Actual and future predicted occupancy of the Black Kite in Spain

- Case study of the course Monitoring and occupancy modelling -

Felix Nößler

2^{nd} semester in master program: Ecology, Evolution and Conservation (M.Sc.),

Registration number: 810578

August 16, 2021
Module Quantitative conservation biogeography
Prof. Dr. Damaris Zurell & Dr. Guillermo Fandos-Guzman

ABSTRACT

The Black Kite is an abundant bird of prey in Spain. The main goal is to access which site covariates are important for the occurrence of the Black Kite in Spain. Furthermore, an actual distribution map of the Black Kite in Spain is presented and compared to the distribution under the assumption of an increase of annual mean temperature and a decrease of total precipitation in 80 years. The selection of site covariates is based on ecological knowledge of the focal species and an exploratory analysis with a Random Forests model. The eBird data set is filtered for repeated surveys and used as the data source for fitting the static occupancy model.

1 Introduction

Climate change affects all levels of biodiversity (Bellard et al. 2012; Garcia et al. 2014). A main task in ecological research is to build accurate models to predict the biological response to climate change (Urban et al. 2016; Araújo and Rahbek 2006).

One of the affected groups by the climate change are birds of prey. Birds of prey play a crucial role in ecosystems. Very often they are top predators. They can be regarded as flagship species, because they are very vulnerable to human activity, because of their role as top predator in food chains and because of the attractiveness of their behavior to humans. Furthermore, they provide regulating, supporting and cultural ecosystem services (Donázar et al. 2016).

A raptor species with a very wide distribution is the Black Kite, *Milvus migrans* (Boddaert, 1783). The distribution of this species ranges from Western Europe to East Asia and Australia (BirdLife International 2021). The species is mainly migratory and the European population stays in the winter in sub-Saharan Africa. The individuals leave the breeding area between July and October and come back back between February and May (Panuccio et al. 2014; BirdLife International 2021). In this case study the focus lies on the Black Kite population in Spain.

Citizen science data are often used in species distribution modelling, because of the high data availability. However there are some challenges in using these data sets, for example the spatio-temporal bias of the of the observations. For instance people tend to watch for birds more in the breeding season and around their own home (Robinson et al. 2018; Reich et al. 2018). Two techniques are available to overcome this difficulties. First, it is possible to filter out low quality observations. Second, it is possible to apply statistical techniques to address sampling bias and observational heterogeneity (Steen et al. 2019). In this study the second option is used. Observational heterogeneity and imperfect detection is tackled with an extension of species distribution model, the so-called occupancy model (MacKenzie, Nichols, et al. 2002).

The model is used to learn more about the ecology of the focal species. This knowledge can be used in conservation planning.

The main questions are (1) which site covariates are most important for the occurrence of the Black Kite and (2) what happens to the Black Kite population in Spain under future climate conditions.

2 Methods

Data of the species occurrences are from eBird data (Sullivan et al. 2009; Cornell Lab of Ornithology 2021). The dataset is filtered for repeated surveys, that were done three to ten times in the same area between April and June of 2019. Only standing or travelling surveys with a total distance up to 5 km with one to five observers were used. To get an overview a map with all observations was produced using QGIS Development Team (2021).

Data preparation, selection of covariates, model selection, model evalutation and prediction were done with R Core Team (2021). Exploratory analysis of the covariates were done in Van Rossum and Drake (2009). An ODMAP-Protocol (Zurell et al. 2020) describing typical steps in species distribution modelling can be found in the electronic appendix.

Spatial resolution of the model

The home range size differs between younger not breeding individuals, that are one to seven years old, so-called floaters, (Blas et al. 2009) and the breeding individuals. Floaters had in south Spain an home range size over 300 km^2, breeding females 43 km^2 and breeding males 80 km^2 (Tanferna et al. 2013). However, these home range values were observed with radio-tracking of individuals and later on the calculation of the minimum convex polygon. The main activity is probably more restricted to centre of the area.

A resolution of 2.5 minutes (roughly 21.5 km^2) was chosen. It is possible to argue that the size is too small for the large home range sizes of the Black Kite. However through spatial subsampling the data loss is higher and the connection between the observations and the site variables becomes less strong.

Selection of detection covariates

Detection covariates that are present in the eBird data are used. These are the day of the year, time when observation started, duration of observation, walking distance and protocol type (standing or travelling). The two most important covariates from the Random Forests model (see section on exploratory analysis of covariates) were used in the final model.

Selection of site covariates

First a preliminary selection of site covariates was compiled. These set include all bioclimatic variables (Fick and Hijmans 2017), land cover fractions, as the tree cover, the grass cover and the bare soil cover (Buchhorn, Smets, et al. 2020), and the distance between the centre of the raster cells to the closest landfill and the closest river or lake.

In the next step ecological hypotheses were formulated based on existing ecological knowledge from the literature. Some of the covariates were selected based on these hypotheses. As an additionally step, all preliminary covariates were used to fit a Random Forests model and to get the importance of all the covariates for predicting the detection probability of the Black Kite. The most important covariates from these analysis were added to the list of covariates.

The covariates from the ecological hypotheses and from the Random Forests model were tested for collinearity with the Variance Inflation Factor (as implemented in Heiberger 2020). Highly collinear covariates were removed. Maps of the final site covariates can be found in the appendix (see Figure 7). The complete selection process is shown in Figure 1 and further explained in the next sections.

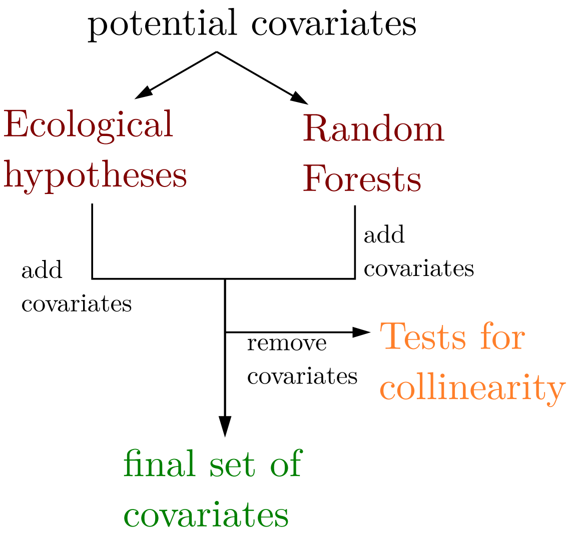

Figure 1: Procedure of selecting site covariates for the occupancy model

Data preparation of site covariates

For all raster operations Hijmans (2021) was used. Bioclimatic variables were download from Fick and Hijmans (2017) in the target resolution. Tree cover, herbaceous vegetation cover and bare soil cover were downloaded (Buchhorn, Lesiv, et al. 2020; Buchhorn, Smets, et al. 2020). The raster images were merged to one file and then cropped to mainland of Spain. The resolution was resampled to the target resolution using a bilinear interpolation.

Land cover data was retrieved in vector format (Copernicus Land Monitoring Service 2018) and cropped to the mainland of Spain. The distance between each centre of the raster cell to all polygons of landfills and rivers or lakes was calculated with Bivand and Rundel (2020). Afterwards, the minimum distance was saved.

Ecological justification of site covariates

The selection of covariates is based on existing ecological knowledge about the focal species. A list of all site covariates and a corresponding simplified hypothesis can be found in Table 1.

Table 1: Site covariates and simplified hypothesized response of occupancy of the Black Kite, arrows symbolize the expected occupancy probability with higher values of the respective covariate

Site covariates
Annual Mean Temperature ↘
Annual precipitation ↗
Tree cover ↘
Grass cover ↗
Bare soil cover ↘
Distance to closest river or lake ↘
Distance to closest landfill ↘

The Black Kite uses a variety of feeding sources, for example birds, fish, crayfish, insects, carrion, vegetable matter and smaller mammals (Sergio and Boto 1999; Vinuela and Veiga 1992; BirdLife International 2021). Part of the prey is found in wetlands and marshes and it was proposed that the Black Kite has a habitat binding to wetlands and marshes (Veiga and Hiraldo 1990; Tanferna et al. 2013). Accordingly, the distance to the closest lake or river was calculated and used as a site covariate. The distance to the closed lake or river was used, because it is more important that lakes or rivers are nearby than that the Black Kite was actually observed in cell with a lake or a river.

It has been shown that Black Kites visit landfills for feeding (De Giacomo and Guerrieri 2008; Blanco 1994). Therefore the distance to the closest landfill were analysed. The landfill cover was not used, because in most of the raster cells landfills are not present and therefore the model fitting will not be optimal.

Black Kites breed in branches of trees. However, closed woodlands are avoided (Tanferna et al. 2013). Therefore the hypothesis is that the occupancy of the Black Kite follows a unimodal distribution with regard to the tree cover, where the occupancy is high at low to intermediate tree cover. In contrast, there should be a positive relationship with the grass cover, because the Black Kite is searching for prey in open landscapes. The bare soil cover is especially high in the mountains, the hypothesis is that the Black Kites is more a lowland species in Spain and avoids areas with high bare soil cover.

The annual mean temperature is added, because it is hypothesized that the Black Kite is negatively affected by too high annual mean temperatures. Additionally, a high annual precipitation may lead to more vegetation and more prey for the Black Kite. Both covariates are especially important for predicting the change of occupancy due to climate change.

Exploratory analysis of site and detection covariates

An exploratory analysis of the importance of the site and detection covariates was done using the machine learning method Random Forests (Ho 1995). All covariates as described before were used. These include all bioclim variables, land cover fractions, cover of water bodies, cover of landfills, distance to closest water body, distance to closest landfill and all detection covariates. From each grid cell one observation was selected and related to the site and detection covariates. Because of the higher number of non-detection an balanced Random Forest Classifier was used (Lemaître et al. 2017, with default parameters). Preparing the data, standardization of data and splitting in train and test data (80 % and 20 %) was done with Pedregosa et al. (2011), Harris et al. (2020), and McKinney (2010). 500 simulation were done to account variability in the model fit to the different data sets (each time one of the up to ten observations per grill cell was chosen). The mean and the standard error of the importance of all covariates was calculated. The effect of the most important covariates on the detection probability was tested. Therefore all other covariates were set to the mean and only the focal covariate was changed. The python script for conducting the analysis can be found in the appendix.

Selection of best model

A nullmodel, a model with only detection covariates, a model with only site covariates and a full model (site and detection covariates) were compared with the Akaike information criterion (AIC) and the Akaike information criterion corrected for small sample size (AICc) with Fiske and Chandler (2011). The model with the lowest AIC and AICc was chosen.

To compare the stability of the model an average model was build from the best model. Subsets of the best model were chosen with Bartoń (2020), all detection covariates were used as fixed terms (present in all subsetted models). Models were sorted according to the lowest AIC and selected with cumulative Akaike weight (Wagenmakers and Farrell 2004) larger than 95 %. The selected models are weighted with the AICs and averaged with Bartoń (2020).

Model evaluation

The predictive power of the best model is evaluated with the R^2-value (Nagelkerke 1991). A parametric bootstrapping approach is used to access the goodness of fit ("parboot" function as implemented in Fiske and Chandler 2011). From the test results the $\hat{c}$-value was calculated. If this values is noticeably larger than one, it indicates overdispersion of the model. With this $\hat{c}$-value a quasi-Akaike information criterion corrected for small sample

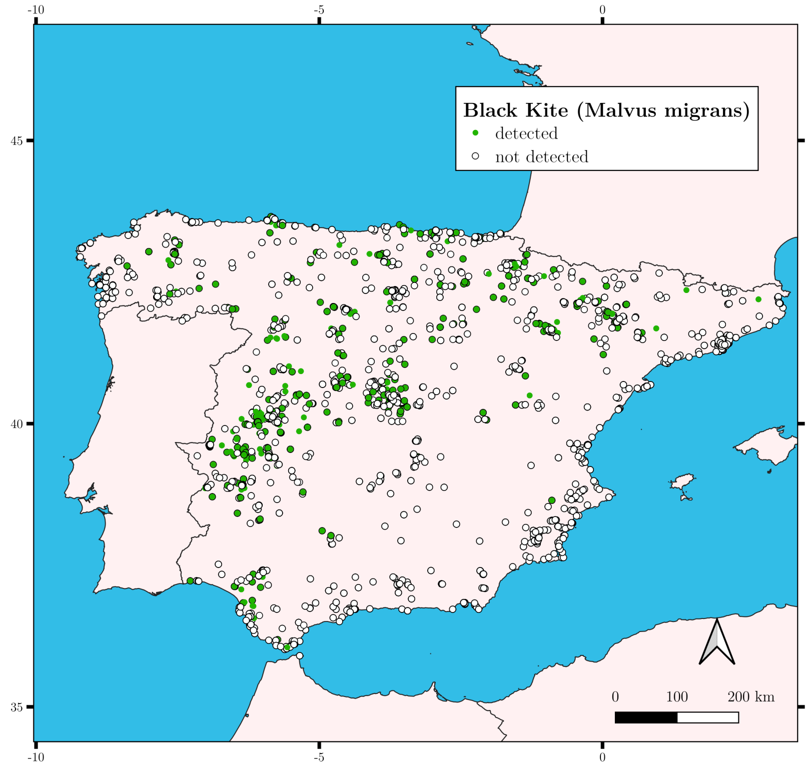

Figure 2: Detection of the Black Kite in the filtered eBird data set in Spain, each dot represents one observation within the repeated surveys, created with QGIS Development Team (2021)

size (qAICc) was retrieved to see if there are big differences to the calculated AICc. Another goodness of fit as proposed by MacKenzie and Bailey (2004) was carried out (implemented in Mazerolle 2020).

Prediction

First, predictions were done first on the grid cells with observational data and secondly with the best model for the mainland of Spain. A map was produced with the predicted occupancy and the standard error of the prediction. The outcome of the best model were compared to the results from the average model.

Prediction of future climate conditions were done accordingly to existing knowledge about the climate change. The annual temperature will likely rise around 3 °C and the annual rainfall will decrease around 10 % from 2020 to 2100 (State Meteorological Agency (AEMET) 2021). Iturbide et al. (2021) show under the SSP2 4.5 scenario (Riahi et al. 2017) an increase between 2 and 3 °C and a decrease of total precipitation between 5 % and 20 % in different areas of Spain (mean values of 32 models, 2081-2100, relative to 1995-2014). The model agreement is high for the temperature increase, however the signal is not robust for the decrease in total precipitation. These data set is used to model the future occupancy of the Black Kite. The interactive atlas provides

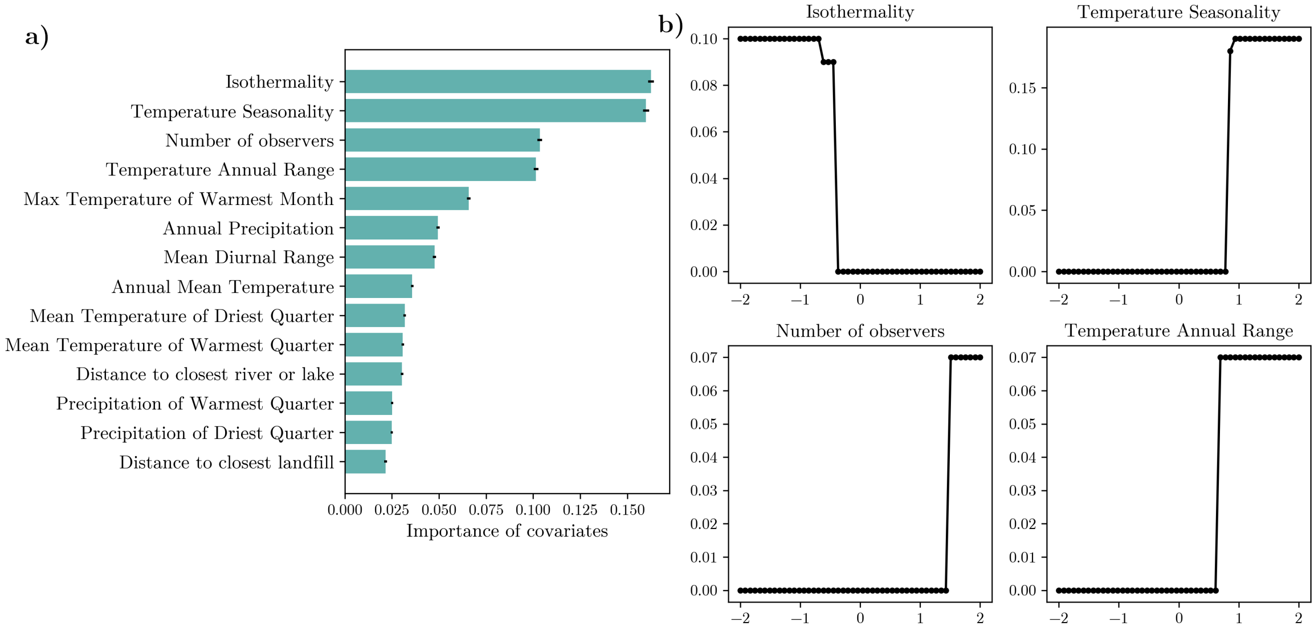

Figure 3: Relationship between covariates and detection probability, (a) importance of the covariates in the Random Forests model (only covariates with importance > 0.02 are included), mean importance values of 500 simulation are shown, black lines represent the standard error, and (b) influence of the four most important covariates on the detection probability of the Black Kite in Spain, the y-axis shows the probability, the x-axis the standardized value of the covariate, created with Hunter (2007)

regional predictions on a spatial scale of 95 km edge length of a raster cell. GeoTIFFs were downloaded for the change in annual mean temperature and the annual total precipitation (change from 1995-2014 to 2081-2100). These raster files were cropped to the mainland of Spain and resampled to the target resolution (2.5 minutes of a degree). Maps of the changes in the target resolution can be seen in the appendix (see Figure 8). The change in temperature and precipitation were applied to the respective bioclimatic variables (Fick and Hijmans 2017).

We will only incorporate the habitat suitability in relation to climate factors and omit all biological mechanisms that are proposed to play a role for prediction by Urban et al. (2016) like demography or evolution of a population. It is therefore a simplistic model, but the goal is to catch the main trend.

The effect of the site and detection covariates were tested. Accordingly, all other covariate than the target covariate were set to the mean value and the predictions were done in the range of the target covariate that the model was fitted to. Occupancy probabilities were produced for this range the uncertainty was measured with the standard error.

3 Results

A map with all filtered observation from the eBird data set is shown in Figure 2. It can be observed that the main distribution is in central Spain. The Black Kite was detected from the coast of the Gulf of Cadiz northwards up to the western Pyrenees. However, in the south-eastern part of Spain detections of the Black Kites are rare.

Exploratory analysis of covariates

The Random Forests model could explain close to 100 % of the detection of the Black Kite in the train and the test data set. An overview of the most important covariates can be found in Figure 3a.

Especially important are covariates that deal with temperature evenness over the year like isothermality, temperature seasonality and temperature annual range. The only detection covariate that really plays a role is the number of observers. More observers improve the chance to detect a Black Kite (see Figure 3b). Land cover fractions are not represented under the most important covariates. The annual precipitation is sixth influential covariate. The distance to the closest river or lake and the distance to the closest landfill play minor role.

The four most important factor, namely the isothermality, temperature seasonality, temperature annual range and maximal temperature of warmest month were added as preliminary covariates.

Occupancy model

Seven site covariates were analysed from the ecological hypotheses and four were added from the Random Forests model. Because of the high collinearity temperature seasonality, temperature annual range and maximal temperature of warmest month were excluded from the final list of covariates. After that the variance inflation factor was for all covariates below 1.9. The highest correlation exists between the tree cover and the annual precipitation (Pearson $\rho = 0.62$).

The final best model is the full model and consists of 20 parameters and has an AIC of 4578. The second best model with only the site covariates has an AIC that is 143 higher (delta AIC). The AICc gives roughly the same result. In the full model there are two detection covariates, namely the number of observers and the duration of observation. The explained variance in the detection of the Black Kite is 35 % (R^2-value).

An average model was build from three single models. These are one model with all site covariates except the distance to closest landfill, the full model and a third model with all site covariates except the annual precipitation and the closest distance to landfills.

All site covariates are fitted as polynomials of two degrees. These are in the full model the annual mean temperature, the isothermality, the annual precipitation, the bare soil cover, the grass cover, the tree cover, the distance to closest river or lake and the distance to closest landfill. All estimated values of the site covariates are significant different from zero, except

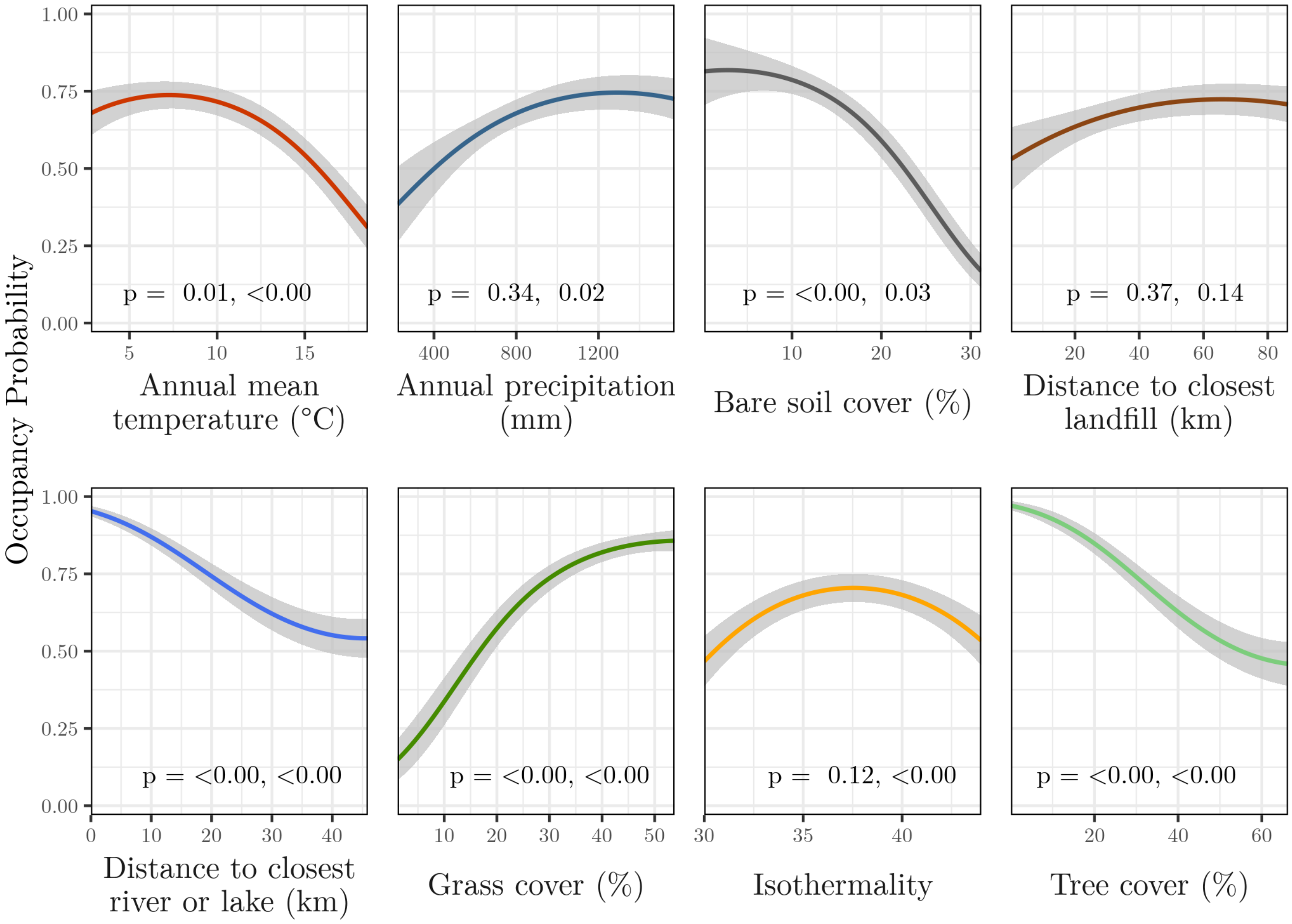

Figure 4: Predicted occupancy of the full model in response to site covariates, for each plot only the focal site covariate was varied, grey areas represent the standard error, the p values show the significance of the first and the second coefficient, created with Wickham (2016)

the first coefficients of the isothermality and the annual precipitation and both coefficients of the distance to the closest landfill. The occupancy probability shows a positive relation with the grass cover and the annual precipitation. A unimodal relationship can be seen between the isothermality and the annual mean temperature and the occupancy probability of the Black Kite. Bare soil cover, distance to closest lake or river and a high tree cover have a negative influence on the occupancy probability (see Figure 4). These results go inline with the outcome of the average model. The uncertainty is high in the site covariates annual precipitation and the distance to closest landfill, the occupancy probability shows the same clear response to all other site covariates as with the full model (see Figure 9 in the appendix).

The duration of the observation has a positive influence on the occupancy probability and the coefficient is significant different from

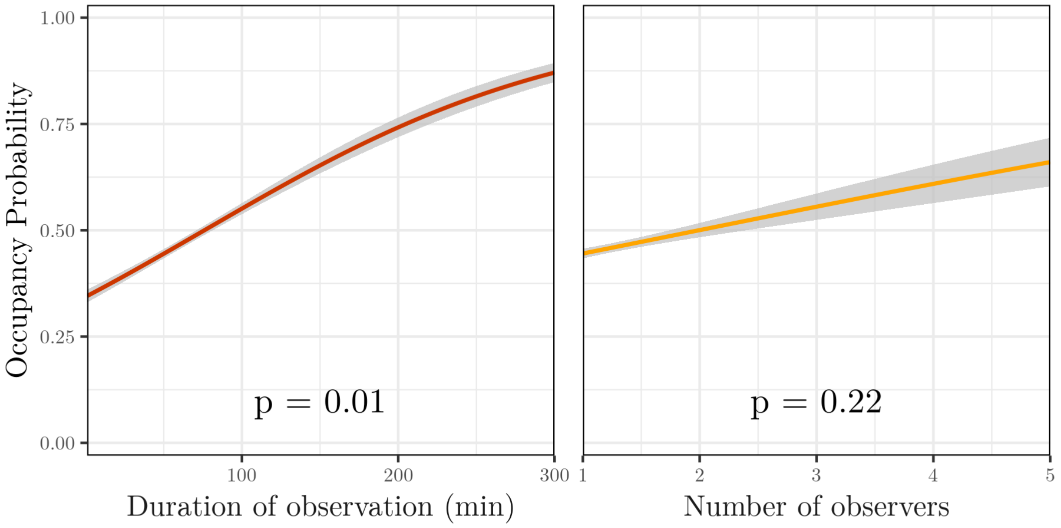

Figure 5: Predicted occupancy in response to detection covariates, for each plot only the focal detection covariate was varied, grey areas represent the standard error, the p values show the significance of the coefficient, created with Wickham (2016)

zero. In contrast, the number of observers does not show a significant result (see Figure 5).

The full model predicts for the grid cell with observational data that 522 out of 1032 grid cells are occupied (51 %, median best unbiased predictor from occurrence state). In 503 grid cells was the Black Kite detected (49 %). The MacKenzie and Bailey (2004) goodness-of-fit test shows that the full model adequately fits the data ($\hat{c} = 0.83$, $p = 1$). Also the parametric bootstrap test does not indicate a lack of fit ($\hat{c} = 0.97$, $p = 0.86$). The $\hat{c}$-values are below zero, for this reason no QAIC and QAICc are calculated (they have the same values as the AIC and AICc when setting the $\hat{c}$-value to one).

For the area of the mainland of Spain, the mean occupancy probability under actual climate conditions is 52.6 % (16748 grid cells with occupancy probability >= 0.5). The mean occupancy probability decreases and is 37.8 % under future climate conditions (10290 grid cells with occupancy probability >= 0.5). This decrease in occupancy is also visible in the map especially between Sevilla and Madrid (see Figure 6). In 92 % of all grid cells there is a lower occupancy probability under future climate conditions, in contrast in only 8 % of all grid cells the occupancy probability is higher under future climate conditions. The average model predicts very similar results (see Figure 10 in the appendix).

4 Discussion

Our results imply that the climate change may negatively affect the occupancy of the Black Kite in Spain. However even if higher annual mean temperature has a clear negative effect on the occupancy of the Black Kite, the uncertainty for the annual precipitation is high. The climate predictions from the IPCC (Iturbide et

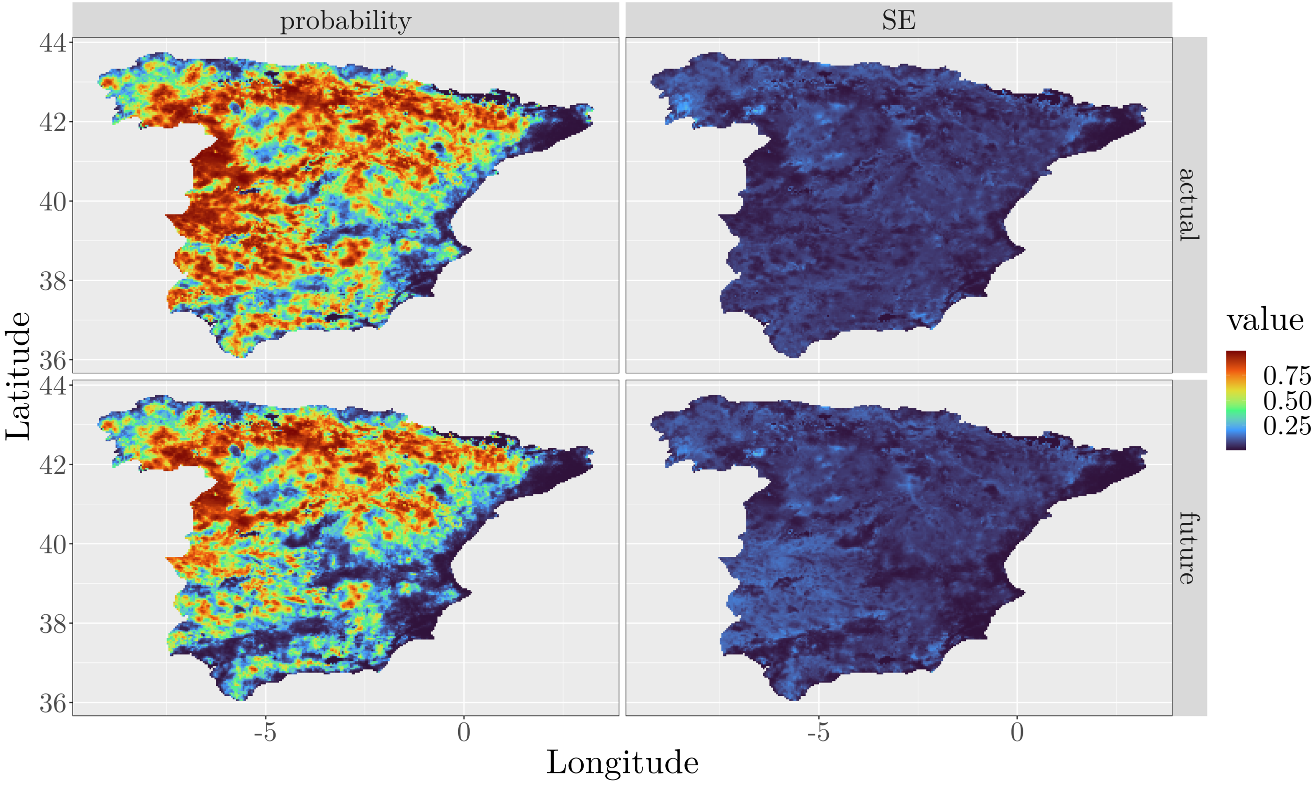

Figure 6: Occupancy map under actual and future climate conditions predicted with the full model, the occupancy probability is shown in the left maps, the corresponding standard error in the plots on the right side, created with Wickham (2016)

al. 2021) had a very coarse scale in comparison with the scale of the occupancy model (95 km versus 4.6 km edge length) and the models had no robust signal about the decrease in annual precipitation in Spain. Furthermore, the annual mean temperature and the annual precipitation are not the only factors that influence the occupancy of the Black Kite. Also changes in the habitat will affect the occurrence of the Black Kite. Additionally, many biological mechanisms of the Black Kite are not covered in this simple model.

Tanferna et al. (2013) point out some challenges in the protection of birds of prey. These are the large home range sizes (especially for younger not breeding individuals), they uses different habitat types in different seasons of the year and they often need nonidentical habitat characteristics on different spatial scales. That is why it is not so easy to derive conservation management measures from this study. Nevertheless, the grass cover and distance to closest river or lakes play an important role for the occupancy probability of the Black Kite in Spain as already mentioned in the literature (Tanferna et al. 2013; Veiga and Hiraldo 1990).

However, no effect of the distance to closest landfill was detected here as proposed by Blanco (1997) and Blanco (1994). It is unlikely that landfills really do not play a role. A possible explanations for this mismatch are that large landfills that are probably well visible in the Corine Land Cover data set (Copernicus Land Monitoring Service 2018) are often covered in reality and not accessible to the Black Kite. It is also possible that the citizen scientist did not look often around landfills and therefore the data fit is not good.

The duration of observation the only significant detection covariate in this study. Maybe other detection covariate are more appropriate than the detection covariates already present in the eBird data set like the weather at the day of the observation.

A possible extension of this static occupancy model is the dynamic occupancy model. A dynamic occupancy model with observations from more than one year can give more insights on the population trends (Green et al. 2019).

References

Araújo, M. B. and C. Rahbek (2006). "How does climate change affect biodiversity?" In: *Science* 313.5792, pp. 1396–1397. DOI: 10.1126/science.1131758.

Bartoń, K. (2020). *MuMIn: Multi-Model Inference.* URL: https://cran.r-project.org/package=MuMIn.

Bellard, C., C. Bertelsmeier, P. Leadley, W. Thuiller, and F. Courchamp (2012). "Impacts of climate change on the future of biodiversity". In: *Ecology Letters* 15.4, pp. 365–377. DOI: 10.1111/j.1461-0248.2011.01736.x.

BirdLife International (2021). *Species factsheet: Milvus migrans.* URL: http://datazone.birdlife.org/species/factsheet/black-kite-milvus-migrans/ (visited on 07/19/2021).

Bivand, R. and C. Rundel (2020). *rgeos: Interface to Geometry Engine - Open Source ('GEOS').* URL: https://cran.r-project.org/package=rgeos.

Blanco, G. (1994). "Seasonal aboundance of Black kites associated with the rubbish dump of Madrid, Spain". In: *Journal of Raptor Research* 28.4, pp. 242–245.

– (1997). "Role of refuse as food for migrant, floater and breeding Black Kites (Milvus migrans)". In: *Journal of Raptor Research* 31.1, pp. 71–76.

Blas, J., F. Sergio, and F. Hiraldo (2009). "Age-related improvement in reproductive performance in a long-lived raptor: A cross-sectional and longitudinal study". In: *Ecography* 32.4, pp. 647–657. DOI: 10.1111/j.1600-0587.2008.05700.x.

Buchhorn, M., M. Lesiv, et al. (2020). "Copernicus Global Land Cover Layers: Collection 2". In: *Remote Sensing* 12.6, p. 1044. DOI: 10.3390/rs12061044.

Buchhorn, M., B. Smets, et al. (2020). *Copernicus Global Land Service: Land Cover 100m: collection 3: epoch 2019: Globe.* DOI: 10.5281/zenodo.3939050.

Copernicus Land Monitoring Service (2018). *Corine Land Cover (CLC) 2018*. European Union, European Environment Agency (EEA). URL: https://land.copernicus.eu/pan-european/corine-land-cover/clc2018.

Cornell Lab of Ornithology (2021). *eBird Basic Dataset*. Ithaca, New York. URL: https://ebird.org/data/download (visited on 03/01/2021).

De Giacomo, U. and G. Guerrieri (2008). "The feeding behavior of the Black Kite (Milvus migrans) in the rubbish dump of Rome". In: *Journal of Raptor Research* 42.2, pp. 110–118. DOI: 10.3356/JRR-07-09.1.

Donázar, J. A. et al. (2016). "Roles of Raptors in a Changing World: From Flagships to Providers of Key Ecosystem Services". In: *Ardeola* 63.1, pp. 181–234. DOI: 10.13157/arla.63.1.2016.rp8.

Fick, S. E. and R. J. Hijmans (2017). "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas". In: *International Journal of Climatology* 37.12, pp. 4302–4315. DOI: 10.1002/joc.5086.

Fiske, I. and R. Chandler (2011). "unmarked: An R Package for Fitting Hierarchical Models of Wildlife Occurrence and Abundance". In: *Journal of Statistical Software* 43.10, pp. 1–23. DOI: 10.18637/jss.v043.i10.

Garcia, R. A., M. Cabeza, C. Rahbek, and M. B. Araújo (2014). "Multiple dimensions of climate change and their implications for biodiversity". In: *Science* 344.6183. DOI: 10.1126/science.1247579.

Green, A. W., D. C. Pavlacky, and T. L. George (2019). "A dynamic multi-scale occupancy model to estimate temporal dynamics and hierarchical habitat use for nomadic species". In: *Ecology and Evolution* 9.2, pp. 793–803. DOI: 10.1002/ece3.4822.

Harris, C. R. et al. (2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

Heiberger, R. M. (2020). *HH: Statistical Analysis and Data Display: Heiberger and Holland*. URL: https://cran.r-project.org/package=HH.

Hijmans, R. J. (2021). *raster: Geographic Data Analysis and Modeling*. URL: https://cran.r-project.org/package=raster.

Ho, T. K. (1995). "Random decision forests". In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* 1, pp. 278–282. DOI: 10.1109/ICDAR.1995.598994.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: 10.1109/MCSE.2007.55.

Iturbide, M. et al. (2021). *Repository supporting the implementation of FAIR principles in the IPCC-WG1 Atlas*. DOI: 10.5281/zenodo.3691645. URL: https://github.com/IPCC-WG1/Atlas (visited on 08/13/2021).

Lemaître, G., F. Nogueira, and C. K. Aridas (2017). “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning”. In: *Journal of Machine Learning Research* 18.17, pp. 1–5. URL: http://jmlr.org/papers/v18/16-365.

MacKenzie, D. I. and L. L. Bailey (2004). “Assessing the fit of site-occupancy models”. In: *Journal of Agricultural, Biological, and Environmental Statistics* 9.3, pp. 300–318. DOI: 10.1198/108571104X3361.

MacKenzie, D. I., J. D. Nichols, et al. (2002). “Estimating site occupancy rates when detection probabilities are less than one”. In: *Ecology* 83.8, pp. 2248–2255. DOI: 10.1890/0012-9658(2002)083[2248:ESORWD]2.0.CO;2.

Mazerolle, M. J. (2020). *AICcmodavg: Model selection and multimodel inference based on (Q)AIC(c).* URL: https://cran.r-project.org/package=AICcmodavg.

McKinney, W. (2010). “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference.* Ed. by S. van der Walt and J. Millman, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

Nagelkerke, N. J. D. (1991). “A note on a general definition of the coefficient of determination”. In: *Biometrika* 78.3, pp. 691–692. DOI: 10.1093/biomet/78.3.691.

Panuccio, M., N. Agostini, U. Mellone, and G. Bogliani (2014). “Circannual variation in movement patterns of the Black Kite (Milvus migrans migrans): A review”. In: *Ethology Ecology and Evolution* 26.1, pp. 1–18. DOI: 10.1080/03949370.2013.812147.

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html.

QGIS Development Team (2021). *QGIS Geographic Information System.* QGIS Association. URL: https://www.qgis.org.

R Core Team (2021). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.r-project.org/.

Reich, B. J., K. Pacifici, and J. W. Stallings (2018). “Integrating auxiliary data in optimal spatial design for species distribution modelling”. In: *Methods in Ecology and Evolution* 9.6, pp. 1626–1637. DOI: 10.1111/2041-210X.13002.

Riahi, K. et al. (2017). “The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview”. In: *Global Environmental Change* 42, pp. 153–168. DOI: 10.1016/j.gloenvcha.2016.05.009.

Robinson, O. J., V. Ruiz-Gutierrez, and D. Fink (2018). “Correcting for bias in distribution modelling for rare species using citizen science data”. In: *Diversity and Distributions* 24.4, pp. 460–472. DOI: 10.1111/ddi.12698.

Sergio, F. and A. Boto (1999). “Nest dispersion, diet, and breeding success of Black Kites (Milvus migrans) in the Italian pre-Alps”. In: *Journal of Raptor Research* 33.3, pp. 207–217.

State Meteorological Agency (AEMET) (2021). *Climate projections for the XXI Century.* URL: http://www.aemet.es/en/serviciosclimaticos/cambio%7B%5C_%7Dclimat/ (visited on 08/02/2021).

Steen, V. A., C. S. Elphick, and M. W. Tingley (2019). “An evaluation of stringent filtering to improve species distribution models from citizen science data”. In: *Diversity and Distributions* 25.12, pp. 1857–1869. DOI: 10.1111/ddi.12985.

Sullivan, B. L. et al. (2009). “eBird: A citizen-based bird observation network in the biological sciences”. In: *Biological Conservation* 142.10, pp. 2282–2292. DOI: 10.1016/j.biocon.2009.05.006.

Tanferna, A., L. López-Jiménez, J. Blas, F. Hiraldo, and F. Sergio (2013). “Habitat selection by Black kite breeders and floaters: Implications for conservation management of raptor floaters”. In: *Biological Conservation* 160, pp. 1–9. DOI: 10.1016/j.biocon.2012.12.031.

Urban, M. C. et al. (2016). “Improving the forecast for biodiversity under climate change”. In: *Science* 353.6304. DOI: 10.1126/science.aad8466.

Van Rossum, G. and F. L. Drake (2009). *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Veiga, J. P. and F. Hiraldo (1990). “Food habits and the survival and growth of nestlings in two sympatric kites (Milvus milvus and Milvus migrans)”. In: *Ecography* 13.1, pp. 62–71. DOI: 10.1111/j.1600-0587.1990.tb00590.x.

Vinuela, J. and J. P. Veiga (1992). “Importance of rabbits in the diet and reproductive success of black kites in southwestern Spain”. In: *Ornis Scandinavica* 23.2, pp. 132–138. DOI: 10.2307/3676440.

Wagenmakers, E. J. and S. Farrell (2004). “AIC model selection using Akaike weights”. In: *Psychonomic Bulletin and Review* 11.1, pp. 192–196. DOI: 10.3758/BF03206482.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org.

Zurell, D. et al. (2020). “A standard protocol for reporting species distribution models”. In: *Ecography* 43.9, pp. 1261–1277. DOI: 10.1111/ecog.04960.

Supplementary Information

The appendix consists of extra graphics and code for repeating the analysis.

Graphics

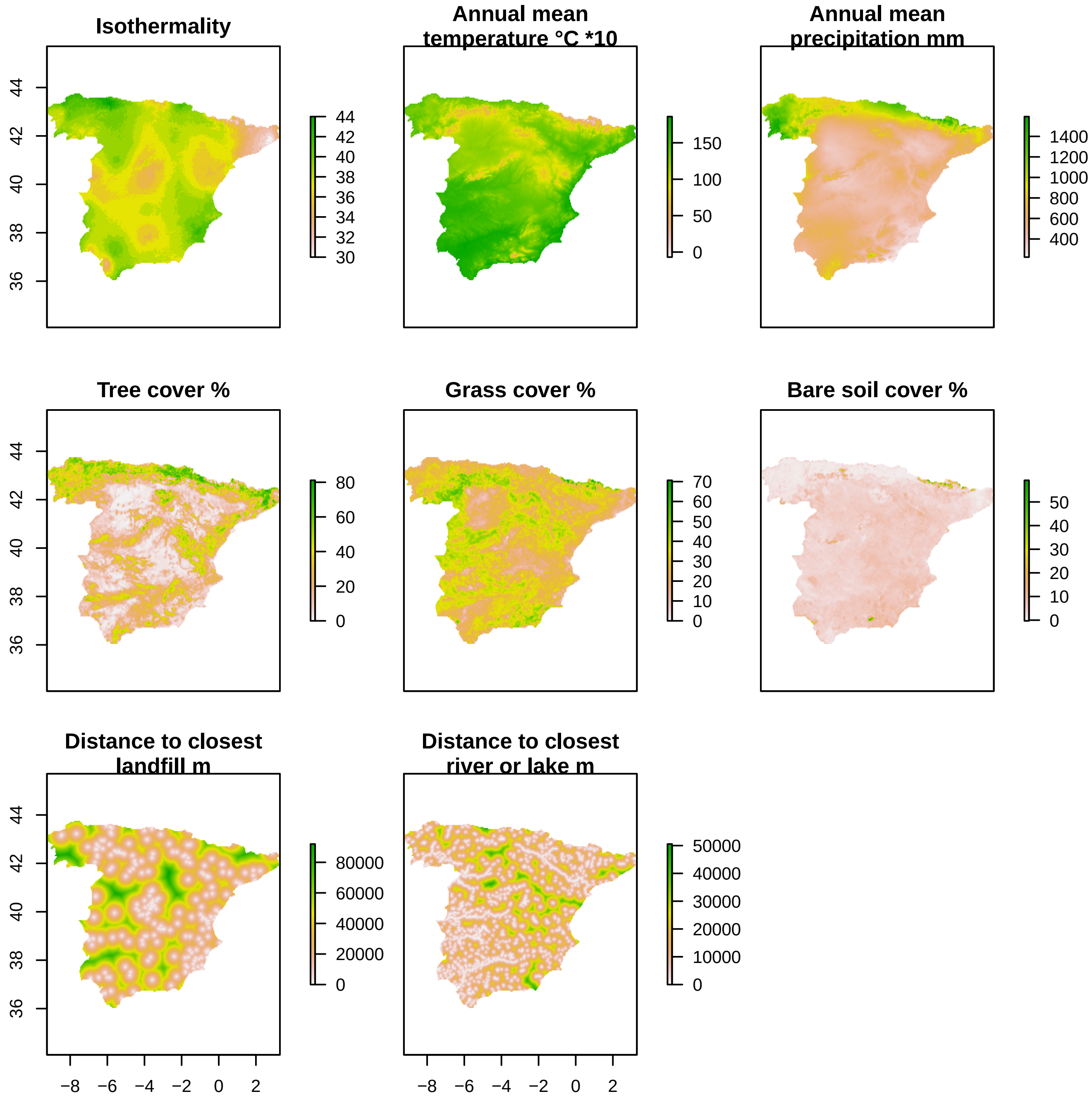

Figure 7: Maps of all site covariates, created with Hijmans (2021)

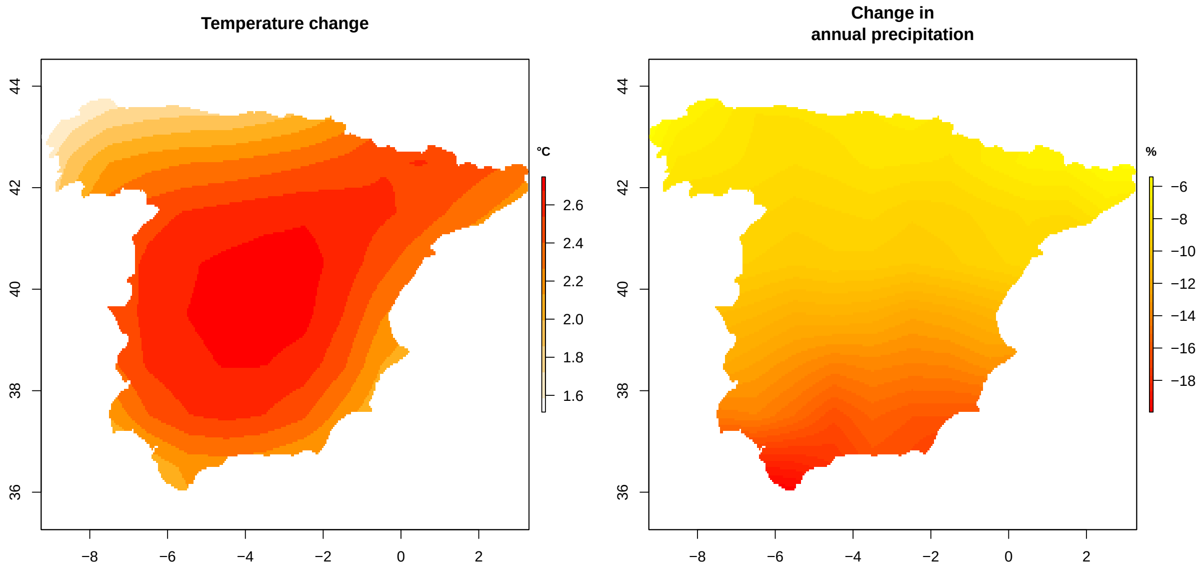

Figure 8: Change in the annual mean temperature and the annual precipitation between 1995-2014 and 2081-2100 with the SSP2 4.5 pathway, mean of 34 models (Iturbide et al. 2021), created with Hijmans (2021)

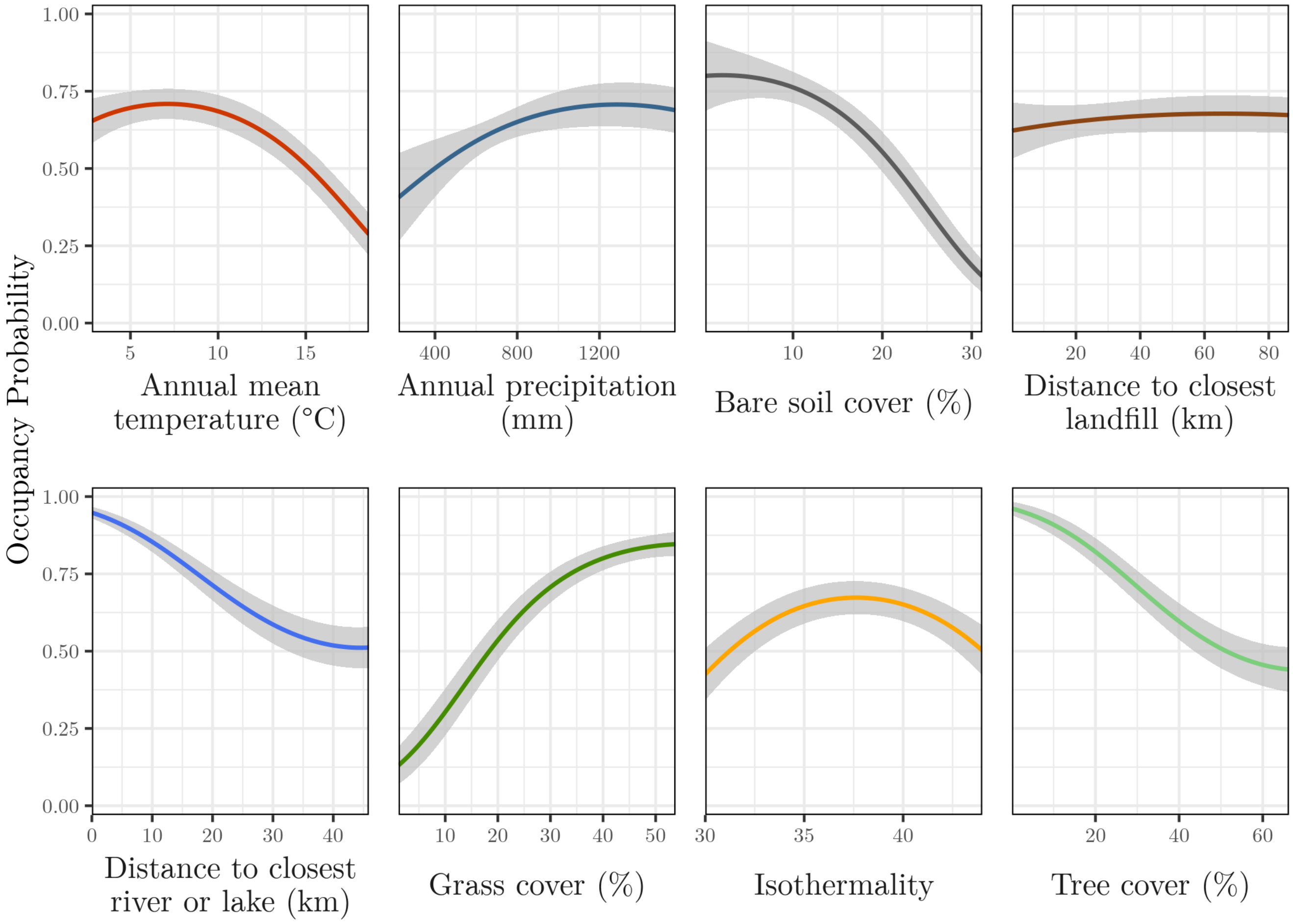

Figure 9: Predicted occupancy of the average model in response to site covariates, for each plot only the focal site covariate was varied, grey areas represent the standard error, created with Wickham (2016)

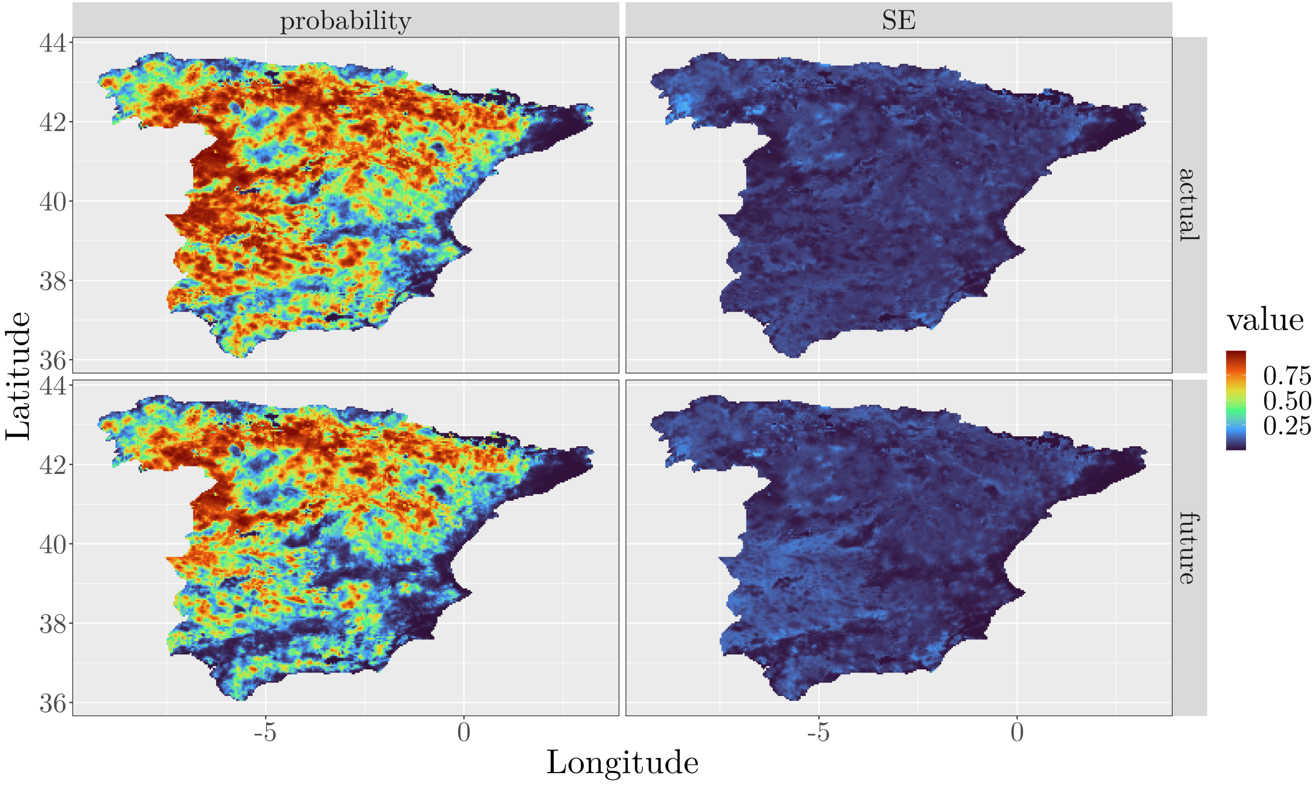

Figure 10: Occupancy map under actual and future climate conditions predicted with the average model, the occupancy probability is shown in the left maps, the corresponding standard error in the plots on the right side, created with Wickham (2016)

Code

The data and the R and Python Code can be found at
https://github.com/FelixNoessler/QCB_Black_Kite

The R-scripts for running the analysis are also embedded here (Python script is further down on page 53):

```r
##########################################################
# First script
# Prepare the environmental data
#   - Land cover data
#   - Bioclimatic variables
#   - Distance to closest landfill and river or lake
#
# save everything as a raster stack in the target
# spatial resolution
#
# Prepare the change in annual temperature and
# in annual precipiytion for the prediction of
# of the futre climate condtions
#
##########################################################



# Install required packages -----------------------------------------------
packages <- c("ggplot2", "gridExtra", "dplyr",
              "tidyr", "purrr", "HH", "psych", "MuMIn",
              "rnaturalearth", "rmapshaper",
              "auk", "unmarked", "AICcmodavg",
              "raster", "rgeos", "sp", "sf")

install.packages(setdiff(packages, rownames(installed.packages())))


# Loading packages --------------------------------------------------------
library(ggplot2)
library(dplyr)
```

```r


# Loading geometries for the mainland of Spain ----------------------------
spain <- rnaturalearth::ne_countries(country = 'spain',
                                     scale = 'medium',
                                     returnclass = 'sf')
# spain %>%
#   ggplot()+
#   geom_sf(fill='black')


spain_crop <- rmapshaper::ms_filter_islands(spain,
                                            min_area = 100000000000,
                                            drop_null_geometries=T)
# plot(sf::st_geometry(spain_crop))


# Prepare climate data ----------------------------------------------------
### Climate data

if (!file.exists('data/environmental_data/clim.grd')) {
  clim <- raster::getData('worldclim',
                          var = 'bio',
                          res = 2.5,
                          download = F,
                          path = 'data/environmental_data')

  raster::xres(clim) * 111.19

  clim <- raster::crop(clim, spain_crop)
  clim <- raster::mask(clim, spain_crop)

  raster::writeRaster(clim,
                      'data/environmental_data/clim.grd',
                      format = 'raster',
                      options = 'INTERLEAVE=BAND',
                      overwrite = TRUE)
} else {
  clim <- raster::brick('data/environmental_data/clim.grd')
```

```r
}


# Prepare land cover data -------------------------------------------------

### Tree cover

if (!file.exists('data/environmental_data/lc_tree.grd')) {
  lc_tree1 <- raster::raster('data/environmental_data/tree_cover1.tif')
  lc_tree1 <- raster::crop(lc_tree1, spain_crop)
  lc_tree1 <- raster::mask(lc_tree1, spain_crop)

  lc_tree2 <- raster::raster('data/environmental_data/tree_cover2.tif')
  lc_tree2 <- raster::crop(lc_tree2, spain_crop)
  lc_tree2 <- raster::mask(lc_tree2, spain_crop)

  lc_tree3 <- raster::raster('data/environmental_data/tree_cover3.tif')
  lc_tree3 <- raster::crop(lc_tree3, spain_crop)
  lc_tree3 <- raster::mask(lc_tree3, spain_crop)

  lc_tree4 <- raster::raster('data/environmental_data/tree_cover4.tif')
  lc_tree4 <- raster::crop(lc_tree4, spain_crop)
  lc_tree4 <- raster::mask(lc_tree4, spain_crop)

  lc_tree <- raster::merge(lc_tree1, lc_tree2, lc_tree3, lc_tree4)
  raster::plot(lc_tree)

  lc_tree_cover <- raster::resample(lc_tree, clim, method = 'bilinear')
  names(lc_tree_cover) <- 'tree_cover'

  raster::writeRaster(lc_tree_cover,
                      'data/environmental_data/lc_tree.grd',
                      format = 'raster',
                      options = 'INTERLEAVE=BAND',
                      overwrite = TRUE)
  rm(lc_tree1, lc_tree2, lc_tree3, lc_tree4, lc_tree)
} else {

  lc_tree_cover <- raster::raster('data/environmental_data/lc_tree.grd')
```

```r
}


### Herbaceous vegetation

if (!file.exists('data/environmental_data/lc_herbs.grd')) {

  lc_herbs1 <- raster::raster(
    'data/environmental_data/herbaceous_vegetation1.tif')
  lc_herbs1 <- raster::crop(lc_herbs1, spain_crop)
  lc_herbs1 <- raster::mask(lc_herbs1, spain_crop)

  lc_herbs2 <- raster::raster(
    'data/environmental_data/herbaceous_vegetation2.tif')
  lc_herbs2 <- raster::crop(lc_herbs2, spain_crop)
  lc_herbs2 <- raster::mask(lc_herbs2, spain_crop)

  lc_herbs3 <- raster::raster(
    'data/environmental_data/herbaceous_vegetation3.tif')
  lc_herbs3 <- raster::crop(lc_herbs3, spain_crop)
  lc_herbs3 <- raster::mask(lc_herbs3, spain_crop)

  lc_herbs4 <- raster::raster(
    'data/environmental_data/herbaceous_vegetation4.tif')
  lc_herbs4 <- raster::crop(lc_herbs4, spain_crop)
  lc_herbs4 <- raster::mask(lc_herbs4, spain_crop)

  lc_herbs <- raster::merge(lc_herbs1, lc_herbs2,
                            lc_herbs3, lc_herbs4)

  # raster::plot(lc_herbs)

  lc_herb_cover <- raster::resample(lc_herbs,
                                    clim,
                                    method = 'bilinear')
  names(lc_herb_cover) <- 'grass_cover'

  raster::writeRaster(lc_herb_cover,
              'data/environmental_data/lc_herbs.grd',
              format = 'raster',
              options = 'INTERLEAVE=BAND',
              overwrite = TRUE)

  rm(lc_herbs1, lc_herbs2, lc_herbs3, lc_herbs4, lc_herbs)
} else {

  lc_herb_cover <- raster::raster('data/environmental_data/lc_herbs.grd')
}


### Bare soil

if (!file.exists('data/environmental_data/lc_bare_soil.grd')) {
  lc_bare1 <- raster::raster('data/environmental_data/bare_soil1.tif')
  lc_bare1 <- raster::crop(lc_bare1, spain_crop)
  lc_bare1 <- raster::mask(lc_bare1, spain_crop)

  lc_bare2 <- raster::raster('data/environmental_data/bare_soil2.tif')
  lc_bare2 <- raster::crop(lc_bare2, spain_crop)
  lc_bare2 <- raster::mask(lc_bare2, spain_crop)

  lc_bare3 <- raster::raster('data/environmental_data/bare_soil3.tif')
  lc_bare3 <- raster::crop(lc_bare3, spain_crop)
  lc_bare3 <- raster::mask(lc_bare3, spain_crop)

  lc_bare4 <- raster::raster('data/environmental_data/bare_soil4.tif')
  lc_bare4 <- raster::crop(lc_bare4, spain_crop)
  lc_bare4 <- raster::mask(lc_bare4, spain_crop)

  lc_bare <- raster::merge(lc_bare1, lc_bare2, lc_bare3, lc_bare4)

  # raster::plot(lc_bare)

  lc_bare_soil <- raster::resample(lc_bare, clim, method = 'bilinear')
  names(lc_bare_soil) <- 'bare_soil'

  raster::writeRaster(lc_bare_soil,
                      'data/environmental_data/lc_bare_soil.grd',
                      format = 'raster',
                      options = 'INTERLEAVE=BAND',
                      overwrite = TRUE)

  rm(lc_bare1, lc_bare2, lc_bare3, lc_bare4, lc_bare)
}else{

  lc_bare_soil <- raster::raster('data/environmental_data/lc_bare_soil.grd')
}



# Calculate distance to closest river or lake -----------------------------


if (!file.exists('data/environmental_data/distance_to_water.grd')) {
  water <- sf::st_read('data/environmental_data/clc2018_vector/clc2018.gpkg',
                       query = "SELECT * FROM clc2018
                        WHERE Code_18 == 511
                        OR Code_18 == 512")

  plot(sf::st_geometry(water),  axes = TRUE)


  ## load only the first
  clim <- raster::raster('data/environmental_data/clim.grd')




  raster_points <-  as(clim,"SpatialPoints")
  water_poly <- as(water, "Spatial")

  water_poly@proj4string
  raster_points@proj4string



  crs1 <- sp::CRS('+proj=laea
                +lat_0=52
                +lon_0=10
                +x_0=4321000
                +y_0=3210000
                +ellps=GRS80
                +units=m
                +datum=WGS84 +no_defs')

raster_points_transformed <- sp::spTransform(raster_points, crs1)
plot(sf::st_as_sf(raster_points_transformed), cex=0.2, pch=15, axes=T)


water_poly_transformed <- sp::spTransform(water_poly, crs1)
plot(sf::st_as_sf(water_poly_transformed),  axes = TRUE)


dist1 <- rgeos::gDistance(raster_points_transformed[1:10000,],
                          water_poly_transformed,
                          byid=T)

dist2 <- rgeos::gDistance(raster_points_transformed[10001:20000,],
                          water_poly_transformed,
                          byid=T)

dist3 <- rgeos::gDistance(raster_points_transformed[20001:30220,],
                          water_poly_transformed,
                          byid=T)


min_distances <- c(apply(dist1,2,min),apply(dist2,2,min),apply(dist3,2,min))
data.frame(min_distances)

raster_points_df <- sp::SpatialPointsDataFrame(raster_points,
                                               data=data.frame(min_distances))


crs2 <- raster::projection(clim)
raster_points_df_backtransformed <- sp::spTransform(raster_points_df,
                                                    crs2)
```

```r

  dist_raster <- raster::rasterFromXYZ(raster_points_df_backtransformed)

  #raster::projection(dist_raster)
  #raster::projection(clim)
  #raster::plot(dist_raster)
  #raster::plot(clim)

  names(dist_raster) <- 'distance_to_water'
  raster::writeRaster(dist_raster,
                      'data/environmental_data/distance_to_water.grd',
                      format = 'raster',
                      options = 'INTERLEAVE=BAND',
                      overwrite = TRUE)

  distance_to_water <- dist_raster
}else {
  distance_to_water <- raster::raster(
    'data/environmental_data/distance_to_water.grd')
}



# Calculate distance to closest landfill ----------------------------------

if (!file.exists('data/environmental_data/distance_to_landfill.grd')) {
  landfills <- sf::st_read(
    'data/environmental_data/clc2018_vector/clc2018.gpkg',
    query = "SELECT * FROM clc2018
    WHERE Code_18 == 132")

  plot(sf::st_geometry(landfills),  axes = TRUE)
  landfills

  ## load only the first
  clim <- raster::raster('data/environmental_data/clim.grd')
```

```r

raster_points <-  as(clim,"SpatialPoints")
landfills_poly <- as(landfills, "Spatial")

landfills_poly@proj4string
raster_points@proj4string



crs1 <- sp::CRS('+proj=laea
                  +lat_0=52
                  +lon_0=10
                  +x_0=4321000
                  +y_0=3210000
                  +ellps=GRS80
                  +units=m
                  +datum=WGS84 +no_defs')

raster_points_transformed <- sp::spTransform(raster_points, crs1)
plot(sf::st_as_sf(raster_points_transformed), cex=0.2, pch=15, axes=T)


landfills_poly_transformed <- sp::spTransform(landfills_poly, crs1)
plot(sf::st_as_sf(landfills_poly_transformed),  axes = TRUE)


dist1 <- rgeos::gDistance(raster_points_transformed[1:10000,],
                          landfills_poly_transformed,
                          byid=T)

dist2 <- rgeos::gDistance(raster_points_transformed[10001:20000,],
                          landfills_poly_transformed,
                          byid=T)

dist3 <- rgeos::gDistance(raster_points_transformed[20001:30220,],
                          landfills_poly_transformed,
                          byid=T)


```

```


  min_distances <- c(apply(dist1,2,min),
                     apply(dist2,2,min),
                     apply(dist3,2,min))
  data.frame(min_distances)

  raster_points_df <- sp::SpatialPointsDataFrame(
    raster_points,
    data=data.frame(min_distances))


  crs2 <- raster::projection(clim)
  raster_points_df_backtransformed <- sp::spTransform(raster_points_df,
                                                      crs2)

  dist_raster <- raster::rasterFromXYZ(raster_points_df_backtransformed)


  #raster::projection(dist_raster)
  #raster::projection(clim)
  #raster::plot(log(dist_raster+1))
  #raster::plot(clim)

  names(dist_raster) <- 'distance_to_landfill'

  raster::writeRaster(dist_raster,
                      'data/environmental_data/distance_to_landfill.grd',
                      format = 'raster',
                      options = 'INTERLEAVE=BAND',
                      overwrite = TRUE)

  distance_to_landfill <- dist_raster
} else {
  distance_to_landfill <- raster::raster(
    'data/environmental_data/distance_to_landfill.grd')
}


```

```r

# Save all site covariates as one raster stack ----------------------------

variables <- raster::stack(clim,
                           lc_tree_cover,
                           lc_herb_cover,
                           lc_bare_soil,
                           distance_to_water,
                           distance_to_landfill)



raster::writeRaster(variables,
            'data/environmental_data/variables_spain.grd',
            format = 'raster',
            options = 'INTERLEAVE=BAND',
            overwrite = TRUE)

# rm(clim, lc_tree_cover, lc_herb_cover, lc_bare_soil,
#   distance_to_landfill, distance_to_water,
#   spain, spain_crop, variables)




# Changes in temperature and precipitation --------------------------------
prec_change <- raster::raster(
  'data/environmental_data/climate_change/precipitation.tiff')
temp_change <- raster::raster(
  'data/environmental_data/climate_change/temperature.tiff')


### Precipitation
prec_change <- raster::crop(prec_change, raster::extent(c(-12,5,30,50)))


#### edge length of one raster cell in km
prec_change_crs_m <- raster::projectRaster(
  prec_change,
  crs = '+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000
  +y_0=3210000 +ellps=GRS80 +units=m +no_defs')
poly1 <- raster::rasterToPolygons(prec_change_crs_m)
sqrt(raster::area(poly1[1,])) / 1000

prec_change_rs <- raster::resample(prec_change, clim, method = 'bilinear')
prec_change_rs <- raster::crop(prec_change_rs, spain_crop)
prec_change_rs <- raster::mask(prec_change_rs, spain_crop)

### Temperature
temp_change <- raster::crop(temp_change, raster::extent(c(-12,5,30,50)))

temp_change_rs <- raster::resample(temp_change, clim, method = 'bilinear')
temp_change_rs <- raster::crop(temp_change_rs, spain_crop)
temp_change_rs <- raster::mask(temp_change_rs, spain_crop)



# Plot the changes in a map -----------------------------------------------
pdf('results/climate_change.pdf', width=15, height=7)
par(mfrow=c(1,2),
    oma = c(3, 4, 1, 2) + 0.1,
    mar = c(0, 4, 4, 2) + 0.1)
pal1 = colorRampPalette(c('white', 'orange', 'red'))
raster::plot(temp_change_rs, main='Temperature change',
             col = pal1(10),
             legend.args = list(text = '°C', side = 3,
                                font = 2, line = 1, cex = 0.8))

pal2 = colorRampPalette(c('red', 'orange', 'yellow'))
raster::plot(prec_change_rs, main='Change in\nannual precipitation',
             col = pal2(30),
             legend.args = list(text = '%', side = 3,
                                font = 2, line = 1, cex = 0.8))

dev.off()
par(mfrow=c(1,1))
```

```r
# Save file as csv --------------------------------------------------------
points <- raster::rasterToPoints(raster::brick(temp_change_rs,
                                               prec_change_rs))
change <- data.frame(points)

# avoid precision loss when saving the data frame
change$x <- sprintf("%.20f",change$x)
change$y <- sprintf("%.20f",change$y)

write.csv(change, "results/change_temp_prec.csv", row.names = FALSE)


# Clean up ----------------------------------------------------------------
rm(list = ls())
```

```r
##########################################################
# Second script
#
#     1. Join the site covariates with the eBird data set
#     2. Standardize the site covariates
#     3. Do spatial subsampling
#     4. Store the data frame in the unmarked format
#
##########################################################


# Loading packages --------------------------------------------------------
library(dplyr)


# Load data ---------------------------------------------------------------

### Site covariates
variables <- raster::stack('data/environmental_data/variables_spain.grd')

### Bird data
milmig <- readr::read_csv('data/milmig.csv')

# filter for observations from the mainland of spain
# milmig <- milmig %>%
#  filter(!state_code=='ES-CN')


# Join eBird data and site covariates -------------------------------------
occ_var <- milmig %>%
  cbind(as.data.frame(
    raster::extract(variables,
                    milmig[,c('longitude', 'latitude')],
                    cellnumbers=T))) %>%
  tidyr::drop_na(bio1, tree_cover)

## save a data frame with not standardized covariates
write.csv(occ_var, "results/milmig_not_std.csv", row.names = FALSE)

```

```

# Standardize site covariates --------------------------------------------

occ_var_std <- occ_var%>%
  dplyr::mutate_at(c('bio1', 'bio2', 'bio3', 'bio4', 'bio5', 'bio6',
                     'bio7', 'bio8', 'bio9', 'bio10', 'bio11', 'bio12',
                     'bio13', 'bio14', 'bio15', 'bio16', 'bio17',
                     'bio18', 'bio19',
                     'tree_cover', 'grass_cover', 'bare_soil',
                     'distance_to_water', 'distance_to_landfill'),
                   ~(scale(.) %>% as.vector))


# Convert data frame to unmarked format -----------------------------------
occ_wide <- auk::format_unmarked_occu(
  occ_var_std, site_id = 'site',
  response = 'species_observed',
  site_covs =c('cells', 'n_observations', 'latitude', 'longitude',
               'bio1', 'bio2', 'bio3', 'bio4', 'bio5', 'bio6',
               'bio7', 'bio8', 'bio9', 'bio10', 'bio11', 'bio12',
               'bio13', 'bio14', 'bio15', 'bio16', 'bio17',
               'bio18', 'bio19',
               'tree_cover', 'grass_cover', 'bare_soil',
               'distance_to_water', 'distance_to_landfill'),
  obs_covs =c('time_observations_started','duration_minutes',
              'effort_distance_km','number_observers', 'protocol_type',
              'day_of_year'))


### Convert the detection histories in 1=# presence/ 0= absence
# instead of TRUE/FALSE
cols <- sapply(occ_wide, is.logical)
occ_wide[, cols] <-lapply(occ_wide[, cols], as.numeric)


# Do spatial subsampling --------------------------------------------------
# We can only have one (3 - 10 times repeated) observation per one grid cell!
occ_wide_clean <- occ_wide[!duplicated(occ_wide$cells),]

```

```r
# Part that is removed:
1- nrow(occ_wide_clean)/nrow(occ_wide)


# Save unmarked data as csv -----------------------------------------------
write.csv(occ_wide_clean, "results/milmig.csv", row.names = FALSE)


# Clean up ----------------------------------------------------------------
rm(list = ls())
```

```
##########################################################
# Third script
#
#  Selection of covariates:
#     ... are based on ecological hypotheses and from
#         from the explanatory analysis with a Random
#         Forest model
#
#     --> here some covariates are excluded because of
#         collinerarity
#
##########################################################


# Load data ---------------------------------------------------------------
occ_wide_clean <- read.csv("results/milmig.csv")


# Selected covariates -----------------------------------------------------
random_forest_selection <- c('bio3', 'bio4', 'bio7', 'bio5')

ecological_selection <- c('bio1', 'bio12',
                          'tree_cover', 'grass_cover', 'bare_soil',
                          'distance_to_landfill',  'distance_to_water')

selected_covariates <- c(random_forest_selection, ecological_selection)
selection <- occ_wide_clean[, selected_covariates]


# Exclude some covariates -------------------------------------------------
selected_names <- names(selection)[! names(selection) %in% c('bio7', 'bio4', 'bio5')]


# Make a cluster dendrogram -----------------------------------------------
## cluster with all selected covariates
cor1 <- abs(as.dist(cor(selection)))
clust1 <- hclust(1- cor1)
plot(clust1)

```

```r
## cluster with some covariates removed
cor1 <- abs(as.dist(cor(selection[, selected_names])))
clust1 <- hclust(1- cor1)
plot(clust1)


# Correlation plots -------------------------------------------------------
## Correlation plots with all covariates
psych::pairs.panels(selection)

## Correlation plots with some covariates removed
psych::pairs.panels(selection[, selected_names])


# Test for collinearity ---------------------------------------------------
## Test for collinearity with all covariates
HH::vif(selection)

## Test for collinearity with some covariates removed
HH::vif(selection[, selected_names])


# Make maps of the final selected covariates ------------------------------
site_covariates <- raster::brick('data/environmental_data/variables_spain.grd')
site_covariates <- site_covariates[[selected_names]]

pdf('results/site_covs.pdf')
par(mfrow=c(1,1),
    oma = c(0, 0, 0, 1) + 0.1,
    mar = c(0, 4, 10, 2) + 0.1)
raster::plot(site_covariates, main=c('Isothermality',
                                     'Annual mean\ntemperature °C *10',
                                     'Annual mean\nprecipitation mm',
                                     'Tree cover %',
                                     'Grass cover %',
                                     'Bare soil cover %',
                                     'Distance to closest\nlandfill m',
                                     'Distance to closest\nriver or lake m'))
dev.off()
```

```r



# Clean up ----------------------------------------------------------------
rm(list = ls())
dev.off(dev.list()["RStudioGD"])
```

```r
##########################################################
# Fourth script
#
# Build models with the unmarked package
#
# --> Compare different models with
#     information criterion
#
# --> evaluate the best model
#
# --> build an average model
#
##########################################################

#
setwd("/home/felix/Dokumente/studium/Potsdam/Module/Biogeography/Black kite/R")

# Load packages -----------------------------------------------------------
library(dplyr)
library(ggplot2)
library(unmarked)

# Load data ---------------------------------------------------------------
occ_um <- formatWide(read.csv("results/milmig.csv"), type = "unmarkedFrameOccu")
summary(occ_um)


# Build models ------------------------------------------------------------
## Null model
occ_null <- occu(~ 1 ~ 1, occ_um)
summary(occ_null)
backTransform(occ_null, "state")

## Model only with detection covariates
detection_cov_model <- occu(~ duration_minutes
                                      + effort_distance_km
                                      ~ 1, data=occ_um)
summary(detection_cov_model)

```

```

## Model only with site covariates
site_cov_model <-occu(~ 1
                                    ~ poly(bio1, 2)
                                    + poly(bio3, 2)
                                    + poly(bio12, 2)
                                    + poly(tree_cover, 2)
                                    + poly(grass_cover, 2)
                                    + poly(bare_soil, 2)
                                    + poly(distance_to_water, 2)
                                    + poly(distance_to_landfill, 2)
                                    , data = occ_um)
summary(site_cov_model)




## Full model with covariates
full_model <- occu(~ duration_minutes
                                 + number_observers
                                 ~ poly(bio1, 2)
                                 + poly(bio3, 2)
                                 + poly(bio12, 2)
                                 + poly(tree_cover, 2)
                                 + poly(grass_cover, 2)
                                 + poly(bare_soil, 2)
                                 + poly(distance_to_water, 2)
                                 + poly(distance_to_landfill, 2), data = occ_um)
summary(full_model)

re <- ranef(full_model)
sum(bup(re, stat="mode"))
sum(bup(re, stat="mean"))


# Model selection ---------------------------------------------------------
## Model selection with AIC
models_list <-list(Null = occ_null,
                   detection = detection_cov_model,
                   site = site_cov_model,
                   full_model = full_model)

un_models <- fitList(fits = models_list)
ModSelect <- modSel(un_models, nullmod = "Null")
ModSelect


## Model selection with AICc
AICcmodavg::aictab(models_list, second.ord = T)


best_model <- full_model



if (!file.exists('models.rda')) {

# Goodness of fit test of best model --------------------------------------
GOF <- parboot(best_model, nsim=500, ncores=8, report=T)
GOF

cHat <- GOF@t0 / mean(GOF@t.star)
cHat

### Another goodnes of fit test
AICcmodavg::mb.gof.test(best_model,
                        nsim=500,
                        plot.hist = F,
                        parallel=T,
                        ncores=8)



### QAICc
GOF1 <- AICcmodavg::aictab(models_list, c.hat = 1)
# --> it is the same as above, because the cHat value is below one


# Build an average model --------------------------------------------------
  ## Get the names of the detection covariates
  det_terms <- MuMIn::getAllTerms(best_model) %>%
    purrr::discard(stringr::str_detect, pattern = "psi")

  ## Get combination of models, detection covariates are always present
  occ_dredge <- MuMIn::dredge(best_model, fixed = det_terms)

  ## Get the best models from the model list
  occ_dredge_95 <- MuMIn::get.models(occ_dredge,
                                      subset = cumsum(weight) <  0.95)

  ## Get the average model based on model weights
  #occ_avg <- MuMIn::model.avg(occ_dredge, fit = TRUE, revised.var = TRUE)
  occ_avg <- MuMIn::model.avg(occ_dredge_95, fit=T)

  ## Calculate the AICc for the average model
  sum(occ_avg$msTable$AICc * occ_avg$msTable$weight)

  ## Model coefficients of the average model
  t(occ_avg$coefficients)

  MuMIn::importance(occ_avg)

  save(occ_avg, best_model, GOF, cHat, GOF1, file='models.rda')
}

# Clean up ----------------------------------------------------------------
rm(list = ls())
```

```r
##########################################################
# Fifth script
#
#   Make all plots:
#       - Maps
#       - response of occupancy due to covariates
#
##########################################################

library(dplyr)
library(ggplot2)

# Load data ---------------------------------------------------------------
load('models.rda')
change <- read.csv('results/change_temp_prec.csv')
variables <- raster::brick("data/environmental_data/variables_spain.grd")


# Prepare data for predictions --------------------------------------------
variables_selection <- c("bio1",
                         "bio3",
                         "bio12",
                         "tree_cover",
                         "grass_cover",
                         "bare_soil",
                         "distance_to_water",
                         "distance_to_landfill")

variables.sel <- variables[[variables_selection]]

p_variables <- data.frame(raster::rasterToPoints(variables.sel) )
p_variables <- p_variables %>%
  tidyr::drop_na(tree_cover, bio1)

change



change_joined <- p_variables %>%
  left_join(change, by = c('x', 'y')) %>%
  select(x,y, temperature, precipitation) %>%
  mutate(temperature = temperature * 10,
         precipitation = 1 + precipitation/100)

mean(change_joined$temperature, na.rm=T)/10
mean(change_joined$precipitation, na.rm=T)



p_variables_std <- p_variables %>%
  mutate_at(variables_selection, ~(scale(.) %>% as.vector))

sd_bio1 <- sd(p_variables$bio1)
mean_bio1 <- mean(p_variables$bio1/ sd(p_variables$bio1))

sd_bio12 <- sd(p_variables$bio12)
mean_bio12 <- mean(p_variables$bio12/ sd(p_variables$bio12))

p_variables_std_future <- p_variables_std
p_variables_std_future$bio1 <-
  (p_variables$bio1 + change_joined$temperature) / sd_bio1 - mean_bio1
p_variables_std_future$bio12 <-
  (p_variables$bio12 * change_joined$precipitation) / sd_bio12 - mean_bio12

# Make predictions --------------------------------------------------------
# actual

# occ_avg, best_model
pred_actual <- unmarked::predict(occ_avg,
                                 newdata = select(p_variables_std,
                                                  -x, -y),
                                 type = "state")

# Predicted, fit
# SE, se.fit
actual_climate <- bind_cols(p_variables_std,
                                   probability = pred_actual$fit,
                                   SE = pred_actual$se.fit) %>%
  select(x, y, probability, SE) %>%
  tidyr::pivot_longer(cols = c(probability, SE))


# future

# occ_avg, best_model
pred_future <- unmarked::predict(occ_avg,
                                 newdata = select(p_variables_std_future,
                                                  -x, -y),
                                 type = "state")

# Predicted, fit
v
future_climate <- bind_cols(p_variables_std_future,
                                   probability = pred_future$fit,
                                   SE = pred_future$se.fit) %>%
  select(x, y, probability, SE) %>%
  tidyr::pivot_longer(cols = c(probability, SE))

# join the data, preparation for plotting
data <- actual_climate %>%
  inner_join(future_climate, by = c("x", "y", "name")) %>%
  rename(actual = value.x,
         future = value.y,
         type = name)  %>%
  tidyr::pivot_longer(cols=c(actual, future))

# results/predictions_best_model.csv, or results/predictions_avg_model.csv
write.csv(data, 'results/predictions_avg_model.csv', row.names = F)

#### Plotting the map


# Plot the map ------------------------------------------------------------

# or load data:
# data <- read.csv('results/predictions_best_model.csv')
# data <- read.csv('results/predictions_avg_model.csv')
```

```

data %>%
  ggplot(aes(x,y, fill=value))+
  geom_raster()+
  scale_fill_viridis_c(name="value", option="turbo")+
  theme(panel.border=element_rect(color="black",fill="transparent"),
        text = element_text(size=20))+
  labs(x="Longitude", y="Latitude")+
  coord_fixed()+
  facet_grid(~name ~ type)+
  theme(text = element_text(size=30, family = "LM Roman 10"))
# 'results/best_model_map.png' or 'results/avg_model_map.png'
ggsave('results/avg_model_map.png', width = 16, height=10)


# Mean probabilities
data %>%
  group_by(name) %>%
  filter(type == 'probability') %>%
  summarise(p = mean(value))

# sum of ells occupied
data %>%
  group_by(name) %>%
  filter(type == 'probability') %>%
  mutate(occ = value >= 0.5) %>%
  summarise(s = sum(occ))

data %>%
  tidyr::pivot_wider(names_from=name, values_from = value) %>%
  filter(type == 'probability') %>%
  mutate(lower = actual > future) %>%
  mutate(higher = actual < future) %>%
  summarise(lower = sum(lower), higher = sum(higher))

# total grid cells
nrow(data) / 4

# Prediction of covariates -----------------------------------------------
```

```

raw_data <- readr::read_csv('results/milmig_not_std.csv')
model_statistics <- readr::read_csv2('results/best_model.csv')

model_labels <- model_statistics %>%
  rename(p = `P(>|z|)`) %>%
  tidyr::pivot_wider(values_from=p, names_from = coef_no, id_cols=name) %>%
  filter(name != '-') %>%
  rename(first = `1`,
         second = `2`) %>%
  mutate(first = ifelse( round(first, 2) == 0,
                         '<0.00',
                         paste('', round(first, 2))),
         second = ifelse(round(second, 2) ==0,
                         '<0.00',
                         paste('', round(second, 2))),
         p =  paste('p = ',first, ', ', second, sep='')) %>%
  select(name, p) %>%
  mutate(name =
           recode(name,
                  bare_soil = "Bare soil cover (%)",
                  tree_cover = "Tree cover (%)",
                  grass_cover = "Grass cover (%)",
                  bio1 = "Annual mean\ntemperature (°C)",
                  bio3 = "Isothermality",
                  bio12 = "Annual precipitation\n(mm)",
                  distance_to_water = "Distance to closest\n
                  river or lake (km)",
                  distance_to_landfill = "Distance to closest\n
                  landfill (km)")) %>%
  mutate(x = c(10, 38, 800, 30, 30, 15, 25, 45),
         y = rep(0.1, 8))

variable_names <- c("bio1",
                    "bio3",
                    "bio12",
                    "bare_soil",
                    "tree_cover",
                    "grass_cover",
```

```r
                    "distance_to_water",
                    "distance_to_landfill")

rm(old_data)
for (i in seq_along(variable_names)) {
  variable_str <- variable_names[i]
  print(variable_str)


  newdata <- setNames(data.frame(
    matrix(ncol = length(variable_names),
           nrow = 1000)),
    variable_names)

  newdata[, i] <- seq(min(raw_data[, variable_str]),
                      max(raw_data[, variable_str]),
                      length.out = 100)

  newdata[is.na(newdata)] <- 0

  sd1 <- sd(newdata[, variable_str])
  mean1<- mean(newdata[, variable_str]/sd(newdata[, variable_str]))

  newdata[, variable_str] <- as.numeric(scale(newdata[, variable_str]) )

  # best_model, occ_avg
  predict_newdataset <- unmarked::predict(occ_avg,
                                          newdata = newdata,
                                          type = "state")
  # Predicted, fit
  # SE, se.fit
  plotting_data <- bind_cols(newdata,
                             occ_prob = predict_newdataset$fit,
                             occ_se = predict_newdataset$se.fit) %>%
    select(matches(variable_str), occ_prob, occ_se)

  plotting_data$x <- (plotting_data[, variable_str] + mean1)* sd1

  if (variable_str %in% c('bio1', 'bio2')) {
    plotting_data$x <- plotting_data$x / 10
  } else if (variable_str == 'landfills'){
    plotting_data$x <- plotting_data$x * 100
  } else if (variable_str %in% c('distance_to_water',
                                 'distance_to_landfill')){
    plotting_data$x <- plotting_data$x /1000
  }

  new_data <- plotting_data %>%
    mutate(lower_se = occ_prob - occ_se,
           upper_se = occ_prob + occ_se) %>%
    select(x, occ_prob, lower_se, upper_se) %>%
    mutate(name = variable_str)

  if (i == 1){
    old_data <- new_data
  } else {
    old_data <- old_data %>%
      bind_rows(new_data)
  }
}



old_data %>%
  mutate(name = recode(name,
         bare_soil = "Bare soil cover (%)",
         tree_cover = "Tree cover (%)",
         grass_cover = "Grass cover (%)",
         bio1 = "Annual mean\ntemperature (°C)",
         bio3 = "Isothermality",
         bio12 = "Annual precipitation\n(mm)",
         distance_to_water = "Distance to closest\nriver or lake (km)",
         distance_to_landfill = "Distance to closest\nlandfill (km)")) %>%
  #filter(name != "Distance to closest\nlandfill (km)") %>%
  ggplot()+
  geom_text(data = model_labels,
            aes(x = x, y = y, label = p),
            family="LM Roman 10",
            size=3.5)+
  geom_ribbon(aes(ymin = lower_se,
                  ymax = upper_se,
                  x = x),
              fill="gray", alpha=0.7) +
  geom_line(aes(x=x, y=occ_prob, color=factor(name)),
            size=0.8)+
  scale_color_manual(values =
                       c("Bare soil cover (%)" = "grey36",
                        "Annual mean\ntemperature (°C)" = "orangered3",
                        "Distance to closest\nriver or lake (km)" = "royalblue2",
                        "Distance to closest\nlandfill (km)" = "chocolate4",
                        "Tree cover (%)" = "palegreen3",
                        "Grass cover (%)" = "chartreuse4",
                        "Annual precipitation\n(mm)" = "steelblue4",
                        "Isothermality" = "orange"))+
  labs(x=NULL,
       y="Occupancy Probability")+
  theme_bw()+
  theme(legend.position = "none",
        panel.border = element_rect(color="black",fill="transparent"),
        text = element_text(size=10, family="LM Roman 10"),
        plot.margin=unit(c(2, 5, -5, 2), "points"),
        panel.spacing = unit(0.8, "lines"),
        strip.background = element_blank(),
        strip.placement = "outside",
        strip.text =  element_text(size=12, face='plain',
                                   margin = margin(t = 0, r = 0, b = 10, l = 0)),
        axis.title = element_text(size=12, face='plain'))+
  scale_x_continuous(expand = expansion(mult = c(0, 0))) +
  scale_y_continuous(limits = c(0,1),expand = expansion(mult = c(0.03, 0.03))) +
  facet_wrap(. ~ name, scales="free_x", strip.position = 'bottom', ncol=4)
# 'results/avg_model_site_covariates.png', 'results/site_covariates.png'
ggsave('results/avg_model_site_covariates.png', width = 7, height = 5)


######### detection covariates

summary(raw_data$duration_minutes)
summary(raw_data$number_observers)

variable_names <- c('number_observers', 'duration_minutes')
newdata = setNames(data.frame(
  matrix(ncol = length(variable_names),
         nrow = 2000)),
  variable_names)
newdata[1:1000, variable_names[1]] <- seq(1, 5, length.out=1000)
newdata[1:1000, variable_names[2]] <- colMeans(raw_data[, variable_names[2]],
                                               na.rm=T)

newdata[1001:2000, variable_names[2]] <- seq(1, 300, length.out=1000)
newdata[1001:2000, variable_names[1]] <- colMeans(raw_data[, variable_names[1]],
                                                  na.rm=T)

best_model

predict_labels <- data.frame(
  name = c('Number of observers', 'Duration of observation (min)'),
  p = c('p = 0.22', 'p = 0.01'),
  x = c(3, 150),
  y = c(0.1, 0.1))


predict_newdataset <- unmarked::predict(best_model,
                                        newdata = newdata,
                                        type = "det")

plotting_data <- bind_cols(newdata,
                           occ_prob = predict_newdataset$Predicted,
                           occ_se = predict_newdataset$SE) %>%
  mutate(lower_se = occ_prob - occ_se,
         upper_se = occ_prob + occ_se,
         name = c(rep('Number of observers', 1000),
                  rep('Duration of observation (min)', 1000)))

plotting_data$x <- c(plotting_data[1:1000, 'number_observers'],
                     plotting_data[1001:2000, 'duration_minutes'])

```

```
plotting_data %>%
  ggplot()+
  geom_text(data = predict_labels,
            aes(x = x, y = y, label = p),
            family="LM Roman 10", size=5)+
  geom_ribbon(aes(ymin = lower_se,
                  ymax = upper_se,
                  x = x),
              fill="gray", alpha=0.7) +
  geom_line(aes(x=x, y=occ_prob, color=factor(name)),
            size=0.8)+
  scale_color_manual(values =
                       c("Number of observers" = "orange",
                         "Duration of observation (min)" = "orangered3"))+
  labs(x=NULL,
       y="Occupancy Probability")+
  theme_bw()+
  theme(legend.position = "none",
        panel.border = element_rect(color="black",fill="transparent"),
        text = element_text(size=10, family="LM Roman 10"),
        plot.margin=unit(c(2, 5, -5, 2), "points"),
        panel.spacing = unit(0.8, "lines"),
        strip.background = element_blank(),
        strip.placement = "outside",
        strip.text =  element_text(size=12, face='plain',
                                   margin = margin(t = 0, r = 0,
                                                   b = 10, l = 0)),
        axis.title = element_text(size=12, face='plain'))+
  scale_x_continuous(expand = expansion(mult = c(0, 0))) +
  scale_y_continuous(limits = c(0,1), expand =
                       expansion(mult = c(0.03, 0.03))) +
  facet_wrap(. ~ name, scales="free_x", strip.position = 'bottom', ncol=4)
ggsave('results/det_covariates.png', width = 6, height = 3)
```

Python script for running the Random Forest model:

```python
from sklearn import model_selection, preprocessing
from imblearn import ensemble
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams.update({
    "font.family": "sans-serif",
    "font.sans-serif": ["Latin Modern Roman"]})

data = pd.read_csv('milmig.csv')

## observational covariates
observational_covariates = data.loc[:, 'time_observations_started.1':'day_of_year.10']
observational_covariates.loc[:,'site'] = data.site
stubnames = ['time_observations_started.', 'duration_minutes.', 'effort_distance_km.',
             'number_observers.','day_of_year.', 'protocol_type.']
obs_covariates_long = pd.wide_to_long(observational_covariates,
                                      stubnames,
                                      i='site',
                                      j='observation')
obs_covariates_long = obs_covariates_long.rename(
    columns={"time_observations_started.": "daytime",
             "duration_minutes.": "duration",
             "effort_distance_km.": "distance_km",
             "number_observers.": "n_observers",
             "day_of_year.": "day_of_year",
             "protocol_type.": "protocol_type"})
obs_covariates_long['protocol_type'] = preprocessing.LabelEncoder().fit_transform(
    obs_covariates_long['protocol_type'])
sc = preprocessing.StandardScaler(with_mean=False, with_std=False)
obs_covariates_transformed = sc.fit_transform(obs_covariates_long)


## site covariates
site_covariates = data.loc[:, 'bio1':'marshes']
site_covariates_long = site_covariates.loc[
    np.tile(np.arange(0, len(site_covariates)), 10)]



## join covariates to one table
covs = np.concatenate((site_covariates_long.values,
                       obs_covariates_transformed),
                      axis=1)
column_names = np.append(site_covariates_long.columns,
                         obs_covariates_long.columns)


## detection
observed = pd.melt(data.loc[:, 'y.1':'y.10'])

## simulation
n_sim = 500

feature_imp = []
pred_data_all = []
for i in range(n_sim):

    random_numbers = np.random.randint(1, data.n_observations)

    y = observed.iloc[random_numbers, 1]
    x = covs[random_numbers, :]

    x_train, x_test, y_train, y_test = model_selection.train_test_split(
        x, y, test_size=0.0001)
    clf = ensemble.BalancedRandomForestClassifier()

    clf.fit(x_train, y_train)
    feature_imp.append(clf.feature_importances_)

    predict_data_list = []
    for name in column_names:
        mean_x = np.mean(x, axis=0)
        dummy_data = np.tile(mean_x[:, None], 50)
        covariate = np.linspace(-2, 2, 50)
        dummy_data[column_names == name] = covariate
        pred = clf.predict_proba(dummy_data.T)
        predict_data_list.append(pred[:, 1])

    pred_data_all.append(predict_data_list)


feature_imp = np.array(feature_imp)
importance = np.mean(feature_imp.T, axis=1)
importance_sd = np.std(feature_imp.T, axis=1) / np.sqrt(n_sim)

proper_names ={"bio1": "Annual Mean Temperature",
              "bio2": "Mean Diurnal Range",
              "bio3": "Isothermality",
              "bio4": "Temperature Seasonality",
              "bio5": "Max Temperature of Warmest Month",
              "bio7": "Temperature Annual Range",
              "bio9": "Mean Temperature of Driest Quarter",
              "bio10": "Mean Temperature of Warmest Quarter",
              "bio12": "Annual Precipitation",
              "bio17": "Precipitation of Driest Quarter",
              "bio18": "Precipitation of Warmest Quarter",
              "distance_to_water": "Distance to closest river or lake",
              "distance_to_landfill": "Distance to closest landfill",
              "n_observers": "Number of observers",
              "duration": "Duration of observation"}

for key, val in zip(proper_names.keys(), proper_names.values()):
    column_names[key == column_names] = val

filter_index = importance > 0.02
selected_columns = column_names[filter_index]
selected_columns

plt.barh(np.arange(len(selected_columns)), np.sort(importance[filter_index]),
         xerr=importance_sd[filter_index][np.argsort(importance[filter_index])],
         align='center', color='#21918C', alpha=0.7)
plt.yticks(np.arange(len(importance[filter_index])),
           selected_columns[np.argsort(importance[filter_index])], size=12)
```

```python
plt.xlabel('Importance of covariates', size=12)
plt.tight_layout()

plt.savefig('importance_features.png', dpi=300)
plt.show()



#############
pred_data_all = np.array(pred_data_all)
plt.figure(figsize=(6, 6))

for i, name in enumerate(
        selected_columns[np.argsort(importance[filter_index])][::-1]):
    if i < 4:
        plt.subplot(2,2, i+1)
        data = pred_data_all[:, column_names == name, :]
        y_pred = np.mean(data, axis=1)[1]

        plt.plot(np.linspace(-2, 2, 50), y_pred, '.-k')
        plt.title(name)
plt.tight_layout()
plt.savefig('response.png', dpi=300)
plt.show()
```