tables_sp, m3_tables_gen, m3_tables_sp,
  m4_tables_gen, m4_tables_sp, m5_tables_gen,
  m5_tables_sp) %>%
  filter(!gen_sp %in%c("b_bar", "m_alc", "p_khl"),
    !variable == "(Intercept)") %>%
  rowwise %>%
  mutate(variable = case_when(str_detect(variable, "suburban") ~ "Suburban",
    str_detect(variable, "urban_3km") ~ "Urban",
    str_detect(variable, "sn_grassland") ~ "Semi-Natural Grassland",
    str_detect(variable, "arable") ~ "Arable/Improved Grassland",
    str_detect(variable, "freshwater") ~ "Freshwater",
    str_detect(variable, "deciduous") ~ "Deciduous Woodland",
    str_detect(variable, "coastal") ~ "Coastal",
    str_detect(variable, "scale:dominant_growth_3kminfilling") ~ "Dominant Growth Pattern: Infilling",
    str_detect(variable, "scale:dominant_growth_3kmoutlying") ~ "Dominant Growth Pattern: Outlying",
    str_detect(variable, "infilling") ~ "Dominant Growth Pattern: Infilling",
    str_detect(variable, "outlying") ~ "Dominant Growth Pattern: Outlying",
    str_detect(variable, "growth") ~ "Urban Growth",),
    genus = case_when(gen_sp %in% c("plectts", "p_aur") ~ "Plecotus",
      str_detect(gen_sp, "^p") ~ "Pipistrelle",
      str_detect(gen_sp, "^m") ~ "Myotis",
      str_detect(gen_sp, "^e") ~ "Eptesicus",
      str_detect(gen_sp, "^n") ~ "Nyctalus",
      str_detect(gen_sp, "^r") ~ "Rhinolophus",
      str_detect(gen_sp, "^b") ~ "Barbastella"),
    odds_ratio = exp(slope),
    odds_lci = exp(lci),
    odds_uci = exp(uci),
    perc_change = (odds_ratio-1)*100,
    sig = if_else(p_value<0.05, "TRUE", ""))

# make table of values for models 2-5

```

```

results_table <- model_tables %>%
  ungroup() %>%
  filter(case_when(gen_sp == "p_pyg" ~ model == "Model 5",
                    gen_sp %in% c("myotis", "m_br_wh") ~ model == "Model 4",
                    .default = model == "Model 2")) %>%
  pivot_wider(names_from = variable, id_cols = gen_sp,
              values_from = c(odds_ratio, odds_lci, odds_uci),
              names_vary = "slowest", names_glue = "{.value}_{variable}") %>%
  select(-c("slope", "std_err", "p_value", "lci", "uci", "model", "genus",
            "perc_change", "sig"))

# make table of values for all

all_results_table <- model_tables %>%
  arrange(factor(gen_sp, levels = c("ppstrll", "p_pip", "p_pyg", "p_nat",
                                    "p_khl", "myotis", "m_br_wh", "m_bch",
                                    "m_nat", "m_dau", "plectts", "p_aur",
                                    "nyctals", "n_lsr", "n_noc", "eptescs",
                                    "r_fer")))) %>%

  ungroup() %>%
  pivot_wider(names_from = variable, id_cols = c(gen_sp, model),
              values_from = c(odds_ratio, odds_lci, odds_uci),
              names_vary = "slowest", names_glue = "{.value}_{variable}") %>%
  arrange(factor(model, levels = c("Model 1", "Model 2", "Model 3", "Model 4",
                                    "Model 5"))))

write_csv(all_results_table, "./all_results.csv")

#### MAKE PLOTS OF MODELS 1 AND 2

# myotis
m1_2_myo_plot <- model_tables %>%
  filter(gen_sp == "myotis",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "Myotis",

```

```

      x = NULL,
      y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "bold.italic"),
) +
coord_flip(ylim = c(0.1,10))

m1_2_m_br_wh_plot <- model_tables %>%
  filter(gen_sp == "m_br_wh",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban","Semi-Natural Grassland",
                              "Arable/Improved Grassland","Deciduous Woodland",
                              "Freshwater","Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "M. mystacinus/M. brandtii",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "italic"),
) +
coord_flip(ylim = c(0.1,10))

m1_2_m_nat_plot <- model_tables %>%
  filter(gen_sp == "m_nat",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci,, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban","Semi-Natural Grassland",
                              "Arable/Improved Grassland","Deciduous Woodland",
                              "Freshwater","Coastal")) +
  labs(title = "M. nattereri",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "italic"),
) +

```

```

coord_flip(ylim = c(0.1,10))

m1_2_m_dau_plot <- model_tables %>%
  filter(gen_sp == "m_dau",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "M. daubentonii",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "italic"),
        ) +
  coord_flip(ylim = c(0.1,10))

# pips

m1_2_pip_plot <- model_tables %>%
  filter(gen_sp == "ppstrll",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "Pipistrellus",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "bold.italic"),
        ) +
  coord_flip(ylim = c(0.1,10))

```

```

m1_2_p_pip_plot <- model_tables %>%
  filter(gen_sp == "p_pip",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                 linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "P. pipistrellus",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "italic"),
        ) +
  coord_flip(ylim = c(0.1,10))

m1_2_p_pyg_plot <- model_tables %>%
  filter(gen_sp == "p_pyg",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                 linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "P. pygmaeus",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "italic"),
        ) +
  coord_flip(ylim = c(0.1,10))

m1_2_p_nat_plot <- model_tables %>%
  filter(gen_sp == "p_nat",
         model %in% c("Model 1", "Model 2")) %>%

```



```

ggplot()+
geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
               linewidth = 1, position = position_dodge(-0.4)) +
geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
           position = position_dodge(-0.4)) +
geom_hline(yintercept = 1, linetype = "dashed") +
theme_classic()+
scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                             "Arable/Improved Grassland", "Deciduous Woodland",
                             "Freshwater", "Coastal")) +
scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
scale_y_log10() +
labs(title = "P. nathusii",
     x = NULL,
     y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "italic"),
) +
coord_flip(ylim = c(0.1,10))

# nyctalus
m1_2_nyc_plot <- model_tables %>%
  filter(gen_sp == "nyctals",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                 linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "Nyctalus",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "bold.italic"),
  ) +
  coord_flip(ylim = c(0.1,10))

m1_2_n_lsr_plot <- model_tables %>%
  filter(gen_sp == "n_lsr",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                 linewidth = 1, position = position_dodge(-0.4)) +

```

```

geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
            position = position_dodge(-0.4)) +
geom_hline(yintercept = 1, linetype = "dashed") +
theme_classic()+
scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                             "Arable/Improved Grassland", "Deciduous Woodland",
                             "Freshwater", "Coastal")) +
scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
scale_y_log10() +
labs(title = "N. leisleri",
      x = NULL,
      y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "italic"),
) +
coord_flip(ylim = c(0.1,10))

m1_2_n_noc_plot <- model_tables %>%
  filter(gen_sp == "n_noc",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                 linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban", "Semi-Natural Grassland",
                              "Arable/Improved Grassland", "Deciduous Woodland",
                              "Freshwater", "Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "N. noctula",
        x = NULL,
        y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "italic"),
) +
  coord_flip(ylim = c(0.1,10))

# Eptesicus
m1_2_ept_plot <- model_tables %>%
  filter(gen_sp == "eptescs",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                 linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +

```

```

theme_classic()+
scale_x_discrete(limits = c("Urban", "Suburban","Semi-Natural Grassland",
                             "Arable/Improved Grassland","Deciduous Woodland",
                             "Freshwater","Coastal")) +
scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
scale_y_log10() +
labs(title = "Eptesicus",
      x = NULL,
      y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "bold.italic"),
) +
coord_flip(ylim = c(0.1,10))

# plecotus
m1_2_ple_plot <- model_tables %>%
  filter(gen_sp == "plectts",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban","Semi-Natural Grassland",
                              "Arable/Improved Grassland","Deciduous Woodland",
                              "Freshwater","Coastal")) +
  scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
  scale_y_log10() +
  labs(title = "Plecotus",
       x = NULL,
       y = "Odds Ratio of Species Presence") +
  theme(plot.title = element_text(hjust= 0.5, face = "bold.italic"),
) +
  coord_flip(ylim = c(0.1,10))

m1_2_p_aur_plot <- model_tables %>%
  filter(gen_sp == "p_aur",
         model %in% c("Model 1", "Model 2")) %>%
  ggplot()+
  geom_linerange(aes(x = variable, ymin = odds_lci, ymax = odds_uci, color = model),
                linewidth = 1, position = position_dodge(-0.4)) +
  geom_point(aes(y = odds_ratio, x = variable, color = model), size = 3,
             position = position_dodge(-0.4)) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  theme_classic()+
  scale_x_discrete(limits = c("Urban", "Suburban","Semi-Natural Grassland",
                              "Arable/Improved Grassland","Deciduous Woodland",

```

```

                                "Freshwater", "Coastal")) +
scale_color_manual(values = c("Model 1" = "#619CFF", "Model 2" = "#F8766D")) +
scale_y_log10() +
labs(title = "P. auritus",
      x = NULL,
      y = "Odds Ratio of Species Presence") +
theme(plot.title = element_text(hjust= 0.5, face = "italic"),
) +
coord_flip(ylim = c(0.1,10))

m1_2_myo_plot + m1_2_pip_plot + m1_2_nyc_plot + m1_2_ple_plot + m1_2_ept_plot +
m1_2_m_br_wh_plot + m1_2_p_pip_plot + m1_2_n_lsr_plot + m1_2_p_aur_plot + plot_spacer() +
m1_2_m_nat_plot + m1_2_p_pyg_plot + m1_2_n_noc_plot + plot_spacer() + plot_spacer() +
m1_2_m_dau_plot + m1_2_p_nat_plot + plot_spacer() + plot_spacer() + plot_spacer() +
plot_layout(ncol = 5,
            axes = "collect_y",
            axis_title = "collect",
            guides = "collect")

ggsave("./images/models1and2results.png")

### Run pairwise comparisons
em_p_pyg <- emmeans(m4_list_sp$p_pyg, ~ dominant_growth_3km, adjust = "Tukey")
pairs_p_pyg <- pairs(em_p_pyg, type = "response")
p_pyg_plot <- plot(pairs_p_pyg, col = "#2c33ef")+
  geom_vline(xintercept =1, linetype = "dashed") +
  labs(title= "P.pygmaeus",
        x = "Odds ratio of estimated marginal means, log scale",
        y = "Factor Contrast") +
  scale_x_log10() +
  coord_cartesian(xlim = c(0.005, 20)) +
  theme_classic() +
  theme(plot.title = element_text(hjust= 0.5, face = "italic"))

em_myo <- emmeans(m4_list_gen$myotis, ~ dominant_growth_3km, adjust = "Tukey")
pairs_myo <- pairs(em_myo, type = "response")
myo_plot <- plot(pairs_myo, col = "#2c33ef")+
  geom_vline(xintercept =1, linetype = "dashed") +
  labs(title= "Myotis",
        x = "Odds ratio of estimated marginal means, log scale",
        y = "Factor Contrast") +
  scale_x_log10() +
  coord_cartesian(xlim = c(0.005, 20)) +
  theme_classic() +
  theme(plot.title = element_text(hjust= 0.5, face = "bold.italic"))

```

```

em_m_br_wh <- emmeans(m4_list_sp$m_br_wh, ~ dominant_growth_3km, adjust = "Tukey")
pairs_m_br_wh <- pairs(em_m_br_wh, type = "response")
m_br_wh_plot <- plot(pairs_m_br_wh, col = "#2c33ef")+
  geom_vline(xintercept=1, linetype = "dashed") +
  labs(title= "M. mystacinus/brandtii",
        x = "Odds ratio of estimated marginal means, log scale",
        y = "Factor Contrast") +
  scale_x_log10() +
  coord_cartesian(xlim = c(0.005, 20)) +
  theme_classic() +
  theme(plot.title = element_text(hjust= 0.5, face = "italic"))

p_pyg_plot + myo_plot + m_br_wh_plot +
  plot_layout(axes = "collect", ncol = 1)

ggsave("./images/pairwise.png")

```

822 A_create_images.R

```

#####
#####
#####          GRAPHS AND PICTURES          #####
#####
#####
#####

# this script creates the graphs and images used within the project.

#####          MAP OF KENT          #####
kent_boundary <- sf::st_read("./data/county_boundaries_2023_bfe/CTYUA_MAY_2023_UK_BFE.shp")
  filter(CTYUA23NM %in% c("Kent", "Medway")) %>%
  st_union() %>%
  st_combine() %>%
  st_as_sf()

# Where is Kent?
gb <- sf::st_read("./data/figures_only/country_border/CTRY_DEC_2023_UK_BFE.shp") %>%
  filter(CTRY23NM %in% c("England", "Scotland", "Wales")) %>%
  st_union() %>%
  st_combine() %>%
  st_as_sf()

locator <- ggplot() +
  geom_sf(data = gb, aes(), color = NA, fill = "grey") +
  geom_sf(data = kent_boundary, aes(), color = NA, fill = "red") +
  theme_void()

```

and what is in it?

```

kent_spat <- vect(kent_boundary)
target_crs <- crs(kent_spat)

cover_rasters <- rast("./data/urban_growth_variables/cover_all_years.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)
urban_extent_1990 <- cover_rasters["1990"]
urban_extent_1990[urban_extent_1990 == 1] <- 1990
urban_extent_1990 <- as.factor(urban_extent_1990)

urban_extent_2007 <- cover_rasters["2007"]
urban_extent_2007[urban_extent_2007 == 1] <- 2007
urban_extent_2007 <- as.factor(urban_extent_2007)

urban_extent_2023 <- cover_rasters["2023"]
urban_extent_2023[urban_extent_2023 == 1] <- 2023
urban_extent_2023 <- as.factor(urban_extent_2023)

greenbelt <- sf::st_read("./data/figures_only/greenbelt_22-23/England_Green_Belt_2022_23")
sf::st_transform(crs = 27700) %>%
  filter(st_intersects(.$geometry, kent_boundary, sparse = TRUE) %>% lengths > 0) %>%
  st_union() %>%
  st_combine() %>%
  st_intersection(kent_boundary)
aonb <- sf::st_read("./data/figures_only/aonb/Areas_of_Outstanding_Natural_Beauty_England")
sf::st_transform(crs = 27700) %>%
  filter(st_intersects(.$geometry, kent_boundary, sparse = TRUE) %>% lengths > 0) %>%
  st_union() %>%
  st_combine() %>%
  st_intersection(kent_boundary)

kent_high_tide <- sf::st_read("./data/county_boundaries_2023_bfc/CTYUA_MAY_2023_UK_BFC.shp")
filter(CTYUA23NM %in% c("Kent", "Medway")) %>%
  st_union() %>%
  st_combine() %>%
  st_as_sf()

overview_map <- ggplot() +
  #geom_sf(data = greenbelt, aes(), color = NA, fill = "lightgreen", alpha = 0.2) +
  #geom_sf(data = aonb, aes(), color = NA, fill = "darkgreen", alpha = 0.2) +
  geom_spatraster(data = urban_extent_2023) +
  #geom_spatraster(data = urban_extent_2007) +
  geom_spatraster(data = urban_extent_1990) +
  geom_sf(data = kent_high_tide, aes(), color = "black", fill = NA) +
  scale_fill_manual(values = c("2023" = "#619CFF", "2007" = "plum",

```

```

        "1990" = "#F8766D", "NaN" = NA,
        "TRUE" = "blue", "FALSE" = "white"),
        na.value = "transparent") +
  annotation_custom(ggplotGrob(locator),
    xmin = 620000, xmax = 650000,
    ymin = 115000, ymax = 145000) +
  labs(fill = "Urban and Suburban Extent") +
  theme_classic() +
  theme(legend.position = "bottom")

#####          OVERALL AND GENERA DETECTION MAPS          #####

# must run '3_standardise_obs_data'
no_years <- obs_data %>%
  filter(!(common_name %in% c("bat", NA))) %>%
  group_by(PLAN_NO) %>%
  summarize(num_years = length(unique(year))) %>%
  left_join(grid_1km)

obs_per_year <- obs_data %>%
  filter(!(common_name %in% c("bat", NA))) %>%
  group_by(year, season) %>%
  summarize(num_grids = length(unique(PLAN_NO)))

obs_hist <- ggplot() +
  geom_col(aes(x = year, y = num_grids, fill = season), data = obs_per_year,
    position = "identity") +
  theme_classic() +
  scale_fill_manual(values = c("active" = "#619CFF", "hibernation" = "#F8766D"),
    labels = c("active" = "Active", "hibernation" = "Hibernation")) +
  labs(fill = "Season",
    x = "Year",
    y = "Number of 1 km x 1 km Grid Squares \n with at least one record") +
  theme(legend.position = "bottom")

observations_by_year <- ggplot() +
  geom_sf(data = no_years, aes(fill = num_years, geometry = geometry), color = NA) +
  geom_sf(data = kent_high_tide, aes(), fill = NA, color = "black") +
  theme_classic() +
  scale_fill_distiller(palette = "Spectral") +
  labs(fill = "Years with at \n least one record") +
  theme(legend.position = "bottom")

observations_by_year + obs_hist + plot_annotation(tag_levels = "A") &
  theme(plot.tag = element_text(size = 25))

ggsave("./images/record_dist.png")

```

```
##### OBSERVATIONS TABLE #####
table_obs_gen <- det_genera %>%
  group_by()

##### URBAN GROWTH CLASSIFICATION IMAGE #####
# create required geoms
ugp_rasters <- rast("./data/urban_growth_variables/all_years.tif") %>%
  project(y=target_crs) %>%
  crop(y=kent_spat, mask = TRUE)

ugp_2020 <- ugp_rasters["2020"]

urban_extent_2019 <- cover_rasters["2019"]
urban_extent_2019[urban_extent_2019 == 1] <- 2019
urban_extent_2019 <- as.factor(urban_extent_2019)

urban_extent_2020 <- cover_rasters["2020"]
urban_extent_2020[urban_extent_2020 == 1] <- 2020
urban_extent_2020 <- as.factor(urban_extent_2020)

ex_urban_extent_2019 <- crop(urban_extent_2019, c(xmin = 545000, xmax = 555000,
                                                  ymin = 127500, ymax = 135000))
ex_urban_extent_2020 <- crop(urban_extent_2020, c(xmin = 545000, xmax = 555000,
                                                  ymin = 127500, ymax = 135000))
ex_ugp_2020 <- crop(ugp_2020, c(xmin = 545000, xmax = 555000,
                                ymin = 127500, ymax = 135000))

plot1 <- ggplot() +
  geom_spatraster(data = urban_extent_2019) +
  geom_spatraster(data = ugp_2020)
  scale_fill_manual(values = c("2019" = "lightgrey", "NA" = NA),
                    na.value = "transparent", "infilling" = "red3", "edge expansion" = "
                    "outlying" = "violet") +

# create four quadrant
# 1st - 2019 urban extent with extent and holes marked

B_1_map <- ggplot() +
  geom_spatraster(data = ex_urban_extent_2019) +
  scale_fill_manual(values = c("2019" = "lightgrey", "NA" = NA),
                    na.value = "transparent") +
  geom_sf(data = base_2019, aes(), fill = NA, color = "black") +
  geom_sf_pattern(data = holes_2019, aes(), fill = NA, color = "red",
                  pattern = "stripe", pattern_color = "red", pattern_fill = "red",
                  pattern_size = 0.05, pattern_spacing = 0.005) +
  coord_sf(xlim = c(550500, 552500), ylim = c(128000, 130000)) +
```



```

theme_classic() +
theme(legend.position = "none",
      axis.title = element_blank(),
      axis.text = element_blank(),
      axis.line = element_blank(),
      axis.ticks = element_blank(),
      panel.border = element_rect(color = "black", fill = NA))

```

```

B_2_map <- ggplot() +
  geom_spatraster(data = ex_urban_extent_2020) +
  geom_spatraster(data = ex_urban_extent_2019) +
  scale_fill_manual(values = c("2019" = "lightgrey", "2020" = "grey29",
                                "NA" = NA),
                    na.value = "transparent") +
  geom_sf(data = base_2019, aes(), fill = NA, color = "black") +
  geom_sf_pattern(data = holes_2019, aes(), fill = NA, color = "red",
                  pattern = "stripe", pattern_color = "red", pattern_fill = "red",
                  pattern_size = 0.05, pattern_spacing = 0.005) +
  coord_sf(xlim = c(550500,552500), ylim = c(128000, 130000)) +
  theme_classic() +
  theme(legend.position = "none",
        axis.title = element_blank(),
        axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.border = element_rect(color = "black", fill = NA))

```

```

B_3_map <- ggplot() +
  geom_spatraster(data = ex_ugp_2020) +
  scale_fill_manual(values = c("infilling" = "grey29", "edge expansion" = "grey29",
                                "outlying" = "grey29", "NA" = NA),
                    na.value = "transparent") +
  geom_sf(data = base_2019, aes(), fill = NA, color = "black") +
  geom_sf_pattern(data = holes_2019, aes(), fill = NA, color = "red",
                  pattern = "stripe", pattern_color = "red", pattern_fill = "red",
                  pattern_size = 0.05, pattern_spacing = 0.005) +
  coord_sf(xlim = c(550500,552500), ylim = c(128000, 130000)) +
  theme_classic() +
  theme(legend.position = "none",
        axis.title = element_blank(),
        axis.text = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        panel.border = element_rect(color = "black", fill = NA))

```

```

B_4_map <- ggplot() +
  geom_spatraster(data = ex_ugp_2020) +

```

```

scale_fill_manual(values = c("infilling" = "red3", "edge expansion" = "limegreen",
                             "outlying" = "violet", "NA" = NA),
                  na.value = "transparent") +
geom_sf(data = base_2019, aes(), fill = NA, color = "black") +
geom_sf_pattern(data = holes_2019, aes(), fill = NA, color = "red",
               pattern = "stripe", pattern_color = "red", pattern_fill = "red",
               pattern_size = 0.05, pattern_spacing = 0.005) +
coord_sf(xlim = c(550500, 552500), ylim = c(128000, 130000)) +
theme_classic() +
theme(legend.position = "none",
      axis.title = element_blank(),
      axis.text = element_blank(),
      axis.line = element_blank(),
      axis.ticks = element_blank(),
      panel.border = element_rect(color = "black", fill = NA))

# create table grob

B_table <- tibble("patch" = c("New urban patch 1", "New urban patch 2",
                             "New urban patch 3"),
                 `intersects hole` = c("TRUE", "FALSE", "FALSE"),
                 `intersects extent` = c("-", "TRUE", "FALSE"),
                 pattern = c("infilling", "edge expansion", "outlying"))

B_table_grob <- gridExtra::tableGrob(B_table, rows = NULL)

tiff(filename = "./images/urban_growth.tiff", units = "in", width = 5, height = 5, res = 300)

(B_1_map | B_2_map) /
(B_3_map | B_4_map) +
plot_annotation(tag_levels = "A",
               theme = theme(plot.margin = margin())) &
theme(plot.tag = element_text(size = 50),
      plot.tag.location = "panel",
      plot.tag.position = "bottomleft")

dev.off()

B_table <- tibble("patch" = c("New urban patch 1", "New urban patch 2",
                             "New urban patch 3"),
                 `intersects hole` = c("TRUE", "FALSE", "FALSE"),
                 `intersects existing` = c("-", "TRUE", "FALSE"),
                 pattern = c("infilling", "edge expansion", "outlying"))

B_table_grob <- gridExtra::tableGrob(B_table, rows = NULL)

ggsave("./images/urban_growth_table.tiff", B_table_grob)

```

```
##### COVARIATE PLOTS #####

ggplot() +
  geom_point(aes(x = total_arable_3km_perc, y = total_im_grassland_3km_perc),
    data = det_genera)

ggplot() +
  geom_point(aes(x = total_urban_3km_perc, y = total_suburban_3km_perc),
    data = det_genera)
```

823 E - Risk Assessment



Risk Assessment

824

Summary

Reference: RA092833/1

Sign-off Status: Authorised

Date Created:	01/05/2024	Confidential?	No
Assessment Title:			
Assessment Outline: The assessment covers a largely desk-based project. There will be computer-based work and electronics lab work. There will also be a non-data collecting field trip.			
Area Responsible (for management of risks)		Location of Risks	
Division, School, Faculty, Institute:	Faculty of Life Sciences	Building:	Off-Site
Department:	Div of Biosciences	Area:	
Group/Unit:	Genetics, Evolution & Environment	Sub Area:	
Further Location Information:	Most desk-based work undertaken in the Centre for Biodiversity and Environment Research. Some desk-based work from home. Electronics work in location tbc (UCL East). Field visits to various locations in Kent.		
Is additional GM or HG approval required? Only relevant to specialist biological risk assessments (GMM2, GMM3, HG2, HG3, GM animals, GM plants) except GMM class 1.:	Click SELECT to change ONLY if this is a GMM Class 2, GMM Class 3, HG2, HG3, GM animals or GM plants risk assessment		
Assessment Start Date:	01/05/2024	Review or End Date:	01/05/2025
Relevant Attachments:			
Description of attachments:			
Location of non-electronic documents:			
Assessor(s):			
Approver(s): HAYLEY BOAKES			
Signed Off: HAYLEY BOAKES (17/07/2024 14:40)			
29/07/2024 20:06)			

PEOPLE AT RISK (from the Activities covered by this Risk Assessment) *

CATEGORY
Employees
Post-Graduates
Women of Child-bearing Age
Other Vulnerable Persons



1. Computational Work	
Description of Activity:	Desk-based analysis of data and writing up research project. Using laptop and connected monitor within
Hazard 1. Eye strain	
Poor DSE workstation set up, lack of regular breaks from the computer screen can lead to eye strain, headaches, blurred vision	Existing Control Measures Regular breaks taken, 20m for 20s every 20 minutes rule followed. DSE assessment taken on Risknet. Eye test taken within the last six months.
Hazard 2. Fire	
In the event of fire, participant could become trapped and could suffer ill health from smoke inhalation, burns, or death	Existing Control Measures Regular inspections are conducted of fire alarm systems in the workspace and building. Full Building Fire Risk assessments are undertaken by UCL Safety Services Fire department. Fire drills are carried out annually. Fire Evacuation Marshalls are appointed and trained.
Hazard 3. Poor lighting	
Poor lighting conditions may result in headaches or sore eyes. The lighting levels may not be suitable.	Existing Control Measures Lighting in office space is suitably controlled, with glare from artificial light sources minimised.
Hazard 4. Electricity	
Electric shock/burn	Existing Control Measures No unauthorised alterations or repairs allowed to University equipment.
Hazard 5. Hot surface/material	
Hot/boiled water from taps/kettles and microwaved food. Boiling water tap.	Existing Control Measures Care to be taken when transporting hot food and drink. Not too many items to be carried at once.



Risk Assessment

Hazard 6. Static or poor body posture	
Postural issues may give rise to discomfort or injury and can arise through a poor or inadequate workstation setup.	Existing Control Measures DSE assessment taken on Risknet. Workstation appropriately set up and office chair used. Regular breaks to be taken. Lighting and temperature suitably controlled.
Hazard 7. Trip hazard	
Injury from trips/slips over items, e.g. trailing cables, low steps, etc.	Existing Control Measures Good housekeeping practices observed by all within the office space. Adequate lighting levels maintained. Walkways kept clear from obstructions.
Hazard 8. Repetitive Movements	
Staff risk discomfort or injury to hands and arms from overuse/improper user or poorly designed workstation layouts	Existing Control Measures Completion of DSE assessment on Risknet. Rest breaks regularly taken. n/a

Risk Level

With Existing Controls:

Risk Level

A - Very Low / Trivial

2. Electronic Lab Work	
Description of Activity:	Development of sensor including wiring, use of 3D printers and laser cutters. Possible soldering. Software writing on personal laptop.



Risk Assessment

Hazard 1. Electricution	
Incorrect wiring may lead to electrocution of participant.	Existing Control Measures Work only to be undertaken at low voltages to minimise severity of consequence. Care to be taken when connecting wires.
Hazard 2. Burns	
Potential for burns from soldering.	Existing Control Measures Gloves and eye cover to be worn when suitable. Soldering equipment to have been checked.
Hazard 3. Desk-based risks	
All risks as outlined in the computational work activity.	Existing Control Measures As in the computational work activity

Risk Level

With Existing Controls:

Risk Level

B - Low / Tolerable

3. Field Visits	
Description of Activity:	Field visits to bat hibernation roosts in Kent. Emergence surveys at dusk.
Hazard 1. Falling	
Falling down deneholes/in caves/mines	Existing Control Measures Proper footwear to be worn. Correct equipment to be used where appropriate. Suitable lighting to be used. Participants should always be accompanied.



Risk Assessment

Hazard 2. Trips in low light	
Low light during emergence surveys increase risk of trips and falls.	Existing Control Measures Locations to be identified and assessed during daylight hours. Movement in unknown locations to be minimised. No lone working.
Hazard 3. Travel	
Public transport to be used, trains. Risk of injury from failing trains, slips and trips, etc.	Existing Control Measures All advice of operating companies and stations to be followed. Travel will not occur alone.

Risk Level

With Existing Controls:

Risk Level

B - Low / Tolerable