University of St Andrews
Masters Thesis

Predicting Mediterranean Monk Seal Cave Occupancy in the Central Ionian Isles, Greece:

The Application of Open-Source Hardware and Artificial Intelligence to Conservation

---

Author:
200017683

Supervisor:
Dr. Luke Rendell

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in

Marine Mammal Science

17th August 2021

Declaration of Authorship:

I, 200017683 hereby certify that this dissertation, which is approximately 10,500 words in length, has been composed by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a degree.

This project was conducted by me, 200017683, at the University of St Andrews from January 2021 to August 2021 towards fulfilment of the requirements of the University of St Andrews for the degree of Master of Science in Marine Mammal Science under the supervision of Dr Luke Rendell.

Signed:

Date: 17th August 2021

Table of Contents

Abstract

Mediterranean monk seals (*Monachus* monachus; hereafter monk seals) are among the most endangered pinnipeds globally. In Eastern Mediterranean waters, monk seal haul outs and pupping sites consist almost entirely of remote and inaccessible marine caves. Non-invasive monitoring of monk seals is required for effective conservation. Inaccessibility of marine caves also necessitates autonomous data collection. Autonomous technologies such as camera traps are associated with large datasets or so-called 'big data'. In a big-data era, constraints no longer lie within data collection and cost of technology, but rather in analysing voluminous data effectively and efficiently with tools such as artificial intelligence.

Monk seal presence and vessel traffic were recorded at 3 cave sites in the Central Ionian Islands, Greece, with data collected between 23rd May – 30th November 2019. Open-source hardware was used to create solar powered autonomous camera traps, with data transmitted via mobile networks. Each site had two camera traps capturing the external and internal cave environment at regular 15-minute intervals. A total of 93,218 images were capture, of which 44,913 were internal and 48,215 were external. Seal presence was manually classified in images from inside caves, whereas an artificial intelligence approach of neural network image processing was used to detect vessels in the external cave environs. The rate of vessel presence in images was significant predictor of monk seal cave presence at both diel and seasonal scales ($p < 0.001$) using Generalised Additive Modelling. At 2 out of 3 sites, cave usage by monk seals peaked during September at a seasonal scale. Understanding impacts of vessel presence on monk seal cave usage can support the implementation of seasonal anchorage and mooring restrictions to limit disturbance to monk seal at haul out and pupping sites. Further expansion of camera trap monitoring to additional sites can inform conservation management actions.

[*Words*: *297*]

Acknowledgements

I would like to thank the Octopus Foundation for providing the dataset used for this thesis. I particularly would like to thank Julien Pfyffer and Andy Guinand for the time they have devoted to make this project possible. I would also like to thank all who supported the Octopus Foundation: Tethys Research Institute & Ionian Dolphin Project, Kosamare, Fiskardo Divers, Odyssey Outdoor Activities, Octanis and the Smile Wave Fund.

I am also grateful to all teaching staff on the MSc Marine Mammal Science program, who enabled me to develop the skills required to undertake such a project. I am especially grateful for the continued support and imparted knowledge from Dr Luke Rendell throughout the duration of this project and the academic year.

1. Introduction

Ecological composition is changing world-wide due to increased rates of extinction and the decline of top predators as a result of increasing human activity (Gonzalez et al., 2016; Ripple et al., 2014). To counteract increasing rates of extinction, urgent conservation of species most at risk is needed. Substantial knowledge and understanding of a species' life history, movements, habitat usage and population trends are required for the development and effective implementation of conservation measures (Guerra et al., 2019). Consequently, current conservation efforts prioritise species with sufficient data (Junker et al., 2020).

To gain sufficient conservation-relevant knowledge of endangered species with small population sizes, even minimally invasive methods such as externally attached telemetry devices may be detrimental (Zemanova, 2020). There is a growing need for non-invasive monitoring techniques for species at risk of extinction.

1.1. New Emerging Technologies in Conservation

The development of novel technologies and innovative solutions has long been integral to the study and conservation of endangered species (Arts et al., 2015; Berger-Tal and Lahoz-Monfort, 2018; Marvin et al., 2016). Emerging technologies and innovation such as remote sensing and autonomous data collection present ever-increasing opportunities to address key gaps in conservation-relevant knowledge (Arts et al., 2015). This can include identifying critical habitat, quantifying human impacts, and long-term population monitoring (Booth et al., 2020). Advancing technologies can improve both the cost-effectiveness and quality of data collection because of reduced costs of instrumentation, increased device sensitivity and increased sampling rates (Chan et al., 2021). Examples of emerging technologies being used to study marine mammals include remote sensing to monitor marine mammals through satellite imagery (Fretwell et al., 2014), animal-borne telemetry (Harcourt et al., 2019), and use of Unmanned Aerial Vehicles (UAVs) or drones to assess body condition of whales (Christiansen et al., 2020; Durban et al., 2015). Additionally, improved accessibility to cloud computing, increased computational power, and digital data networks have lessened geographical and physical constraints to data collection, allowing for more comprehensive and complete scientific evidence (Berger-Tal and Lahoz-Monfort, 2018; Edgar et al., 2016).

Technological advances have enabled the collection of large datasets, or so-called 'big data', at a minimal cost without requiring in-person data collection. Financial constraints now lie within the ability to fund the handling and analysis of large quantities of data, as opposed to the physical collection of data. However, limited funding can often constrain the consistency, scale and analysis of datasets to inform conservation (Walls, 2018). To maintain the integrity of inferences drawn from automated data collection, multiple checks by numerous people are needed at a 'big data' scale, increasing cost of analysis.

1.1.2. Autonomous Image Collection

Direct observations of animals in their natural habitat are advantageous and preferable to ex-situ or captive study, and the latter may not even be possible for endangered species (Meek et al., 2014). However, in-situ observations can be labour intensive and costly, with the added consideration of observer bias (Nazir et al., 2018). Additionally, in-situ monitoring techniques such as line transect sampling may result in few observations when species are sparce or rare, restricting the ability to draw robust inferences (Newey et al., 2015; Sollmann, 2018). Autonomous imagery allows for researchers to monitor and observe species in absentia over long deployment periods, regardless of inaccessibility of study location, weather conditions or researcher availability (Nazir et al., 2018). In recent years, autonomous cameras have become an increasingly popular tool in the study of behaviour and occupancy of large mammals due to the autonomous and non-invasive nature of remote data collection, and minimal human disturbance (Caravaggi et al., 2017; Dorning and Harris, 2019; Mccallum, 2013).

Autonomous camera traps can be programmed to take successive photographs at a given time interval, facilitating multiple observations of the same individual and observations of social interactions between animals (Marvin et al., 2016). However, regular image captures can greatly increase the volume of data collected compared to a camera that only triggers with animal presence. Higher rates of image capture require greater computational power, larger data storage capacity and place increased demand on battery provision (Arts et al., 2015). Additionally, the larger volumes of data associated with regular triggering intervals will take longer for processing and analysis, increasing research costs if done manually.

Commercially available passive camera traps can be costly; financial restraints may limit temporal coverage due to technological capabilities such as lower-cost batteries, or spatial coverage due to limitations on the number of camera traps across different sites (Arts et al., 2015). However, the wide range of applications of autonomous camera technology can inform conservation management of declining species and enable long term monitoring, making camera traps a valuable resource(Caravaggi et al., 2017; Edgar et al., 2016). Camera traps can nonetheless provide very useful data for multiple applications including study of long-term population trends, identifying factors that influence species distribution, species life history, habitat use and population demography (Murphy et al., 2018; Walls, 2018)

The rising popularity of camera traps as a leisure product for personal interest has made the technology more accessible and affordable, despite limitations in battery life and quality compared to high-end commercial camera traps (Newey et al., 2015). Furthermore, open-source hardware (OSH) systems such as Raspberry Pi or Arduino have eliminated the need for costly commercial camera traps, allowing custom and bespoke camera traps built specifically to species or research questions at low-cost (Rico-Guevara and Mickley, 2017). Rico-Guevara and Mickley used OSH to build specialized, portable camera traps capable of capturing high-speed feeding behaviours of hummingbirds. Rico-Guevara and Mickley highlight OSH usage to overcome limitations of commercial camera traps.

Additionally, the transition from film to digital cameras has also reduced limitations previously common to camera trap studies. Previously, human land-based observations have been used in conjunction with an infrared monitoring system to examine monk seal cave usage patterns and photo-identification (Gücü et al., 2004). However, in Gücü et al. (2004), all data collected at two sites was unusable due to mechanical damage from storms, meaning cave usage patterns were determined from a single cave site. Additionally, 35-mm film cameras used in Gücü et al. (2004) limited image captures due to length of film.

Recently, digital camera traps were used to calculate monk seals population estimations and examine demographic structure (Kurt and Gücü, 2021). The advancement in camera technology from Gücü et al. (2004) to Kurt and Gücü (2021) demonstrates how advancing technology can enable the collection of more comprehensive and complete scientific evidence, and fill knowledge gaps for the conservation of endangered species.

1.2. Artificial Intelligence and Machine Learning

Artificial intelligence (AI) is the ability of computing systems to perform tasks that previously only humans could undertake, such as image classification and decision-making (Munim et al., 2020). Manual interpretation and image classification of sizable camera traps datasets is financially inefficient and time-consuming (Lamba et al., 2019). AI presents an alternative, time-effective solution compared to manual processing of 'big data' collected by autonomous monitoring systems such as camera traps (Tabak et al., 2019).

AI is becoming more accessible than ever, with implementation possible in multiple programming languages including Python, Matlab and R (Nguyen et al., 2019). Despite major time-cost benefits of AI use in the analysis voluminous datasets, models can be more error-prone and less sensitive than human cognition (Stowell et al., 2019). Validation of AI results using data subsets is common practise to ensure adequate accuracy, consistency, and reproducibility of AI models (Tabak et al., 2019; Tao et al., 2019). However, the quantity of data required for manual training of an AI algorithm can vary with study objectives and model complexity, requiring from hundreds to tens of thousands of labelled examples (Tabak et al., 2019). As a result, dataset size can limit model complexity, making AI approaches most suitable for vast datasets (Tao et al., 2019).

Machine learning is a branch of AI dependent on trained computational models which learn repeated patterns from input data; models are trained using manually defined parameters or labelled examples from a subset of data (Mac Aodha et al., 2018). Supervised machine learning approaches initially require manual analysis by a trained individual to classify data, however trained model algorithms can then be applied to much larger datasets with little human input (Lamba et al., 2019). Machine learning techniques include deep learning and neural networks, which utilize computer algorithms to label features within individual images, sounds or text (Nguyen et al., 2019). Capabilities of neural networks include species identification, facial recognition software and object detection (Agarwal and Dhar, 2014). The application of deep learning and neural networks for automated image classification is rapidly becoming a critical tool for the analysis of large image datasets such as camera trap imagery (Tabak et al., 2019).

1.3. Mediterranean Monk Seals

Mediterranean monk seals (*Monachus monachus*) are one of the most endangered pinnipeds globally, with only ~800 individuals remaining (Karamanlidis et al., 2016). The Mediterranean monk seal (hereafter monk seal) are the only resident pinniped within the Mediterranean Sea (Karamanlidis et al., 2016). Historically, monk seal occurrence was continuous throughout the Mediterranean and Black Seas, with populations extending as far as Northern Spain and Morocco (González, 2015; Karamanlidis et al., 2016). However, modern-day monk seal distribution is highly fragmented (Karamanlidis et al., 2021).

Decline and fragmentation of monk seal populations has been attributed to historical hunting, and more recently deliberate killing, entanglement, and habitat loss (Karamanlidis et al., 2016). Three significant but isolated monk seal populations remain: Cabo Blanco Peninsula, and the Archipelago of Madeira, in the North Atlantic (Martínez-Jauregui et al., 2012; Pires et al., 2008), and the Eastern Mediterranean population (Karamanlidis et al., 2016). Eastern Mediterranean subpopulations occur in Greece, Cyprus and Turkey and comprises of an estimated ~300 mature individuals, occupying 90% of the area known to be habituated by monk seals (Karamanlidis et al., 2021). However, recent localised estimates of monk seal populations are scarce. The most recent estimate of monk seals in the Ionian Islands, Greece, dates to 1997 with an estimated 30-40 individuals (Marchessaux and Duguy, 1977). Mammals with reduced population size as well as reduced and fragmented geographic range are more vulnerable to extinction (Crooks et al., 2017)

Mediterranean monk seals are currently listed as 'Endangered' by the International Union for Conservation of Nature (Karamanlidis and Dendrinos, 2015). The IUCN criteria underlying this status includes small population size, decline in geographic range and population fragmentation. Under the EU Habitats Directive (92/43/EEC), monk seals are a qualifying feature of 82 protected areas in Greek coastal waters. Obligations under Annex II of the EU Habitats Directive (92/43/EEC) require strict protection of monk seals, including designation of marine protected areas and implementation of planning and development policies. Monk seals are also listed as a species at high risk of extinction under Appendix I of the Migratory Species of Wild Animals Convention (CMS, 2015). Bonn Convention obligations for monk seals recovery are implemented through the UNEP Mediterranean Action Plan, which includes protection of key habitats and establishment of population monitoring.

Long-term monitoring and accurate population estimates are crucial for understanding conservation status, impacts of management actions and recovery of a species (Campbell et al., 2002; Kurt and Gücü, 2021). However, factors such as small population sizes and inaccessibility of study locations have resulted in large gaps in the biological understanding of monk seals (Karamanlidis et al., 2016). The retreat of monk seals from open beaches to isolated marginal habitat, for example marine caves, has been recorded in the Eastern Mediterranean (Johnson and Lavigne, 1999). Historical records indicate that monk seals once frequented coastal beaches to haul out and pup (Johnson and Lavigne, 1999).

Human disturbance and habitat loss are associated with the retreat of monk seals to secluded caves (Karamanlidis et al., 2016). In protected areas such as Gyaros, Greece or Cabo Blanco, Mauritania where human activity is low, monk seals haul out on open beaches (Dendrinos et al., 2008; Gilmartin and Forcada, 2002). The entire Cabo Blanco subpopulation is known to use less than five cave systems for hauling out and pupping (Martínez-Jauregui et al., 2012). In contrast, monk seal cave usage has been observed in 34 caves in the Northern Sporades Archipelago, Greece (Dendrinos et al., 2007). Outside areas of low human activity, monk seal pupping and resting occurs almost exclusively in remote and inaccessible marine caves, peaking in September- October (Dendrinos et al., 2008, 2007).

Marine caves have therefore become important monk seal breeding habitat, despite evidence suggesting they are suboptimal (Beton et al., 2021; Gazo et al., 2000). Survival rate of pups decreases in cave-breeding individuals compared to those breeding on open beaches (Gazo et al., 2000). The main cause of pup mortality is physical injuries from impact against cave walls during large swells or storms (Gazo et al., 2000). We can also expect seals to be selective in their use of caves for breeding in order to reduce the risks, and cave morphology is a key predictor of monk seal cave selection and usage (Dendrinos et al., 2007; Karamanlidis et al., 2004b). Dendrinos et al. (2007) showed that daylight availability within caves and the visibility of haul out areas from the caves' exterior were the most important predictors of monk seal cave usage. In decreasing order of importance, other predictors included substrate type (sandy gravel being preferred), the area available for hauling out, the depth of cave entrance and the extent of human activity nearby (Dendrinos et al., 2007). Human activity and vessel presence apparently disrupt breeding, with pupping where human activity is lowest (Karamanlidis et al., 2004a).

Pinniped responses to vessel traffic can include displacement, avoidance, reduced cave occupancy and behavioural disruption (Dendrinos et al., 2007; Mpougas et al., 2019). Consequently, at-sea monitoring from survey vessels can produce biased observations, as well as being expensive. Monk seal susceptibility to human disturbance and vessel presence, combined with the rarity of the species, only increases the difficulty of studying them from sea-going vessels. Additionally, remote and unobservable cave locations limits the effectiveness of land-based or aerial surveys. Recently, motion-sensor activated cameras were used at known monk seal haul outs in the North-eastern Mediterranean (Kurt and Gücü, 2021). Kurt and Gücü (2021) observed monk seal behaviour and calculated population estimations using photo-identification mark-recapture methods from camera-trap data. These studies highlight the potential of new technologies to implement non-invasive data collection and monitoring.

1.4. Aims and Objectives

Given the need for non-invasive ecological monitoring of endangered species, this thesis aims to demonstrate how low cost open-source hardware and artificial intelligence can be used as a bespoke solution to answer research questions. Monk seals are known to inhabit and reproduce within the central Ionian region, however fine-scale temporal trends of monk seal cave occupancy are unknown in part due to inaccessibility of marine caves. Therefore, the following tasks were determined as the objectives of this study to be fulfilled:

i. Demonstrate how low-cost, 'build-your-own' instrumentation and artificial intelligence technologies can play a role in the monitoring of endangered pinnipeds.
ii. Examine trends of Mediterranean monk seal cave presence at diel and seasonal scales in the central Ionian Isles, Greece.
iii. Determine if proximal vessel presence is a predictor of monk seal presence within central Ionian marine caves.

2. Methodology

2.1. Data Collection

Remotely captured images were collected by the Octopus Foundation (https://octopusfoundation.org) at 3 locations within the central Ionian islands, Greece between 23rd May and 30th November 2019. Infrared autonomous cameras using solar power and mobile data networks for data transmission (Section 2.3) were used to log monk seal presence inside three marine caves at 15-minute intervals (Figure 1).

Figure 1 - Example of seal presence in marine caves using camera trap imagery. © Octopus Foundation 2019

Additional autonomous cameras were placed at the exterior of each marine cave, capturing vessel presence within the external environs of the marine caves. External camera installations captured vessel presence images at 15-minute intervals concurrently with cameras with each cave. Caves were accessed through use of SCUBA equipment for camera installation using during May, avoiding peak monk seal breeding months of September and October (Dendrinos et al., 2008). Non-operational days were typically as a result of technical malfunction or extreme weather events; for example, storms in the central Ionian Islands in October 2019 resulted in Site 1 (outside) being non-operational between 01/10/2019 – 01/11/2019. Images captured as a result of camera malfunction, identified by non-quarterly hour timestamps (e.g. 12:23 or 18:47) were removed from subsequent analyse so the sampling interval between images would remain the same across functional data collection periods.

2.2. Study Area

Three cave sites within the central Ionian Isles, Greece were selected by the Octopus Foundation for remote image collection. Cave sites were selected as study locations from anecdotal local knowledge and confirmed reported sightings of monk seals. Exact locations of the study sites are undisclosed due to the conservation implications and sensitivities associated with revealing pupping and haul out locations of a vulnerable and endangered species. Site 1 is a North-facing cave exposed to prevailing winds, located at the base of a remote cliff with a sub-marine entrance used for monk seal egress to the cave. Site 1 has a second, larger cave entrance above the water line which is inaccessible to humans without vessel use. Site 2 is distant from human habitation or mooring sites, with a submerged entrance and is typically sheltered from prevailing winds, except south-westerly winds. Site 3 is near a popular anchorage and mooring site. Haul out areas within the cave systems at Site 1 and 3 are large, gravelly beaches, whereas Site 2 consists of rocky outcrops above the water line.

2.3. Autonomous Camera Installations

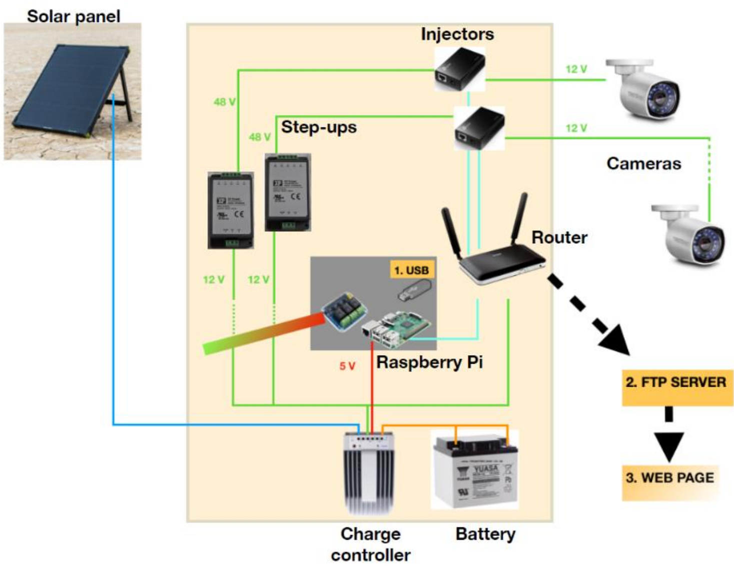


Figure 2 – Components of autonomous camera systems for both internal and external cameras used at each cave site. © Octopus Foundation 2019

Figure 2 gives an overview of the open-source hardware and components used for camera installations within this study. Camera installations at each site were powered by an 100 Watt 12V solar panel in conjunction with a 50Ah AGM battery, Epever MT50 remote meter and Epever Tracer B series 20a charge controller (Figure 2).

A Raspberry Pi 3 model B+ microcontroller was used to control memory and input/outputs of the camera installations. The central control system of the installation consisted of a Raspberry Pi 3 model B+ (https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief), Waveshare Relay board, an 8GB SD card preinstalled with NOOBS (New Out Of Box Software; https://github.com/raspberrypi/noobs) and a 32GB USB flash drive (Grey Box; Figure 2).

PoE (Power over Ethernet) cameras can exchange both power and data via Ethernet cables, over a distance of up to 100m the central control system. POE Trendnet TV-IP-316PI cameras were used at internal and external locations at each of the 3 cave sites. Data from the POE cameras were relayed to the central control system to be stored locally (USB flash drive). Additionally, a D-Link DWR-921 3G/4G router and GSM Antenna was used to transmit data via mobile network to an FTP (File Transfer Protocol) Server to allow remote access and real-time monitoring of data. Supplementary information about the open-source monitoring installations can be found at: http://octopusfoundation.org/en/project/ mediterranean-monk-seal-greece-iucn/#technologie.

2.4. Data Classification

2.4.1. Seal Classification by Human Eye

Seal presence or absence within each image was manually classified by the Octopus Foundation with the human eye as data were collected (Figure 1). Seal presence was determined by the same trained individual for all data to limit observer bias. Human visual ability is more successful at classifying low-quality images compared to machine learning (Geirhos et al., 2017). Low quality images can occur in low-light conditions, which were common to this dataset due to a lack of natural light within cave systems. Following manual classification of internal cave images by the Octopus Foundation, two folders of images were provided; one folder contained all internal cave images whilst the other contained images where monk seals were present. Additionally, photo-identification was carried out by the Octopus Foundation to identify individual monk seals from images using distinguishing features such as pelage patterns and scarring (Samaranch and González, 2000). From individuals identified, descriptive statistics were calculated to determine typical length of seal cave occupancy. Photo-identification was not possible at Site 1 due to camera distance from the haul out beach preventing detailed images of seal pelage.

2.4.2. Artificial Intelligence and Vessel Classification

A machine-learning app 'Mirador 1.0' for MacOS, developed by Faktoria (www.faktoria.ch), was used to apply deep learning and neural network techniques to label vessel presence and vessel type within external cave images (Figure 3). Supervised machine learning was used to train model 'seadetect_outdoor_1908_4' by Faktoria, using labelled examples from a subset of autonomous vessel images across all 3 sites. Mirador outputs included images with boxed vessel detections, labelled vessel type and detection confidence values (Figure 3) as well as a summary table of image vessel detections. Following visual inspection of trialled vessel detection, false-positive rates were measured and a detection threshold of 50% confidence was applied to the AI model to minimise false-positive detections.

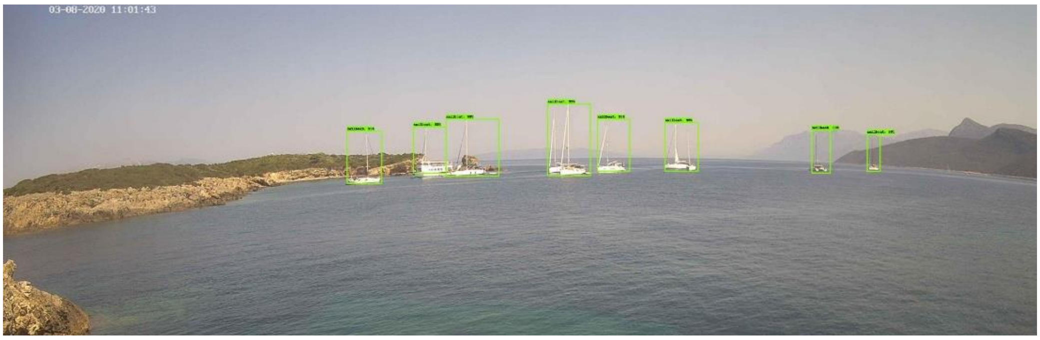


Figure 3 – Examples of boxed and labelled vessel detections using neural network image processing in Mirador 1.0 at Site 3.

Overnight external images were excluded from subsequent analyses as Mirador 1.0 is limited in capacity to determine vessels in night-time images due to a lack of natural lighting. Night-time images were removed prior to analysis, using custom script and packages 'opencv' (Bradski, 2000), 'numpy' (Harris et al., 2020), and 'skimage' (Walt et al., 2014) in Python 3.8.8 (Python Software Foundation, 2021). The average Red-Green-Blue (RGB) value of each row of image pixels and the RGB average of all image pixel rows was calculated as an indicator of lack of natural light (Appendix A). Visual inspection showed that night-time images did not always have an RGB of [0, 0, 0], indicating the absence of total darkness. A lack of total darkness within night-time images was often due to natural factors such a moonlight or insect presence reflecting infrared light. Trial and error were used with a random subset of images to determine a suitable RGB threshold of [86, 86, 86] to exclude night-time images; an RGB value of [86, 86, 86] is representative of a dark grey colour (Appendix A). Excluding night-time images through use of an RGB image value, as opposed to time of day, accounts for seasonal variation in sunset times throughout the dataset.

2.3. Collation of Data

A Custom R script was used in RStudio (Version 1.2.5001; RStudio Team, 2020) and the stringr library (Wickham, 2019), to extract date, time, cave site, camera location and seal presence from image filenames (Appendix A). Image filenames took the format of 'OF_s1i1_19_05_28_17_15.jpg' where 's1' represents Site 1; 'i1' signifies that the camera was internal; '19_05_28' represents date of image capture and '17_15' represents time of image capture. Using image filenames, internal images within the monk seal 'presence' folder, as classified by the Octopus Foundation, were assigned a presence value of 1. Images outwith the presence folder were assigned presence values of 0.

Internal and external images captured at the same time and site have identical filenames with the exception of 'i1' within the filename representing an internal cave image, whereas 'e1' is representative of an external image. To relate Mirador vessel detections to seal presence data by corresponding filenames, additional R script was implemented using the following libraries: plyr (Wickham, 2011); dplyr (Wickham et al., 2021); readr (Wickham et al., 2018); and tidyverse (Wickham et al., 2019).

Total vessel count was calculated by totalling motorboat, sailing boat and kayak detections for each image. To account for variation in daily sampling effort due to technical issues and varying length of daylight, photographic detections were converted into an index of vessel activity (Sollmann, 2018). The total number of vessel detections within a 24hr period was divided by the number of daylight images taken in the same period to produce a daily vessel index. A random subset of external images (10%) from all 3 sites was used to manually validate Mirador detection of vessel presence in MATLAB R2019a (version 9.6.0; MATLAB, 2019). Manual classification of vessel presence was then compared with Mirador classification by image filename in RStudio (Version 1.2.5001; RStudio Team, 2020), using custom script to obtain false negative and false positive rates (Appendix A).

2.4. Statistical Analysis

Temporal occurrence of monk seals and vessel detections was modelled over both seasonal and diel time scales. Diel patterns of monk seal presence were likely to be indicative of behavioural activity throughout the day, while the distribution of monk seal presence over a seasonal scale was more likely to identify long-term occupancy trends and identify key habitat usage (Ikeda et al., 2016). In subsequent modelling, only data collected whilst cameras at all 3 sites were operational (1st June 2019 – 30th September 2019) was used. This excludes the initial camera installation period and technological damage due to storm events in October 2019. Generalised Additive Models (GAM; Hastie and Tibshirani, 1990) were used to allow for natural fluctuation and complex relationships likely within time series data (Li et al., 2018; Suzuki and Ando, 2019a, 2019b). GAMs were implemented using the 'mgcv' library (version 1.8-36; Wood, 2011) within RStudio (Version 1.2.5001; RStudio Team, 2020).

2.4.1. Diel Analysis

To model data over a diel scale, each image taken at 15-minute intervals was assigned a value of '1' with monk seal presence or '0' with absence. Bernoulli successes and failures were then calculated for each 15-minute interval. Subsequent trials at each time occurred on separate dates, 24 hours apart, therefore samples were assumed to be independent. For example, if a trial occurred at 09:00 on 1st October, the subsequent trial where time of day was 09:00 would occur on 2nd October, 24hrs after the previous trial. Each 15-minute time interval was assigned a consecutive numerical value representative of the equivalent time of image capture (e.g. 00:00 = 1 and 23:45 = 96).

A binomial family GAM with a complimentary loglog ('cloglog') function (Fisher, 1944) was used to model probability of monk seal occurrence within camera trap images over time. The model equation used within the gam function from the mgcv library in R was:

$$cbind(success, failure) \sim s(time\_number, bs \; = \text{“}cc\text{"})$$

The cloglog function was used to allow for asymmetry in successes and failures (Bowler et al., 2019), as monk seal absences were more common to the dataset. Monk seal absences accounted for 79% of internal cave photographs. Monk seal occurrence was modelled with numerical time of day as a cyclic penalised regression spline to remove discontinuity between 23:00 and 00:00. Within the dataset, autocorrelation was suspected as a common feature of biological and longitudinal time series data (Brown et al., 2011; Jebb and Tay, 2016; Tomaŝových and Kidwell, 2011). The autocorrelation function (ACF) from the 'mgcv' library was then used to examine suspected autocorrelation common to longitudinal time series data (Appendix B). Correlograms indicated autocorrelation was insignificant at a threshold of 1 hour and 15 minutes (Appendix B). Consequently, monk seal occurrence data was pooled into hourly intervals.

Hourly vessel data was constrained by Mirador's requirement of natural light to identify vessels within images, therefore only images between 6:00 and 22:00 were used in subsequent analysis. For each hourly period, a vessel index was calculated as the total hourly Mirador vessel detections, divided by the number of external images captured within that hour. Vessel index was used for diel models as photographic effort may have been lower in extremes of time due to fluctuation in day length and low light levels.

To investigate effects of vessel presence on monk seal cave occupancy over time, a GAM assuming binomial error distribution and using the cloglog function was fitted monk seal presence data from camera trap images, with vessel index over time and time of day modelled as thin-plate regression splines. Thin-plate regression splines were used instead of cyclic penalised regression spline as previous due to the removal of night time data, thus removing continuity of time between 24 hour periods.

The linear predictor of the model used to examine influence of vessel presence and hour of day over a diel scale took the following form:

$$\mu = \beta^0 + f(vessel\_index^{site\ 1}) + f(vessel\_index^{site\ 2}) + f(vessel\_index^{site\ 3}) + f(hour\_of\_day)$$

Where f represents a spline function as applied within the GAM framework. The complimentary log-log ('cloglog') link function was used to convert values on the linear predictor scale to probabilities of observing seals,

$$y_i = CLogLog(\mu_i) = \ln(-\ln(1-\mu_i))$$

where the inverse link function was:

$$\mu_i = CLogLog^{-1}(y_i) = 1 - e^{-ey_i}$$

The success/failure trials of monk seal presence and absence within images (*y)* follows Bernoulli distribution:

$$y_i \sim Binomial\ (Pr\ (y_i)\ )$$

If both vessel and time smooth terms were modelled with site as a factor variable, it resulted in fewer unique covariate combinations than the specified maximum degrees of freedom. Consequently, only the vessel smooth term was modelled by site, as the predictor that explained the most variation in data and gave the lowest UBRE score. Thin-plate regression splines were used unlike in exploratory GAMs of seal presence/absence, as only data within the range of 6:00 to 22:00 rather than continuous time. Correlograms produced by the ACF from the 'mgcv' library showed that when data was modelled at hourly intervals, autocorrelation was insignificant at a threshold of 20hrs, which was less than the 24hrs period between time intervals. For example, the presence of seals at 09:00 on day d was likely to be independent from presence at 09:00 of day $d - 1$. Thus counting presence or absence at the same time on consecutive days constituted independent trials as per the assumptions of binomial family GAMs. Model predictions were implemented using the 'predict' function from the mgcv library.

2.4.2. Seasonal Analysis

For seasonal trends, presence and absence monk seals data was pooled by date as Bernoulli success/failure trials to examine autocorrelation. A sampling interval of 24hrs was chosen due to the lack of overnight vessel data within each 24hr period. Dates were consecutively numbered by day.

A GAM with the cloglog function and Bernoulli family was used to model probability of monk seal occurrence within camera trap images by date (Appendix B). Camera site was used as a factor variable, with numbered day as a smooth term modelled as a thin-plate regression spline. The model equation implanted using the gam function from the mgcv package in R was:

$$cbind(success, failure) \sim s(day\_of\_year,\ \ by = as.factor(Site)\ )$$

The autocorrelation function (ACF) from the 'mgcv' library was then used to produce correlograms of seal presence, revealing that autocorrelation was insignificant at a threshold of 2.67 days (Appendix B). Consequently, monk seal data was pooled over 72hr periods for data points to be considered independent samples. Each 72hr period was then assigned a consecutive numerical value according to relative date of that 72hr period.

Vessel index was calculated, as previous, for each 72hr period. However, visual inspection of vessel index outliers revealed over-inflated vessel index during 72hr periods with low image coverage. Consequently, data points where cameras were operational for less than 50% of each 72hr period were removed from subsequent analyses, removing only 5 data points from the original 117 data points.

A GAM with the cloglog function and binomial family was used to model probability of monk seal occurrence within camera trap images by 72hr period and vessel index (Appendix B). Camera site was used as a factor variable for both smooth terms, with numbered 72hr period and vessel index modelled as a thin-plate regression splines. The linear predictor of the final model used to examine influence of vessel presence over a seasonal scale across sites took the following form:

$$\mu_i = \beta^0 + f(vessel\_index^{site\,1}) + f(vessel\_index^{site\,2}) + f(vessel\_index^{site\,3}) + f(72hr\_period^{site\,1}) + f(72hr\_period^{site\,2}) + f(72hr\_period^{site3})$$

The complimentary log-log ('cloglog') function was used, as per the previous diel model:

$$\Pr(y_i) = CLogLog(\mu_i) = \ln(-\ln(1-\mu_i))$$

The success/failure trials of monk seal presence within images (*y)* follows a Binomial distribution:

$$y_i \sim Binomial\,(Pr\,(y_i))$$

Model predictions for each site were generated using the 'predict' function from the mgcv library and the original dataframe subsetted by site. A new dataframe was not constructed to make predictions as vessel index was a function of seasonality and site.

3. Results

3.1. Image Collection

A total of 93,128 images were autonomously collected between 23/05/2019 and 30/11/2019. Of the 93,128 images collected, 44,913 images captured the internal cave environment at Site 1 (14,860 images), Site 2 (14,027 images) and Site 3 (16,026 images). 48,215 images captured the external environment at each cave site. Images of the external environment totalled 17,841, 16,267, and 14,107 images at Sites 1 to 3 respectively. During 2019, autonomous cameras were operational for an average period of 152 days (range 147-192 days; Table 1).

Table 1 – Summaries of the camera trap operational period at each cave site.

	Start Date	End Date	Total Operational Days	Possible Operational Days
Site 1 (inside)	28/05/2019	30/11/2019	187	187
Site 1 (outside)	28/05/2019	30/11/2019	156	187
Site 2 (inside)	31/05/2019	03/11/2019	147	157
Site 2 (outside)	29/05/2019	30/11/2019	148	186
Site 3 (inside)	23/05/2019	30/11/2019	152	192
Site 3 (outside)	23/05/2019	13/11/2019	153	175
		Average	157	181

3.1.2. Mediterranean Monk Seals

Seal presence within caves was identified in 9,429 images by a human observer across all 3 sites. Site 2 had the highest occurrence of seals, with seal presence in 7,276 images (51.9%) of 14, 027 images. At Site 1 and Site 3, monk seals were present in 774 of 14,860 images (5.21%) and 591 of 16,026 images (3.69%) respectively. Across sites 1-3, rates of seal presence per 100 images was 5, 52 and 4, in that order.

3.1.3. Seal Photo-Identification

Manual photo-identification of monk seals was completed by the Octopus Foundation and manually verified by Student 20017683 for the purposes of this study. In total, 10 individuals were identified across 241 days, predominately at Site 2. Estimations of seal age were possible by comparison to rocky outcrops of a known length used for hauling out. Sexually dimorphic pelage patterns allowed for identification of a mature adult male, an adult female suspected to be heavily pregnant, and numerous juvenile seals. The length of cave occupancy by individual seals was calculated from photo-identification. Length of cave occupancy by an individual seal ranged from 1 day to a maximum of 10 days with a mean duration of 2.74 days (Appendix A). The longest, 10-day, period of occupancy was by an adult female likely to be pregnant from size observations. Cave usage by identified seals was highest in September and October, typically by juvenile and sub-adult seals (86% of occupancy bouts).

3.2. Vessel Detections

Of the 48,215 images taken of external environs of cave sites, 27,349 photographs exceeded an RGB threshold of [86, 86, 86] indicating adequate natural light for vessel detection (Appendix A). Images with sufficient daylight totalled 10,010, 9,261 and 8,065 across sites 1-3 respectively. Vessel detection in these images took a total of 238 processing hours using 'Mirador 1.0', to label 17,731 vessels across all 3 study locations. Vessels were most commonly detected at Site 3 with 86 vessels per 100 images. Rates of vessel detections at sites 1 and 2 per 100 images were 27 and 4 respectively. An overall total of 12,196 vessel detections comprised of 9952 sailing boats, 7735 motorboats and 44 kayaks, however many of these detections would have been repeat detections if vessels was moored for an extended time period.

A subset of 2,735 external images (10% of those available) were randomly selected from all sites to validate neural network detections. Across sites 1 to 3, vessel presence was manually classified using 972 (9.71%), 956 (10.32%) and 808 (10.02%) images respectively. Site 1 had the highest rate of false-positive vessel detections (Table 2). Upon visual inspection of the labelled images, the source of false-positive detections was determined to be a large rocky outcrop contrasting against the sea (Appendix A). Model validation showed agreement rates of 77%, 92% and 89% between Mirador and manual classifications of vessel presence across sites 1-3 respectively (Table 2). Overall, a comparison between manual and model classifications gave an 86% agreement rate of true classifications. The average percentage of total false classifications across all 3 sites was 14%, adding uncertainty within the data.

Table 2 – Agreement rates between manual classification of vessel presence/absence and classifications using neural network image processing.

Mirador 1.0	Site 1	Site 2	Site 3	All Sites
True Positive	0.18	0.04	0.53	0.23
False Positive	0.19	0.05	0.08	0.10
True Negative	0.59	0.88	0.36	0.62
False Negative	0.05	0.03	0.04	0.04
Total True Classifications	0.77	0.92	0.89	0.86
Total False Classifications	0.23	0.08	0.11	0.14

3.4. General Trends

On an hourly scale, the highest levels of vessel activity and lowest occurrence of seal presence were observed at Site 3. At Site 3, the maximum recorded hourly seal presence was 28 images of an average possible 624 images (4.5%). Vessel indexes of 0.08 – 2.51 were observed at Site 3, with a mean value of 1.13 (Figure 4a). Seal presence at Site 1 was comparable to Site 3, with total images of seal presence not exceeding 37 images out of an average of 624 images. However, mean seal presence was slightly higher at Site 1 (20 images) than Site 3 (13 images). Site 1 vessel index was typically between 0.06 and 0.09 (Figure 4a). Site 2 had the lowest vessel index with a mean value of 0.07 and typical range of 0.06 and 0.09 (Figure 4a). At Site 2, hourly vessel index did not exceed 0.16, equating to 16 vessel detections per 100 images (Figure 4a). Site 2 also had highest occurrence of monk seals (Figure 4a), with seals typically present in 310-351 (50%) images (Figure 4a). Seal presence at Site 2 was 25x more common than Site 1, and 16x more common than Site 3.

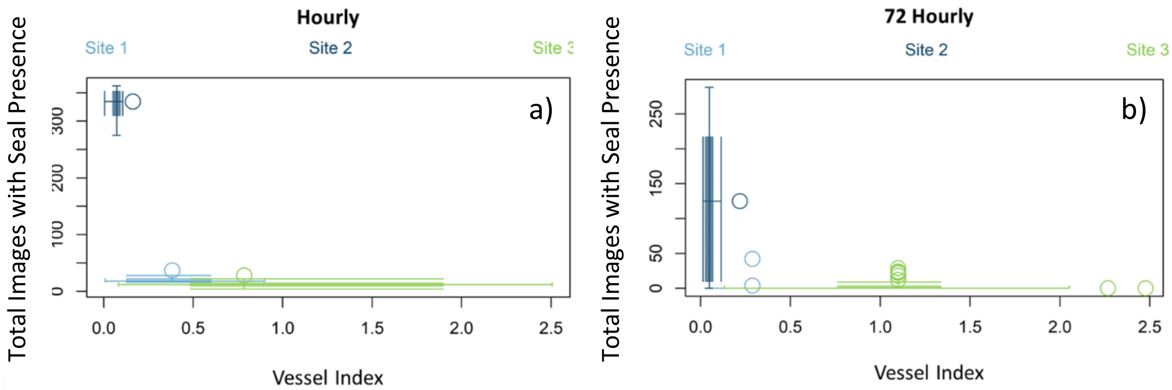


Figure 4 – Two-dimensional boxplots depicting hourly and 72-hourly seal presence and vessel index at each site, where Os are outliers, whiskers represent the min and max values, shaded boxes indicate the interquartile range, and whisker intercepts represent median values.

On a 72-hourly scale, Site 2 also had the highest occurrence of monk seals with a mean seal presence of 122 images of a possible 288 (42%). Mean seal presence occurrence at Site 1 was 1 image and 5 images at Site 3 (Figure 4b). Site 2 had the lowest vessel index, ranging from 0.014 – 0.22 (Figure 4b). Comparatively, Site 3 had a mean vessel index of 1.12, with vessels detected in every image on average. The maximum observed vessel index at Site 3 was 2.48, averaging 2.5 vessels in all images within that 72hr period (Figure 4b). Site 3 had the lowest seal presence, with a maximum 72hr seal presence of 29 out of 288 images (Figure 4b). At Site 1, vessel indexes of 0.076 – 0.57 were recorded, with a mean vessel index of 0.27 (Figure 4b). Average seal presence at Site 1 was 1 of 288 images per 72 hours at Site 1. Maximum seal presence at Site 1 was 42 images in 72 hours (Figure 4b).

3.4. Modelling Results

3.4.1. Diel Cave Occupancy Patterns

Autocorrelation was no longer significant above a threshold of 1 hour and 15 minutes, therefore diel data was pooled into hourly intervals (Appendix B). A multivariate GAM over hourly intervals gave a deviance explained (DE) of 99.7% and an adjusted (Adj.) R^2 of 0.999 (Table 3). Vessel index (calculated as the number of vessel detections over the total number of images taken within hourly intervals) and hour of day were both significant predictors of monk seal presence (Table 3). Vessel index was a significant predictor of monk seal presence at all 3 sites ($p<0.001$; Table 3), with likelihood of monk seal presence declining with increasing vessel index (Figure 6). At Site 1, the relationship between vessel index and likelihood of monk seal presence was linear (Estimated degrees of freedom (edf) = 1.00; Table 3). Monk seal presence at Site 1 was unlikely after vessel index exceeded ~0.5, equating to 1 detected vessel in half of all images within a given hour (Figure 5a).

*Table 3 - Model summary statistics for the diel monk seal presence GAM, with vessel index as a function of vessel detections and hour of day as model predictors. Significance levels: 0.001 '***', 0.01 '**', 0.05 '*', 0.1 '·'.*

Parametric Coefficients				
	Estimate	Standard Error	z value	Pr(>|z|)
Intercept	-3.55545	0.05318	-64.68	<2e-16 ***
Approximate significance of smooth terms				
	Estimated df	Reference df	Chi Squared	p value
s(vessel_index):as.factor(site)1	1.000	1.000	24.97	5.83e-07 ***
s(vessel_index):as.factor(site)2	2.581	2.798	3716.26	<2e-16 ***
s(vessel_index):as.factor(site)3	2.450	3.030	51.18	5.17e-11 ***
s(hour_of_day)	2.654	3.332	14.24	0.00544 **
% Deviance Explained = 99.7%	*Adjusted R^2 = 0.999*			

Non-linear relationships between vessel index and seal presence were observed at Sites 2 and 3, with edf values of 2.58 and 2.45, respectively (Table 3). Similar to Site 1, at Site 3 monk seal presence was unlikely after vessel index exceeded ~0.5 (Figure 5c). Monk seal presence was highly likely at Site 2 if low vessel index values were observed (Figure 5b). The largest decline in likelihood of monk seal presence was observed at Site 2, however large confidence intervals at the upper end of the observed vessel index range were likely due to vessel detections being less common at Site 2. The non-linear relationship between monk seal presence and hour of day was also significant across data from all sites ($p<0.05$; Table 3). Likelihood of seal presence was slightly higher during daylight hours (9:00-18:00; Figure 5d).

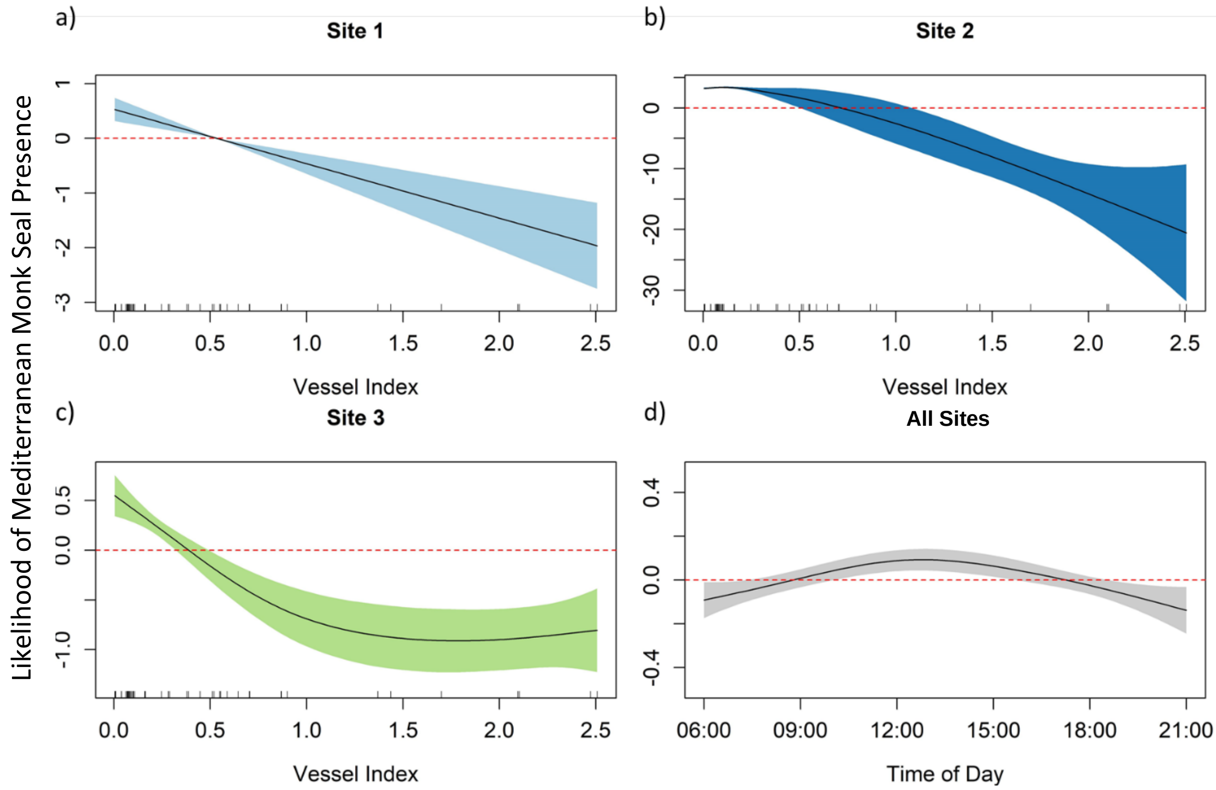


Figure 5 – Diel GAM predictions of likelihood of monk seal presence with hourly vessel index as a function of vessel detections and time of day.

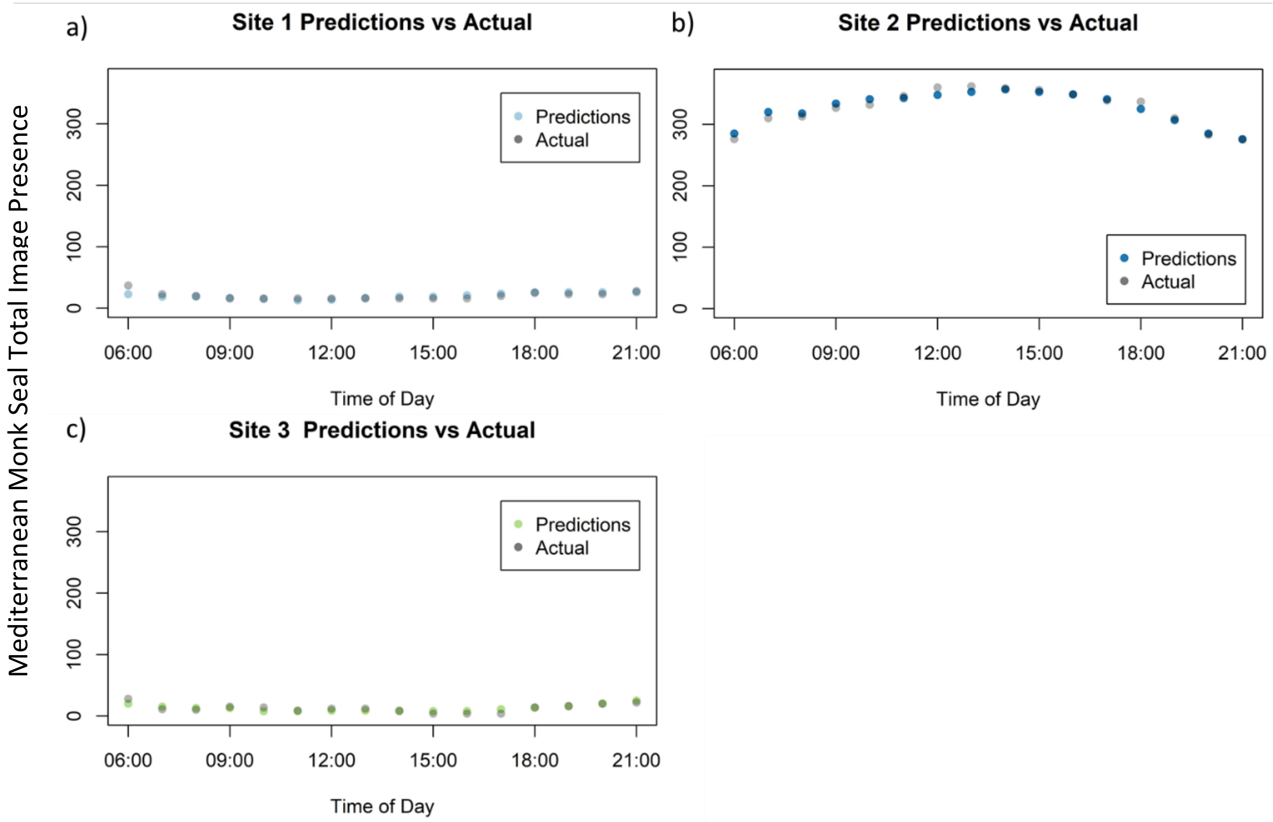


Figure 6 – Diel model predictions versus actual occurrence of seals as number of photos per hourly period for the whole dataset.

Model predictions showed that monk seal presence was typically homogenous throughout daylight hours at both Sites 1 and 3 (Figure 6a, 6c). Predictions of monk seal presence at Sites 1 and 3 were comparatively lower than predicted and observed presence at Site 2 (Figure 6). Predicted monk seal presence slightly increased throughout midday at Site 2 (Figure 6b), so patterns observed in Figure 5d can largely be attributed to data from this site. The limited number of monk seal sightings at both Sites 1 and 3 may explain the high DE shown in Table 3, as low numbers of monk seals made it easy to make model predictions.

3.4.2. Seasonal Cave Occupancy Patterns

At a seasonal scale, autocorrelation was no longer significant above a threshold of 2.67 days (Appendix B), therefore data was pooled over 72 hourly (3 day) intervals. An autocorrelation threshold of 72hrs is also reflected in the 2.74 day mean length of cave occupancy by monk seals (Appendix A) A multivariate seasonal GAM gave a DE of 82.8% and an Adj. R^2 of 0.835 (Table 4). Date as 72hr periods and vessel index were both significant predictors of monk seal presence (Table 4). Date was a significant predictor at site 1 ($p<0.001$) and site 2 ($p<0.001$), the highest probability of monk seal presence occurred from August to October (Figure 7a, 7b). Probability of monk seal presence typically increased with date at sites 1 and 2.

*Table 4 - GAM model summary statistics for the seasonal monk seal presence, with 72hr period as a numerical value and vessel index as a function of vessel detections as model predictors. Significance levels: 0.001 '***', 0.01 '**', 0.05 '*', 0.1 '·'.*

Parametric Coefficients				
	Estimate	Standard Error	z value	Pr(>|z|)
Intercept	-3.9726	0.1162	-34.17	<2e-16 ***
Approximate significance of smooth terms				
	Estimated df	Reference df	Chi Squared	p value
s(data_72hr):as.factor(site)1	3.990	4.000	280.04	<2e-16 ***
s(data_72hr):as.factor(site)2	3.992	4.000	2647.28	<2e-16 ***
s(data_72hr):as.factor(site)3	3.905	3.993	50.53	2.91e-10 ***
s(vessel_index)	8.762	8.982	1420.94	<2e-16 ***
% Deviance Explained = 82.8%	*Adjusted R^2 = 0.835*			

Date was also a significant predictor of monk seal presence at site 3 (p<0.001; Table 4). However, at Site 3 likelihood of monk seal presence increased throughout June before plateauing from July to September and then falling until October where data collection ended. Monk seal presence was most likely at Site 3 between mid-June and the end of August (Figure 7c). The relationship between monk seal presence and vessel index was significant over 72-hour periods, with values of >0.5 being associated with reduced seal presence, as in the diel analysis (p<0.001; Table 4). Likelihood of monk seal presence steeply declined with increasing vessel index (Figure 7d). If a 72-hourly vessel index of ~0.25 was observed, equating to a single vessel present in 1 of every 4 photographs, then it was much less likely that monk seal presence would occur (Figure 7d).

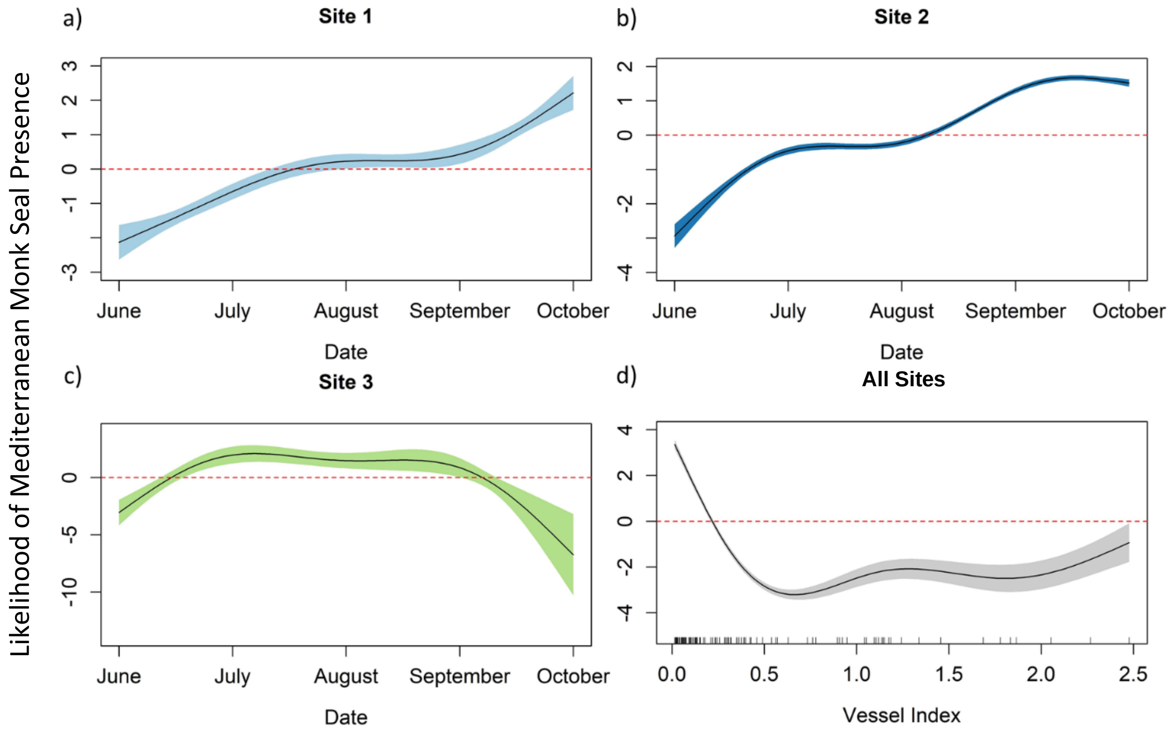


Figure 7 - Seasonal GAM predictions of likelihood of monk seal presence with date and 72-hourly vessel index.

Model predictions at Site 1 were consistently low with occasional fluctuations (Figure 8a). Whilst the model predicted low seal presence well, it was less successful at predicting intermittent fluctuations. Predictions of monk seal presence were low at both Sites 1 and 3 (Figure 8a, 8c), as expected from data exploration. Comparison of actual data and model predictions at Site 2 showed considerable natural fluctuation in seal presence (Figure 8b). However, a general trend of increasing monk seal presence, peaking in September, was observed in both actual and predicted data (Figure 8b).

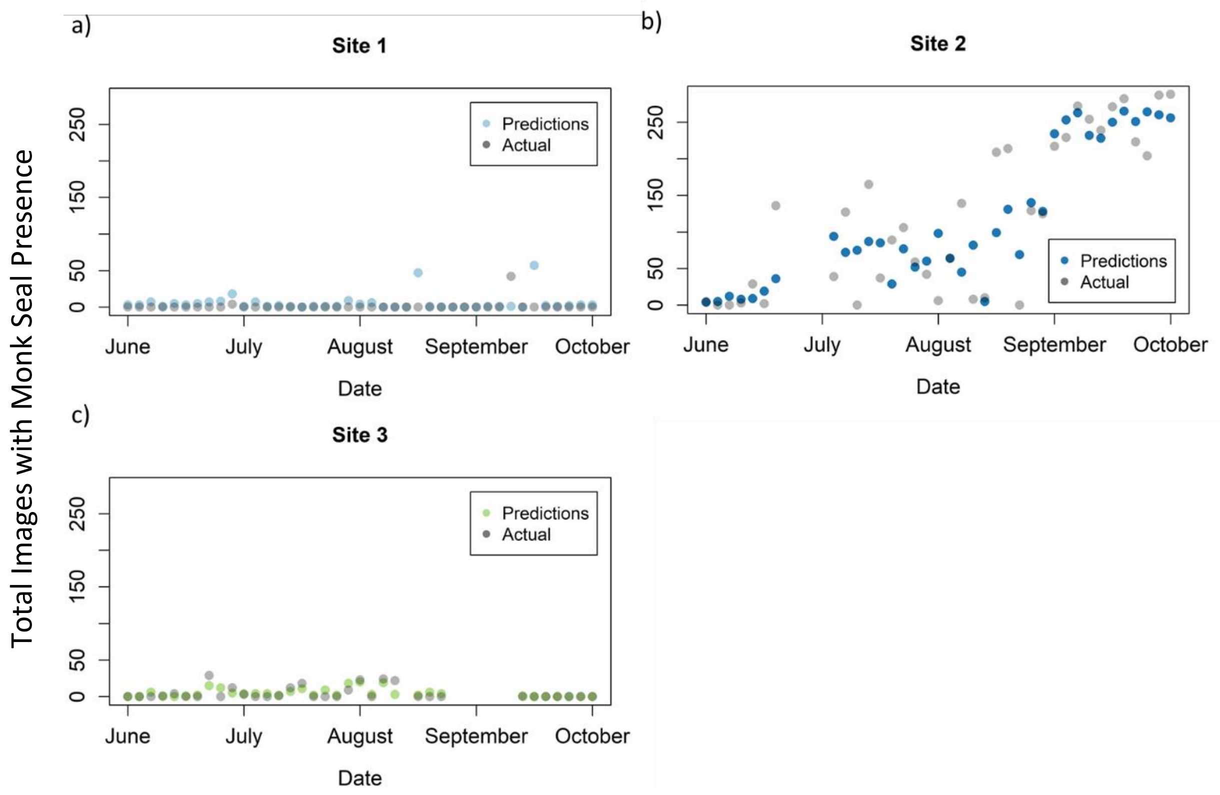


Figure 8 – Seasonal model predictions versus actual occurrence of monk seals in photographs per 72-hour period.

4. Discussion

The first aim of this thesis was achieved by demonstrating how low-cost instrumentation components can be used in place of commercially available camera traps to monitor monk seals within remote cave systems. The first aim was also met by using AI and deep neural networks to detect vessels within autonomously captured images. AI proved a crucial tool in the analysis of large volumes of data associated with camera traps. The second aim of examining trends of monk seal presence within caves was achieved over both diel and seasonal scales by producing GAMs over both hourly and 72-hourly periods. The third aim of this thesis was also achieved through the implementation of GAMs, with vessel index a significant predictor of monk seal presence over both temporal scales.

Camera traps made from open-source hardware proved to be a practical and valuable tool for the study of monk seals in remote marine caves. The customisable nature of open-source hardware allowed the camera traps to be designed to endure the challenging conditions found within marine caves such as high humidity, limited natural light and corrosivity of saline water. Additionally, camera traps could be designed specifically to each site. For instance, in standard camera traps, wire length may not have been sufficient to run from the solar powered unit above the cave to the camera located within the cave chamber. The low cost of components compared to high-end commercially available traps also allowed for wider spatial coverage, with more cameras over more sites for the same cost.

It was integral to this study that methods used were minimally invasive to avoid disruption to an endangered species. Two potential causes of disruption from camera traps were identified; noise disturbance due to shutter sound when an image is taken and visual disturbance due to infrared light produced by the cameras. Monk seal habituation to shutter noise was likely due to the regularity of image capture at 15-minute intervals. Shutter noise from image capture typically ranges from 40 to 70 dB re 1 µPa. Wild phocid seals have shown avoidance at received noise levels 135-146 dB re. 1 µPa (Götz and Janik, 2010), almost double the maximum source level produced by typical camera shutters (70dB). Habituation to sounds even at 146 dB re. 1 µPa has been shown in captive phocids (Götz and Janik, 2010). Due to distance between camera location and seal haul outs, the received noise level would also be lower than the 40-70 dB re. 1 µPa range produced by the shutter, therefore it is unlikely to cause disturbance to monk seals.

Furthermore, visual disturbance to monk seals from infrared light emitted by cameras during image capture was unlikely. Similar sized phocid seals such as the harbour seal (*Phoca vitulina*) have a visual acuity of 2.6 arcmins in water compared to a resolution of 5.6 arcmins in air (Mass and Supin, 2018). Autonomous cameras were installed within air-filled chambers of marine caves. A consequence of differences in visual acuity between air and water means that infrared lighting from autonomous cameras were less likely to cause visual disturbance when installed in air compared to underwater.

Cone receptors within pinniped retinae are sparse, indicating reduced colour sensitivity in pinniped vision (Crognale et al., 1998; Peichl et al., 2001). A cone sensitivity in harbour seals of 501nm indicates sensitivity towards blue spectral wavelengths, as common in marine mammals due to dominance of blue wavelengths in the marine environment. Rod receptors determine vision at low light levels. Harbour seals have the highest known rod sensitivity of phocid seals at 495nm (Mass and Supin, 2018). Infrared light emitting diodes of cameras used in this study produced wavelengths of 850nm, avoiding the highest visual sensitivity and causing minimal disturbance to phocids. Disruption was further minimised by installing cameras during May, avoiding the height of monk seal occurrence.

Previous studies that captured monk seal presence with autonomous imagery had technological constraints such as film roll length (Gücü et al., 2004). Technology no longer inhibits the collection of voluminous camera trap datasets (Arts et al., 2015; Berger-Tal and Lahoz-Monfort, 2018). The large volume of data used in this study meant artificial intelligence was essential to quantifying vessel presence in external images. This study was the first to use Mirador neural network image processing for vessel detection at such a scale, and across multiple sites. Without neural network image processing it would not have been possible to classify the 27,349 external daylight images within time constraints.

There were multiple sources of uncertainty in classified vessel detections by AI. Firstly, the ability to detect vessels in low light levels and overnight exceeded Mirador capabilities. Additionally, it was not possible to detect vessels overnight from navigational lights as many vessels captured in images did not display navigational lights overnight when anchored. In future studies it may be beneficial to cross-reference vessel detections with Automatic Identification System (AIS) data, however AIS is not a requirement for cruising yachts or small commercial vessels.

Secondly, a comparison of manual versus Mirador vessel classification of presence/absence revealed that the false-positive rate at Site 1 was 4x higher than at Site 2 and more than double the rate at Site 3. Site 1 had a false-positive detection rate of 19%. Consequently, the rate of vessel presence at Site 1 could be lower than expected. For example, if 15 vessels were detection within 100 images at Site 1, the real number of vessels present at Site 1 would be 12 vessels as ~20% of detections would be false-positive. Upon visual inspection of images at Site 1, the source of false-positive vessel detections was determined to be a lightly coloured rocky outcrop. Adjustments to the structure of the Mirador neural network model could reduce the number of false-positive detections at Site 1. The TensorFlow package for Python 3.8.8 underpins the detection of vessels within the Mirador outputs include an image with labelled common objects such as vessels, a bounding box surrounding the detected object and the image pixel coordinates of the bounding box. Vessel detection within pixel coordinates of the rocky outcrop could be restricted, however this would also prevent the detection of any vessels in front of the rocky outcrop.

Thirdly, Mirador validation was carried out for detection of vessel presence/absence only due to time constraints. In diel and seasonal models predicting likelihood of monk seal presence, vessel index as a function of vessel count was used as a predictor. For future validation, comparison between manual vessel count and Mirador vessel detections should be used to calculate agreement rates. Agreement rates were not calculated for seal detections as classification was completed manually, therefore high confidence in classification was assumed. In continuation of this study, validation of seal detections would be completed to account for potential observer bias.

Additional uncertainty resulted from variation in survey effort between sites due to camera malfunction arising from technological issues or inclement weather. Consequently, vessel detections from Mirador were converted to a vessel index, as the number of vessel detections per total number of images taken within a given time-period, to account for effort. Furthermore, it should be considered in the interpretation of results that multiple vessel detections may have occurred if vessels were moored within the camera view for extended time periods.

When modelling data over a diel scale, only observations from daylight hours were used as vessel detections were not available overnight. As expected, serial temporal autocorrelation was observed in diel success/failure trials of seal presence were sampled at 15-minute intervals and modelled within a GAM framework. Consequently, data was pooled into hourly intervals to be considered independent samples. At all 3 sites, vessel index was a significant predictor of monk seal presence over an hourly scale. Monk seal presence had a strong negative correlation with increasing vessel index, as previously shown in monk seal pupping habitat use (Dendrinos et al., 2007). Observations of active avoidance of speedboats by monk seals have previously been documented, however accounts were anecdotal rather than quantified (Akkaya et al., 2016).

There was a slight increase in likelihood of monk seal presence during midday hours across all sites, however observations were only across 3 sites. Sheltering in caves at midday when outdoor temperatures can exceed 30^{o}C may minimise risk of overheating (Watts, 1992). In Turkish coastal waters, Akkaya et al. (2016) mostly observed monk seal sightings between 06:00 and 10:00, typically consisting of foraging and travelling behaviours. Only 8% of sightings by Akkaya et al. (2016) displayed signs of resting behaviour, giving additional confidence to resting and haul out behaviours within marine caves typically occurring after 10:00. Deviance explained for the diel model was high (99.7%), however there were over 600 success/failure trials of seal presence for each independent hourly interval. Additional examination of code and data reinforced that the high deviance explained value for the diel model was correct.

Over a seasonal scale, as expected, autocorrelation was significant within success/failure trials of seal presence sampled at daily intervals when modelled in a GAM framework. Therefore data was pooled into 72-hour (3 day) periods to be consider independent from the previous sample. The average length of cave occupancy by identified individual seals was 2.74 days, which supported the pooling of data into 3-day periods to be considered independent samples. It should be considered that the autocorrelation threshold may exceed 3 days depending on cave use, as previous studies have shown continuous cave usage for 3-4 months by mother-pup pairs (Karamanlidis et al., 2021).

Over a seasonal scale, GAMs predicted that seal presence were most likely to occur at Site 2, peaking during the month of September. Monk seal surveys in Turkish coastal waters reported the highest rate of monk seal encounters during September (Akkaya et al., 2016). Previous studies have also shown that monk seal pupping and cave usage peaks between September - October (Dendrinos et al., 2008, 2007). Seasonally, the rate of monk seal presence was highest at Site 2, with the lowest rate of vessel occurrence. Likelihood of monk seal presence declined with increasing vessel presence across all sites. Previously, vessel mooring, travelling and anchorage as an index of human activity has proved a significant predictor of pupping habitat selection for monk seals (Dendrinos et al., 2007). However, in Dendrinos et al (2007), cave substrate was ranked as a more important predictor of cave selection than human activity. Caves used for pupping were dominated by sand/gravel substrates as opposed to rock platforms (Dendrinos et al., 2007).

In this study, Sites 1 and 3 comprised of gravelly beaches but had the lowest occurrence of monk seals. Site 2 had the highest occurrence of seals, despite the haul out area consisting of rocky platforms. However, Sites 1 and 3 had high rates of vessel presence compared to Site 2, suggesting that vessel activity may be a more important predictor of monk seal cave presence than sediment type. This reflects the documented retreat of monk seals into suboptimal breeding habitat as a result of human activity (Johnson and Lavigne, 1999; Karamanlidis et al., 2016). Rock platforms found at Site 2 are suboptimal for pupping and hauling out due to increased injury from cave walls and increased pup mortality (Gazo et al., 2000). However, vessel activity is likely to be limiting use of caves with more suitable sediment such as Sites 1 and 3. To increase the validity of this study, it would be beneficial to increase the spatial coverage of camera traps to additional sites with a range of substrate types.

At present, seal detections were classified manually but the development of an AI model for automated seal detection is underway. Neural network image processing to classify seal presence in internal cave images will be essential as the number of study sites, and consequently volumes of data, increases. Custom R script used to extract key information from image filenames was developed with increasing study sites in mind (Appendix A), therefore would not require alteration to accommodate new camera traps.

Similarly to the diel model, the deviance explained value for the seasonal model was high. Seal presence was typically higher at site 2 compared to sites 1 and 3, which were typically being low with only occasional fluctuations. Consequently, the model predictions of low or high seal presence were predicted easily, hence a high deviance explained value, unlike predictions of fluctuating presence which were less successful. Within each independent 72hr period sampling interval in the seasonal model, successive trials within the 72hr period were not independent from one another. Improvements should be made to the modelling framework in this study to account for autocorrelation in successive trials within each independent sampling interval.

To account for longitudinal autocorrelation and clustering, a Generalized Estimating Equation (GEE) approach could be taken (Pirotta et al., 2014; Scales et al., 2014; Wang, 2014). Using GAMS with GEEs would be a more robust way to model serial autocorrelation within each 72-hour period at a seasonal scale or hourly interval on a diel scale. GEE Generalised Linear Models (GLM) can be fitted using the 'geeglm' function from the 'geepack' package in R, before constructing splines within the GEE-GLM using the splines library to produce GEE-GAMs (Pirotta et al., 2011).

Despite the limitations and sources of uncertainty as outlined above, this study was able to investigate effects of vessel presence and other factors on monk seal cave occupancy. Whilst it is likely that vessel index is inflated due to false-positive detections and modelling approaches, it is likely to be representative that vessel presence influences monk seal cave occupancy at both seasonal and diel scales. Over a seasonal scale, monk seal presence was lower with high rates of vessel presence, potentially influencing longer-term cave selection for hauling out by monk seals in addition to pupping (Dendrinos et al., 2007). At a diel scale, it is likely that vessel presence influences short-term behaviour of monk seals, as well as temporally effecting cave ingress and egress (Ikeda et al., 2016; Karamanlidis et al., 2004). Bonn Convention obligations for the recovery of monk seals require protection of key habitats, such as the protection of high-usage caves from vessel disturbance. Photographic documentation of monk seals can also contribute to Bonn Convention population monitoring requirements. Documented cave usage by monk seals through camera trap images can support the proposal of marine protected areas for monk seals, as obliged under Annex II of the EU Habitats Directive, however continued data collection is required.

Monk seal presence peaked in the month of September, therefore it would be beneficial to introduce seasonal restrictions on anchorage and mooring in proximity to study sites, such as limiting the number mooring vessels during autumnal months. Further study is needed to determine accurate thresholds of vessel presence that limit monk seal cave ingress and ingress, as well as long-term cave selection. To determine thresholds of vessel influence over monk seal cave use, methodology and instrumentation used in this study can be expanded to additional cave sites to increase spatial data coverage.

5. Conclusion

The three objectives of this thesis were achieved. Firstly, this thesis demonstrated that instrumentation built from open-source hardware and artificial intelligence technologies can play a role in the monitoring of endangered pinnipeds. Without the use of low cost open-source hardware for autonomous data collection, the collection of a dataset as large as the one in this study, with minimal disturbance to an endangered species, would not have been possible. Additionally, artificial intelligence to automate vessel detections proved essential to classify vessel presence within time constraints. Secondly, diel and seasonal trends of Mediterranean monk seal cave presence in the central Ionian Isles, Greece were investigated through Generalised Additive Modelling. Thirdly, vessel presence was determined to be a significant predictor of monk seal cave presence at both diel and seasonal scales, despite sources of uncertainty within this study. Recommendations for the conservation management of Mediterranean monk seals in the Central Ionian Islands were suggested, as well as recommendations and improvements for further study.

6. References

Agarwal, R., Dhar, V., 2014. Editorial—Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research. https://doi.org/10.1287/isre.2014.0546 25, 443–448. https://doi.org/10.1287/ISRE.2014.0546

Akkaya, A., Piludu, N., Lagoa, J., Atchoi, E., 2016. Recent Sightings of the Mediterranean Monk Seal Monachus monachus and Evaluation of Anthropogenic Activities with Recommended Conservation Implications in Antalya Bay, Turkey.

Arts, K., Wal, R. van der, Adams, W.M., 2015. Digital technology and the conservation of nature. Ambio 2015 44:4 44, 661–673. https://doi.org/10.1007/S13280-015-0705-1

Berger-Tal, O., Lahoz-Monfort, J.J., 2018. Conservation technology: The next generation. Conservation Letters 11, e12458. https://doi.org/10.1111/conl.12458

Beton, D., Broderick, A.C., Godley, B.J., Kolaç, E., Ok, M., Snape, R.T.E., 2021. New monitoring confirms regular breeding of the Mediterranean monk seal in Northern Cyprus. Oryx 55, 522–525. https://doi.org/10.1017/S0030605320000848

Booth, C.G., Sinclair, R.R., Harwood, J., 2020. Methods for Monitoring for the Population Consequences of Disturbance in Marine Mammals: A Review. Frontiers in Marine Science 7, 115. https://doi.org/10.3389/fmars.2020.00115

Bowler, D.E., Nilsen, E.B., Bischof, R., O'Hara, R.B., Yu, T.T., Oo, T., Aung, M., Linnell, J.D.C., 2019. Integrating data from different survey types for population monitoring of an endangered species: the case of the Eld's deer. Scientific Reports 2019 9:1 9, 1–14. https://doi.org/10.1038/s41598-019-44075-9

Bradski, G., 2000. The OpenCV Library.

Brown, C.J., Schoeman, D.S., Sydeman, W.J., Brander, K., Buckley, L.B., Burrows, M., Duarte, C.M., Moore, P.J., Pandolfi, J.M., Poloczanska, E., Venables, W., Richardson, A.J., 2011. Quantitative approaches in climate change ecology. Global Change Biology 17, 3697–3713. https://doi.org/10.1111/J.1365-2486.2011.02531.X

Campbell, S.P., Clark, J.A., Crampton, L.H., Guerry, A.D., Hatch, L.T., Hosseini, P.R., Lawler, J.J., O'connor, R.J., 2002. An Assessment of Monitoring Effort in Endangered Species Recovery Plans. Ecological Applications 12, 674–681. https://doi.org/10.1890/1051-0761(2002)012

Caravaggi, A., Banks, P.B., Burton, A.C., Finlay, C.M. v., Haswell, P.M., Hayward, M.W., Rowcliffe, M.J., Wood, M.D., 2017. A review of camera trapping for conservation behaviour research. Remote Sensing in Ecology and Conservation 3, 109–122. https://doi.org/10.1002/rse2.48

Chan, K., Schillereff, D.N., Baas, A.C.W., Chadwick, M.A., Main, B., Mulligan, M., O'Shea, F.T., Pearce, R., Smith, T.E.L., van Soesbergen, A., Tebbs, E., Thompson, J., 2021. Low-cost electronic sensors for environmental research: Pitfalls and opportunities. Progress in Physical Geography 45, 305–338. https://doi.org/10.1177/0309133320956567

Christiansen, F., Dawson, S.M., Durban, J.W., Fearnbach, H., Miller, C.A., Bejder, L., Uhart, M., Sironi, M., Corkeron, P., Rayment, W., Leunissen, E., Haria, E., Ward, R., Warick, H.A., Kerr, I., Lynn, M.S., Pettis, H.M., Moore, M.J., 2020. Population comparison of right whale body condition reveals poor state of the North Atlantic right whale. Marine Ecology Progress Series 640, 1–16. https://doi.org/10.3354/meps13299

CMS, 2015. Appendices I and II of the Convention on the Conservation of Migratory Species of Wild Animals .

Crognale, M.A., Levenson, D.H., Ponganis, P.J., Deegan, J.F., Jacobs, G.H., 1998. Cone spectral sensitivity in the harbor seal (Phoca vitulina) and implications for color vision. Canadian Journal of Zoology 76, 2114–2118. https://doi.org/10.1139/Z98-129

Crooks, K.R., Burdett, C.L., Theobald, D.M., King, S.R.B., Marco, M. di, Rondinini, C., Boitani, L., 2017. Quantification of habitat fragmentation reveals extinction risk in terrestrial mammals. Proceedings of the National Academy of Sciences 114, 7635–7640. https://doi.org/10.1073/PNAS.1705769114

Dendrinos, P., Karamanlidis, A.A., Kotomatas, S., Legakis, A., Tounta, E., Matthiopoulos, J., 2007. Pupping habitat use in the Mediterranean monk seal: A long-term study. Marine Mammal Science 23, 615–628. https://doi.org/10.1111/j.1748-7692.2007.00121.x

Dendrinos, P., Karamanlidis, A.A., Kotomatas, S., Paravas, V., Adamantopoulou, S., 2008. Report of a new Mediterranean monk seal (Monachus monachus) breeding colony in the Aegean Sea, Greece. Aquatic Mammals 34, 355–361. https://doi.org/10.1578/AM.34.3.2008.355

Dorning, J., Harris, S., 2019. The challenges of recognising individuals with few distinguishing features: Identifying red foxes Vulpes vulpes from camera-trap photos. PLoS ONE 14, e0216531. https://doi.org/10.1371/journal.pone.0216531

Durban, J., Fearnbach, H., Barrett-Lennard, L., Perryman, W., Leroi, D., Barrett-Lennard Vancouver, L., LeRoi, D., 2015. Photogrammetry of killer whales using a small hexacopter launched at sea 1. J. Unmanned Veh. Syst 3, dx. https://doi.org/10.1139/juvs-2015-0020

Edgar, G.J., Bates, A.E., Bird, T.J., Jones, A.H., Kininmonth, S., Stuart-Smith, R.D., Webb, T.J., 2016. New Approaches to Marine Conservation Through the Scaling Up of Ecological Data. Annual Review of Marine Science 8, 435–461. https://doi.org/10.1146/annurev-marine-122414-033921

Fisher, R.A., 1944. On the Mathematical Foundations of Theoretical Statistics. Philosophical Transactions of the Royal Society of London 222:, 309–368.

Fretwell, P.T., Staniland, I.J., Forcada, J., 2014. Whales from space: Counting southern right whales by satellite. PLoS ONE 9, 88655. https://doi.org/10.1371/journal.pone.0088655

Gazo, M., Aparicio, F., Cedenilla, M.A., Layna, J.F., González, L.M., 2000. Pup Survival In The Mediterranean Monk Seal (Monachus monachus) Colony At Cabo Blanco Peninsula (Western Sahara-Mauritania). Marine Mammal Science 16, 158–168. https://doi.org/10.1111/J.1748-7692.2000.TB00910.X

Geirhos, R., Janssen, D.H.J., Schütt, H.H., Rauber, J., Bethge, M., Wichmann, F.A., 2017. Comparing deep neural networks against humans: Object recognition when the signal gets weaker. arXiv.

Gilmartin, W.G., Forcada, J., 2002. Monk seal Monachus Monachus, M. tropicalis, and M. schauinslandi, in: Perrin, W.F., Würsig, B., Thewissen, J.G.M. (Eds.), Encyclopedia of Marine Mammals. Academic Press, London, UK, pp. 756–759.

Gonzalez, A., Cardinale, B.J., Allington, G.R.H., Byrnes, J., Endsley, K.A., Brown, D.G., Hooper, D.U., Isbell, F., O'Connor, M.I., Loreau, M., 2016. Estimating local biodiversity change: a critique of papers claiming no net loss of local diversity. Ecology 97, 1949–1960. https://doi.org/10.1890/15-1759.1

González, L.M., 2015. Prehistoric and historic distributions of the critically endangered Mediterranean monk seal (Monachus monachus) in the eastern Atlantic. Marine Mammal Science 31, 1168–1192. https://doi.org/10.1111/MMS.12228

Götz, T., Janik, V.M., 2010. Aversiveness of sounds in phocid seals: Psycho-physiological factors, learning processes and motivation. Journal of Experimental Biology 213, 1536–1548. https://doi.org/10.1242/jeb.035535

Gücü, A.C., Gücü, G., Orek, H., 2004. Habitat use and preliminary demographic evaluation of the critically endangered Mediterranean monk seal (Monachus monachus) in the Cilician Basin (Eastern Mediterranean). Biological Conservation 116, 417–431. https://doi.org/10.1016/S0006-3207(03)00234-9

Guerra, C.A., Pendleton, L., Drakou, E.G., Proença, V., Appeltans, W., Domingos, T., Geller, G., Giamberini, S., Gill, M., Hummel, H., Imperio, S., McGeoch, M., Provenzale, A., Serral, I., Stritih, A., Turak, E., Vihervaara, P., Ziemba, A., Pereira, H.M., 2019. Finding the essential: Improving conservation monitoring across scales. Global Ecology and Conservation 18, e00601. https://doi.org/10.1016/J.GECCO.2019.E00601

Harcourt, R., Sequeira, A.M.M., Zhang, X., Roquet, F., Komatsu, K., Heupel, M., McMahon, C., Whoriskey, F., Meekan, M., Carroll, G., Brodie, S., Simpfendorfer, C., Hindell, M., Jonsen, I., Costa, D.P., Block, B., Muelbert, M., Woodward, B., Weise, M., Aarestrup, K., Biuw, M., Boehme, L., Bograd, S.J., Cazau, D., Charrassin, J.B., Cooke, S.J., Cowley, P., de Bruyn, P.J.N., Jeanniard du Dot, T., Duarte, C., Eguíluz, V.M., Ferreira, L.C., Fernández-Gracia, J., Goetz, K., Goto, Y., Guinet, C., Hammill, M., Hays, G.C., Hazen, E.L., Hückstädt, L.A., Huveneers, C., Iverson, S., Jaaman, S.A., Kittiwattanawong, K., Kovacs, K.M., Lydersen, C., Moltmann, T., Naruoka, M., Phillips, L., Picard, B., Queiroz, N., Reverdin, G., Sato, K., Sims, D.W., Thorstad, E.B., Thums, M., Treasure, A.M., Trites, A.W., Williams, G.D., Yonehara, Y., Fedak, M.A., 2019. Animal-borne telemetry: An integral component of the ocean observing toolkit. Frontiers in Marine Science. https://doi.org/10.3389/fmars.2019.00326

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 2020 585:7825 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hastie, T., Tibshirani, R., 1990. Generalized additive models.

Ikeda, T., Uchida, K., Matsuura, Y., Takahashi, H., Yoshida, T., Kaji, K., Koizumi, I., 2016. Seasonal and Diel Activity Patterns of Eight Sympatric Mammals in Northern Japan Revealed by an Intensive Camera-Trap Survey. PLOS ONE 11, e0163602. https://doi.org/10.1371/JOURNAL.PONE.0163602

Jebb, A.T., Tay, L., 2016. Introduction to Time Series Analysis for Organizational Research: Methods for Longitudinal Analyses. Organizational Research Methods 20, 61–94. https://doi.org/10.1177/1094428116668035

Johnson, W.M., Lavigne, D., 1999. MASS TOURISM and the MEDITERRANEAN MONK SEAL The role of mass tourism in the decline and possible future extinction of Europe's most endangered marine mammal, Monachus monachus.

Junker, J., Petrovan, S.O., Arroyo-RodrÍguez, V., Boonratana, R., Byler, D., Chapman, C.A., Chetry, D., Cheyne, S.M., Cornejo, F.M., CortÉs-Ortiz, L., Cowlishaw, G., Christie, A.P., Crockford, C., Torre, S.D.L., de Melo, F.R., Fan, P., Grueter, C.C., GuzmÁn-Caro, D.C., Heymann, E.W., Herbinger, I., Hoang, M.D., Horwich, R.H., Humle, T., Ikemeh, R.A., Imong, I.S., Jerusalinsky, L., Johnson, S.E., Kappeler, P.M., Kierulff, M.C.M., KonÉ, I., Kormos, R., Le, K.Q., Li, B., Marshall, A.J., Meijaard, E., Mittermeier, R.A., Muroyama, Y., Neugebauer, E., Orth, L., Palacios, E., Papworth, S.K., Plumptre, A.J., Rawson, B.M., Refisch, J., Ratsimbazafy, J., Roos, C., Setchell, J.M., Smith, R.K., Sop, T., Schwitzer, C., Slater, K., Strum, S.C., Sutherland, W.J., Talebi, M., Wallis, J., Wich, S., Williamson, E.A., Wittig, R.M., KÜhl, H.S., 2020. A Severe Lack of Evidence Limits Effective Conservation of the World's Primates. BioScience 70, 794–803. https://doi.org/10.1093/BIOSCI/BIAA082

Karamanlidis, A.A., Dendrinos, P., 2015. Monachus monachus (Mediterranean Monk Seal) [WWW Document]. The IUCN Red List of Threatened Species. URL https://dx.doi.org/10.2305/IUCN.UK.2015-4.RLTS.T13653A45227543.en. (accessed 7.27.21).

Karamanlidis, A.A., Dendrinos, P., Larrinoa, P.F. de, Gücü, A.C., Johnson, W.M., Kiraç, C.O., Pires, R., 2016. The Mediterranean monk seal Monachus monachus: status, biology, threats, and conservation priorities. Mammal Review 46, 92–105. https://doi.org/10.1111/MAM.12053

Karamanlidis, A.A., Dendrinos, P., Tounta, E., Kotomatas, S., 2004a. Monitoring human activity in an area dedicated to the protection of the endangered Mediterranean monk seal. Coastal Management 32, 293–306. https://doi.org/10.1080/08920750490448523

Karamanlidis, A.A., Pires, R., Silva, N.C., Neves, H.C., 2004b. The availability of resting and pupping habitat for the Critically Endangered Mediterranean monk seal Monachus monachus in the archipelago of Madeira. Oryx 38, 180–185. https://doi.org/10.1017/S0030605304000328

Karamanlidis, A.A., Dendrinos, P., Trillmich, F., 2021. Maternal behavior and early behavioral ontogeny of the Mediterranean monk seal Monachus monachus in Greece. Endangered Species Research 45, 13–20. https://doi.org/10.3354/esr01114

Karamanlidis, A.A., Skrbinšek, T., Amato, G., Dendrinos, P., Gaughran, S., Kasapidis, P., Kopatz, A., Stronen, A.V., 2021. Genetic and demographic history define a conservation strategy for earth's most endangered pinniped, the Mediterranean monk seal Monachus monachus. Scientific Reports 2021 11:1 11, 1–10. https://doi.org/10.1038/s41598-020-79712-1

Kurt, M., Gücü, A., 2021. Demography and population structure of Northeastern mediterranean monk seal population. Mediterranean Marine Science 22, 79–87. https://doi.org/https://doi.org/10.12681/mms.21913

Lamba, A., Cassey, P., Segaran, R.R., Koh, L.P., 2019. Deep learning for environmental conservation. Current Biology. https://doi.org/10.1016/j.cub.2019.08.016

Li, X., Bleisch, W. v., Jiang, X., 2018. Using large spatial scale camera trap data and hierarchical occupancy models to evaluate species richness and occupancy of rare and elusive wildlife communities in southwest China. Diversity and Distributions 24, 1560–1572. https://doi.org/10.1111/DDI.12792 mac Aodha, O., Gibb, R., Barlow, K.E., Browning, E., Firman, M., Freeman, R., Harder, B., Kinsey, L., Mead, G.R., Newson, S.E., Pandourski, I., Parsons, S., Russ, J., Szodoray-Paradi, A., Szodoray-Paradi, F., Tilova, E., Girolami, M., Brostow, G., Jones, K.E., 2018. Bat detective—Deep learning tools for bat acoustic signal detection. PLoS Computational Biology 14. https://doi.org/10.1371/journal.pcbi.1005995

Marchessaux, D., Duguy, R., 1977. Le phoque moine, Monachus monachus (Hermann, 1779), en Grèce. Mammalia 41, 419–440. https://doi.org/10.1515/mamm.1977.41.4.419

Martínez-Jauregui, M., Tavecchia, G., Cedenilla, M.A., Coulson, T., Fernández De Larrinoa, P., Muñoz, M., González, L.M., 2012. Population resilience of the Mediterranean monk seal Monachus monachus at Cabo Blanco peninsula. Marine Ecology Progress Series 461, 273–281. https://doi.org/10.3354/MEPS09811

Marvin, D.C., Koh, L.P., Lynam, A.J., Wich, S., Davies, A.B., Krishnamurthy, R., Stokes, E., Starkey, R., Asner, G.P., 2016. Integrating technologies for scalable ecology and conservation. Global Ecology and Conservation. https://doi.org/10.1016/j.gecco.2016.07.002

Mass, A.M., Supin, A.Y., 2018. Vision, in: Encyclopedia of Marine Mammals. Elsevier, pp. 1035–1044. https://doi.org/10.1016/B978-0-12-804327-1.00266-1

MATLAB, 2019. MATLAB; version 9.6.0.

Mccallum, J., 2013. Changing use of camera traps in mammalian field research: Habitats, taxa and study types. Mammal Review. https://doi.org/10.1111/j.1365-2907.2012.00216.x

Meek, P.D., Ballard, G.A., Fleming, P.J.S., Schaefer, M., Williams, W., Falzon, G., 2014. Camera traps can be heard and seen by animals. PLoS ONE 9, 110832. https://doi.org/10.1371/journal.pone.0110832

Mpougas, E., Waggitt, J.J., Dendrinos, P., Adamantopoulou, S., Karamanlidis, A.A., 2019. Mediterranean monk seal (Monachus monachus) behavior at sea and interactions with boat traffic: Implications for the conservation of the species in Greece. Aquatic Mammals. https://doi.org/10.1578/AM.45.4.2019.419

Munim, Z.H., Dushenko, M., Jimenez, V.J., Shakil, M.H., Imset, M., 2020. Big data and artificial intelligence in the maritime industry: a bibliometric review and future research directions. https://doi.org/10.1080/03088839.2020.1788731 577–597. https://doi.org/10.1080/03088839.2020.1788731

Murphy, A.J., Farris, Z.J., Karpanty, S., Kelly, M.J., Miles, K.A., Ratelolahy, F., Rahariniaina, R.P., Golden, C.D., 2018. Using camera traps to examine distribution and occupancy trends of ground-dwelling rainforest birds in north-eastern Madagascar. Bird Conservation International 28, 567–580. https://doi.org/10.1017/S0959270917000107

Nazir, S., Hamdoun, H., Verdicchio, F., Fairhurst, G., 2018. Autonomous monitoring framework for resource-constrained environments. Cyber-Physical Systems 4, 137–155. https://doi.org/10.1080/23335777.2018.1499673

Newey, S., Davidson, P., Nazir, S., Fairhurst, G., Verdicchio, F., Irvine, R.J., van der Wal, R., 2015. Limitations of recreational camera traps for wildlife management and conservation research: A practitioner's perspective. Ambio; Royal Swedish Academy of Sciences 44, 624–635. https://doi.org/10.1007/s13280-015-0713-1

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á.L., Heredia, I., Malík, P., Hluchý, L., 2019. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review 2019 52:1 52, 77–124. https://doi.org/10.1007/S10462-018-09679-Z

Peichl, L., Behrmann, G., Kröger, R.H.H., 2001. For whales and seals the ocean is not blue: A visual pigment loss in marine mammals. European Journal of Neuroscience 13, 1520–1528. https://doi.org/10.1046/j.0953-816x.2001.01533.x

Pires, R., Neves, H.C., Karamanlidis, A.A., 2008. The Critically Endangered Mediterranean monk seal Monachus monachus in the archipelago of Madeira: priorities for conservation. Oryx 42, 278–285. https://doi.org/10.1017/S0030605308006704

Pirotta, E., Matthiopoulos, J., MacKenzie, M., Scott-Hayward, L., Rendell, L., 2011. Modelling sperm whale habitat preference: a novel approach combining transect and follow data. Marine Ecology Progress Series 436, 257–272. https://doi.org/10.3354/MEPS09236

Pirotta, E., Thompson, P.M., Miller, P.I., Brookes, K.L., Cheney, B., Barton, T.R., Graham, I.M., Lusseau, D., 2014. Scale-dependent foraging ecology of a marine top predator modelled using passive acoustic data. Functional Ecology 28, 206–217. https://doi.org/10.1111/1365-2435.12146

Python Software Foundation., 2021. Python Language Reference.

Rico-Guevara, A., Mickley, J., 2017. Bring your own camera to the trap: An inexpensive, versatile, and portable triggering system tested on wild hummingbirds. Ecology and Evolution 7, 4592–4598. https://doi.org/10.1002/ece3.3040

Ripple, W.J., Estes, J.A., Beschta, R.L., Wilmers, C.C., Ritchie, E.G., Hebblewhite, M., Berger, J., Elmhagen, B., Letnic, M., Nelson, M.P., Schmitz, O.J., Smith, D.W., Wallach, A.D., Wirsing, A.J., 2014. Status and Ecological Effects of the World’s Largest Carnivores. Science 343. https://doi.org/10.1126/SCIENCE.1241484

RStudio Team, P., 2020. RStudio: Integrated Development for R. RStudio.

Samaranch, R., González, L.M., 2000. Changes in morphology with age in Mediterranean monk seals (Monachus monachus). Marine Mammal Science 16, 141–157. https://doi.org/10.1111/J.1748-7692.2000.TB00909.X

Scales, K.L., Miller, P.I., Embling, C.B., Ingram, S.N., Pirotta, E., Votier, S.C., 2014. Mesoscale fronts as foraging habitats: composite front mapping reveals oceanographic drivers of habitat use for a pelagic seabird. Journal of The Royal Society Interface 11. https://doi.org/10.1098/RSIF.2014.0679

Sollmann, R., 2018. A gentle introduction to camera-trap data analysis. African Journal of Ecology 56, 740–749. https://doi.org/10.1111/aje.12557

Stowell, D., Wood, M.D., Pamuła, H., Stylianou, Y., Glotin, H., 2019. Automatic acoustic detection of birds through deep learning: The first Bird Audio Detection challenge. Methods in Ecology and Evolution 10, 368–380. https://doi.org/10.1111/2041-210X.13103

Suzuki, K.K., Ando, M., 2019a. Early and efficient detection of an endangered flying squirrel by arboreal camera trapping. Mammalia 83, 372–378. https://doi.org/10.1515/MAMMALIA-2018-0055

Suzuki, K.K., Ando, M., 2019b. Tree trunk camera trapping for a small dormouse. Mammal Research 2019 64:4 64, 479–484. https://doi.org/10.1007/S13364-019-00436-3

Tabak, Michael A., Norouzzadeh, M.S., Wolfson, D.W., Sweeney, S.J., Vercauteren, K.C., Snow, N.P., Halseth, J.M., Salvo, P.A. di, Lewis, J.S., White, M.D., Teton, B., Beasley, J.C., Schlichting, P.E., Boughton, R.K., Wight, B., Newkirk, E.S., Ivan, J.S., Odell, E.A., Brook, R.K., Lukacs, P.M., Moeller, A.K., Mandeville, E.G., Clune, J., Miller, R.S., 2019. Machine learning to classify animal species in camera trap images: Applications in ecology. Methods in Ecology and Evolution 10, 585–590. https://doi.org/10.1111/2041-210X.13120

Tao, C., Gao, J., Wang, T., 2019. Testing and Quality Validation for AI Software-Perspectives, Issues, and Practices. IEEE Access 7, 120164–120175. https://doi.org/10.1109/ACCESS.2019.2937107

Tomaŝových, A., Kidwell, S.M., 2011. Accounting for the effects of biological variability and temporal autocorrelation in assessing the preservation of species abundance. Paleobiology 37, 332–354. https://doi.org/10.1666/09506.1

Walls, S.C., 2018. Coping with constraints: Achieving effective conservation with limited resources. Frontiers in Ecology and Evolution 6, 24. https://doi.org/10.3389/fevo.2018.00024

Walt, S. van der, Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. scikit-image: image processing in Python. PeerJ 2, e453. https://doi.org/10.7717/PEERJ.453

Wang, M., 2014. Generalized Estimating Equations in Longitudinal Data Analysis: A Review and Recent Developments. Advances in Statistics 2014, 1–11. https://doi.org/10.1155/2014/303728

Watts, P., 1992. Thermal constraints on hauling out by harbour seals (Phoca vitulina). Canadian Journal of Zoology 70, 553–560. https://doi.org/10.1139/Z92-083

Wickham, H., 2019. stringr: Simple, Consistent Wrappers for Common String Operations.

Wickham, H., 2011. The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software 40, 1–29.

Wickham, H., Averick, M., Bryan, J., Chang, W., D'Agostino McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Lin Pedersen, T., Miller, E., Milton Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D.P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., Yutani, H., 2019. Welcome to the tidyverse. Journal of Open Source Software 4, 1686.

Wickham, H., François, R., Henry, L., Müller, K., 2021. ddplyr: A Grammar of Data Manipulation.

Wickham, H., Hester, J., Francois, R., 2018. readr: Read Rectangular Text Data.

Wood, S.N., 2011. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B) 73, 3–36.

Zemanova, M.A., 2020. Towards more compassionate wildlife research through the 3Rs principles: moving from invasive to non-invasive methods. https://doi.org/10.2981/wlb.00607 2020. https://doi.org/10.2981/WLB.00607

Appendix A

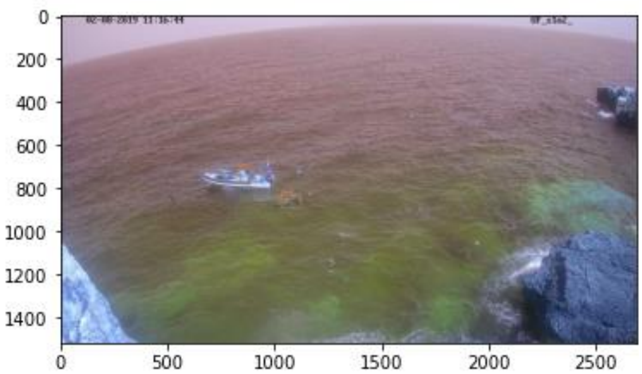
200017683

Exploration of implementation of TensorFlow for neural network detection of vessels

- Launch Python 3.8.8 in Spyder using Anaconda

```
!pip install opencv-python
!pip install cvlib
!pip install matplotlib
!pip install tensorflow

# import necessary modules
import cv2
import matplotlib.pyplot as plt
import cvlib as cv
from cvlib.object_detection import draw_bbox

# load image
im = cv2.imread('e:\python_test\OF_s1o2_19_08_02_11_15.jpg')
# box/label vessels in image using cv common object detection (includes boats)
bbox, label, conf = cv.detect_common_objects(im)

# save output image
output_image = draw_bbox(im, bbox, label, conf)

#  view labelled image                          plt.imshow(output_image)
plt.show()
```

Examples of labelled vessels:



- Missed lots of vessels, also labelled divers in water as vessel.

Remove night-time images in Python 3.8.8

- Launch Python 3.8.8 in Spyder using Anaconda

```
import cv2
import numpy as np
import os
from skimage import io

# actual loop attempt with all data
os.chdir('E:\\boats_only')
cwd = os.getcwd()
cwd

# set image colour threshold
color_thres = ([86, 86, 86])
# loop over all files in the working directory.
for filename in os.listdir('.'):
    myimg = cv2.imread(filename)

# python will use BGR when using loaded images rather than RGB
  # calculate the mean colour of each pixel row [BGR]
    avg_color_per_row = np.average(myimg, axis=0)

    # calculate the mean colour of all pixel row [BGR]
    avg_color = np.average(avg_color_per_row, axis=0)

    # print(avg_color) will print BGR value of average image colour

    if (avg_color > color_thres).all():
        # swap working directory to folder for daytime images
        os.chdir('E:\\boats_final')
        # converts image from GBR used by cv to RGB (standard coloured image)
        myimg_rgb = cv2.cvtColor(myimg, cv2.COLOR_BGR2RGB)
        # save RGB image with same file name
        io.imsave(str(filename), myimg_rgb)
        # swap back to original working directory
        os.chdir('E:\\boats_only')
```

- Colour of RGB threshold [86, 86, 86] (left) and night-time image from 21:45 on 01/08/2019 at Site 3 with a colour average that exceeded this threshold (right).

For-loop to manually classified vessel detections in a random subset of images for model validation.

- Uses MATLAB 9.6.0.1099231 (R2019a)

```
files = dir('E:\boats_final\*.jpg') ;   % all jp gimages in folder
N = length(files) ;      % total files
if isfile('image_validation_working.mat')  % if workspace save in directory
    load image_validation_working.mat      % then use existing workspace
else
    idx = randperm(N) ;      % random order of numbers till N
    counter = zeros(N,1);          % create matrix for results
    filenames =strings(N,1);
    boatPresence = zeros(N,1);
end

%want three columns - one row per image file - a counter index, the
%filename and 1/0 for correct or not not in classifiation of
presence/absence of boats

start = find(counter==0,1);  % counts number of images classified manually

for i = start:N              % loop for each file
    image = files(idx(i)).name;
    image_file = convertCharsToStrings(image);
    filenames(i) = image;
    figure
    imshow(image)      % display image

    % creates button to click yes/no and records output as numbers
        % 1 == Yes; 2 == No; 3 == End;
        % 3 (end) used to end loop
    input = menu('Are boats present in this image?','Yes','No','End');
    if input==3
        close all force
        save image_validation_working
        break
    elseif input==1
        boatPresence(i) = 1;
    end
    counter(i) = i;

    close all force
    save image_validation_working
    % save directory when finished session
end
% save data
validationTable = table(counter, filenames, boatPresence);

writetable(validationTable,'E://BL5599/descriptive_statistics/validation.cs
v','Delimiter',',')
```

<u>All code from this point henceforth was implemented in RStudio Version 1.2.5001 unless otherwise stated.</u>

Load data as list of filenames and create csv file showing monk seal presence/absence in caves by extracting data from filename.

Monk seal images provided in two folders by the Octopus Foundation , one with ALL images capture, and one folder of images with monk seal presence ONLY.

File produced = monk_seal_data.csv

```
# load stringr package to extract dates from image file name
library(stringr)
## Warning: package 'stringr' was built under R version 3.6.3
# use list files function to list all image files irrelevant of
presence/absence

# list file names of all images from all sites (44953 images) in dataframe
data<-data.frame(list.files(path =
"E:/monk_seal_dataset/2019_datasets/2019_all", pattern = NULL, all.files =
FALSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE,
include.dirs = FALSE, no.. = FALSE))

# name first column as file_name
names(data)[1] <- "file_name"

# extract dates from image file name (**_**_** format) and save to
dataframe
data$date <- c(stringr::str_extract(data$file_name, "([0-
9]{2})[:punct:]([0-9]{2})[:punct:]([0-9]{2})"))

# extract time from image file name (**_** 24hr format) and save to
dataframe
# (?=.jpg) = last matching **_** before '.jpg' in file name
data$time <- c(stringr::str_extract(data$file_name, "([0-
9]{2})[:punct:]([0-9]{2})(?=.jpg)"))


# convert date and time to factors from chr
data$date <- as.factor(data$date)
data$time <- as.factor(data$time)

# check data is now factor format
str(data)
## 'data.frame':    44913 obs. of  3 variables:
##  $ file_name: Factor w/ 44913 levels "OF_s1i1_19_05_28_14_00.jpg",..: 1
2 3 4 5 6 7 8 9 10 ...
##  $ date     : Factor w/ 192 levels "19_05_23","19_05_24",..: 6 6 6 6 6
6 6 6 6 6 ...
##  $ time     : Factor w/ 134 levels "00_00","00_15",..: 89 90 92 94 96
98 99 102 103 105 ...
# convert factor date to date format with YYYY (eg 2019) format
data$date <- format(as.Date(data$date,"%Y_%m_%d"), "20%y-%m-%d")

# convert times from **_** format to **:** format
data$time <- gsub("[_]", ":", data$time)
```

```
# create datetime field
data$datetime <- as.POSIXct(paste(data$date, data$time), format="%Y-%m-%d
%H:%M")

# add site field and extract location/site number from file name
data$site <- c(stringr::str_extract(data$file_name, "([:lower:]{1})([0-
9]{1})"))
# remove s before site number
data$site <- gsub("[s]", "", data$site)

# load seal presence file names
seals<-data.frame(list.files(path =
"E:/monk_seal_dataset/2019_datasets/2019_seals", pattern = NULL, all.files
= FALSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE,
include.dirs = FALSE, no.. = FALSE))
# assign column name
names(seals)[1] <- "file_name"

# create new column in dataframe where if seal presence file name = all
data file name
# then 'presence' assigned value of 1, if not assigned value of 0 for
absence
presence = (data$file_name %in% seals$file_name)
data$presence <- ifelse(presence=='TRUE',"1",0)

# create csv
write.csv(data,"E:/BL5599/monk_seal_data.csv", row.names = FALSE)
```

Script to match seal images with classified boat images using datetime/filename

File produced = final_dataset.csv

```
# Load packages
library(plyr)
library(readr)
# Read in seal data
monk_seal_data <-
read.csv('E:/BL5599/descriptive_statistics/monk_seal_data.csv',header=T)

# Read in boat data produced by Mirador output summary
boat_data <-
read.csv('E:/BL5599/descriptive_statistics/all_boats_FINAL.csv',header=T)

# image filename and variable name formats:
# boat data
# ï..file_name
# OF_s1o2_19_05_28_13_36.jpg
# after s1 (site 1) in filename, ‘o’ = outside cave = vessel data

# seal data – i1 = inside cave
# file_name
# OF_s1i1_19_05_28_13_36.jpg
# after s1 (site 1) in filename, ‘i’ = inside cave = seal data
```

```
# make file names the same for both seal and boat data
boat_data$ï..file_name <- gsub('o2', "", boat_data$ï..file_name)
monk_seal_data$file_name <- gsub('i1', "", monk_seal_data$file_name)

# Rename ï..file_name columns to match seal data
names(boat_data)[names(boat_data) == "ï..file_name"] <- "file_name"

# merge monk seal data and boat data into one csv file with NAs for night
time vessel data
joined_data <- merge(boat_data, monk_seal_data, by = "file_name", all =
TRUE)

# subset data into separate sites to calculate boat average over 24hr
period for each site
joined_data1 <- subset(joined_data, site == 1)
joined_data2 <- subset(joined_data, site == 2)
joined_data3 <- subset(joined_data, site == 3)

# create new column for total boat count
joined_data1$boat_total <- joined_data1$sail.boat +
joined_data1$motor.boat + joined_data1$kayak
joined_data2$boat_total <- joined_data2$sail.boat +
joined_data2$motor.boat + joined_data2$kayak
joined_data3$boat_total <- joined_data3$sail.boat +
joined_data3$motor.boat + joined_data3$kayak

# calculate vessel count average over single date for all data at each
site
boats_sum1 <- aggregate(boat_total ~ date.x, joined_data1, sum)
boats_sum2 <- aggregate(boat_total ~ date.x, joined_data2, sum)
boats_sum3 <- aggregate(boat_total ~ date.x, joined_data3, sum)

# count data entries per day
data_count1 <- dplyr::count(joined_data1, date.x)
data_count2 <- dplyr::count(joined_data2, date.x)
data_count3 <- dplyr::count(joined_data3, date.x)

# merges total sum of boats per 24hrs with number of data enteries in
24hrs
boat_date1 <- merge(data_count1, boats_sum1, by = "date.x", all = TRUE)
boat_date2 <- merge(data_count2, boats_sum2, by = "date.x", all = TRUE)
boat_date3 <- merge(data_count3, boats_sum3, by = "date.x", all = TRUE)

# calculates average number of boats over 24hr period as index of vessel
presence
boat_date1$boat_index <- boat_date1$boat_total/boat_date1$n
boat_date2$boat_index <- boat_date2$boat_total/boat_date2$n
boat_date3$boat_index <- boat_date3$boat_total/boat_date3$n

# merge monk seal data with boat data
final_data1 <- merge(joined_data1, boat_date1, by = "date.x", all = TRUE)
final_data2 <- merge(joined_data2, boat_date2, by = "date.x", all = TRUE)
final_data3 <- merge(joined_data3, boat_date3, by = "date.x", all = TRUE)
```

```r
# add all 3 sites together and save as final csv
final_data_all <- rbind(final_data1,final_data2,final_data3)

library(tidyverse)
# remove unneccessary columns from data frame which remain from merges
drop <- c("date.x.x", "time.x", "diver", "swimmer", "spearfisherman",
"monk.seal", "other", "date.y","date.x.y")
FINAL_CSV <- final_data_all[,!(names(final_data_all) %in% drop)]

# simplify names of dataframe columns
FINAL_CSV <- FINAL_CSV %>%  dplyr::rename(date = date.x, time = time.y,
boats_total_15min = boat_total.x, boats_total_24hr = boat_total.y,
data_entry_total = n)

# make csv
write.csv(FINAL_CSV,'E:/BL5599/descriptive_statistics/final_dataset.csv')
```

Condense data onto a daily scale (24hrs) rather than 15-minute intervals

Presence index of monk seals is a photographic rate, e.g. number of images with seals present in 24hrs divided by total number of images in 24hrs

File produced = final_presence_index.csv

```r
# Read in seal data
monk_seal_data <-
read.csv('E:/BL5599/descriptive_statistics/monk_seal_data.csv',header=T)

# Read in boat data produced by Mirador output summary
boat_data <-
read.csv('E:/BL5599/descriptive_statistics/all_boats_FINAL.csv',header=T)

# boat data
# ï..file_name
# OF_s1o2_19_05_28_13_36.jpg

# seal data
# file_name
# OF_s1i1_19_05_28_13_36.jpg
# make file names same for both seal and boat data
boat_data$ï..file_name <- gsub('o2', "", boat_data$ï..file_name)
monk_seal_data$file_name <- gsub('i1', "", monk_seal_data$file_name)

# Rename ï..file_name columns to match seal data
names(boat_data)[names(boat_data) == "ï..file_name"] <- "file_name"

# merge monk seal data and boat data into one csv file with NAs for night
time vessel data
joined_data <- merge(boat_data, monk_seal_data, by = "file_name", all =
TRUE)
```

```r
# subset data into separate sites to calculate boat average over 24hr
period for each site
joined_data1 <- subset(joined_data, site == 1)
joined_data2 <- subset(joined_data, site == 2)
joined_data3 <- subset(joined_data, site == 3)

# count data entries per day
data_count1 <- dplyr::count(joined_data1, date.x)
data_count2 <- dplyr::count(joined_data2, date.x)
data_count3 <- dplyr::count(joined_data3, date.x)

# total monk seal occurrence per day
count_occurence_1 <- joined_data1 %>% group_by(date.y,presence) %>%
tally()
count_occurence_2 <- joined_data2 %>% group_by(date.y,presence) %>%
tally()
count_occurence_3 <- joined_data3 %>% group_by(date.y,presence) %>%
tally()
# add site
count_occurence_1$site <- 1
count_occurence_2$site <- 2
count_occurence_3$site <- 3

# merges total occurence of seals per 24hrs with number of data enteries
in 24hrs
total_data_occurence1 <- merge(count_occurence_1, data_count1, by =
"date.y", all = TRUE)
total_data_occurence2 <- merge(count_occurence_2, data_count2, by =
"date.y", all = TRUE)
total_data_occurence3 <- merge(count_occurence_3, data_count3, by =
"date.y", all = TRUE)

# Calculates presence index by dividing total number of images in 24hrs
where seals occur by the total number of images within the same 24hr
period.
# Multiplying by presence gives 0 as index value for absence.
total_data_occurence1$presence_index <- (total_data_occurence1$n.x/
total_data_occurence1$n.y)*total_data_occurence1$presence
total_data_occurence2$presence_index <- (total_data_occurence2$n.x/
total_data_occurence2$n.y) *total_data_occurence2$presence
total_data_occurence3$presence_index <- (total_data_occurence3$n.x/
total_data_occurence3$n.y) *total_data_occurence3$presence
# add all 3 sites together and save as final csv
final_presence_index <- rbind(total_data_occurence1,
total_data_occurence2, total_data_occurence3)
# simplify names of dataframe columns
final_presence_index <- final_presence_index %>% dplyr::rename(date =
date.y, presence_24hr = n.x,  seal_entries_24hr = n.y)
# make csv file
write.csv(final_presence_index,'E:/BL5599/descriptive_statistics/
final_presence_index.csv')
```

Merge final_presence_index (monk seals) with date_boat / vessel index from 'final_presence_index.csv'

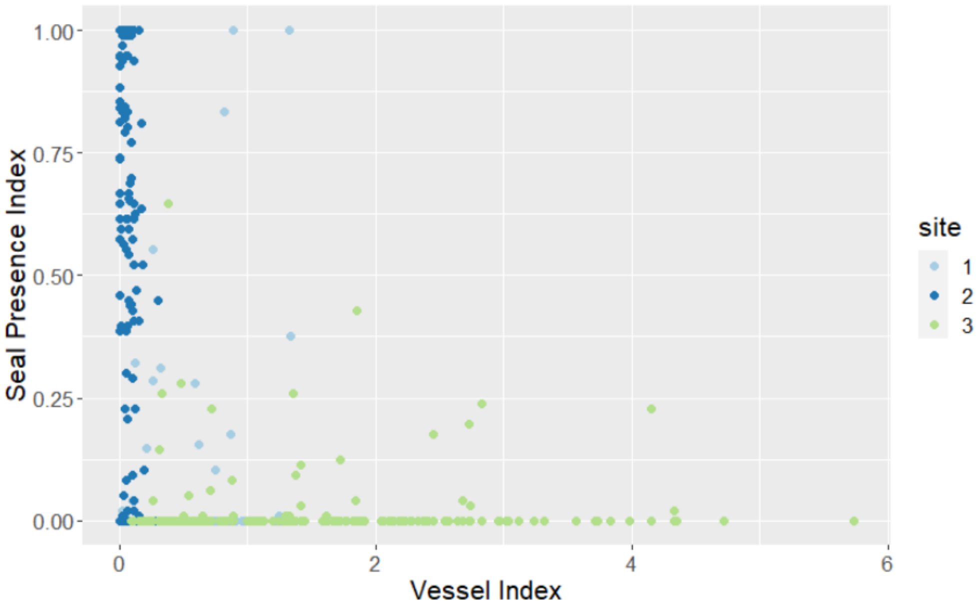
File produced = seal_vessel_index.csv, Environment name = 'new_joined_data'

```
# Read in final data
FINAL_CSV <- read.csv('E:/BL5599/descriptive_statistics/
final_presence_index.csv',header=T)

# reduce FINAL_CSV (all data with boat index) to core variables needed for
further anaylsis:
# date, site, boat index
boat_vars <- c("date","site","boat_index")
new_boat  <- FINAL_CSV[boat_vars]

# Remove duplicates to only keep unqiue values (one vessel presence index
per date and site)
new_boat <- new_boat %>% distinct()

# reduce seal presence index date to only needed columns. Date and site
needed to match to boat data
new_seal <- final_presence_index %>%  select(date, site, presence_index)

# need to merge data - merge won't work unless date is in same format ->
convert to dates as currently factors
str(new_boat)
## 'data.frame':    460 obs. of  3 variables:
##  $ date      : Date, format: "2019-06-01" "2019-07-01" "2019-08-01" ...
##  $ site      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ boat_index: num  0.0984 0.2419 0.431 0.4444 1.0889 ...

str(new_seal)
## 'data.frame':    609 obs. of  3 variables:
##  $ date      : Factor w/ 192 levels "01/06/2019","01/07/2019",..: 1 2 3
4 6 7 8 9 10 12 ...
##  $ site      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ presence_index: num  0 0 0 0 0 0 0 0 0 0 ...

# make sure dates for both sets of data are the same format
new_boat$date <- as.Date(new_boat$date, format = "%d/%m/%Y")
new_seal$date <- as.Date(new_seal$date, format = "%Y-%m-%d")

new_joined_data <- left_join(new_boat, new_seal, by.x=c("date","site"),
by.y=c("date","site"))
# vessel index and seal presence index now merged into one dataframe,
# one index value per date per site

# round presence index and vessel index to 2DP
new_joined_data$presence_index <- round(new_joined_data$presence_index,
digits = 2)
new_joined_data$boat_index <- round(new_joined_data$boat_index, digits =
2)
write.csv(new_joined_data,'E:/BL5599/descriptive_statistics/seal_vessel_in
dex.csv')
```

Exploratory Plots

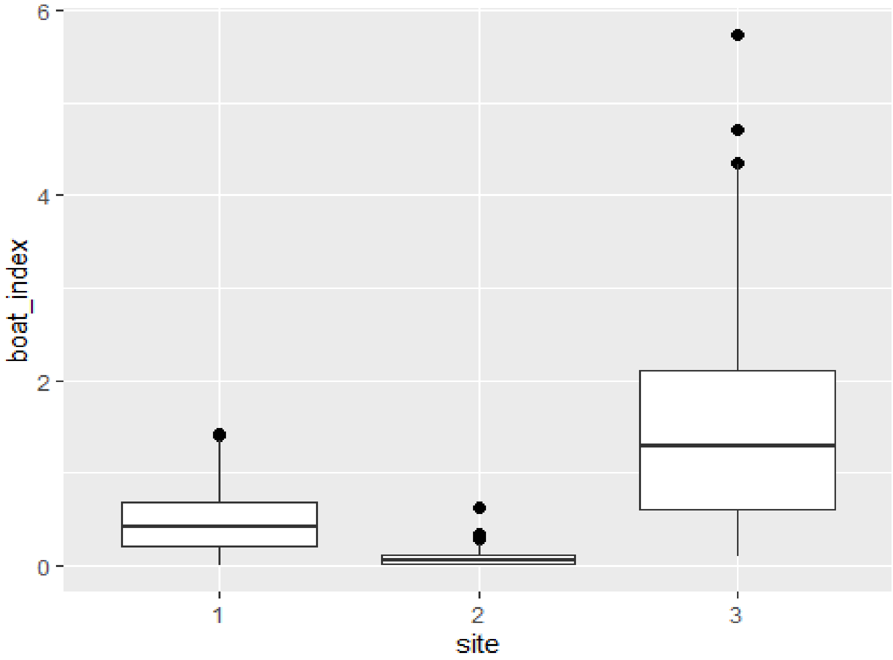
Plot vessel index with seal presence index to view general trends

```
# read in data
new_joined_data <- read.csv('E:/BL5599/descriptive_statistics/
seal_vessel_index.csv', header = T)

# make basic plot
plot(new_joined_data$boat_index, new_joined_data$presence_index)

# replot coloured by site
new_joined_data$site <- as.factor(new_joined_data$site)

# replot using ggplot
library(ggplot2)
plot_index <- ggplot(new_joined_data, aes(boat_index,presence_index,
color=site)) + geom_point(shape=16,size=2.3) +  scale_color_brewer(palette
= "Paired") + xlab("Vessel Index") + ylab("Seal Presence Index") +
theme(text = element_text(size=16))
plot_index
```



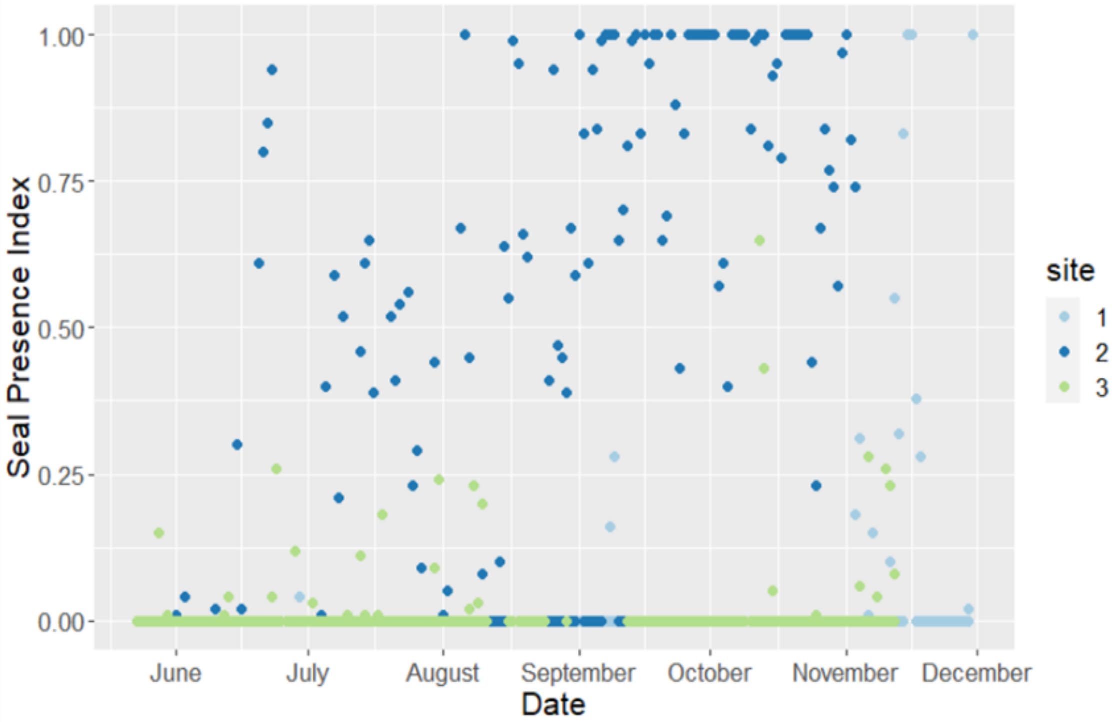
Plot basic box plot of difference in vessel index between sites

```
# plot basic boxplot of data by site
ggplot(new_joined_data, aes(x=site, y=boat_index)) +
geom_boxplot(outlier.colour="black", outlier.shape=16, outlier.size=2)
```



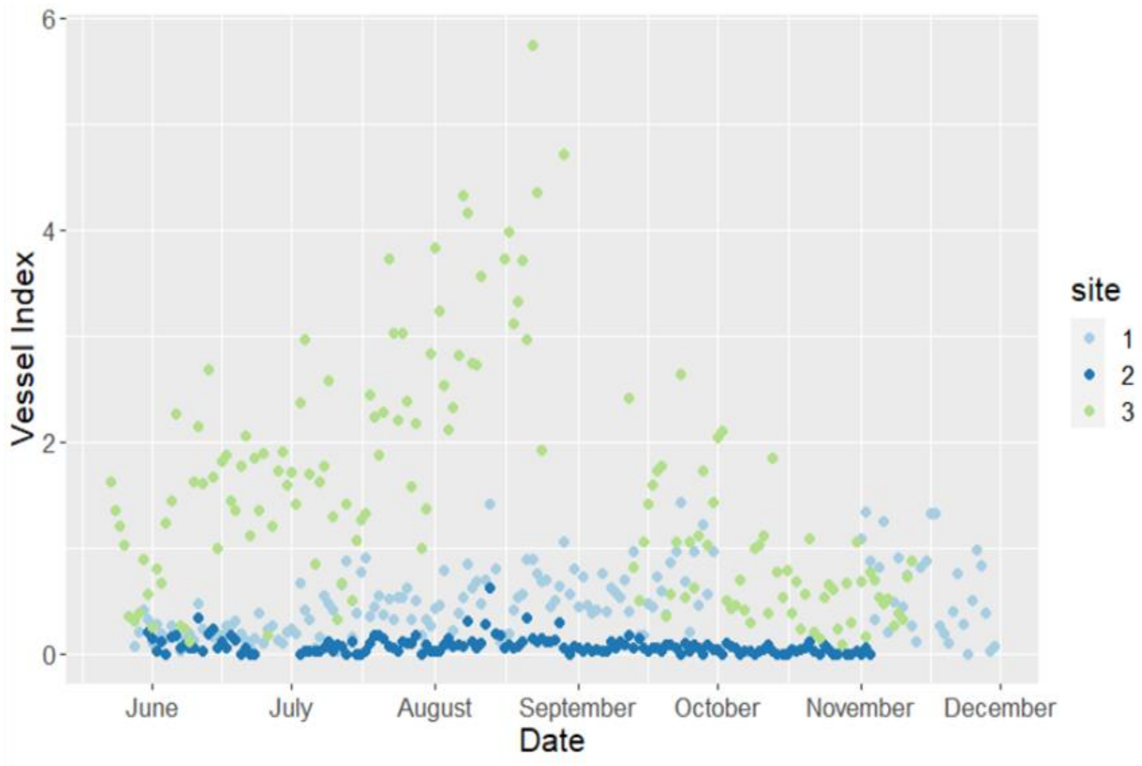
Plot seal trends by date plot using ggplot

```
# set axis breaks
month_breaks <- c("2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01",
"2019-10-01", "2019-11-01", "2019-12-01")
month_breaks <- as.Date(month_breaks)

# plot seal presence index by date
plot_date <- ggplot(new_joined_data, aes(date,presence_index, color=site))
+ geom_point(shape=16,size=2.3) +  scale_color_brewer(palette = "Paired")
+ xlab("Date") + ylab("Seal Presence Index") + theme(text =
element_text(size=16))
plot_date
plot_date  +  scale_x_date(breaks = month_breaks, labels = c("June",
"July", "August", "September", "October", "November", "December"))
```



Plot vessel trends by date plot using ggplot

```
# plot vessel index by date
plot_date_boat <- ggplot(new_joined_data, aes(date,boat_index,
color=site)) + geom_point(shape=16,size=2.3) +  scale_color_brewer(palette
= "Paired") + xlab("Date") + ylab("Vessel Index") + theme(text =
element_text(size=16))
plot_date

# use axis breaks set above (month_breaks)
# plot neat ggplot of data

plot_date_boat + scale_x_date(breaks = month_breaks, labels = c("June",
"July", "August", "September", "October", "November", "December"))
```



Calculate agreement and error between manual and machine classifications of vessel presence

```
# load packages
library(dplyr)
# load Mirador classification data and rename
machine <-
read.csv('E:/BL5599/descriptive_statistics/all_boats_FINAL.csv',header=T)
machine <- machine %>%  dplyr::rename(filenames = ï..file_name)

# load manual classifications completed through MATLAB script (Appendix A)
manual <-
read.csv('E:/BL5599/descriptive_statistics/validation.csv',header=T)

agreement <- merge(machine, manual, by = "filenames")
# create new column for total boat count
agreement$boat_total <- agreement$sail.boat + agreement$motor.boat +
agreement$kayak
agreement$boat_presence_machine <- ifelse(agreement$boat_total >= 1, 1, 0)

# make csv
write.csv(agreement,'E:/BL5599/descriptive_statistics/agreement.csv')
```

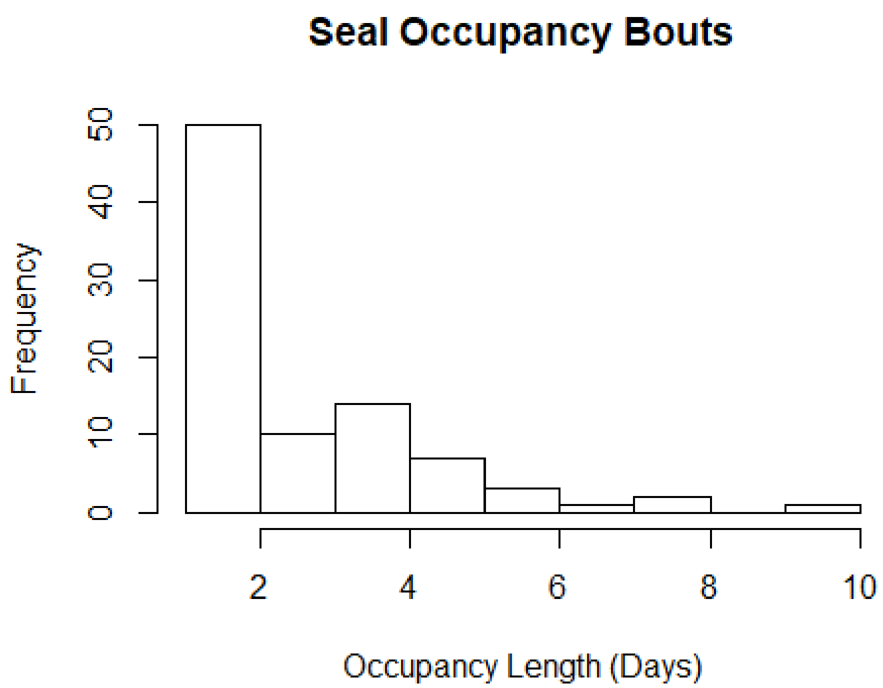
```
# calculate total agreement %
sum(agreement$boatPresence == agreement$boat_presence_machine)
## [1] 2344
2344/2735
## [1] 0.8570384
# false positives
false_pos <- sum(agreement$boatPresence == 0 &
agreement$boat_presence_machine == 1)
false_pos
## [1] 286

286/2735
## [1] 0.1045704
true__pos <- sum(agreement$boatPresence == 1 &
agreement$boat_presence_machine == 1)
true__pos
## [1] 636

636/2735
## [1] 0.2325411

false_neg <-  sum(agreement$boatPresence == 1 &
agreement$boat_presence_machine == 0)
false_neg
## [1] 105

105/2735
## [1] 0.03839122
true__neg <- sum(agreement$boatPresence == 0 &
agreement$boat_presence_machine == 0)
true__neg
## [1] 1708
1708/2735
## [1] 0.6244973
# checking right number of entries (2735 = 10% of data)
false_pos + true__pos + false_neg + true__neg
## [1] 2735
# create subsets and recalculate agreement for each site using filename
agreement_s1 <- agreement[grep("OF_s1", agreement$filenames), ]
agreement_s2 <- agreement[grep("OF_s2", agreement$filenames), ]
agreement_s3 <- agreement[grep("OF_s3", agreement$filenames), ]

# calculate false positives by site
false_pos1 <- (sum(agreement_s1$boatPresence == 0 &
agreement_s1$boat_presence_machine == 1))/nrow(agreement_s1)
false_pos2 <- (sum(agreement_s2$boatPresence == 0 &
agreement_s2$boat_presence_machine == 1))/nrow(agreement_s2)
false_pos3 <- (sum(agreement_s3$boatPresence == 0 &
agreement_s3$boat_presence_machine == 1))/nrow(agreement_s3)
# calculate true positives by site
true_pos1 <- (sum(agreement_s1$boatPresence == 1 &
agreement_s1$boat_presence_machine == 1))/nrow(agreement_s1)
true_pos2 <- (sum(agreement_s2$boatPresence == 1 &
```

```
agreement_s2$boat_presence_machine == 1))/nrow(agreement_s2)
true_pos3 <- (sum(agreement_s3$boatPresence == 1 &
agreement_s3$boat_presence_machine == 1))/nrow(agreement_s3)

# calculate false negatives by site
false_neg1 <-  (sum(agreement_s1$boatPresence == 1 &
agreement_s1$boat_presence_machine == 0))/nrow(agreement_s1)
false_neg2 <-  (sum(agreement_s2$boatPresence == 1 &
agreement_s2$boat_presence_machine == 0))/nrow(agreement_s2)
false_neg3 <-  (sum(agreement_s3$boatPresence == 1 &
agreement_s3$boat_presence_machine == 0))/nrow(agreement_s3)

# calculate true negatives by site
true_neg1 <-  (sum(agreement_s1$boatPresence == 0 &
agreement_s1$boat_presence_machine == 0))/nrow(agreement_s1)
true_neg2 <-  (sum(agreement_s2$boatPresence == 0 &
agreement_s2$boat_presence_machine == 0))/nrow(agreement_s2)
true_neg3 <-  (sum(agreement_s3$boatPresence == 0 &
agreement_s3$boat_presence_machine == 0))/nrow(agreement_s3)

# check calculations are correct (total should = 1.0 = 100%)
false_pos1 + true_pos1 + false_neg1 + true_neg1
## [1] 1
false_pos2 + true_pos2 + false_neg2 + true_neg2
## [1] 1
false_pos3 + true_pos3 + false_neg3 + true_neg3
## [1] 1

# Get agreement values
false_pos1; false_pos2; false_pos3
## [1] 0.186214
## [1] 0.0460733
## [1] 0.07549505
true_pos1; true_pos2; true_pos3
## [1] 0.1759259
## [1] 0.04188482
## [1] 0.5259901
false_neg1; false_neg2; false_neg3
## [1] 0.0462963
## [1] 0.03141361
## [1] 0.03712871
true_neg1; true_neg2; true_neg3
## [1] 0.5915638
## [1] 0.8806283
## [1] 0.3613861
```

Source of Site 1 false-positive detections was identified as rocky outcrop within the camera frame.

Seal Occupancy Bouts

```
# occupancy bouts in days from Octopus Foundation Photo-ID pdf.
# Data manually extracted into csv format
seal_ID <- read.csv("E:/BL5599/seal_occupancy.csv", header = TRUE)


summary(seal_ID$ï..Length)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.739   4.000  10.000

# average length of seal occupancy = 2.739 days

# plot histogram
hist(seal_ID$ï..Length, xlab="Occupancy Length (Days)", main = "Seal
Occupancy Bouts")
```




```
# make high resolution plot
jpeg("E:/BL5599/high_res_plots/seal_occupancy_hist.jpeg", width = 6,
height = 4, units = 'in', res = 600)
hist(seal_ID$ï..Length, xlab="Occupancy Length (Days)", main = "Seal
Occupancy Bouts")
dev.off()

## png
##   2
```

Appendix B

200017683

Convert data for binomial distribution for Binomial GAM instead of proportional data (seal presence index)

Calculate vessel success/failures first to be able to calculate vessel index

Convert date to day number (1:365)

File produced = seal_vessel_binomial.csv

```
# load required packages
library(dplyr)

# Read in boat data as total number of images (trials) per day already
calculated
boat_data <- read.csv('E:/BL5599/descriptive_statistics/
all_boats_FINAL.csv', header = T)

# Read in final_csv data as vessels count per 15 mins (1 trial) already
calculated (boats_total_15mins)
binomial_data <- FINAL_CSV
# convert from count to presence/absence
binomial_data$boat_p_a <- ifelse(binomial_data$boats_total_15min >= 1, 1,
0)
# calculate total successes in 24 hour period


# subset data into separate sites to calculate vessel successes over 24hr
period for each site
binomial_data1 <- subset(binomial_data, site == "1")
binomial_data2 <- subset(binomial_data, site == "2")
binomial_data3 <- subset(binomial_data, site == "3")

# boat_data only has total count of vessel detected
# need total number of successes (vessel present in image) in given day
succ_occurence1 <- aggregate(boat_p_a ~ date, binomial_data1, sum)
succ_occurence2 <- aggregate(boat_p_a ~ date, binomial_data2, sum)
succ_occurence3 <- aggregate(boat_p_a ~ date, binomial_data3, sum)

# succ_occurence1     "date"
# boat_date1          "date.x"    "n"

# merges total vessel data entries with total number of successes (vessel
present) per 24hrs
vessel_successes1 <- select(left_join(succ_occurence1, boat_date1, by =
c("date" = "date.x")), -c(boat_total, boat_index))
vessel_successes2 <- select(left_join(succ_occurence2, boat_date2, by =
c("date" = "date.x")), -c(boat_total, boat_index))
vessel_successes3 <- select(left_join(succ_occurence3, boat_date3, by =
c("date" = "date.x")), -c(boat_total, boat_index))
```

```r
# add in site
vessel_successes1$site <- 1
vessel_successes2$site <- 2
vessel_successes3$site <- 3

# add all 3 sites together and save as final csv
final_vessel_successes <- rbind(vessel_successes1, vessel_successes2,
vessel_successes3)

write.csv(final_vessel_successes, 'E:/BL5599/descriptive_statistics/
vessel_binomial.csv')

##########################################################################
###
# Do the same for seal total trials (n of images) and total successes
(seal present)

# Read in monk seal data for presence/absence index
final_presence_index <- read.csv('E:/BL5599/descriptive_statistics/
final_presence_index.csv', header = T)

# Remove duplicates to only keep unique values (one vessel presence index
per date and site)
# subset into presence/absence = 1s/0s
seal_presence <- subset(final_presence_index, presence == 1)
seal_absence  <- subset(final_presence_index, presence == 0)

# Load plyr library for match_df and anti_join
library(plyr)

# find rows where 0s and 1s recorded on same day -> only count 1s
(successes) within 24hrs
remove_absence  <- match_df(seal_absence, seal_presence, on = c("date",
"site"))
# double checked reverse to make sure values had presence duplicates
remove_absence2 <- match_df(seal_absence, seal_presence, on = c("date",
"site"))

# now delete rows in seal_absence that match remove_absence
# this will leave only total number of successes per date
# and remove total number of failures if successes also occurred that day
final_presence_index <- anti_join(final_presence_index, remove_absence)
# rename to avoid confusion with previous csv
final_seal_successes <- final_presence_index


# 0/1s for presence, but need to translate into number of successful
trials
# eg, presence = 0 (seals absence), in  50 images out of 50 images
# if presence = 0, number of successful trials needs to be 0

final_seal_successes$succ_trials <- ifelse(final_seal_successes$presence
== 0, 0, final_seal_successes$presence_24hr)
# join 0/1 seals per date with 0/1 boats per date
```

```
# dates in different formats - will not join df unless dates are same
format
# seals date   =2019-05-28
# vessels date = 28/05/2019
final_seal_successes$date <- as.Date(final_seal_successes$date, format =
"%Y-%m-%d")
final_vessel_successes$date <- as.Date(final_vessel_successes$date, format
= "%Y-%m-%d")

final_successes_all <- left_join (final_seal_successes,
final_vessel_successes, bx.x=c("date","site"), by.y=c("date","site"))

# convert date to day number
doy <- strftime(final_successes_all$date, format = "%j")
final_successes_all$day_num <- as.numeric(doy)

# save csv
write.csv(final_successes_all,'E:/BL5599/descriptive_statistics/seal_vesse
l_binomial.csv')
```

Calculate Bernoulli trials of seal presence (successes) and seal absence (failures)

seal_entries_24hr - succ_trials = failures (seal absent)

File Produced = seal_vessel_binomial_V2.csv, Environment Name = final_successes_all

```
# calculate failures
final_successes_all$seal_failures <- final_successes_all$seal_entries_24hr
- final_successes_all$succ_trials

# include vessel index
final_successes_all$vessel_index <-
(final_successes_all$boat_p_a/final_successes_all$n)


# create vessels total successes/total failures
final_successes_all$vessel_failures <- final_successes_all$n -
final_successes_all$boat_p_a

# save csv
write.csv(final_successes_all,'E:/BL5599/descriptive_statistics/seal_vesse
l_binomial_V2.csv')
```

GAM Exploratory Plots

Model Bernoulli trials of seal presence/absence

```
# load mgcv library
library(mgcv)

# load data
final_successes_all <- read.csv ('E:/BL5599/descriptive_statistics/
seal_vessel_binomial_V2.csv', header = T)

# model binomial trials of seal presence with vessel index
seals_vessels <- gam(cbind(succ_trials, seal_failures) ~ s(vessel_index),
data=final_successes_all, family=binomial(link="cloglog"))
plot(seals_vessels)
summary(seals_vessels)
gam.check(seals_vessels)
```

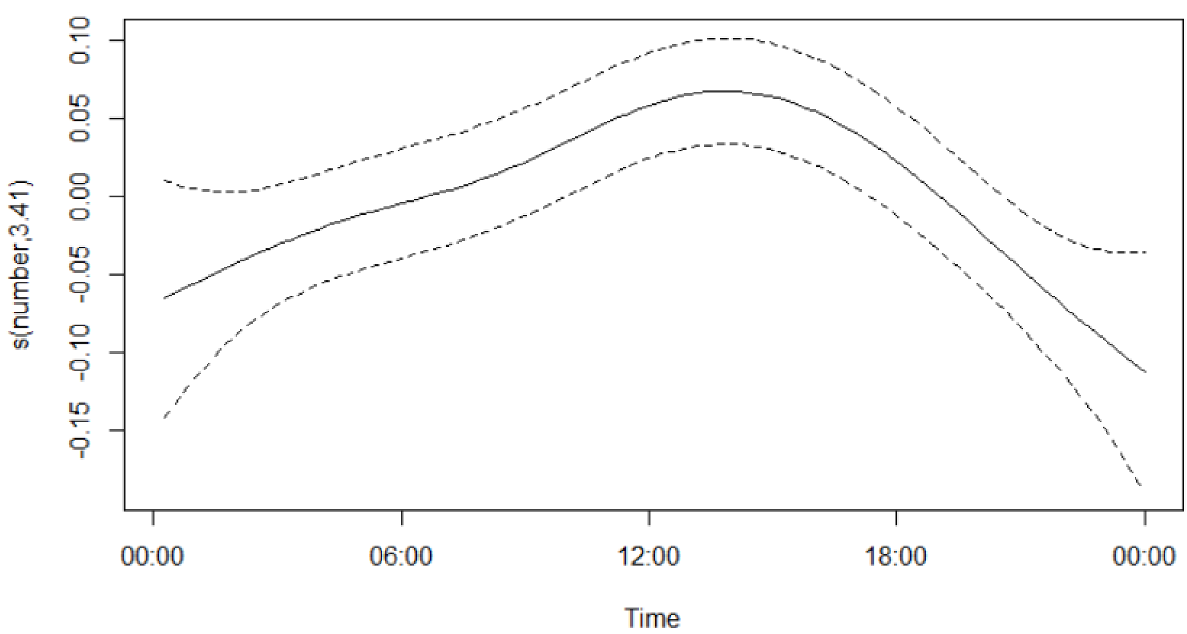
Model Bernoulli trials of seal presence/absence by site

```
# model binomial trials of seal presence with vessel index by site

seals_vessels_site <- gam(cbind(succ_trials, seal_failures) ~
s(vessel_index, by= as.factor(site)), data=final_successes_all,
family=binomial(link="cloglog"))
plot(seals_vessels_site)
summary(seals_vessels_site)
gam.check(seals_vessels_site)
```

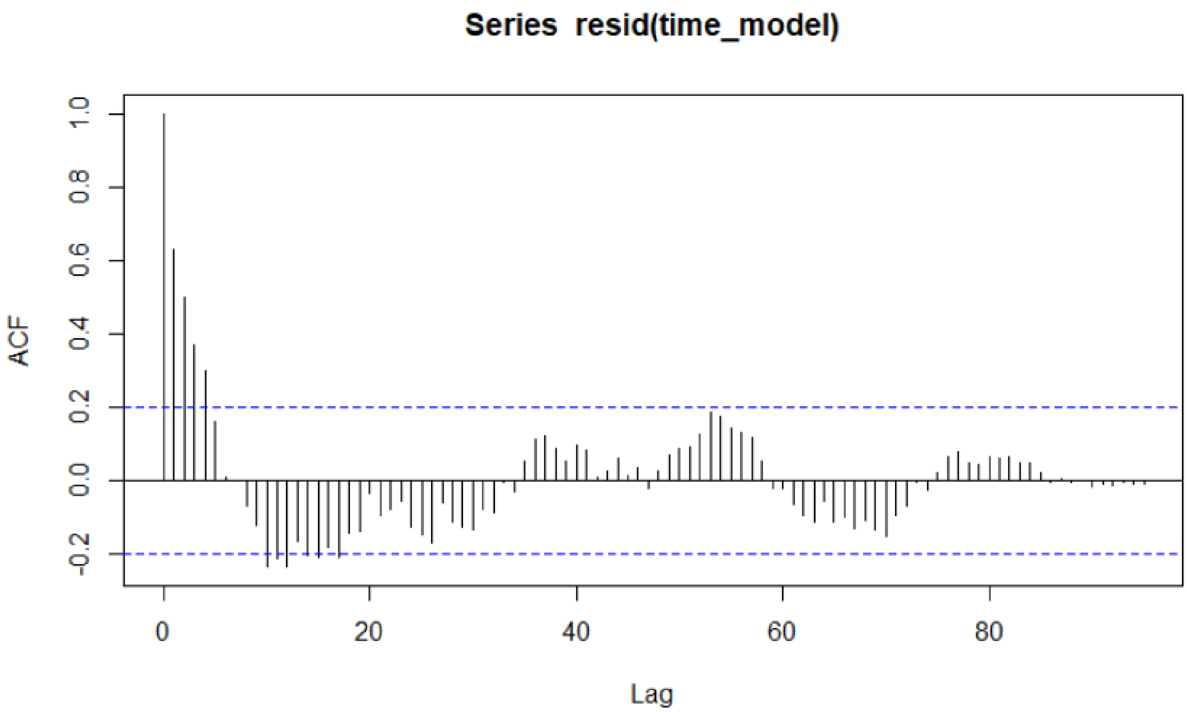
Examine autocorrelation for seals and vessels over time and date

```
# Look at autocorrelation within 24hr period
# create binomial trials for each time within given day
# cyclic spline for times
# times -> 1:96 (4 images per hour for 24hrs)

# load packages
library(dplyr)

# load seal data
monk_seal_data <- read.csv
('E:/BL5599/descriptive_statistics/final_dataset.csv',header=T)

# monk seals with time
# need to calculate total number of monk seal occurances for each time
period
# count total number of occurances of monk seals

# calculate total number of data entries per time (n)
```

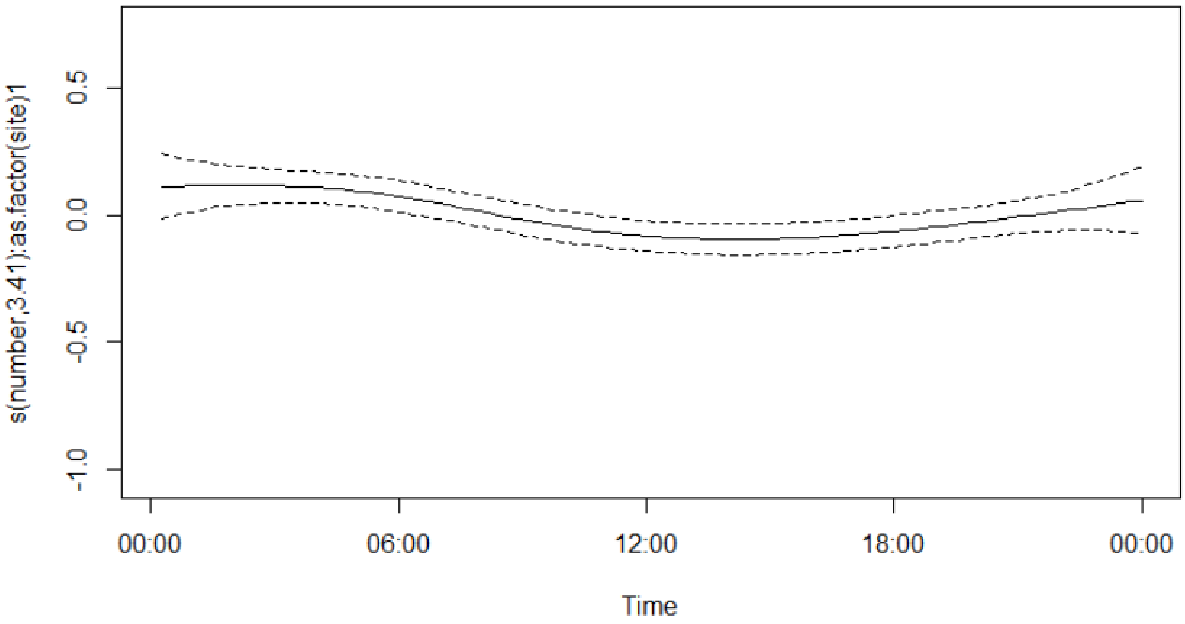
```r
time_seals <- monk_seal_data %>% group_by(time) %>% tally()
# calculate total number of failures (seal absent) and join to n
time_seals_failures<- monk_seal_data %>% group_by(time) %>%
tally(presence==0)
time_seals <- left_join(time_seals,time_seals_failures, by="time")
# calculate number of successful trials (seal present)
time_seals$success <- time_seals$n.x - time_seals$n.y

# add numerical value to time for spline
time_seals$number <- 1:96

time_model <- gam(cbind(success, n.y) ~ s(number, bs = "cc"),
data=time_seals,family=binomial(link="cloglog"))

plot(time_model, xlab="Time", xaxt="n")
label_x <- c ("00:00", "06:00", "12:00", "18:00","00:00")
axis(1, at = c("0", "24", "48", "72", "96"), labels = label_x)
```

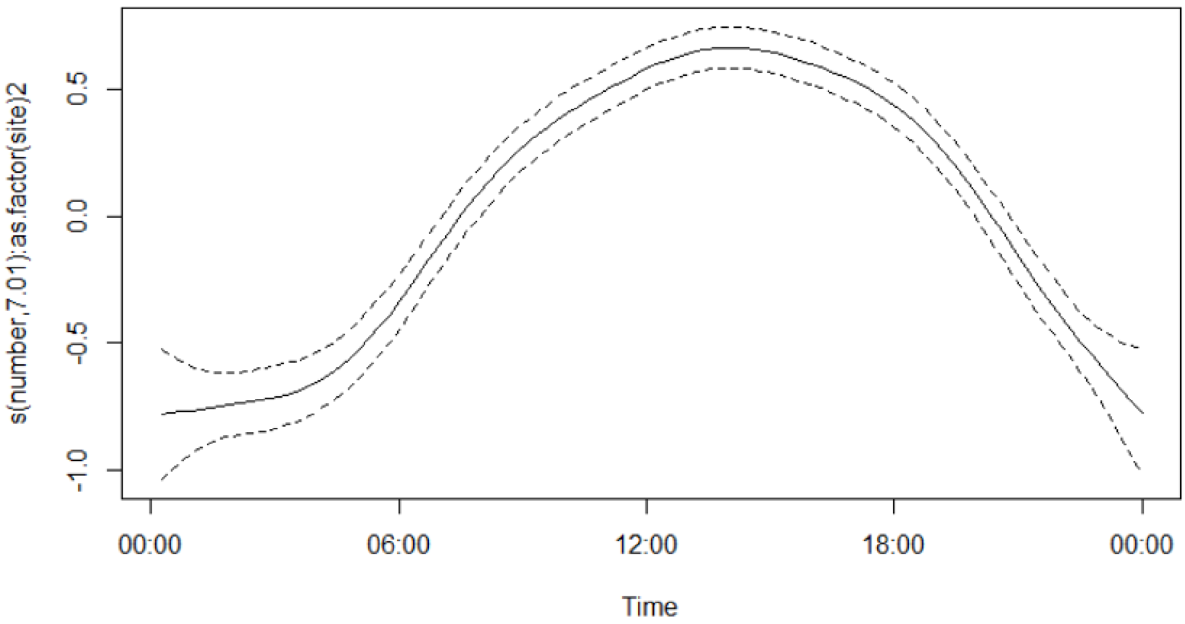


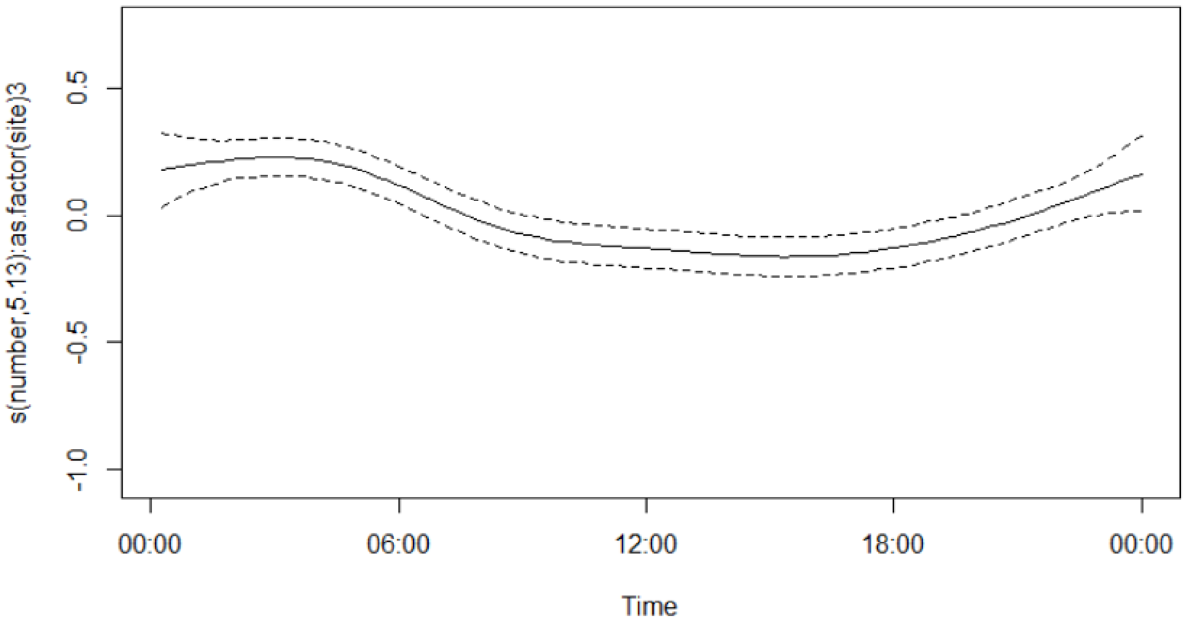

```r
ACF  <-  acf(resid(time_model), lag.max = 96)
# 1 hr 15 minute lag
```



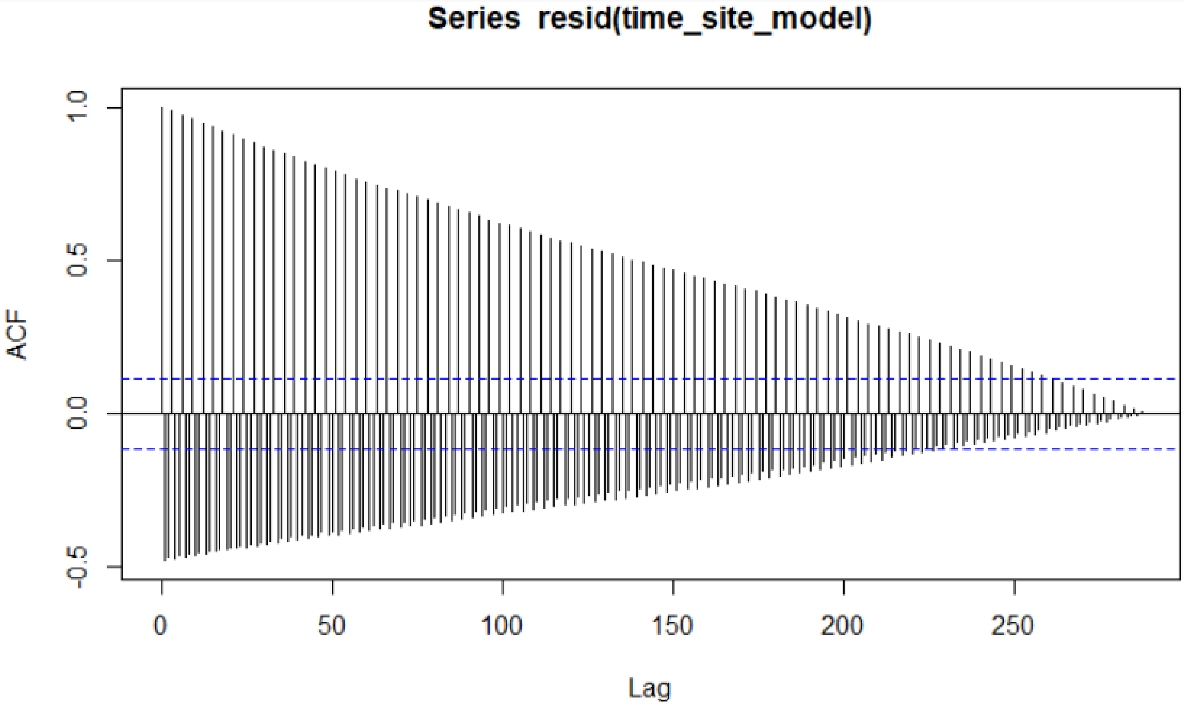

```
# now for each site
time_seals<- monk_seal_data %>% group_by(time,site) %>% tally()

# calculate total number of failures (seal absent) and join to n
time_seals_failures<- monk_seal_data %>% group_by(time,site) %>%
tally(presence==0)
time_seals <- left_join(time_seals,time_seals_failures,
by=c("time","site"))
# calculate number of successful trials (seal present)
time_seals$success <- time_seals$n.x - time_seals$n.y

# add numerical value to time for circular spline
time_seals$number <- rep(1:96, each=3)

time_site_model <- gam(cbind(success, n.y) ~ s(number, bs = "cc",
by=as.factor(site)), data=time_seals,family=binomial(link="cloglog"))

plot(time_site_model,xlab="Time", xaxt="n")
label_x <- c("00:00", "06:00", "12:00", "18:00","00:00")
axis(1, at = c("0", "24", "48", "72", "96"), labels = label_x)
```


s(number,3.41):as.factor(site)1
0.5
0.0
-0.5
-1.0
00:00
06:00
12:00
18:00
00:00
Time


s(number,7.01):as.factor(site)2
0.5
0.0
-0.5
-1.0
00:00
06:00
12:00
18:00
00:00
Time

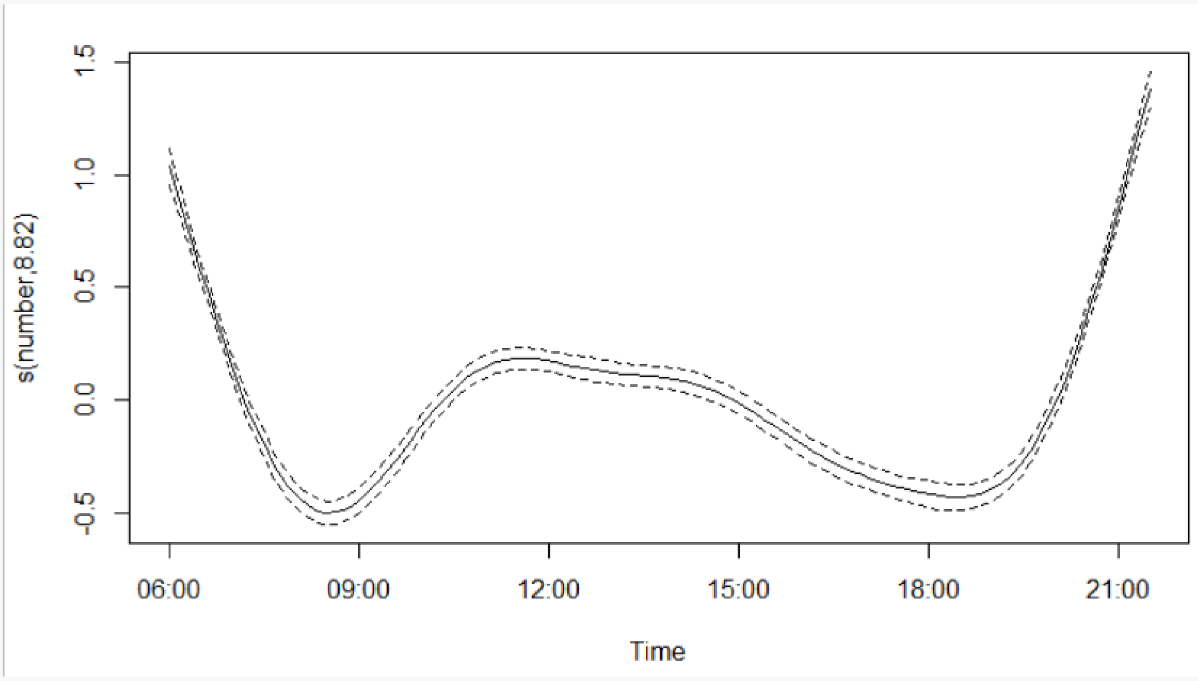



```r
# 96 * 30 = 2880 (30 days)
ACF  <-  acf(resid(time_site_model), lag.max = 2880)
```

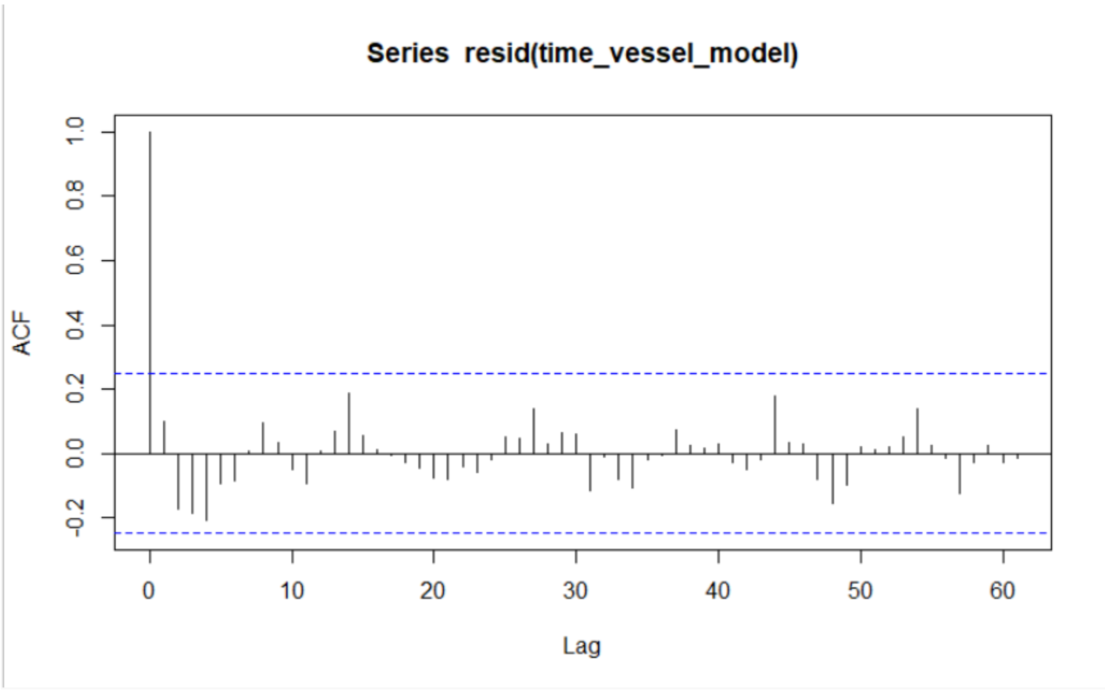



```r
# Repeat for vessel data across 24hrs

# use final_data_all
```

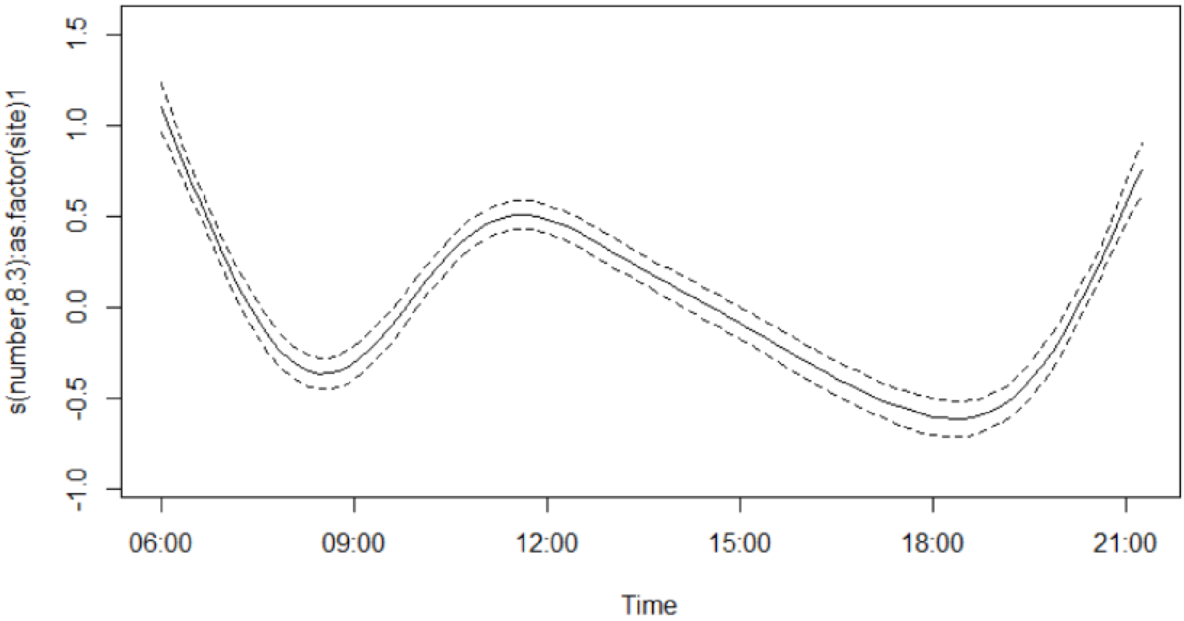
```
# find all data entries per time
time_vessels <- final_data_all %>% group_by(time.y) %>% tally()

# find total failures
time_vessels_failures<- final_data_all %>% group_by(time.y) %>%
tally(boat_total.x==0)

time_vessels <- left_join(time_vessels,time_vessels_failures, by="time.y")
# calculate number of successful trials (seal present)
time_vessels$success <- time_vessels$n.x - time_vessels$n.y

# add numerical value to time for circular spline
time_vessels$number <- 1:96

# remove rows where time is <06:00 or >21:30 (night time)
# 6:00 = 25, 21:30 = 87
time_vessels<-time_vessels[25:87,]

# model vessel success/failure trials by time
time_vessel_model <- gam(cbind(success, n.y) ~ s(number, bs = "cc"),
data=time_vessels,family=binomial(link="cloglog"))

plot(time_model,xlab="Time", xaxt="n")
label_x <- c("06:00", "09:00", "12:00", "15:00", "18:00", "21:00")
axis(1, at = c("25", "37", "49", "61", "73", "85"), labels = label_x)
```



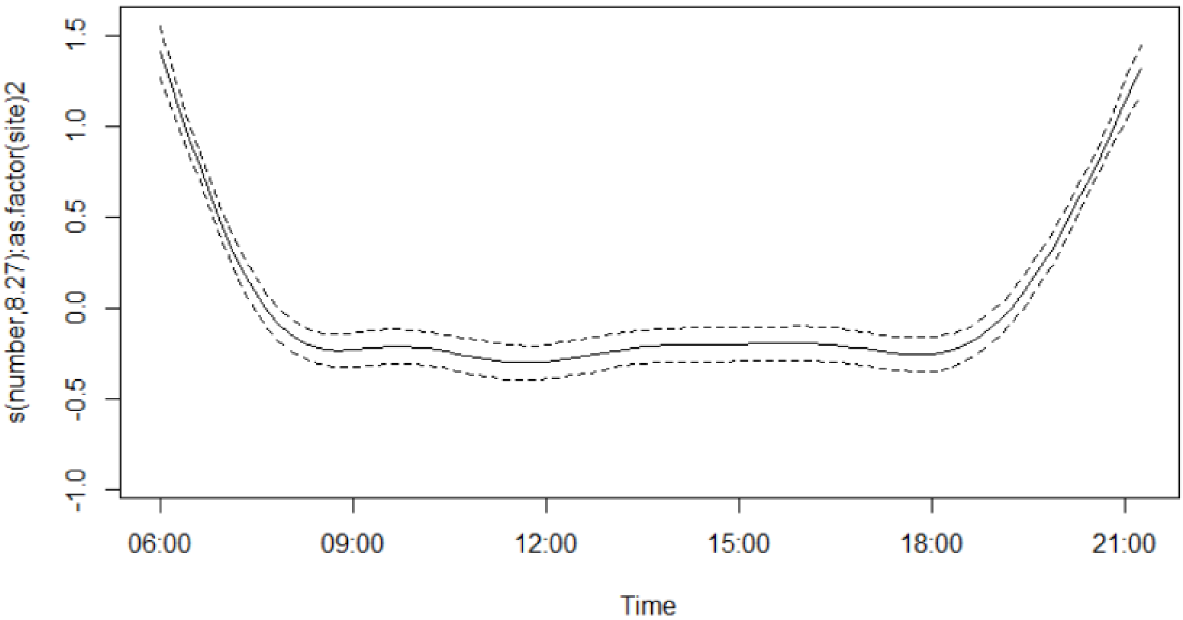

```
# calculate max lag of 1 day
max(time_vessels$number) - min(time_vessels$number)
## [1] 61
ACF  <-  acf(resid(time_vessel_model), lag.max = 61)
```



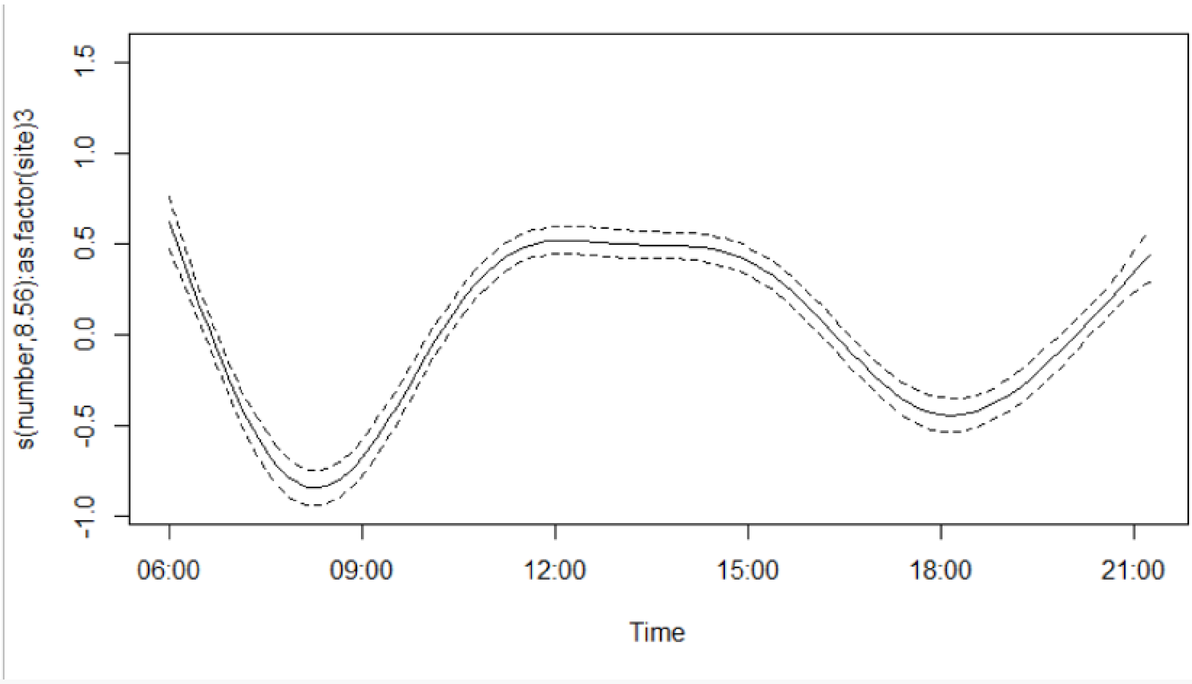

```
# Repeat to include Site
# find all data entries per site for each time
time_vessels <- final_data_all %>% group_by(time.y, site) %>% tally()

# find total failures
time_vessels_failures<- final_data_all %>% group_by(time.y, site) %>%
tally(boat_total.x==0)

time_vessels <- left_join(time_vessels,time_vessels_failures,
by=c("time.y","site"))
# calculate number of successful trials (seal present)
time_vessels$success <- time_vessels$n.x - time_vessels$n.y

# add numerical value to time for circular spline
time_vessels$number <- rep(1:96, each=3)

# remove rows where time is <06:00 or >21:30 (night time)
# 6:00 = 73, 21:30 = 87
time_vessels<-time_vessels[73:258,]

time_vessel_model <- gam(cbind(success, n.y) ~ s(number, bs =
"cc",by=as.factor(site)),
data=time_vessels,family=binomial(link="cloglog"))
plot(time_ vessel_model,xlab="Time", xaxt="n")
label_x <- c("06:00", "09:00", "12:00", "15:00", "18:00", "21:00")
axis(1, at = c("25", "37", "49", "61", "73", "85"), labels = label_x)
```



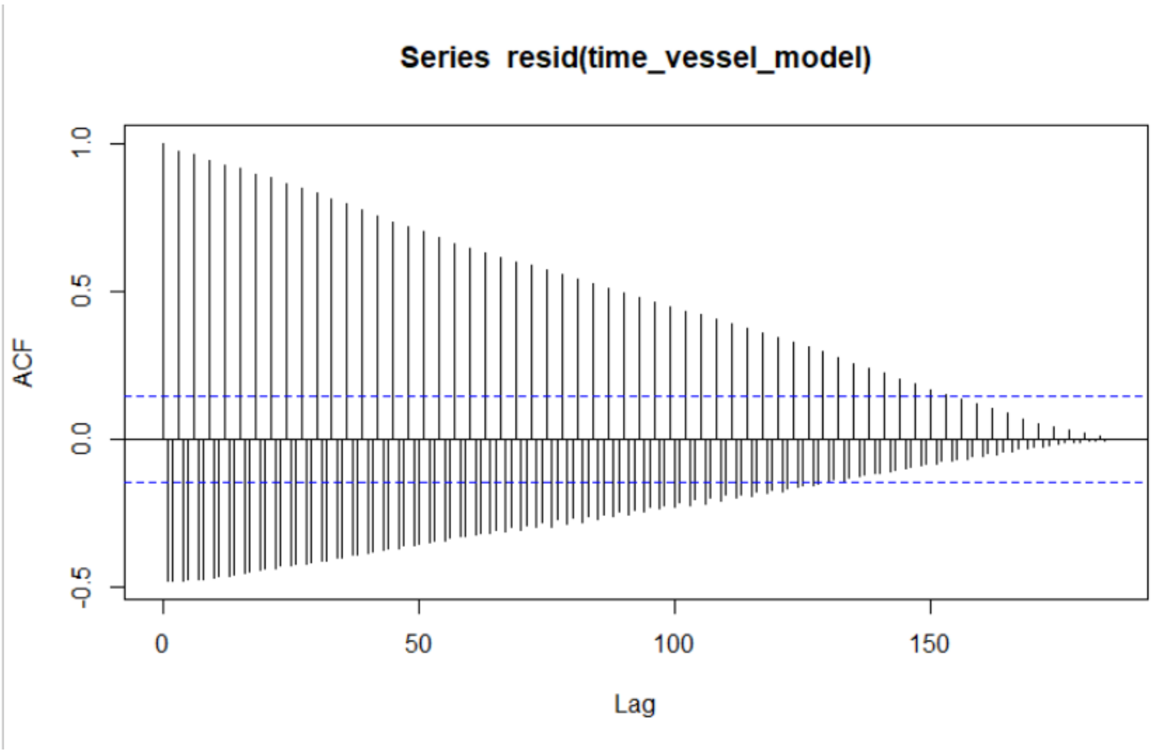
s(number,8.3):as.factor(site)1
1.5
1.0
0.5
0.0
-0.5
-1.0
06:00
09:00
12:00
15:00
18:00
21:00
Time



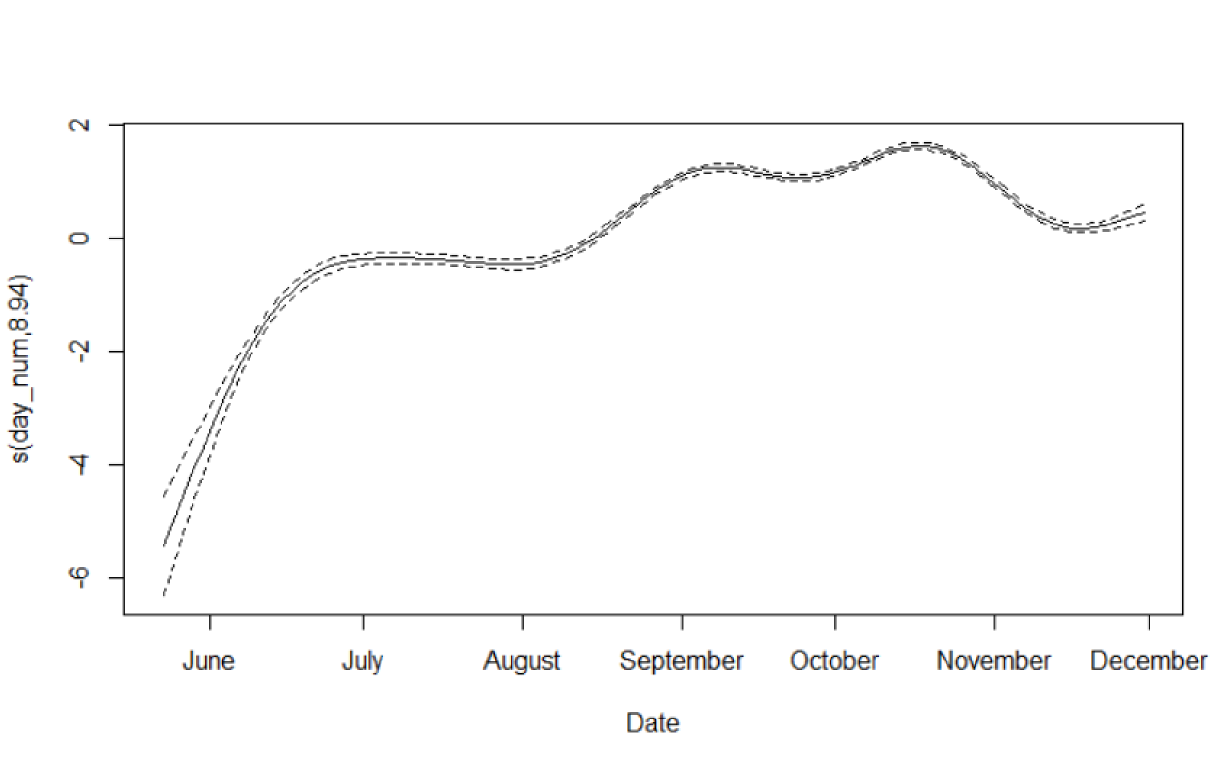
s(number,8.27):as.factor(site)2
1.5
1.0
0.5
0.0
-0.5
-1.0
06:00
09:00
12:00
15:00
18:00
21:00
Time



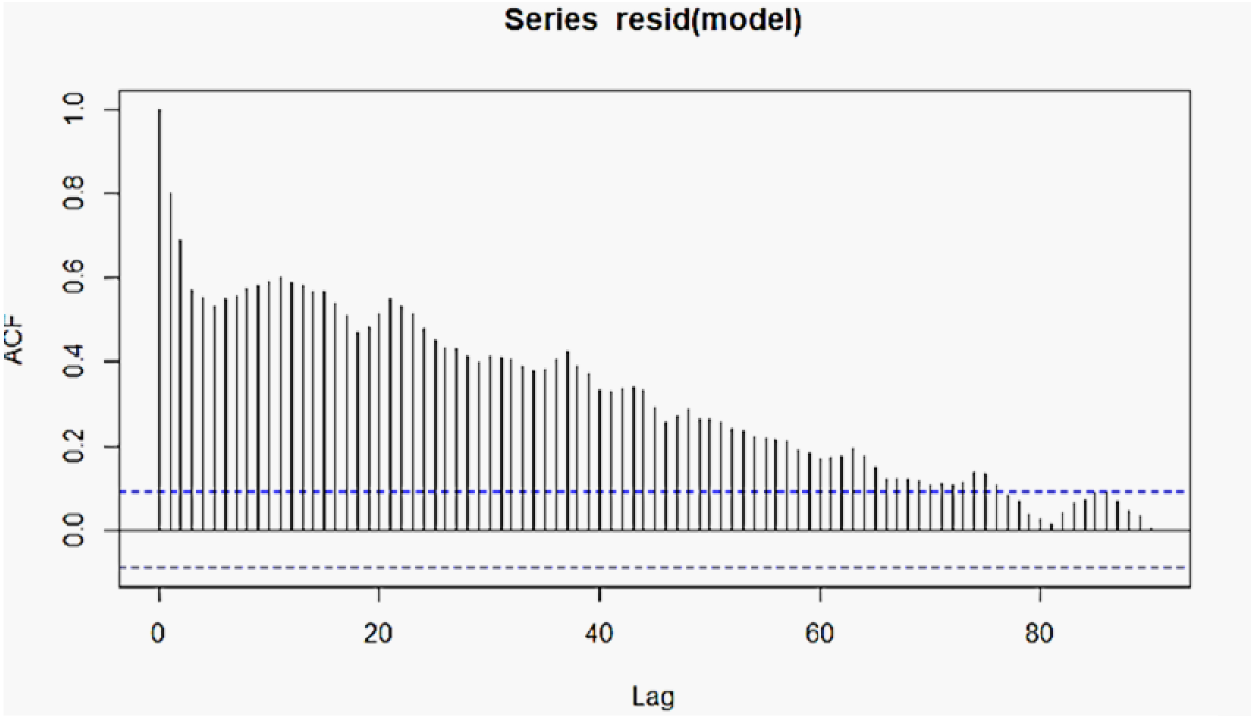

```
# plot autocorrelation
# calculate max lag of 1 day
max(time_vessels$number) - min(time_vessels$number)
# [1] 61

# lag max = 30 days (61*30 = 1830)
ACF  <-  acf(resid(time_vessel_model), lag.max = 1830)
```



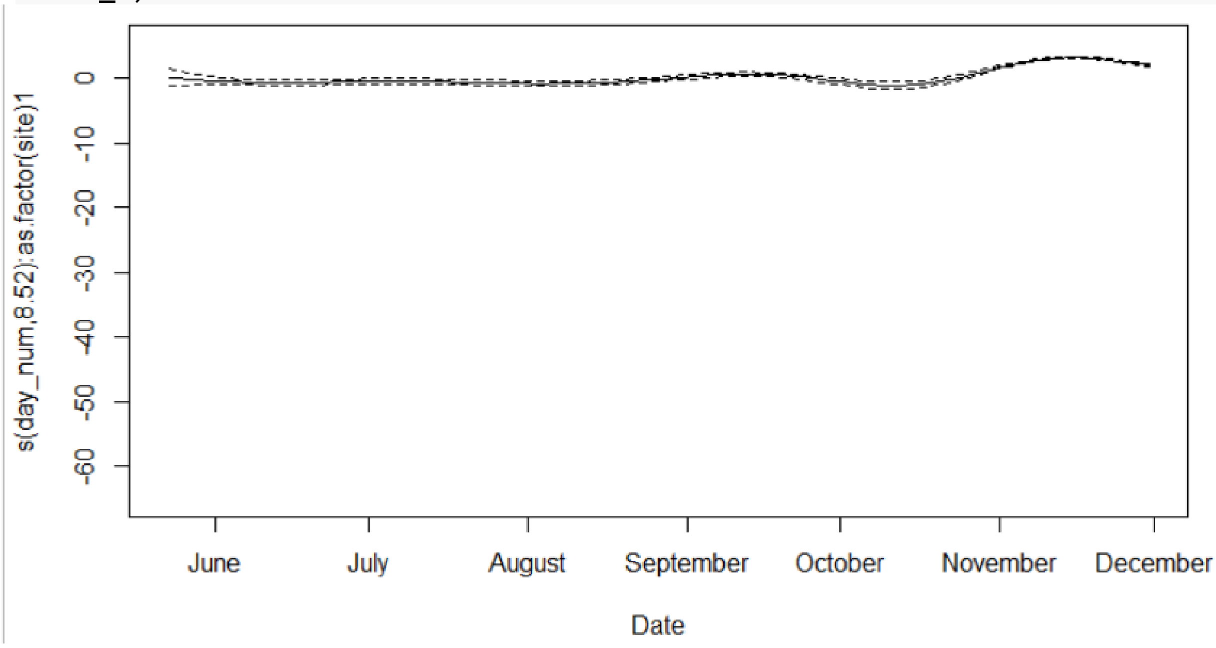
```
# seal data at 24hr intervals
date_seal_model <- gam(cbind(succ_trials, seal_failures) ~ s(day_num, bs =
"cc"), data = final_successes_all, family=binomial(link="cloglog"))

plot(date_seal_model,xlab= "Date", xaxt= "n")
label_x <- c("June", "July", "August", "September", "October", "November",
"December")
axis(1, at = c("152", "182", "213", "244", "274", "305", "335"), labels =
label_x)
```

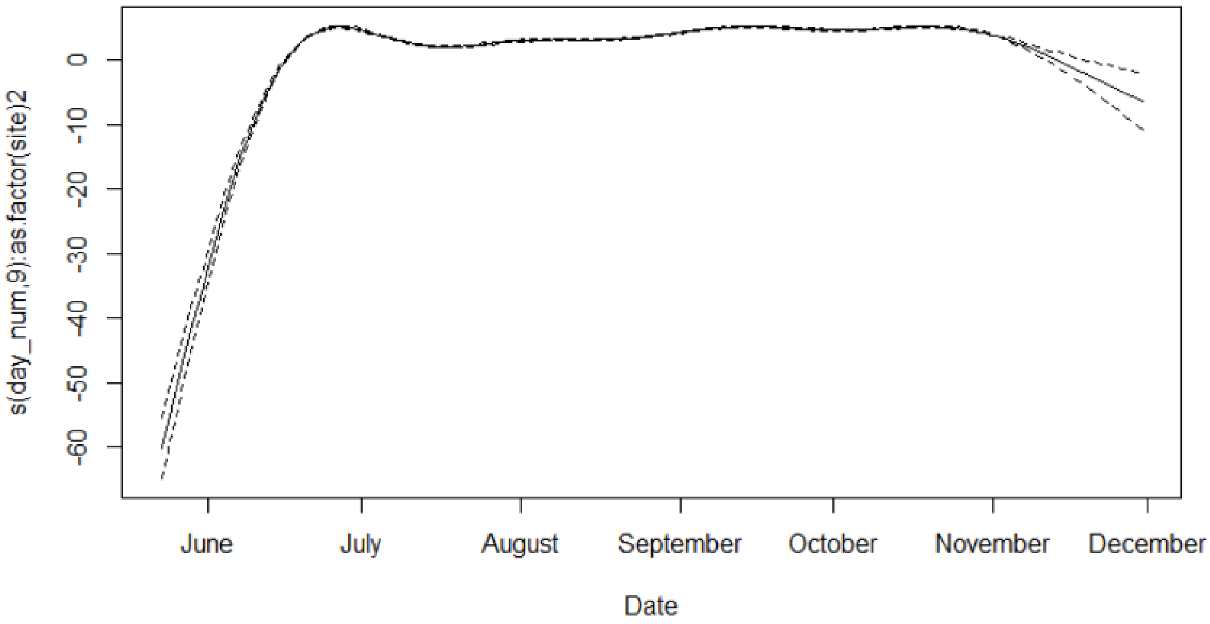


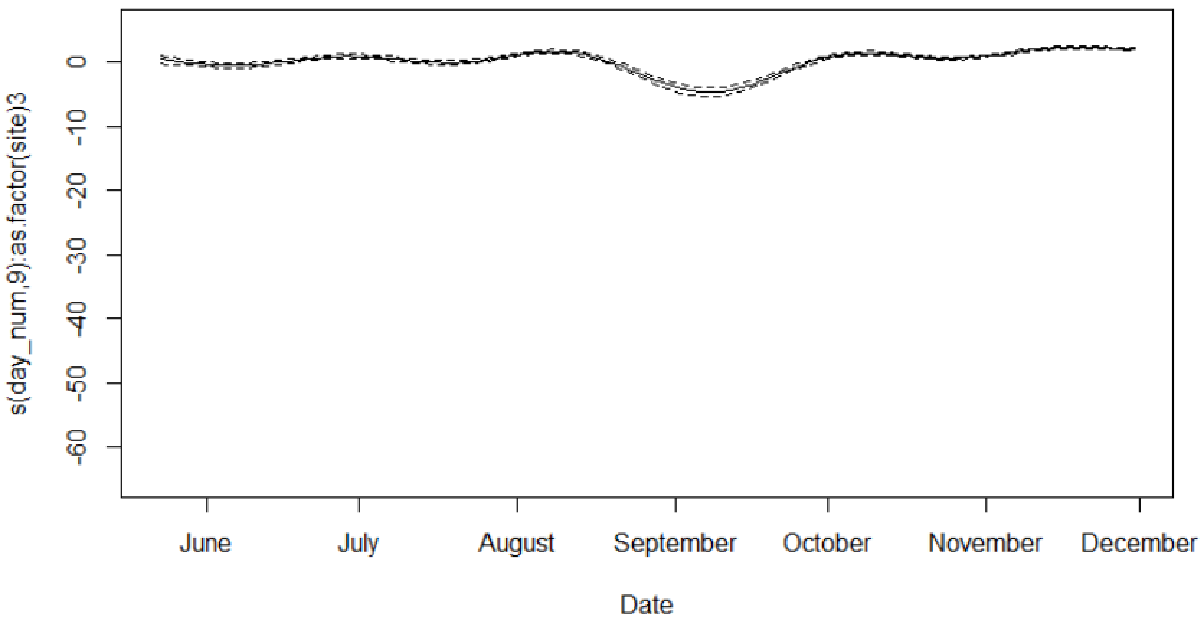

```
# plot autocorrelation
ACF  <-  acf(resid(date_seal_model))
```



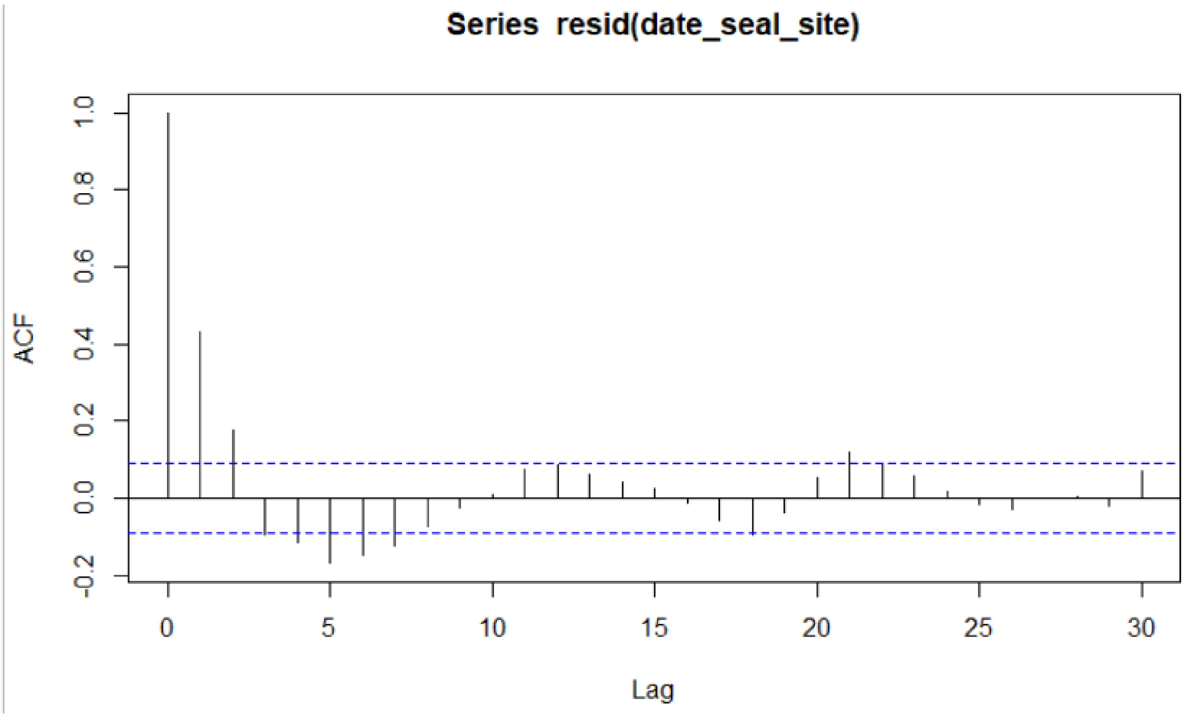

```
#  look at seal data by 24hrs/date and site
date_seal_site <- gam(cbind(succ_trials, seal_failures) ~ s(day_num, bs =
"cc", by= as.factor(site)), data=final_successes_all,
family=binomial(link= "cloglog"))

plot(date_seal_site,xlab="Date", xaxt="n")
label_x <- c("June", "July", "August", "September", "October","November",
"December")
axis(1, at = c("152", "182", "213", "244","274", "305", "335"), labels =
label_x)
```



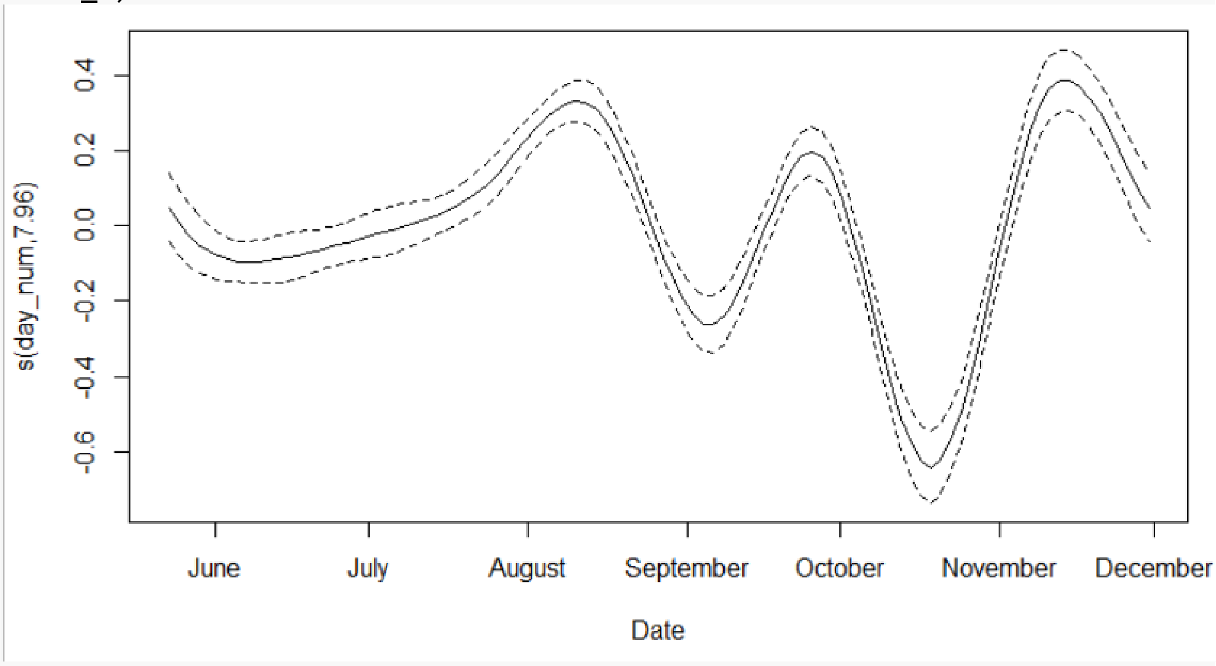




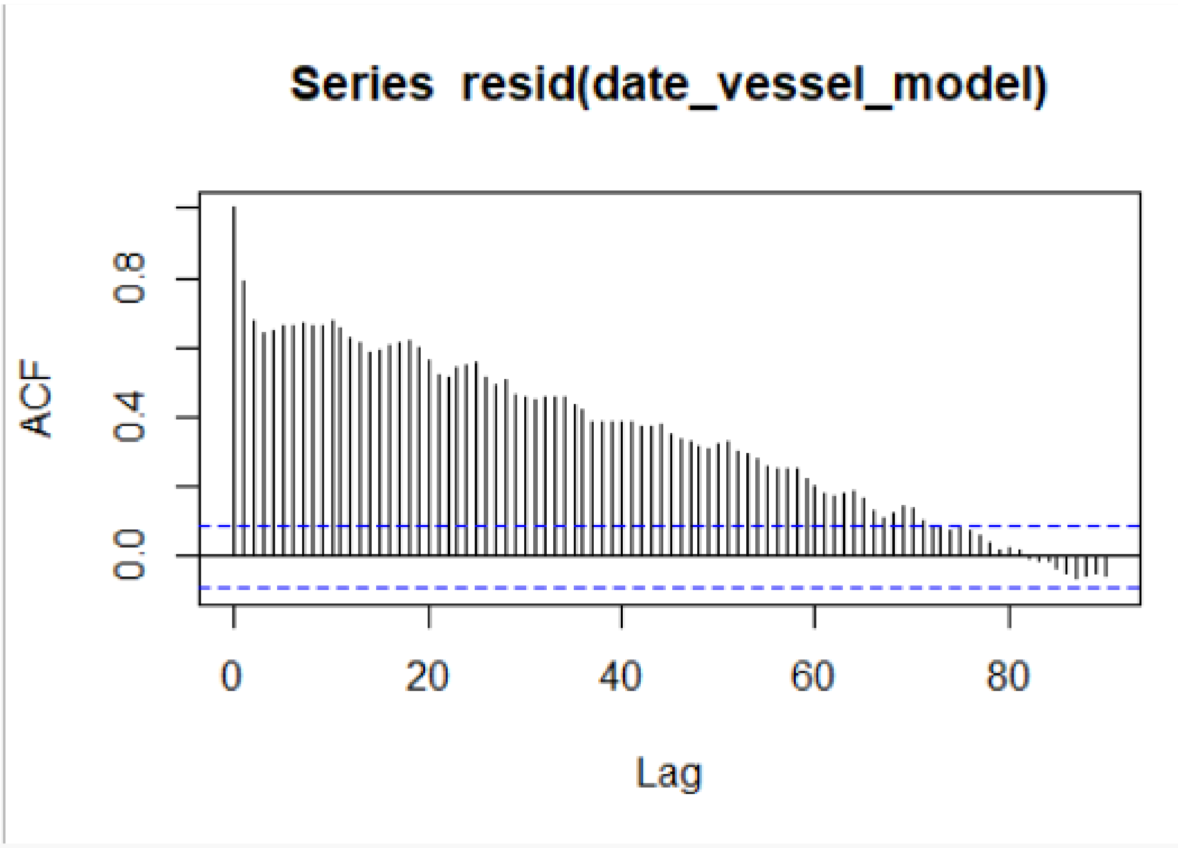

```r
# plot autocorrelation
ACF  <-  acf(resid(date_seal_site), lag.max = 30)
```



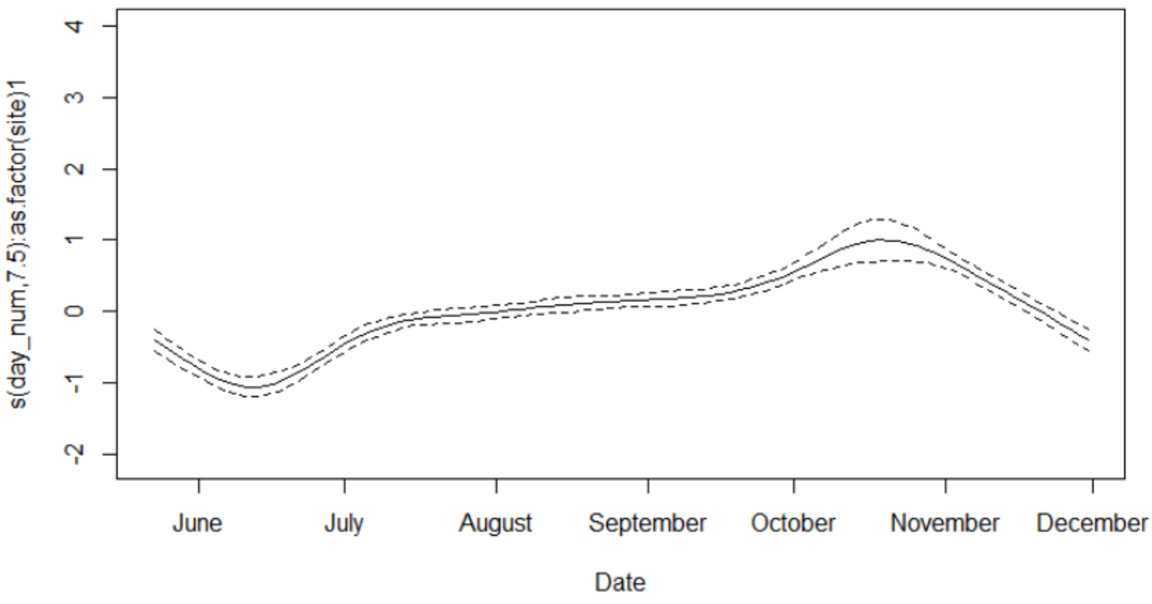

```
# vessel data at 24hr intervals
date_vessel_model <- gam(cbind(boat_p_a, vessel_failures) ~ s(day_num, bs
= "cc"), data = final_successes_all, family=binomial(link="cloglog"))

plot(date_vessel_model,xlab="Time", xaxt="n")
label_x <- c("June", "July", "August", "September", "October","November",
"December")
axis(1, at = c("152", "182", "213", "244","274", "305", "335"), labels =
label_x)
```

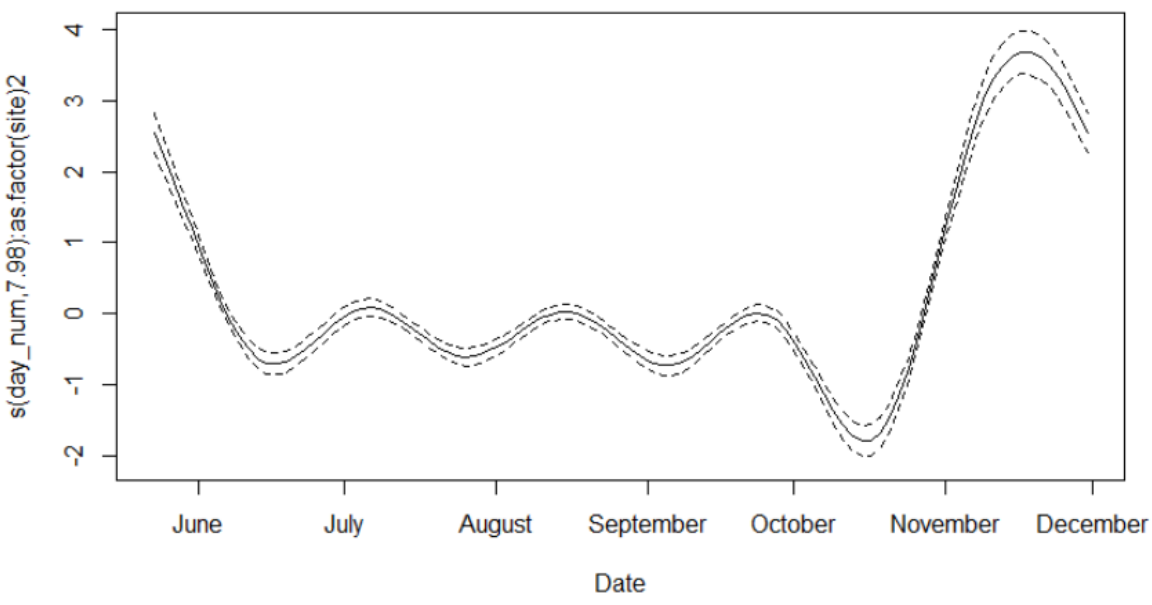


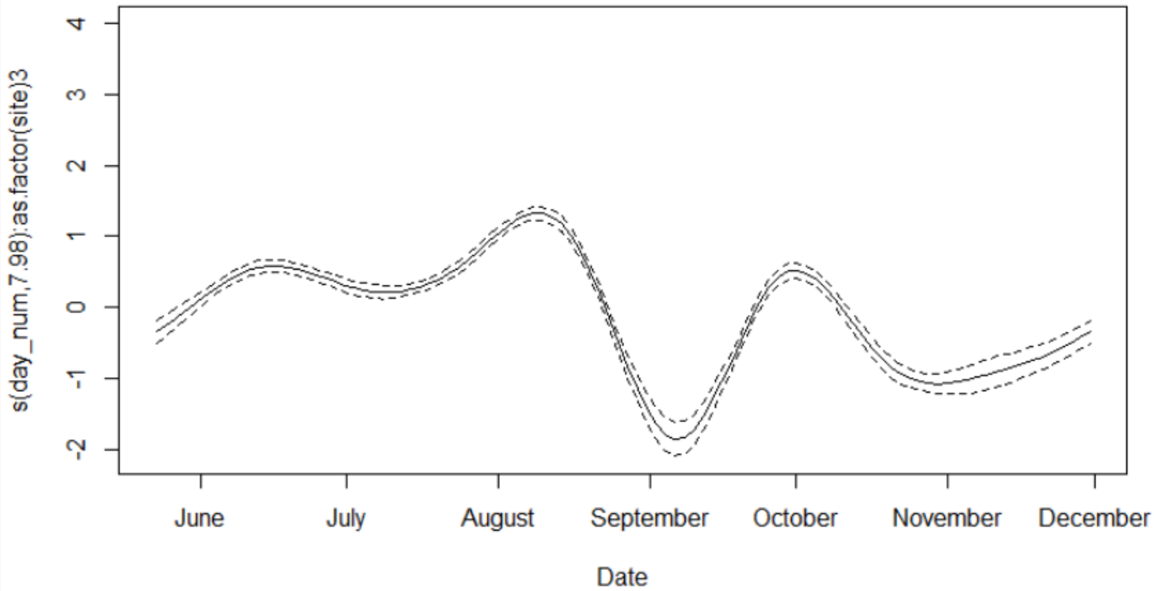

```
# plot autocorrelation
ACF  <-  acf(resid(date_vessel_model), lag.max = 30)
```



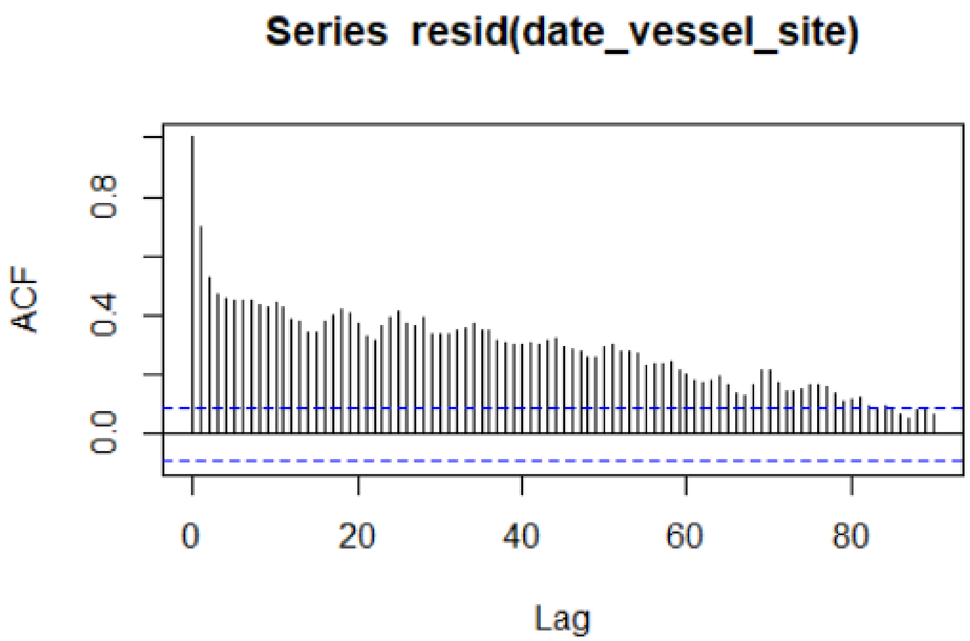

```
#  look at vessel data by 24hrs/date and site

date_vessel_site <- gam(cbind(boat_p_a, vessel_failures) ~ s(day_num, bs =
"cc",by=as.factor(site)), data = final_successes_all,
family=binomial(link="cloglog"))

# plot vessel data by date and site
plot(date_vessel_site, xlab= "Date", xaxt= "n")
label_x <- c("June", "July", "August", "September", "October","November",
"December")
axis(1, at = c("152", "182", "213", "244","274", "305", "335"), labels =
label_x)
```


s(day_num,7.5):as.factor(site)1
-2
-1
0
1
2
3
4
June
July
August
September
October
November
December
Date


s(day_num,7.98):as.factor(site)2
-2
-1
0
1
2
3
4
June
July
August
September
October
November
December
Date




```
# check autocorrelation
ACF  <-  acf(resid(date_vessel_site), lag.max = 90)
```



Convert data to 0s/1s for autocorrelation groupings

Convert data to 0s/1s for autocorrelation groupings

File produced = seasonal = vessel_seal_data_72hrs.csv
(accounts for 3 day autocorrelation within dates)

= diel = all_data_hourly.csv
(accounts for hourly autocorelation within times)

```
# hourly models = seals+time -> 1s/0s, 1 hour intervals (autocorrelation)

# seasonal models = seals+time -> 1s/0s,
# then 72 hour/3 day intervals (autocorrelation)

##########################################################################
#### seasonal data over 72hr period
##########################################################################
###

# load and view data
final_data_all <-
read.csv('E:/BL5599/descriptive_statistics/final_dataset.csv',header=T)
view(final_data_all)
final_data_all$date <- as.Date(final_data_all$date, format= "%d/%m/%Y")

# convert date to day number
doy <- strftime(final_data_all$date, format = "%j")
final_data_all$day_num <- as.numeric(doy)

# min date = 2019-05-23 = day  number 143
# max date = 2019-11-30 = day  number 334

# calculate number of days
334-142
## [1] 192
# calculate number of 72 hour groups
192/3
## [1] 64

# create new dataframe
# create number to identfy each 72hr grouping
data_72hr  <- rep(1:64, each=3)
data_72hr  <- data.frame(data_72hr)
data_72hr$day_num <- as.numeric(143:334)
# add sequence of dates in 1 day intervals
data_72hr$date <- seq(as.Date( "2019-05-23"), as.Date( "2019-11-30"),
"day")
# join 72 hour group number to main data
final_data_all <- left_join(final_data_all, data_72hr, by=("date"))

# subset data into separate sites
```

```
data_72hr_1 <- subset(final_data_all, site == "1")
data_72hr_2 <- subset(final_data_all, site == "2")
data_72hr_3 <- subset(final_data_all, site == "3")

# calculate total number of images
seals_72count1 <- dplyr::count(data_72hr_1, data_72hr)
seals_72count2 <- dplyr::count(data_72hr_2, data_72hr)
seals_72count3 <- dplyr::count(data_72hr_3, data_72hr)

# calculate total presence for each 72hr group
data_72hr_1 <- data_72hr_1 %>% group_by(data_72hr) %>%
tally(presence=="1")
data_72hr_2 <- data_72hr_2 %>% group_by(data_72hr) %>%
tally(presence=="1")
data_72hr_3 <- data_72hr_3 %>% group_by(data_72hr) %>%
tally(presence=="1")

# give 1 value if total count of presence is larger or equal to 1
data_72hr_1$presence_new <- ifelse(data_72hr_1$n >=1 , 1, 0)
data_72hr_2$presence_new <- ifelse(data_72hr_2$n >=1 , 1, 0)
data_72hr_3$presence_new <- ifelse(data_72hr_3$n >=1 , 1, 0)

# assign site
data_72hr_1$site <- 1
data_72hr_2$site <- 2
data_72hr_3$site <- 3

# join dataframes
data_72hr_1 <- left_join(data_72hr_1, seals_72count1, by= "data_72hr")
data_72hr_2 <- left_join(data_72hr_2, seals_72count2, by= "data_72hr")
data_72hr_3 <- left_join(data_72hr_3, seals_72count3, by= "data_72hr")

# rejoin seperate sites
data_72hr <- rbind(data_72hr_1,data_72hr_2,data_72hr_3)

# new presence/absence values assigned
# join 72 hour group number to main data
final_data_all <- left_join(final_data_all, data_72hr,
by=c("data_72hr","site"))

# create csv
write.csv(final_data_all,"E:/BL5599/seals_72hrs.csv", row.names = FALSE)

# now need to calculate vessel index over same 72 hour period

# subset data into separate sites
boats_72hr_1 <- subset(final_data_all, site == "1")
boats_72hr_2 <- subset(final_data_all, site == "2")
boats_72hr_3 <- subset(final_data_all, site == "3")


# calculate vessel index for each 72hr group

# create new column for total boat count
```

```
boats_72hr_1$boat_total <- boats_72hr_1$sail.boat +
boats_72hr_1$motor.boat + boats_72hr_1$kayak
boats_72hr_2$boat_total <- boats_72hr_2$sail.boat +
boats_72hr_2$motor.boat + boats_72hr_2$kayak
boats_72hr_3$boat_total <- boats_72hr_3$sail.boat +
boats_72hr_3$motor.boat + boats_72hr_3$kayak

# calculate vessel count average over 72 hours for all data at each site -
correct
boats_72sum1 <- aggregate(boats_total_15min ~ data_72hr, boats_72hr_1,
sum)
boats_72sum2 <- aggregate(boats_total_15min ~ data_72hr, boats_72hr_2,
sum)
boats_72sum3 <- aggregate(boats_total_15min ~ data_72hr, boats_72hr_3,
sum)

# count data entries per 72 hours
boats_72count1 <- dplyr::count(boats_72hr_1, data_72hr)
boats_72count2 <- dplyr::count(boats_72hr_2, data_72hr)
boats_72count3 <- dplyr::count(boats_72hr_3, data_72hr)

# merges total sum of boats per 72 hours with number of data entries in
72hrs
boats_72date1 <- merge(boats_72sum1, boats_72count1, by = "data_72hr", all
= TRUE)
boats_72date2 <- merge(boats_72sum2, boats_72count2, by = "data_72hr", all
= TRUE)
boats_72date3 <- merge(boats_72sum3, boats_72count3, by = "data_72hr", all
= TRUE)

# site
boats_72date1$site <- 1
boats_72date2$site <- 2
boats_72date3$site <- 3

# calculates average number of boats over 72 hour period as index of
vessel presence
boats_72date1$boat_index <-
boats_72date1$boats_total_15min/boats_72date1$n
boats_72date2$boat_index <-
boats_72date2$boats_total_15min/boats_72date2$n
boats_72date3$boat_index <-
boats_72date3$boats_total_15min/boats_72date3$n

# add all 3 sites together
boats_72 <- rbind(boats_72date1, boats_72date2, boats_72date3)

# merges total sum of boats per 72 hours with monk seal presence over
72hrs
all_data_72 <- left_join(data_72hr, boats_72, by = c("data_72hr","site"),
all = TRUE)

# create csv
```

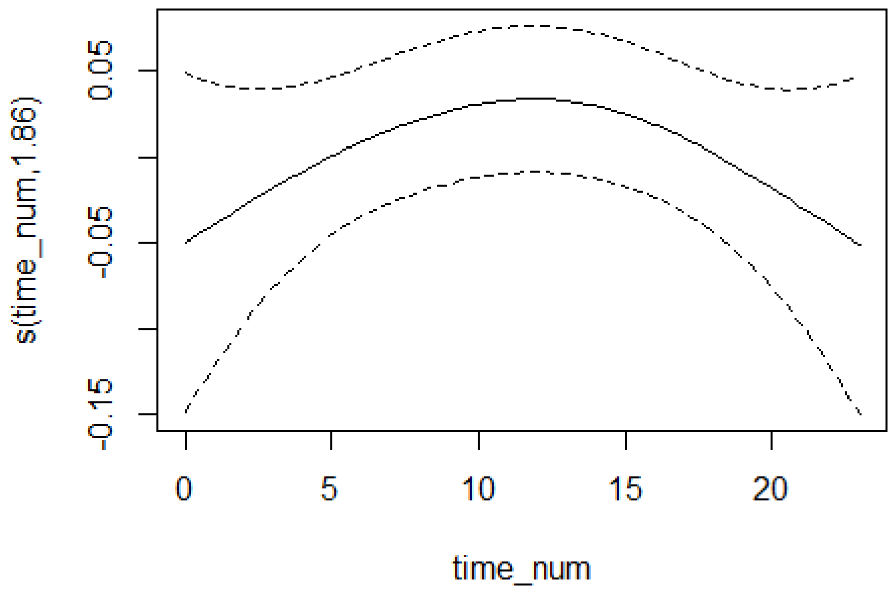
```
write.csv(all_data_72,"E:/BL5599/vessel_seal_data_72hrs.csv", row.names =
FALSE)
########################################################################
###
# time intervals = occur on seperate days -> independent trials unlike
# grouping trials by date
# autocorrelation in seal data = 1hr 15 mins
# group into hourly intervals, convert presence/absence trials into 0s/1s
########################################################################
###

# load seal data
data <-
read.csv('E:/BL5599/descriptive_statistics/final_dataset.csv',header=T)

# monk seals with time - need to calculate total number of monk seal
occurrences for each time period
# separate into sites
all_seals1 <- subset(data, site == "1")
all_seals2 <- subset(data, site == "2")
all_seals3 <- subset(data, site == "3")

# calculate total number of data entries per time (n)
time_seals1 <- all_seals1 %>% group_by(time) %>% tally()
time_seals2 <- all_seals2 %>% group_by(time) %>% tally()
time_seals3 <- all_seals3 %>% group_by(time) %>% tally()

# group into 1 hour intervals by representative time number
time_seals1$time_num  <- rep(0:23, each=4)
time_seals2$time_num  <- rep(0:23, each=4)
time_seals3$time_num  <- rep(0:23, each=4)

# calculate total number of data entries per 1 hour interval (n)
n_time_seals1 <- aggregate(n ~ time_num, time_seals1, sum)
n_time_seals2 <- aggregate(n ~ time_num, time_seals2, sum)
n_time_seals3 <- aggregate(n ~ time_num, time_seals3, sum)

# add total data entries per hour to each time entry (need each entry to
aggregate presence)
time_seals1 <- left_join(time_seals1, n_time_seals1, by = ("time_num"))
time_seals2 <- left_join(time_seals2, n_time_seals2, by = ("time_num"))
time_seals3 <- left_join(time_seals3, n_time_seals3, by = ("time_num"))

# add site
time_seals1$site  <- 1
time_seals2$site  <- 2
time_seals3$site  <- 3

# rbind into one dataframe
seals_hourly <- rbind(time_seals1, time_seals2, time_seals3)

# merge
all_seals_hourly <- merge(seals_hourly, data, by = c("time","site"), all =
TRUE)
```

```
# subset data into separate sites
data_72hr_1 <- subset(final_data_all, site == "1")
data_72hr_2 <- subset(final_data_all, site == "2")
data_72hr_3 <- subset(final_data_all, site == "3")

# calculate total presence for each 72hr group
time_seals1 <- all_seals1 %>% group_by(data_72hr) %>% tally(presence=="1")
time_seals2 <- all_seals2 %>% group_by(data_72hr) %>% tally(presence=="1")
time_seals3 <- all_seals3 %>% group_by(data_72hr) %>% tally(presence=="1")

# rbind into one dataframe
seals_hourly <- rbind(time_seals3, time_seals3, time_seals3)

# merge presence with hourly grouping
all_seals_hourly <- merge(seals_hourly, data, by = c("time", "site"), all
= TRUE)

# if presence >= 1, assign new presence value of 1
all_seals_hourly <- all_seals_hourly %>% group_by(date, site, time_num)
%>% tally(presence=="1")

# give 1 value if total count of presence is larger or equal to 1
# max value = 4 as 4 images per hour
all_seals_hourly$presence_new <- ifelse(all_seals_hourly$n >=1 , 1, 0)

# save as csv and create copy of dataframe
write.csv(all_seals_hourly,"E:/BL5599/seal_data_hourly.csv", row.names =
FALSE)
all_seals_hourly_COPY <- all_seals_hourly

###########################################################################
###
# calculate new vessel index for each 4x 15 min block per site

# revert back to main data merged with hourly grouping numbers
all_vessels <- merge(seals_hourly, data, by = c("time","site"), all =
TRUE)

write.csv(all_vessels,"E:/BL5599/all_vessels_hourly.csv", row.names =
FALSE)

# split into site
all_vessels1 <- subset(all_vessels, site == "1")
all_vessels2 <- subset(all_vessels, site == "2")
all_vessels3 <- subset(all_vessels, site == "3")

# calculate vessel count average at each hourly group for all data at each
site and date
boats_hrsum1 <- all_vessels1 %>% group_by(date, time_num) %>%
tally(boats_total_15min)
boats_hrsum2 <- all_vessels2 %>% group_by(date, time_num) %>%
tally(boats_total_15min)
```

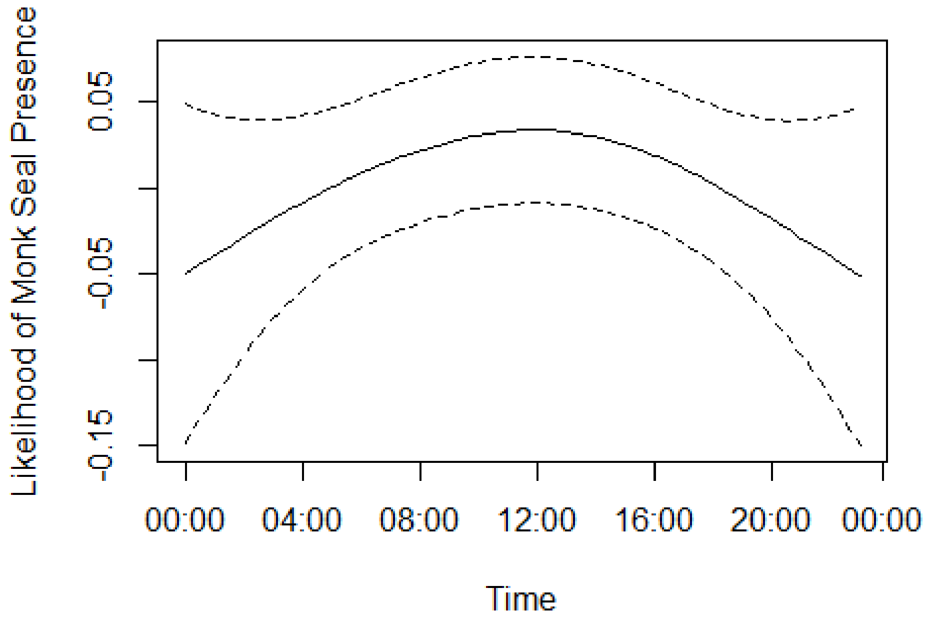
```r
boats_hrsum3 <- all_vessels3 %>% group_by(date, time_num) %>%
tally(boats_total_15min)

# calculate total number of data entries per 1 hour interval (max = 4)
boats_hrcount1 <- all_vessels1 %>% group_by(date, time_num) %>% tally()
boats_hrcount2 <- all_vessels2 %>% group_by(date, time_num) %>% tally()
boats_hrcount3 <- all_vessels3 %>% group_by(date, time_num) %>% tally()

# join count of data entries and total sum of vessels for each hourly
block
boats_hrsum1 <- left_join(boats_hrsum1, boats_hrcount1,
by=c("date","time_num"))
boats_hrsum2 <- left_join(boats_hrsum2, boats_hrcount2,
by=c("date","time_num"))
boats_hrsum3 <- left_join(boats_hrsum3, boats_hrcount3,
by=c("date","time_num"))

# calculate hourly vessel_index
boats_hrsum1$vessel_index_hr <- boats_hrsum1$n.x / boats_hrsum1$n.y
boats_hrsum2$vessel_index_hr <- boats_hrsum2$n.x / boats_hrsum2$n.y
boats_hrsum3$vessel_index_hr <- boats_hrsum3$n.x / boats_hrsum3$n.y

# create new vessel presence/absence 0s/1s column for hourly groups
boats_hrsum1$v_presence <- ifelse(boats_hrsum1$n.x >=1 , 1, 0)
boats_hrsum2$v_presence <- ifelse(boats_hrsum2$n.x >=1 , 1, 0)
boats_hrsum3$v_presence <- ifelse(boats_hrsum3$n.x >=1 , 1, 0)

# add site
boats_hrsum1$site <- 1
boats_hrsum2$site <- 2
boats_hrsum3$site <- 3

# join sites into one dataframe
all_vessels_hourly <- rbind(boats_hrsum1, boats_hrsum2, boats_hrsum3)

# save as csv
write.csv(all_vessels_hourly, "E:/BL5599/vessel_data_hourly.csv",
row.names = FALSE)

# combine hourly seal data with hourly vessel data
all_data_hourly <- left_join(all_seals_hourly, all_vessels_hourly,
by=c("site","date","time_num"))

# save as csv
write.csv(all_data_hourly,"E:/BL5599/all_data_hourly.csv", row.names =
FALSE)
```

Examine autocorrelation for seals and vessels over time and date using new grouped data

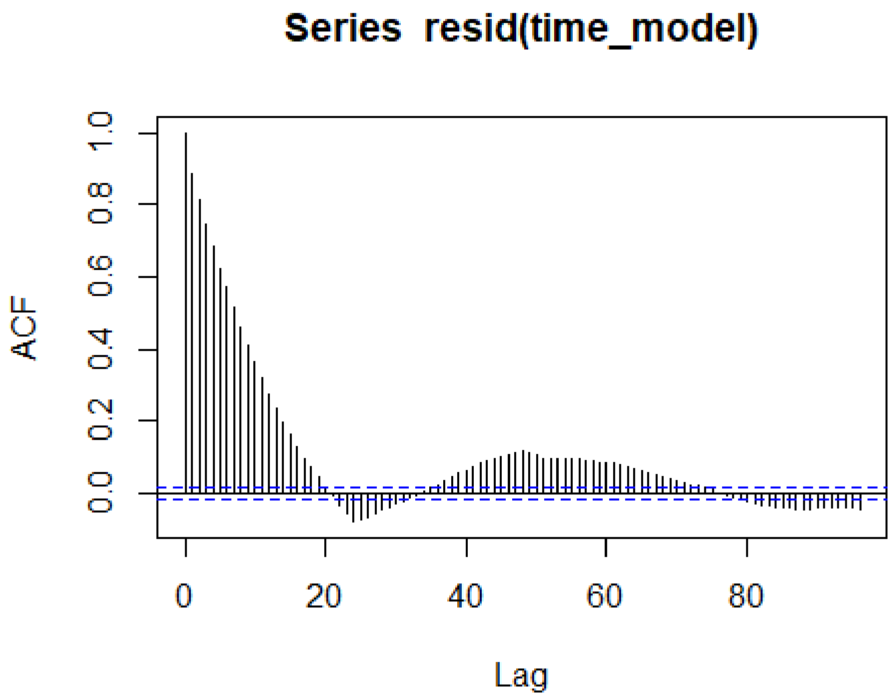
```
library(mgcv)

# Load date
all_data_72 <- read.csv('E:/BL5599/vessel_seal_data_72hrs.csv', header =
T)
all_data_hourly <- read.csv('E:/BL5599/all_data_hourly.csv', header = T)

time_model <- gam(presence_new ~ s(time_num),
data=all_data_hourly,family=binomial(link="cloglog"))
plot(time_model)
```




```
# plot with new axis and labels
plot(time_model, xlab="Time",ylab="Likelihood of Monk Seal Presence",
xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```

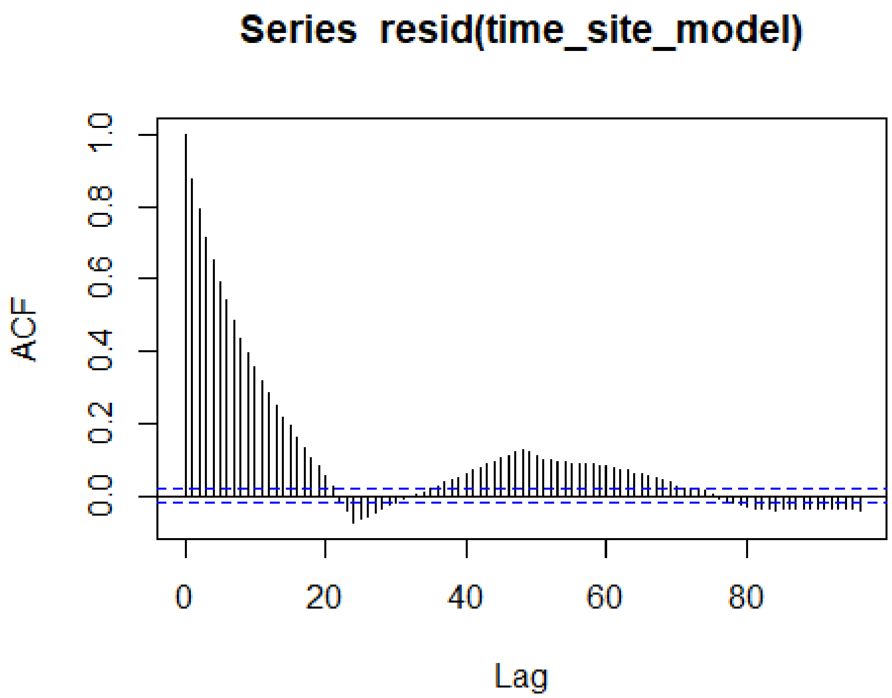



```
# check autocorrelation
ACF  <-  acf(resid(time_model), lag.max = 96)
```




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

# ACF plot shows autocorrelation threshold <24 hrs (lag = ~20 hours)
```

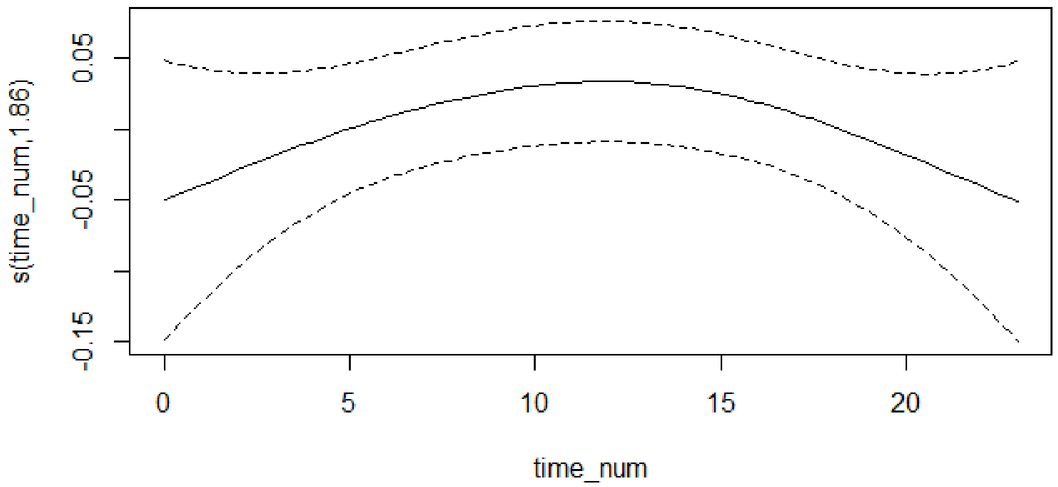
```
###########################################################################
###

# Look at seal autocorrelation within 24hr period by site, using hourly
data
time_site_model <- gam(presence_new ~ s(time_num, by = as.factor(site)),
data = all_data_hourly, family=binomial(link="cloglog"))

# plot with new axes and labels and save as jpeg images
jpeg("E:/BL5599/high_res_plots/seals_hourly_site1.jpg", width = 6, height
= 4, units = 'in', res = 600)
plot(time_site_model, select = 1, main = "Site 1",
xlab="Time",ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/seals_hourly_site2.jpg", width = 6, height
= 4, units = 'in', res = 600)
plot(time_site_model, select = 2, main = "Site 2",
xlab="Time",ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/seals_hourly_site3.jpg", width = 6, height
= 4, units = 'in', res = 600)
plot(time_site_model, select = 3, main = "Site 3",
xlab="Time",ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
dev.off()

## png
##   2

# check autocorrelation
ACF  <-  acf(resid(time_site_model), lag.max = 96)
```



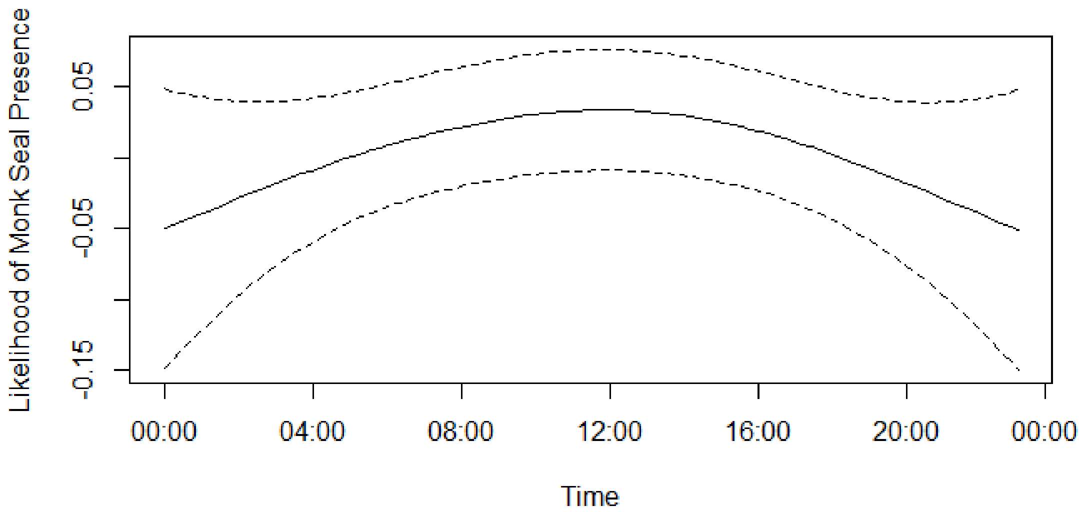

```
# 20 hour lag
```

Examine autocorrelation for seals and vessels over time and date using new grouped data

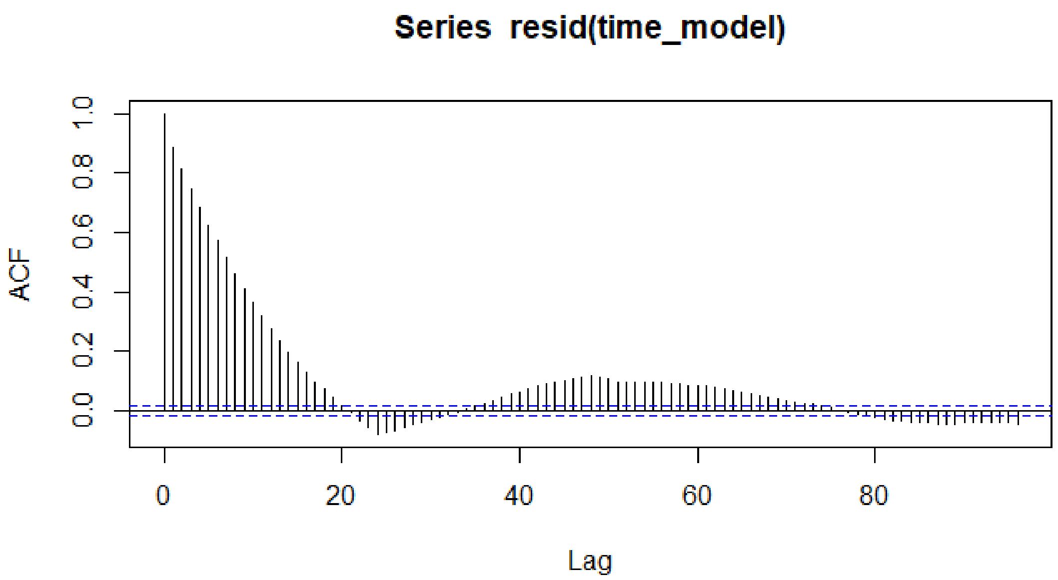
```
library(mgcv)

# Load data
all_data_72 <- read.csv('E:/BL5599/vessel_seal_data_72hrs.csv', header =
T)
all_data_hourly <- read.csv('E:/BL5599/all_data_hourly.csv', header = T)

time_model <- gam(presence_new ~ s(time_num),
data=all_data_hourly,family=binomial(link="cloglog"))
plot(time_model)
```



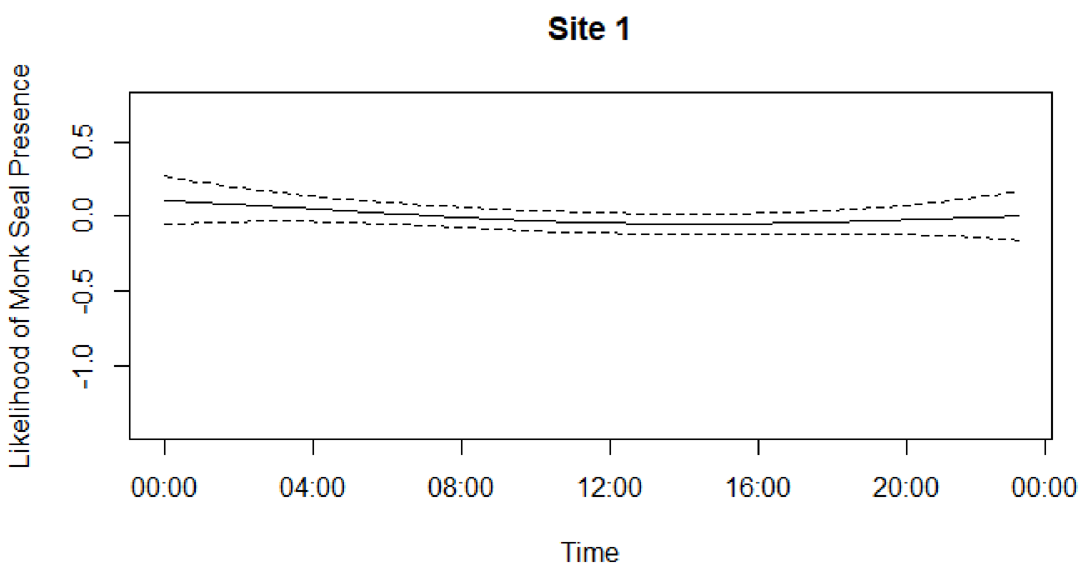

```
# plot with new axis and labels
plot(time_model, xlab="Time",ylab="Likelihood of Monk Seal Presence",
xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```



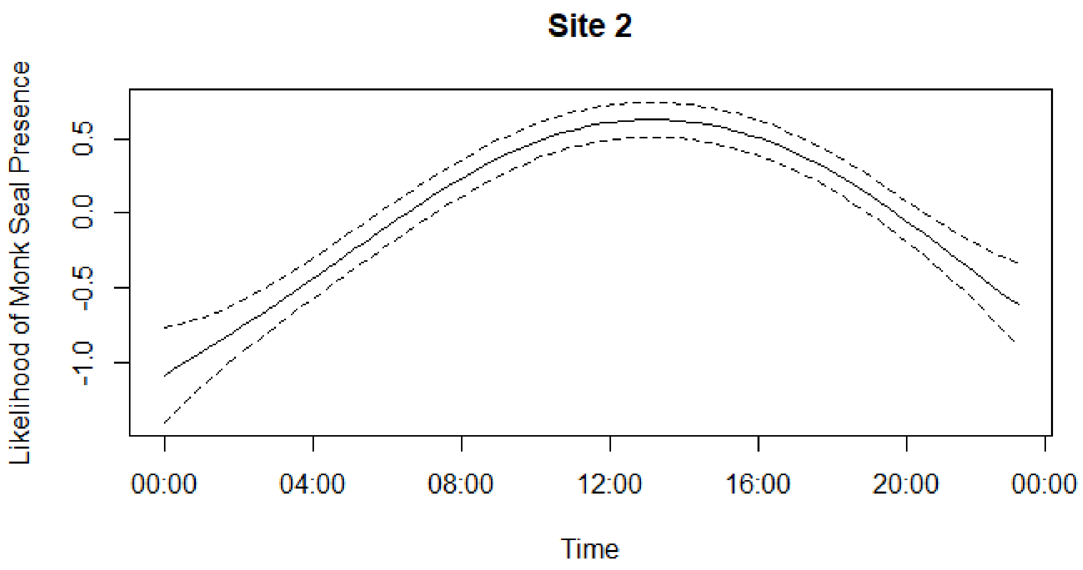

```
# check autocorrelation
ACF  <-  acf(resid(time_model), lag.max = 96)
```



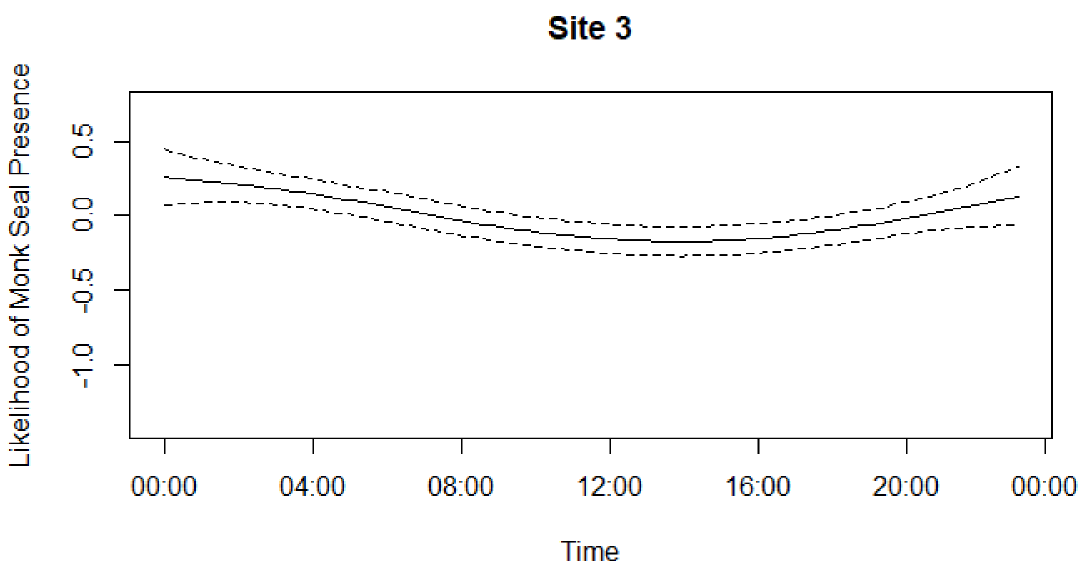

```
################################################################################
###

# Look at seal autocorrelation within 24hr period by site, using hourly
data
time_site_model <- gam(presence_new ~ s(time_num, by = as.factor(site)),
data = all_data_hourly, family=binomial(link="cloglog"))

# plot with new axes and labels
plot(time_site_model, select = 1, main = "Site 1",
xlab="Time",ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```



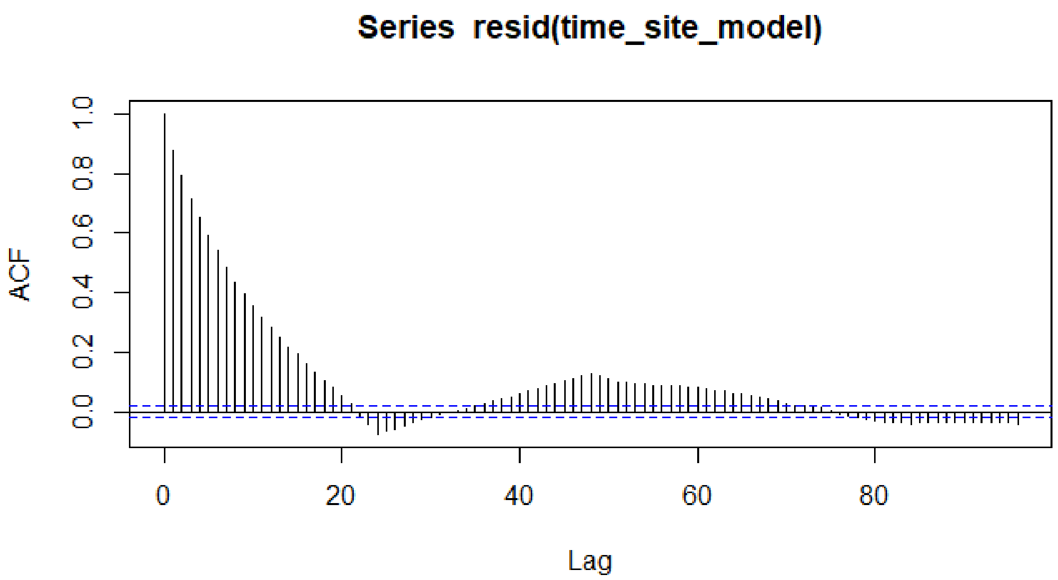
```
plot(time_site_model, select = 2, main = "Site 2",
xlab="Time",ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```



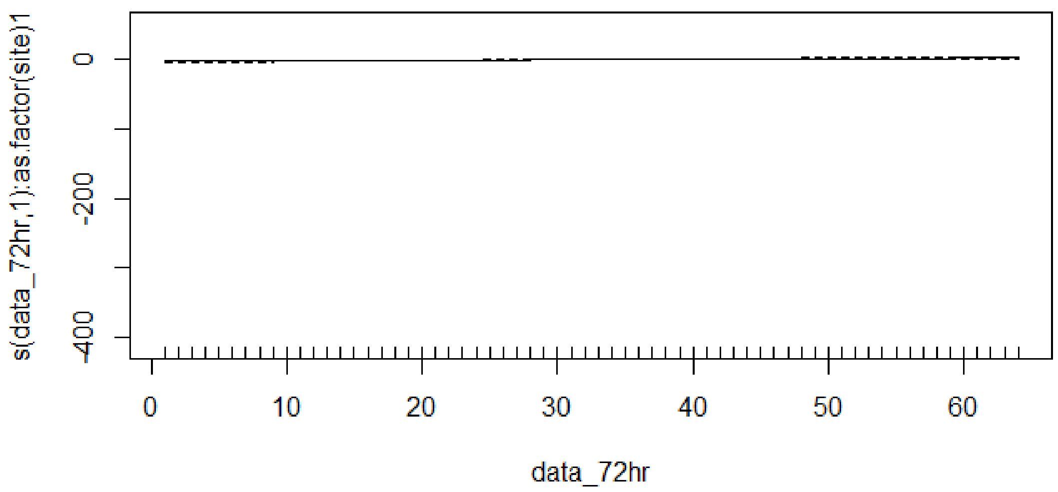

```
plot(time_site_model, select = 3, main = "Site 3",
xlab="Time",ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00", "20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```



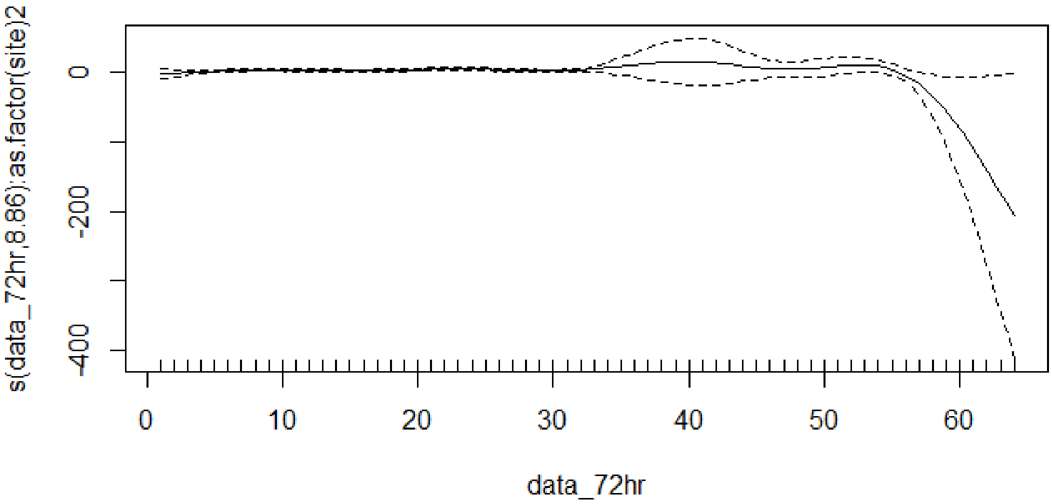

```
# check autocorrelation
ACF  <-  acf(resid(time_site_model), lag.max = 96)
```

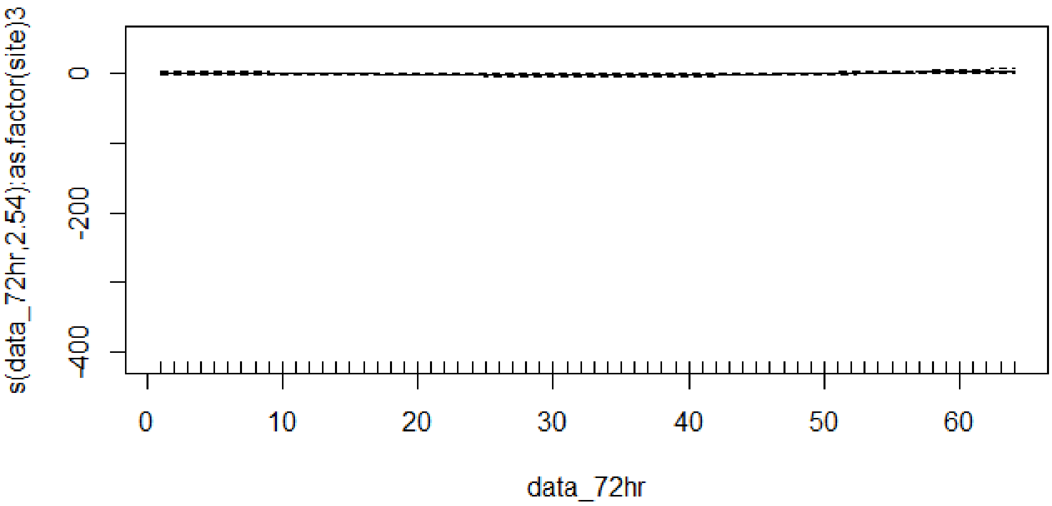



```
# 20 hour(?) lag

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

#  look at seal data by date and site at 72 hour groupings
date_seal_site <- gam(presence_new ~ s(data_72hr,by=as.factor(site)), data
= all_data_72, family=binomial(link="cloglog"))
```

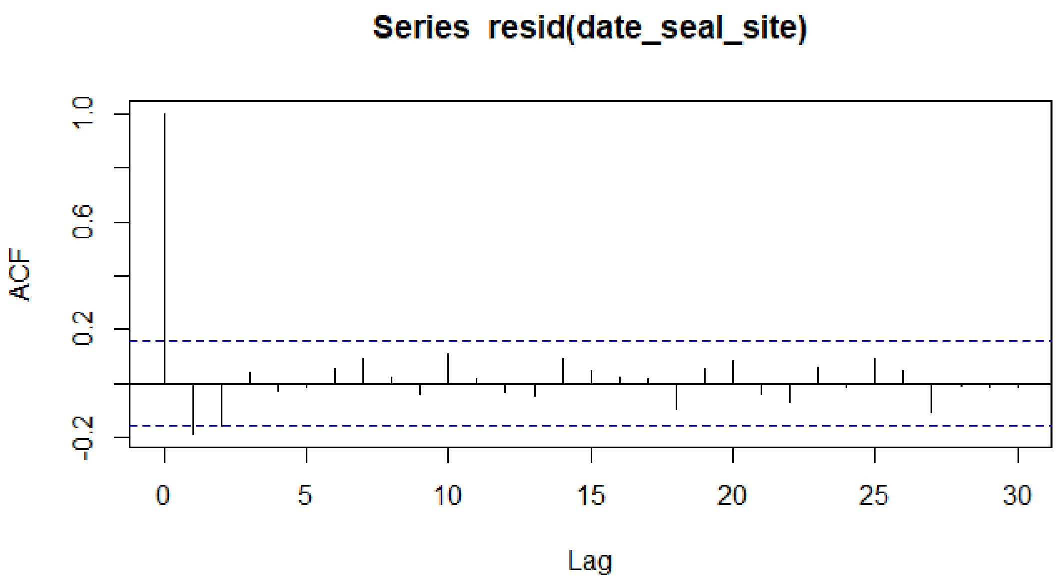



```
plot(date_seal_site)
```

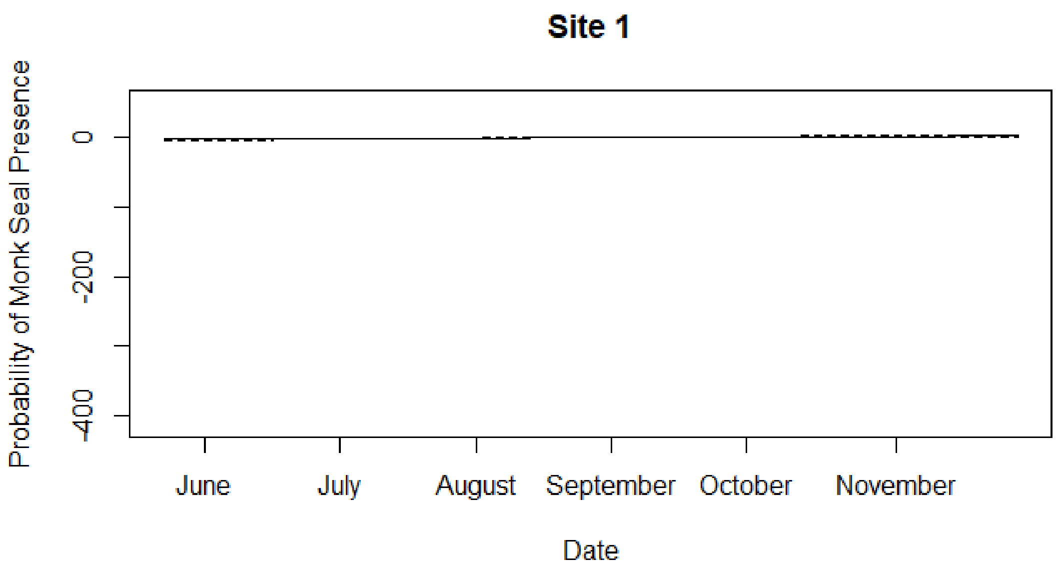




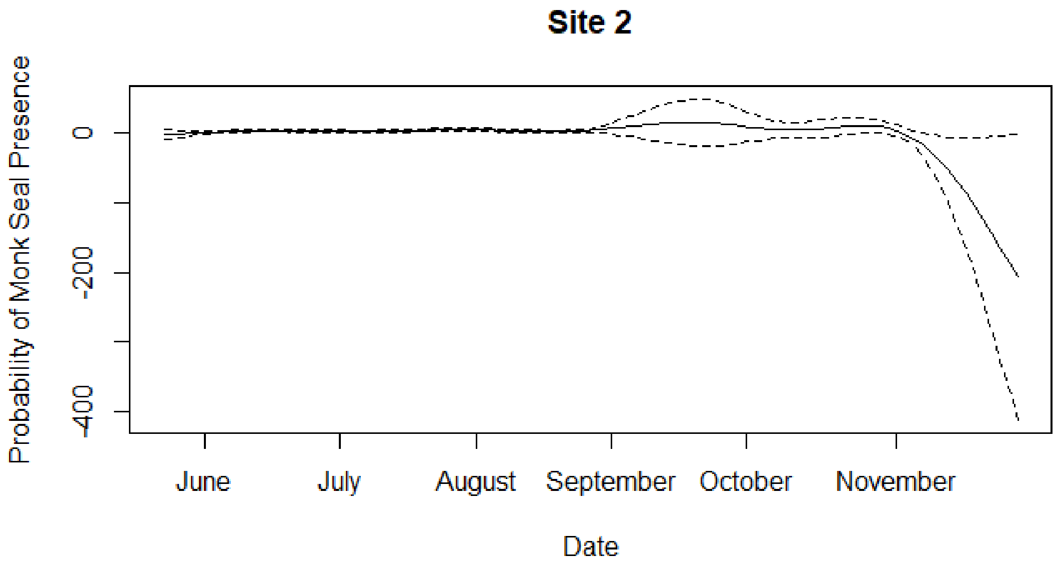

```
# plot autocorrelation
ACF  <-  acf(resid(date_seal_site), lag.max = 30)
```



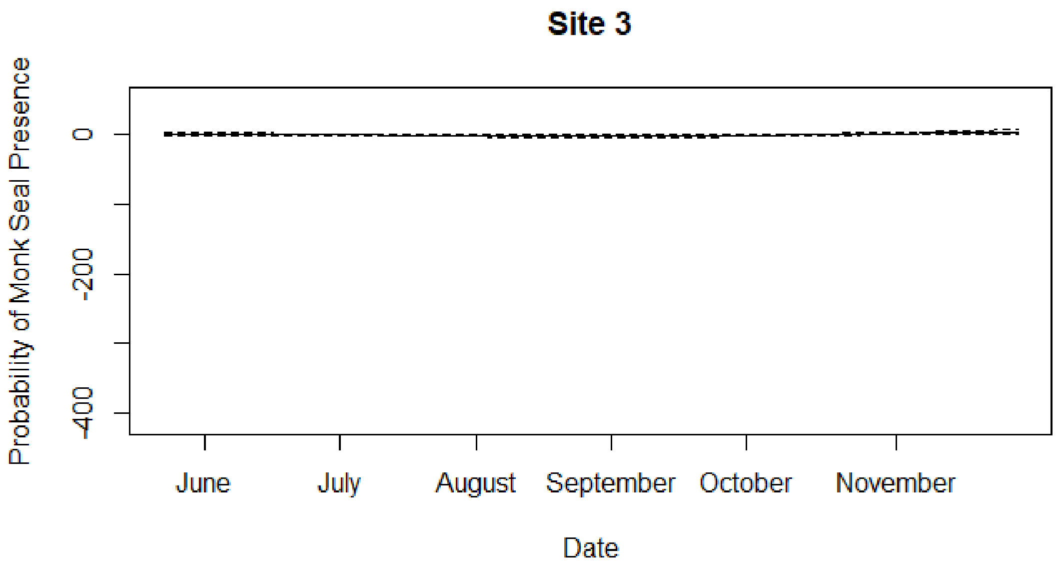

```
# replot
plot(date_seal_site, select = 1, main= "Site 1", xlab="Date",
ylab="Probability of Monk Seal Presence", xaxt="n")
label_x <- c("June", "July", "August","September","October","November")
axis(1, at = c("4", "14", "24","34","44","55"), labels = label_x)
```




```
plot(date_seal_site, select = 2, main= "Site 2", xlab="Date",
ylab="Probability of Monk Seal Presence", xaxt="n")
label_x <- c("June", "July", "August","September","October","November")
axis(1, at = c("4", "14", "24","34","44","55"), labels = label_x)
```




```
plot(date_seal_site, select = 3, main= "Site 3", xlab="Date",
ylab="Probability of Monk Seal Presence", xaxt="n")
label_x <- c("June", "July", "August","September","October","November")
axis(1, at = c("4", "14", "24","34","44","55"), labels = label_x)
```

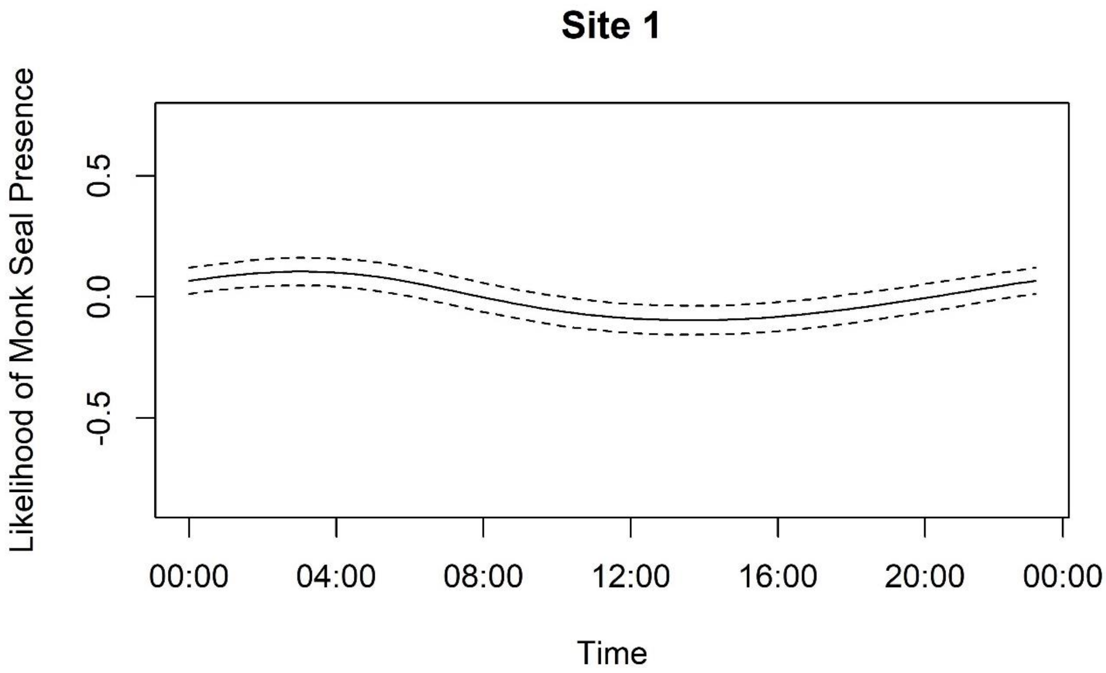


Create a new dataframe with vessel success/failures as each hour trial are independent, as trials occur on separate days

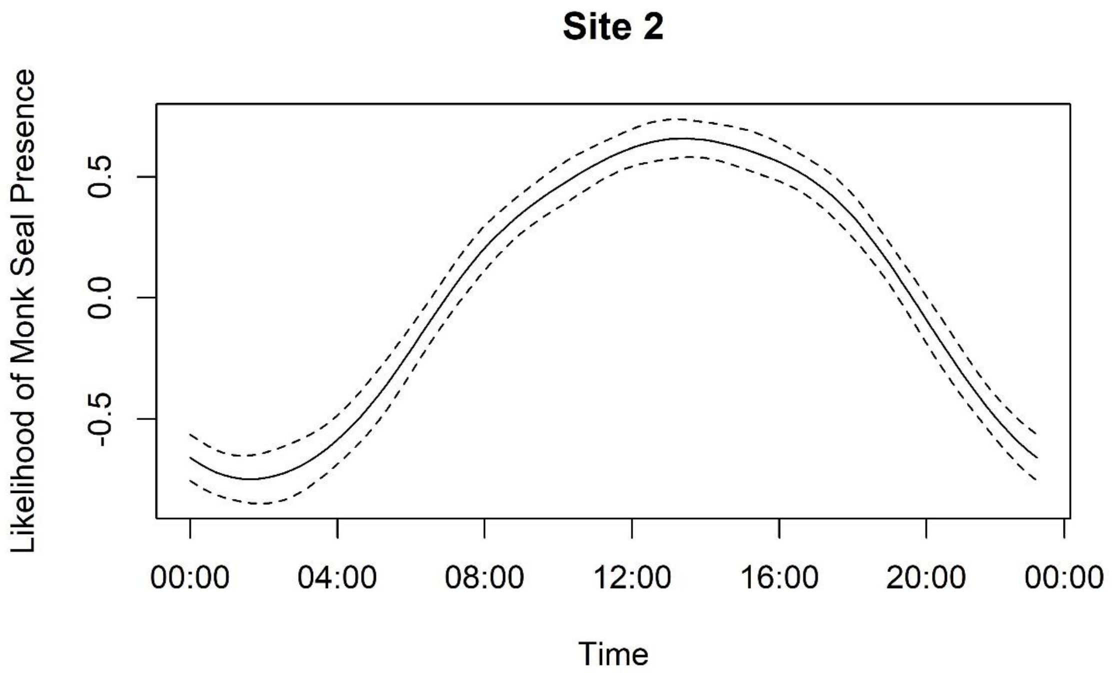
```r
# load data
all_vessels <- read.csv("E:/BL5599/all_vessels_hourly.csv", header =
T)

# split into site
all_vessels1 <- subset(all_vessels, site == "1" )
all_vessels2 <- subset(all_vessels, site == "2" )
all_vessels3 <- subset(all_vessels, site == "3" )


# count boat data entries per hour
boats_hr_count1 <- dplyr::count(all_vessels1, time_num)
boats_hr_count2 <- dplyr::count(all_vessels2, time_num)
boats_hr_count3 <- dplyr::count(all_vessels3, time_num)

# calculate number of successes
# calculate total presence for each 72hr group
v_success1  <- all_vessels1 %>% group_by(time_num) %>%
tally(boats_total_15min >= "1")
v_success2  <- all_vessels2 %>% group_by(time_num) %>%
tally(boats_total_15min >= "1")
v_success3  <- all_vessels3 %>% group_by(time_num) %>%
tally(boats_total_15min >= "1")

# join dataframes
boats_hr_count1 <- left_join(boats_hr_count1,v_success1, by="time_num")
boats_hr_count2 <- left_join(boats_hr_count2,v_success2, by="time_num")
boats_hr_count3 <- left_join(boats_hr_count3,v_success3, by="time_num")

# calculate vessel failures
boats_hr_count1$failures <- boats_hr_count1$n.x - boats_hr_count1$n.y
boats_hr_count2$failures <- boats_hr_count2$n.x - boats_hr_count2$n.y
boats_hr_count3$failures <- boats_hr_count3$n.x - boats_hr_count3$n.y

# add site
boats_hr_count1$site <- 1
boats_hr_count2$site <- 2
boats_hr_count3$site <- 3

# add seal successes/failures
data <-
read.csv('E:/BL5599/descriptive_statistics/final_dataset.csv',header=T)

# separate seal data into sites
all_seal1 <- subset(data, site == "1" )
all_seal2 <- subset(data, site == "2" )
all_seal3 <- subset(data, site == "3" )

# calculate total number of data entries per time (n)
time_seals1 <- all_seals1 %>% group_by(time) %>% tally()
time_seals2 <- all_seals2 %>% group_by(time) %>% tally()
```

```
time_seals3 <- all_seals3 %>% group_by(time) %>% tally()

# group into 1 hour intervals by representative time number
time_seals1$time_num  <- rep(0:23, each= 4)
time_seals2$time_num  <- rep(0:23, each= 4)
time_seals3$time_num  <- rep(0:23, each= 4)

# calculate total number of data entries per time (n)
time_seals1 <- time_seals1 %>% group_by(time_num) %>%
summarise_at(vars(n), list(name = sum))
time_seals2 <- time_seals2 %>% group_by(time_num) %>%
summarise_at(vars(n), list(name = sum))
time_seals3 <- time_seals3 %>% group_by(time_num) %>%
summarise_at(vars(n), list(name = sum))

# add site
time_seals1$site <- 1
time_seals2$site <- 2
time_seals3$site <- 3


############
# calculate total presence for each time
s_success1  <- all_seals1 %>% group_by(time) %>% tally(presence == 1)
s_success2  <- all_seals2 %>% group_by(time) %>% tally(presence == 1)
s_success3  <- all_seals3 %>% group_by(time) %>% tally(presence == 1)

# group into 1 hour intervals by representative time number
s_success1$time_num  <- rep(0:23, each= 4)
s_success2$time_num  <- rep(0:23, each= 4)
s_success3$time_num  <- rep(0:23, each= 4)

# calculate total number of successes by time_num
s_success1 <- s_success1 %>% group_by(time_num) %>% summarise_at(vars(n),
list(name = sum))
s_success2 <- s_success2 %>% group_by(time_num) %>% summarise_at(vars(n),
list(name = sum))
s_success3 <- s_success3 %>% group_by(time_num) %>% summarise_at(vars(n),
list(name = sum))

# join dataframes
time_seals1 <- left_join(time_seals1,s_success1, by="time_num")
time_seals2 <- left_join(time_seals2,s_success2, by="time_num")
time_seals3 <- left_join(time_seals3,s_success3, by="time_num")

s_success_hr <- rbind(time_seals1,time_seals2,time_seals3)

s_success_hr <- s_success_hr %>%  dplyr::rename(trials=name.x,
successes=name.y)
s_success_hr$failures <- s_success_hr$trials - s_success_hr$successes

# subset to daylight hours to join with vessel data later
s_success_hr_6_22 <- subset(s_success_hr, time_num >= 6 & time_num < 22)
# 48 obs.
```

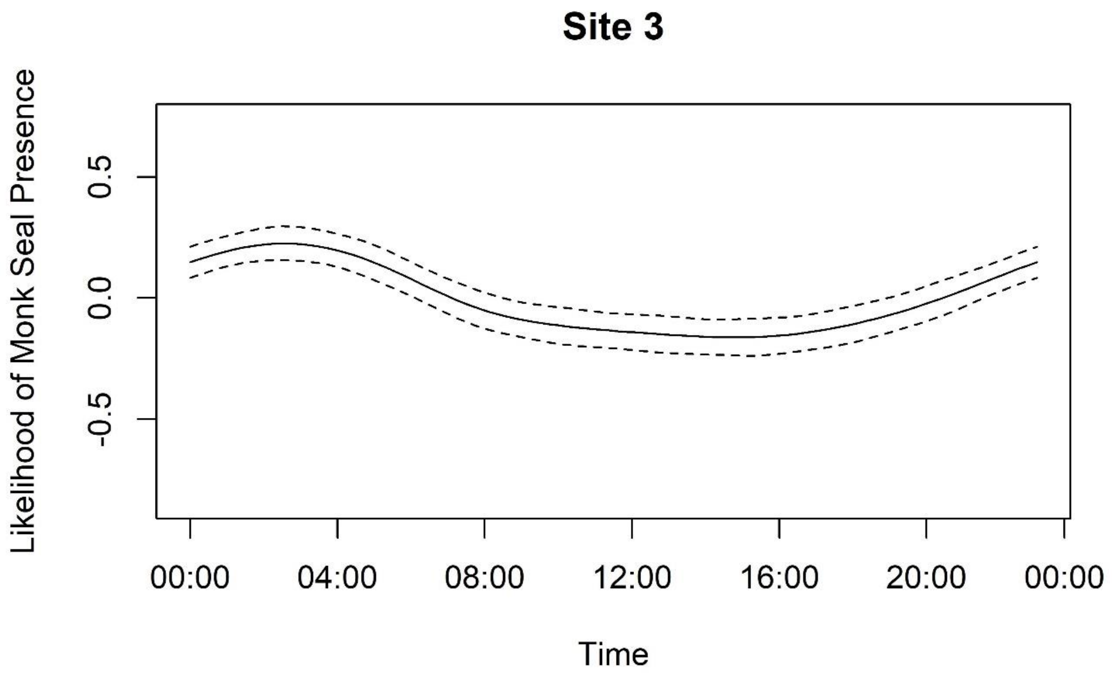
```
seals_hourly <- gam(cbind(successes, failures) ~ s(time_num,
by=as.factor(site)), data = s_success_hr, family =
binomial(link="cloglog"))

# plot with new axes and labels and save as jpeg images
plot(seals_hourly, select = 1, main = "Site 1", xlab="Time",
ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00","20:00","00:00")
```



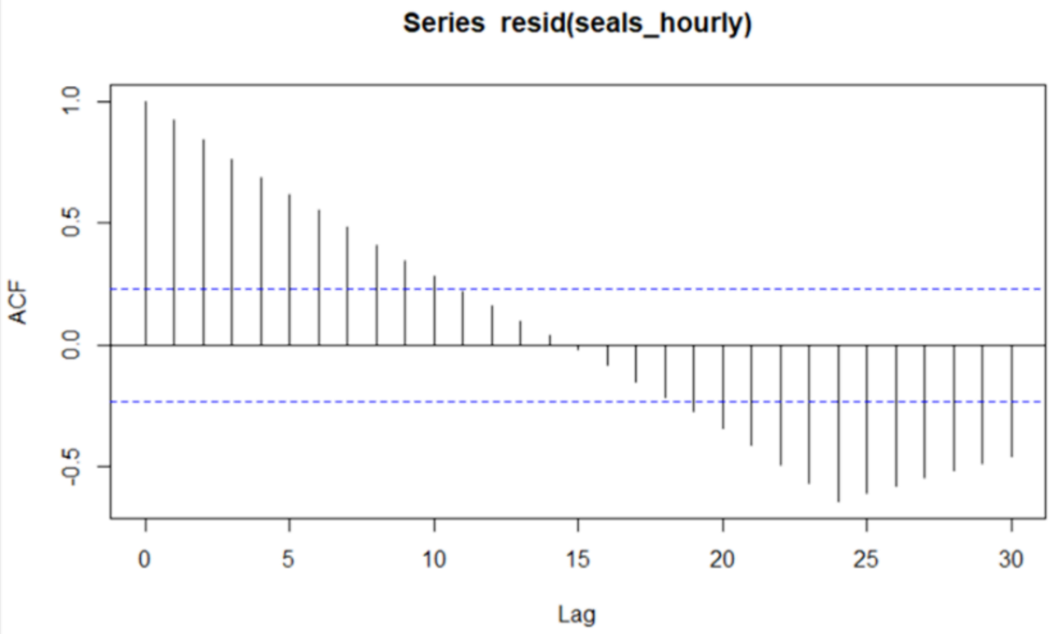

```
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)

plot(seals_hourly, select = 2, main = "Site 2", xlab="Time",
ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00","20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```




```
plot(seals_hourly, select = 3, main = "Site 3", xlab="Time",
ylab="Likelihood of Monk Seal Presence", xaxt="n")
label_x <- c("00:00","04:00","08:00","12:00","16:00","20:00","00:00")
axis(1, at = c("0", "4", "8", "12", "16", "20", "23.75"), labels =
label_x)
```

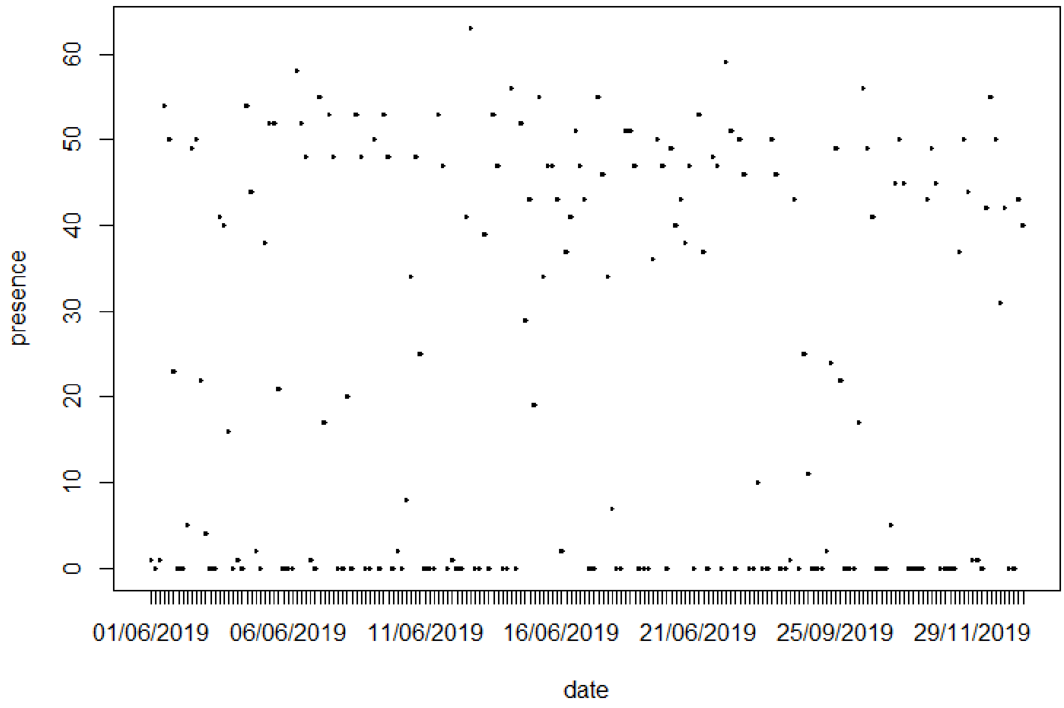



```
# plot autocorrelation
ACF  <-  acf(resid(seals_hourly), lag.max = 30)
```

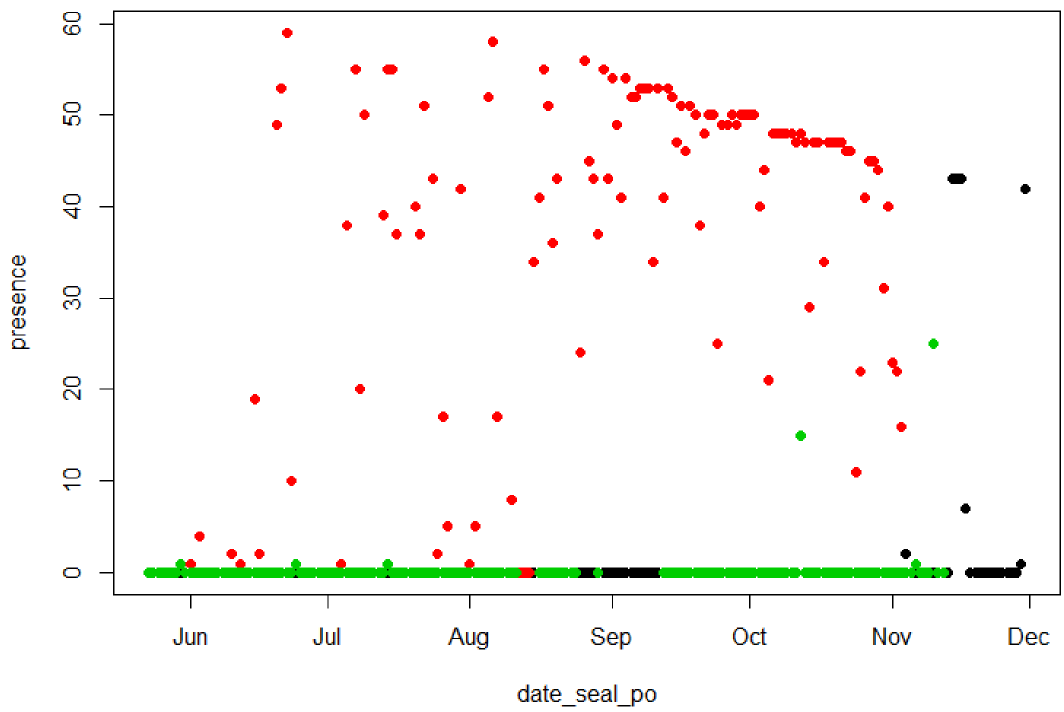


Plot RAW seasonal seal data vs 72 hourly data

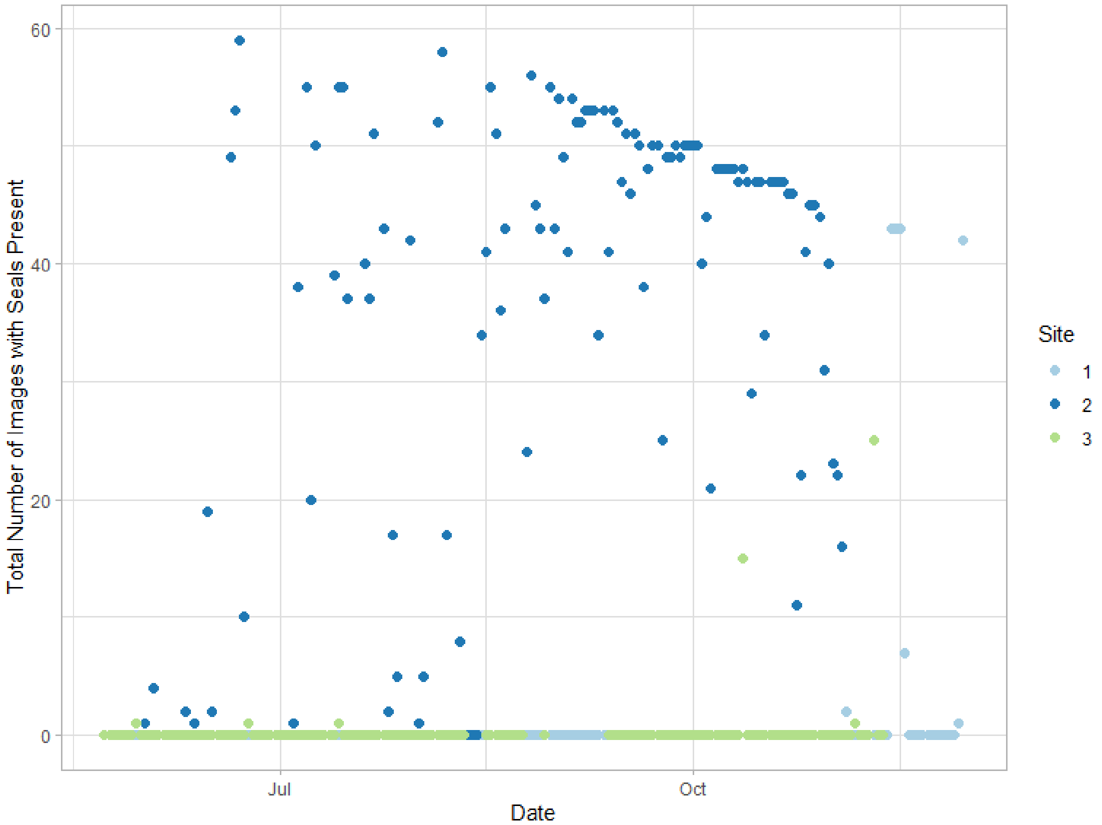
```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

# Raw data  -> use 'data'
data <- read.csv('E:/BL5599/descriptive_statistics/final_dataset.csv',
header = T)

# 72hr data -> use 'all_data_72'
all_data_72 <- read.csv('E:/BL5599/vessel_seal_data_72hrs.csv', header =
T)

# monk seals with date
# need to calculate total number of monk seal occurances for each date
date_seal <- aggregate(presence ~ date, data=data, sum)
plot(presence~date,data=date_seal)
```




```r
# times are factor
str(date_seal)

## 'data.frame':    192 obs. of  2 variables:
##  $ date    : Factor w/ 192 levels "01/06/2019","01/07/2019",..: 1 2 3 4
5 6 7 8 9 10 ...
##  $ presence: int  1 0 1 54 50 23 0 0 5 49 ...

# convert times to POSIXct to plot time series
date_seal$date_seal_po<-as.POSIXct(date_seal$date,format='%d/%m/%y')

# plot by date AND site
date_seal <- aggregate(presence ~ date + site, data=data, sum)
# convert times to POSIXct to plot time series
date_seal$date_seal_po<-as.POSIXct(date_seal$date,format="%d/%m/%Y")

# plot date coloured by site
plot(presence~date_seal_po,data=date_seal, pch = 16, cex = 1, col =
date_seal$site)
```



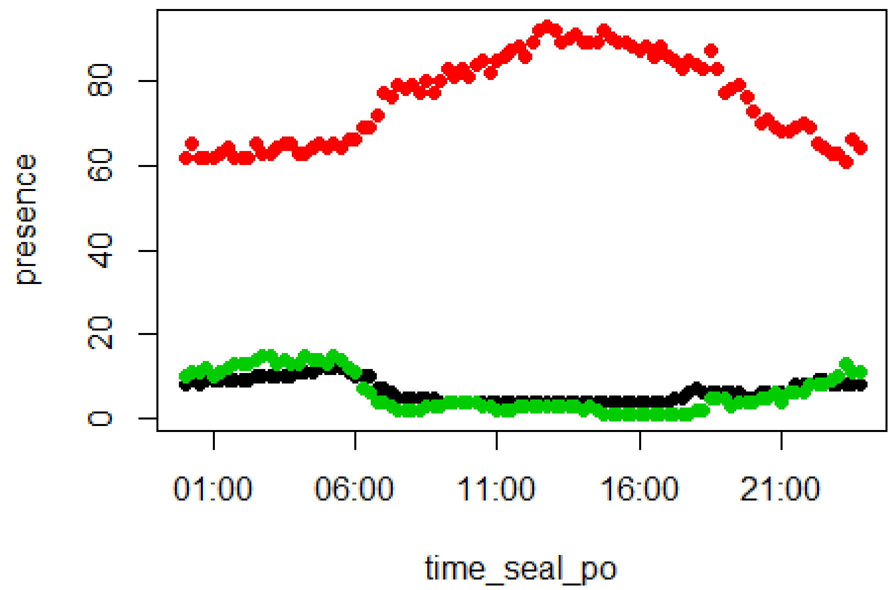

```
# plot date coloured by site FINAL
# convert site from factor to integer
date_seal$Site <- as.factor(date_seal$site)
 plot_date <- ggplot(date_seal, aes(date_seal_po,presence, color=Site)) +
geom_point(shape=16,size=2.3) +  scale_color_brewer(palette = "Paired") +
xlab("Date") + ylab("Total Number of Images with Seals Present") +
theme(text = element_text(size=16))

plot_date  + theme_light()
```




```
# relabel x axis labels to show month
date_breaks <- as.POSIXct(c("2019-06-01 00:00:00 BST", "2019-07-01
00:00:00 BST", "2019-08-01 00:00:00 BST", "2019-09-01 00:00:00 BST",
"2019-10-01 00:00:00 BST", "2019-11-01 00:00:00 BST", "2019-12-01 00:00:00
BST"))

plot_date + scale_y_continuous(name="Total Number of Images with Seals
Present", limits=c(0, 100)) + scale_x_datetime(breaks= date_breaks,
labels=c("June", "July", "August", "September", "October",
"November","December")) + ggtitle("24 Hour Sampling Rate") + theme_light()
```

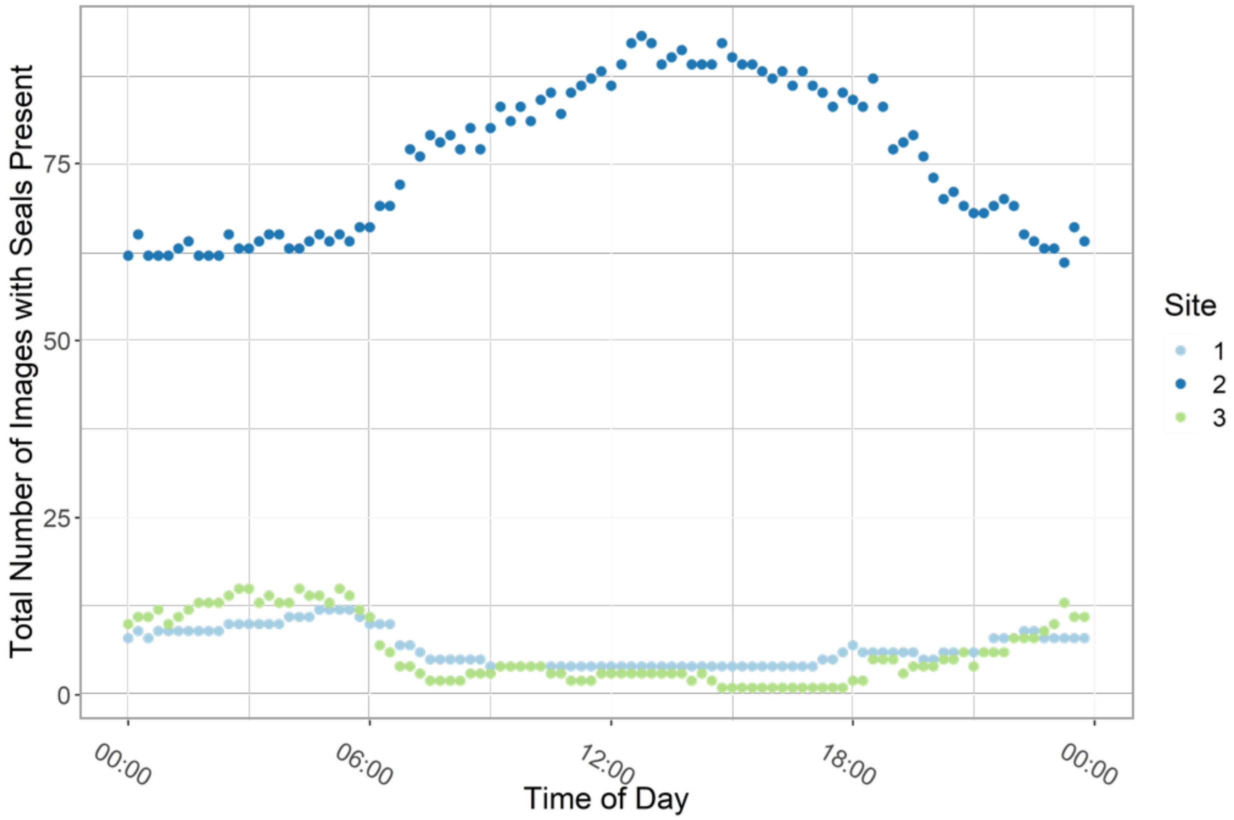
Plot RAW seal data vs hourly data

```r
# load packages
library(ggplot2)

library(dplyr)

# load data
data <-
read.csv('E:/BL5599/descriptive_statistics/final_dataset.csv',header=T)
all_data_hourly <- read.csv("E:/BL5599/all_data_hourly.csv", header = T)


# plot by time AND site
time_seal <- aggregate(presence ~ time_round + site, data=data, sum)
# convert times to POSIXct to plot time series
time_seal$time_seal_po<-as.POSIXct(time_seal$time_round,format='%H:%M')

# plot date coloured by site
plot(presence~time_seal_po,data=time_seal, pch = 16, cex = 1, col =
time_seal$site)
```




```r
# change site to Site for legend header
time_seal$Site <- as.factor(time_seal$site)

  # plot time coloured by site FINAL
plot_time_seal <- ggplot(time_seal, aes(time_seal_po,presence,
color=Site)) + geom_point(shape=16,size=2.3) +  scale_color_brewer(palette
= "Paired") + xlab("Time of Day") + ylab("Total Number of Images with
Seals Present") + theme(text = element_text(size=16), axis.text.x =
element_text(angle = 330)) + theme_light()
```

```
# specify axis ticks
time_breaks <- as.POSIXct(c("2021-07-12 00:00:00 BST", "2021-07-12
06:00:00 BST", "2021-07-12 12:00:00 BST", "2021-07-12  18:00:00 BST",
"2021-07-13 00:00:00 BST"))

# plot and relabel x axis
plot_time_seal + scale_x_datetime(breaks= time_breaks, labels=c("00:00",
"06:00", "12:00", "18:00", "00:00")) + theme_light()
```

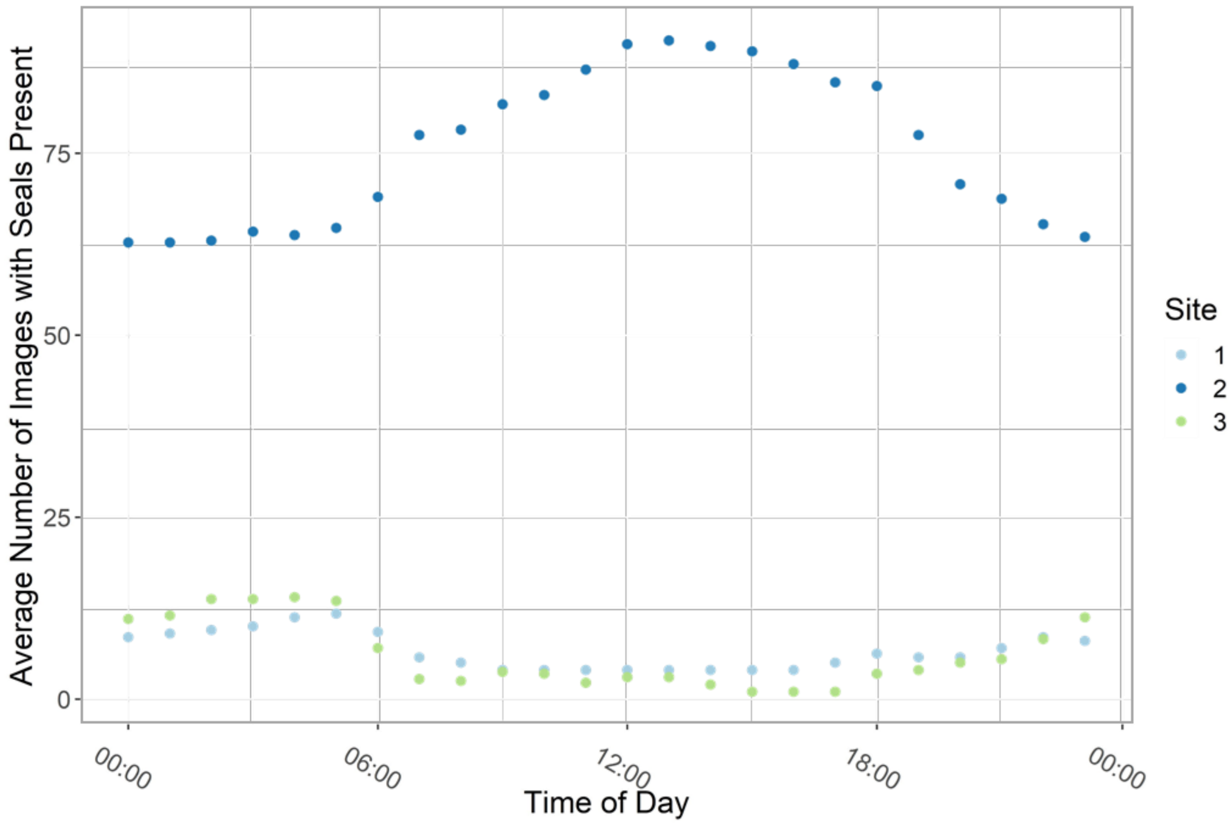



```
# save high quality plot
jpeg("E:/BL5599/high_res_plots/seals_RAW.jpg", width = 9, height = 6,
units = 'in', res = 600)
plot_time_seal + scale_x_datetime(breaks= time_breaks, labels=c("00:00",
"06:00", "12:00", "18:00", "00:00")) + theme_light()
dev.off()

## png
##   2

# plot hourly seal data - subset into sites
time_seal_1 <-  subset(time_seal, site == "1")
time_seal_2 <-  subset(time_seal, site == "2")
time_seal_3 <-  subset(time_seal, site == "3")

# site 1 obs = 96, site 2 obs = 96, site 3 obs = 96
# add time num for hourly groupings
time_seal_1$time_num <- rep(0:23, each=4)
time_seal_2$time_num <- rep(0:23, each=4)
time_seal_3$time_num <- rep(0:23, each=4)
```

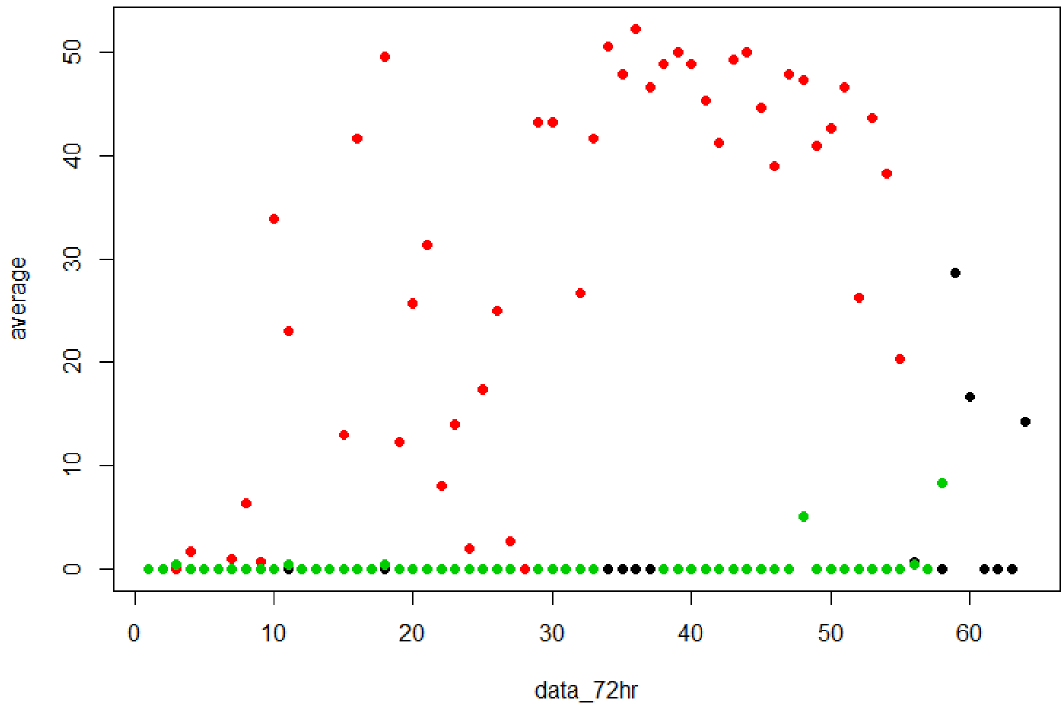
```
# calculate total seal presence occurance count for each hourly interval
hrly_seal_1 <- time_seal_1 %>% group_by(time_num) %>%
summarise_at(vars(presence), list(name = sum))
hrly_seal_2 <- time_seal_2 %>% group_by(time_num) %>%
summarise_at(vars(presence), list(name = sum))
hrly_seal_3 <- time_seal_3 %>% group_by(time_num) %>%
summarise_at(vars(presence), list(name = sum))

# add site
hrly_seal_1$Site <- 1
hrly_seal_2$Site <- 2
hrly_seal_3$Site <- 3

# redo to add in total data entry count for each time_num
all_data_hourly$site = as.factor(all_data_hourly$site)
all_data_hourly1 <- subset(all_data_hourly, site == 1)
all_data_hourly2 <- subset(all_data_hourly, site == 2)
all_data_hourly3 <- subset(all_data_hourly, site == 3)

# calculate total data entries
hrly_count1 <- dplyr::count(all_data_hourly1, time_num)
hrly_count2 <- dplyr::count(all_data_hourly2, time_num)
hrly_count3 <- dplyr::count(all_data_hourly3, time_num)

# join hrly count with hrly seal presence
hrly_seal_1 <- left_join(hrly_seal_1, hrly_count1, by= "time_num")
hrly_seal_2 <- left_join(hrly_seal_2, hrly_count2, by= "time_num")
hrly_seal_3 <- left_join(hrly_seal_3, hrly_count3, by= "time_num")

# calculate hourly average
hrly_seal_1$data_x4  <- hrly_seal_1$n * 4
hrly_seal_2$data_x4  <- hrly_seal_2$n * 4
hrly_seal_3$data_x4  <- hrly_seal_3$n * 4

# calculate seal failures
hrly_seal_1$failures <- hrly_seal_1$data_x4 - hrly_seal_1$name
hrly_seal_2$failures <- hrly_seal_2$data_x4 - hrly_seal_2$name
hrly_seal_3$failures <- hrly_seal_3$data_x4 - hrly_seal_3$name

# join seperate sites back into one dataframe
hrly_seal_all <- rbind(hrly_seal_1, hrly_seal_2, hrly_seal_3)
hrly_seal_all$Site <- as.factor(hrly_seal_all$Site)

# save as csv
write.csv(hrly_seal_all, "E:/BL5599/seals_all_hourly.csv")

# now plot hourly data to compare to raw data
plot_time_seal <- ggplot(hrly_Seal_all, aes(time_num, average_seals,
color=Site)) + geom_point(shape=16,size=2.3) +  scale_color_brewer(palette
= "Paired") + xlab("Time of Day") + ylab("Average Number of Images with
Seals Present") + theme(text = element_text(size=16), axis.text.x =
element_text(angle = 330)) + theme_light()
```

```
# relabel x axis with actual times rather than time_num
time_breaks <- c("0", "6", "12", "18", "23.9")
time_breaks <- as.numeric(time_breaks)
```

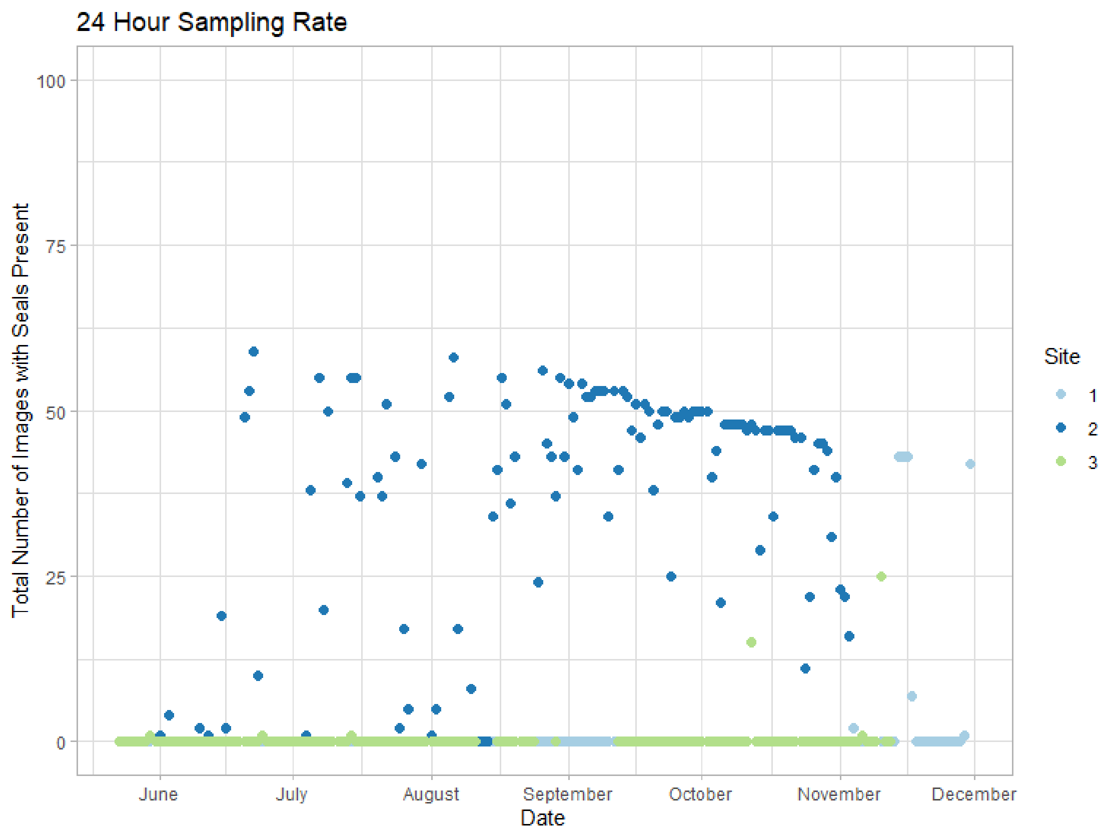
```
plot_time_seal + scale_x_discrete(breaks= time_breaks, labels=c("00:00",
"06:00", "12:00", "18:00", "00:00")) + theme_light()
```




```
# Plot 72 hourly data

# monk seals with 72 hourly grouping number
# need to calculate total number of monk seal occurances for each 72hr
period
date_seal72 <- aggregate(n.x ~ data_72hr + site, data=all_data_72, sum)

# will need to convert presence total to average across 3 days/72 hrs to
be comparable to daily plot
# 24 hr average
date_seal72$average <- date_seal72$n.x/3

# plot averaged data
plot(average~data_72hr,data=date_seal72, pch = 16, cex = 1, col =
date_seal72$site)
```

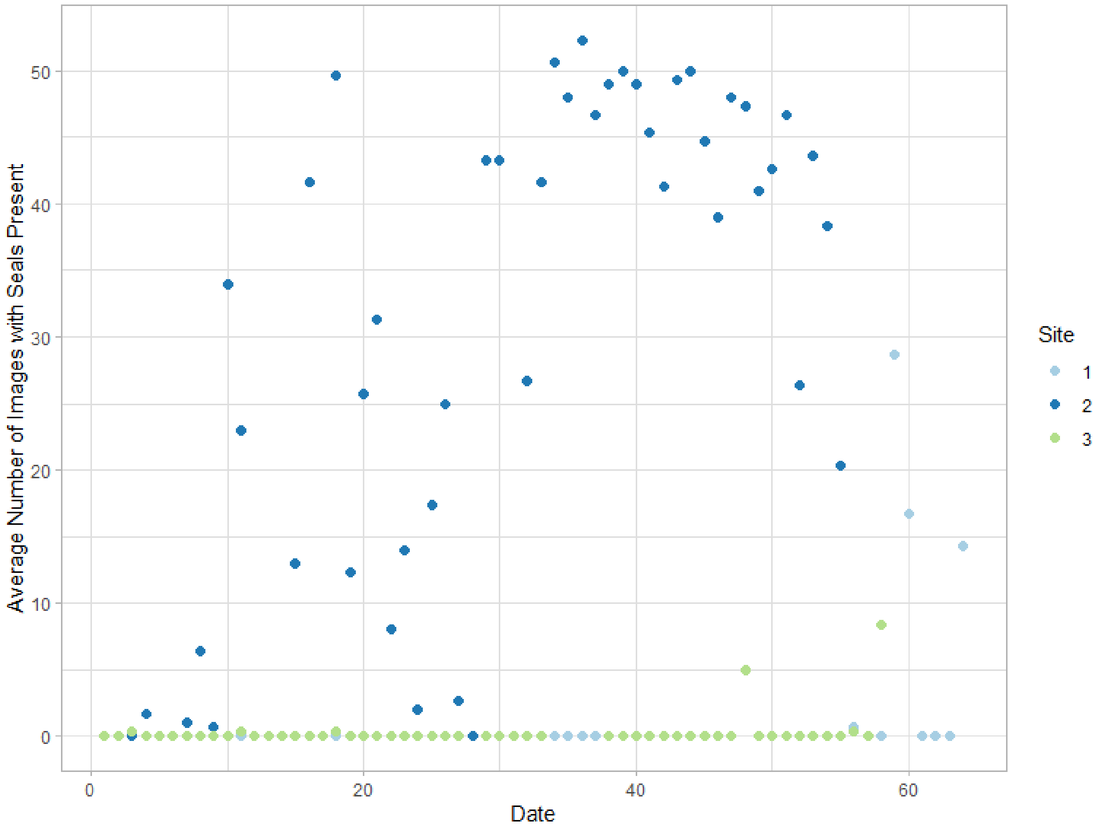



```
# check data format
```

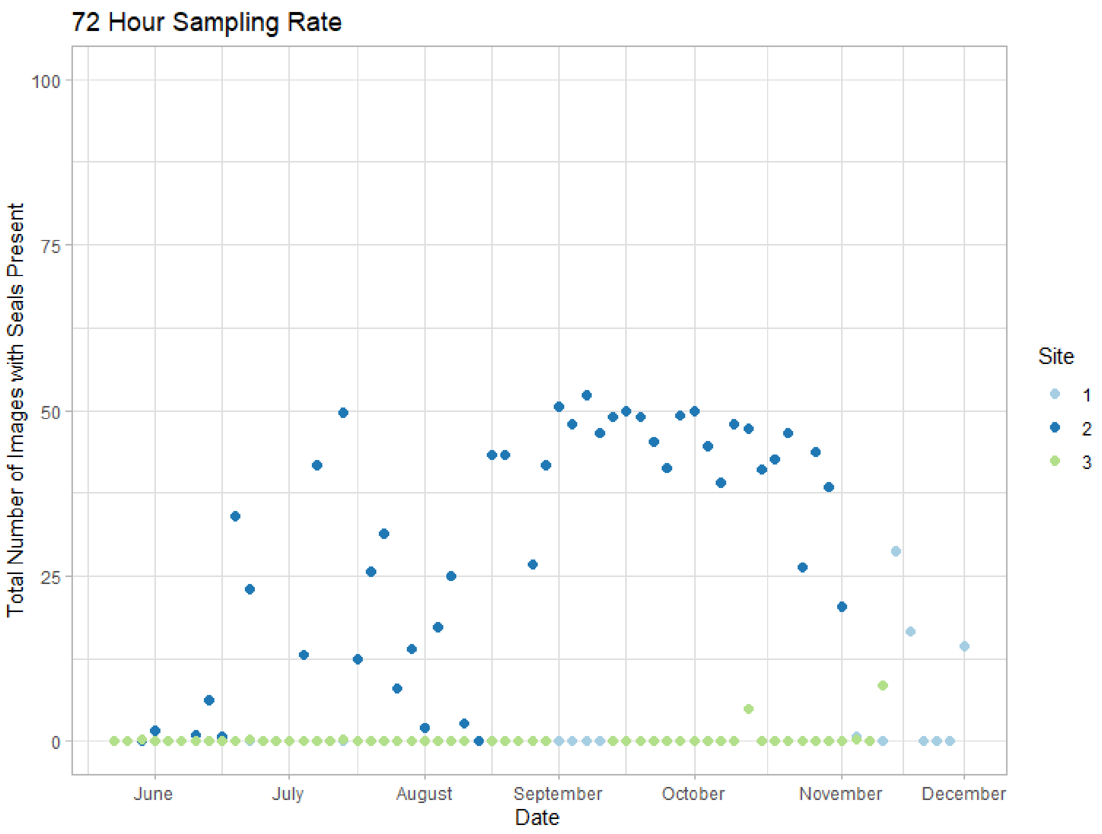



```
str(date_seal72)

## 'data.frame':    157 obs. of  4 variables:
##  $ data_72hr: int  2 3 4 5 6 7 8 9 10 11 ...
##  $ site     : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ n.x      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ average  : num  0 0 0 0 0 0 0 0 0 0 ...

# convert site from integer to factor for plot colour
# capitalise for legend title
date_seal72$Site <- as.factor(date_seal72$site)

# plot seals with 72hr period coloured by site
# convert site from factor to integer
plot_seal_72 <- ggplot(date_seal72, aes(data_72hr, average, color=Site)) +
geom_point(shape=16,size=2.3) +  scale_color_brewer(palette = "Paired") +
xlab("Date") + ylab("Average Number of Images with Seals Present") +
theme(text = element_text(size=16))
plot_seal_72  + theme_light()
```




```
# relabel x axis with month
date_breaks72 <- as.numeric(c("4","14","24","34","44","55","64"))
plot_seal_72 + scale_y_continuous(name="Total Number of Images with Seals
Present", limits=c(0, 100)) + scale_x_continuous(breaks= date_breaks72,
labels=c("June", "July", "August", "September", "October",
"November","December")) + ggtitle("72 Hour Sampling Rate") + theme_light()
```


72 Hour Sampling Rate
Total Number of Images with Seals Present
100
75
50
25
0
June
July
August
September
October
November
December
Date
Site
1
2
3

Remodel data using only June - end of September to exclude the majority of periods where not all sites are operational

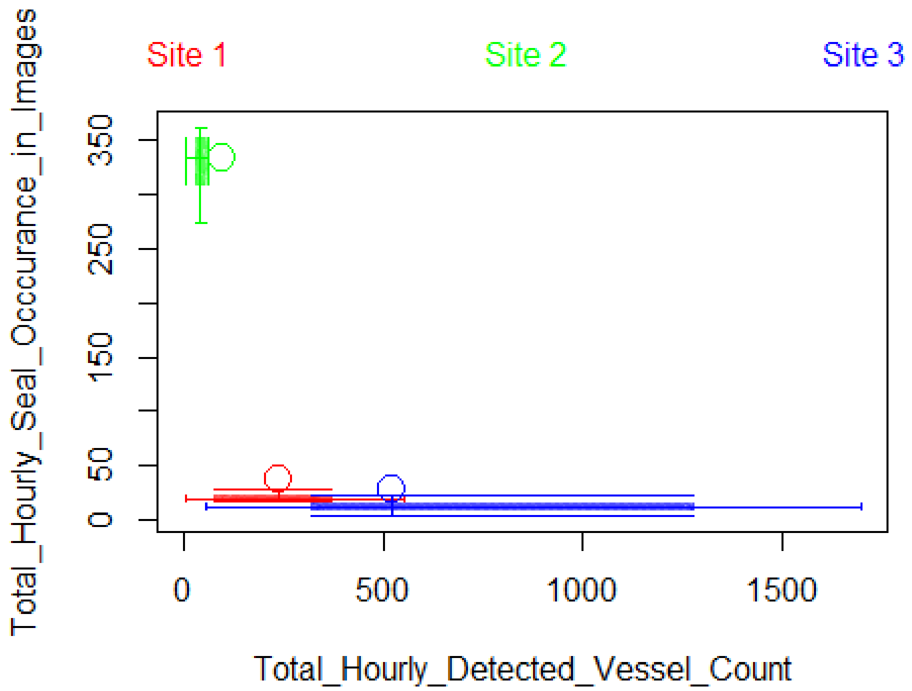
```
# visual inspection of data shows all 3 sites typically operational
between date_num (72hr period
# grouping) 3 - 44  == 29/05/2019 -> 30/09/2019
# site 2 camera down in Oct due to storms
# camera operation across all 3 sites is varied in November
```

72hr_num	total_seal_presence	0_1_presence	site	total_boat_count	total_v_images	vessel_index
24	23	1	3	591	288	2.052083333
25	0	0	3	329	288	1.142361111
26	24	1	3	653	288	2.267361111
27	22	1	3	423	251	1.685258964
29	0	0	3	364	147	2.476190476
30	0	0	3	528	288	1.833333333
31	0	0	3	512	288	1.777777778
32	0	0	3	69	63	1.095238095
33	0	0	3	132	28	4.714285714
38	0	0	3	141	153	0.921568627
39	0	0	3	155	288	0.538194444
40	0	0	3	261	288	0.90625

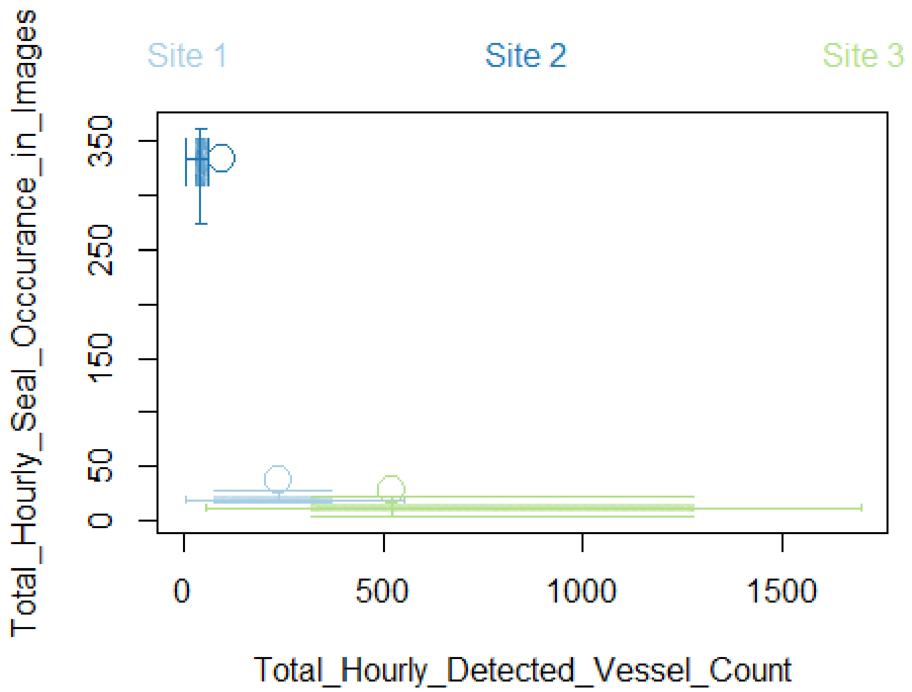
```
# subset by time_num
all_data_72_Jun_Sept <- subset(all_data_72, data_72hr >= 4 & data_72hr <=
44)
```

Plot medians and interquartile ranges by site with seal presence on x axis and vessel index on y axis

```
#load data
final_72hr_data <-
read.csv('E:/BL5599/vessel_seal_data_72hrs_21_07_21.csv',header=T)
all_data_hourly_6_22 <-
read.csv('E:/BL5599/all_data_hourly_6_22.csv',header=T)

final_72hr_data1 <- subset(final_72hr_data, site == "1")
final_72hr_data2 <- subset(final_72hr_data, site == "2")
final_72hr_data3 <- subset(final_72hr_data, site == "3")

# load pacakge
# install.packages("boxplotdbl")
library(boxplotdbl)

# create boxplot dataframe as boxplotdbl does not allow specification of
xlab and ylab
boxplot_hourly <- all_data_hourly_6_22
boxplot_hourly$site <- ifelse(boxplot_hourly$site == 1, "Site 1",
boxplot_hourly$site)
boxplot_hourly$site <- ifelse(boxplot_hourly$site == 2, "Site 2",
boxplot_hourly$site)
boxplot_hourly$site <- ifelse(boxplot_hourly$site == 3, "Site 3",
boxplot_hourly$site)

boxplot_hourly$Total_Hourly_Seal_Occcurance_in_Images <-
boxplot_hourly$successes
boxplot_hourly$Total_Hourly_Detected_Vessel_Count <- boxplot_hourly$name

# hourly vessel index
boxplot_hourly$Vessel_Index_Hourly <- boxplot_hourly$vessel_index

# create double boxplot
boxplotdou( Total_Hourly_Detected_Vessel_Count ~ site, boxplot_hourly,
Total_Hourly_Seal_Occcurance_in_Images ~ site, boxplot_hourly,shading=500,
factor.labels= F, name.on.axis=F)
```

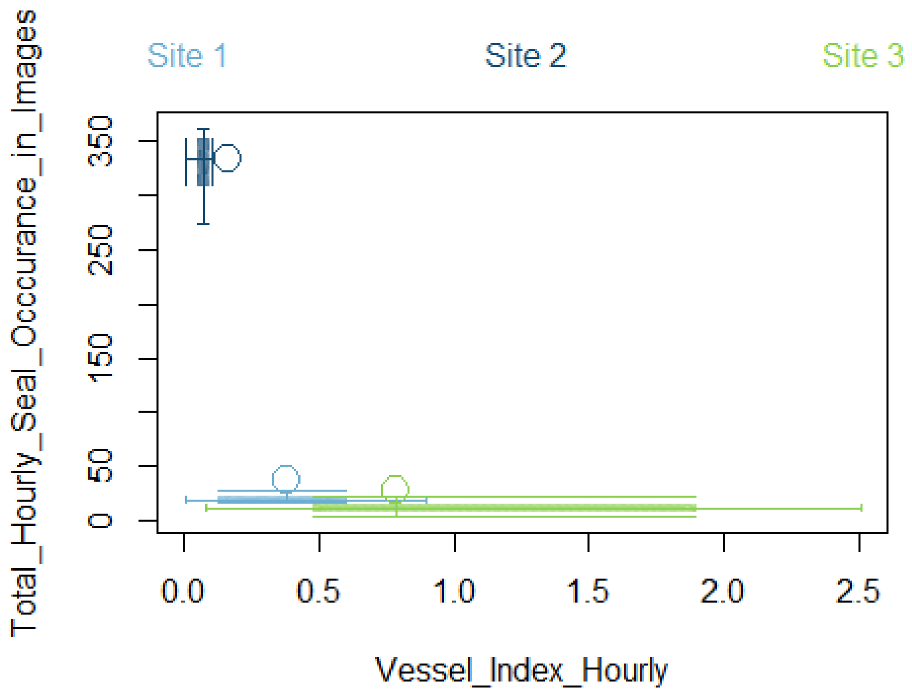



```
# colour by site colours
boxplotdou( Total_Hourly_Detected_Vessel_Count ~ site, boxplot_hourly,
Total_Hourly_Seal_Occcurance_in_Images ~ site, boxplot_hourly,shading=500,
factor.labels= F, name.on.axis=F, col = c("#A6CEE3","#1F78B4","#B2DF8A"))
```



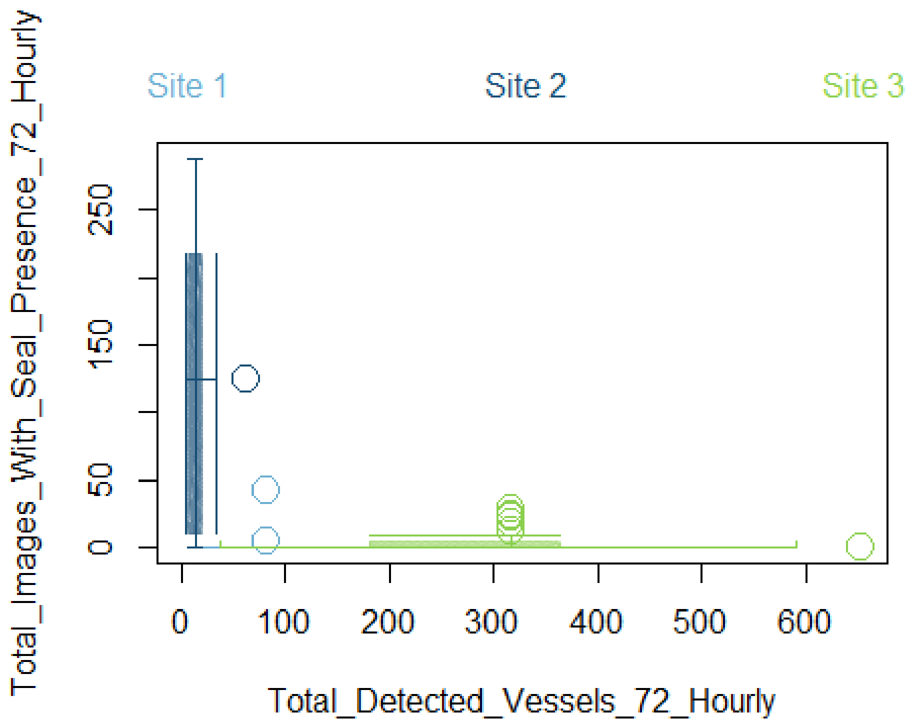

```
# needs brighter colours for visibility
jpeg("E:/BL5599/high_res_plots/2D_boxplot_hourly.jpg", width = 6, height =
4, units = 'in', res = 600)
boxplotdou( Total_Hourly_Detected_Vessel_Count ~ site, boxplot_hourly,
Total_Hourly_Seal_Occcurance_in_Images ~ site, boxplot_hourly,shading=500,
```

```
factor.labels= F, name.on.axis=F, col = c("#6caed1","#144c73","#8acf4e"))
dev.off()

## png
##   2

# hourly vessel index
boxplotdou( Vessel_Index_Hourly ~ site, boxplot_hourly,
Total_Hourly_Seal_Occcurance_in_Images  ~ site,
boxplot_hourly,shading=500, factor.labels= F, name.on.axis=F, col =
c("#6caed1","#144c73","#8acf4e"))
```



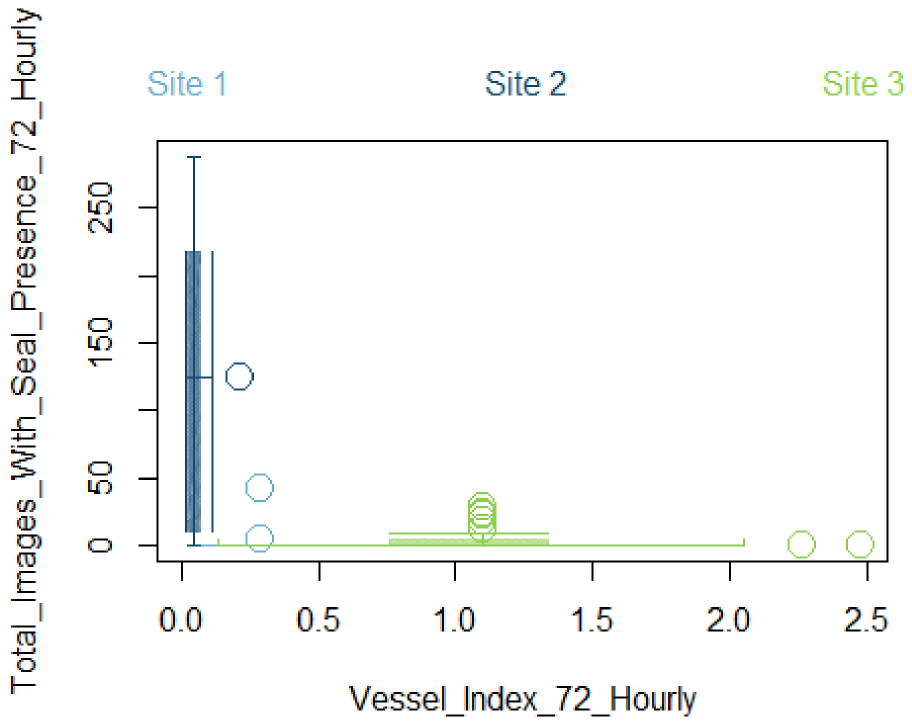

```
# save plot
jpeg("E:/BL5599/high_res_plots/2D_boxplot_hourly_index.jpg", width = 6,
height = 4, units = 'in', res = 600)
boxplotdou( Vessel_Index_Hourly ~ site, boxplot_hourly,
Total_Hourly_Seal_Occcurance_in_Images  ~ site,
boxplot_hourly,shading=500, factor.labels= F, name.on.axis=F, col =
c("#6caed1","#144c73","#8acf4e"))
dev.off()

## png
##   2

##
# repeat over seasonal scale
# rename variables
boxplot_seasonal <- final_72hr_data
boxplot_seasonal$site <- ifelse(boxplot_seasonal$site == 1, "Site 1",
boxplot_seasonal$site)
boxplot_seasonal$site <- ifelse(boxplot_seasonal$site == 2, "Site 2",
boxplot_seasonal$site)
```

```
boxplot_seasonal$site <- ifelse(boxplot_seasonal$site == 3, "Site 3",
boxplot_seasonal$site)

boxplot_seasonal$Total_Detected_Vessels_72_Hourly <-
boxplot_seasonal$boats_total_15min
boxplot_seasonal$Total_Images_With_Seal_Presence_72_Hourly <-
boxplot_seasonal$n.x

boxplotdou( Total_Detected_Vessels_72_Hourly ~ site, boxplot_seasonal,
Total_Images_With_Seal_Presence_72_Hourly  ~ site,
boxplot_seasonal,shading=500, factor.labels= F, name.on.axis=F, col =
c("#6caed1","#144c73","#8acf4e"))
```

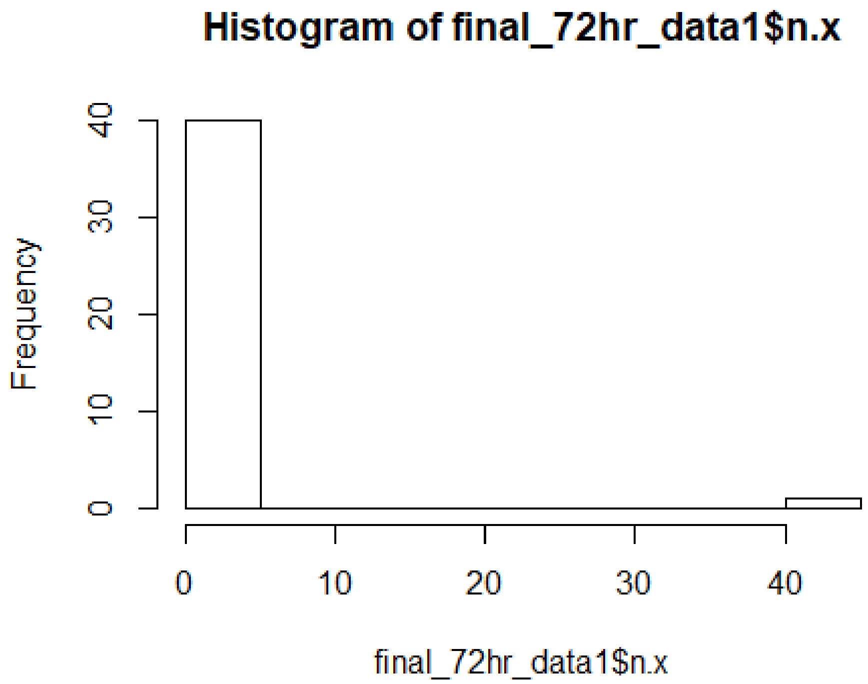



```
# produce boxplot
jpeg("E:/BL5599/high_res_plots/2D_boxplot_seasonal.jpg", width = 6, height
= 4, units = 'in', res = 600)
boxplotdou( Total_Detected_Vessels_72_Hourly ~ site, boxplot_seasonal,
Total_Images_With_Seal_Presence_72_Hourly  ~ site,
boxplot_seasonal,shading=500, factor.labels= F, name.on.axis=F, col =
c("#6caed1","#144c73","#8acf4e"))
dev.off()

## png
##   2

# vessel index instead of boat count
boxplot_seasonal$Vessel_Index_72_Hourly <- boxplot_seasonal$NA_index

boxplotdou( Vessel_Index_72_Hourly ~ site, boxplot_seasonal,
Total_Images_With_Seal_Presence_72_Hourly  ~ site,
```

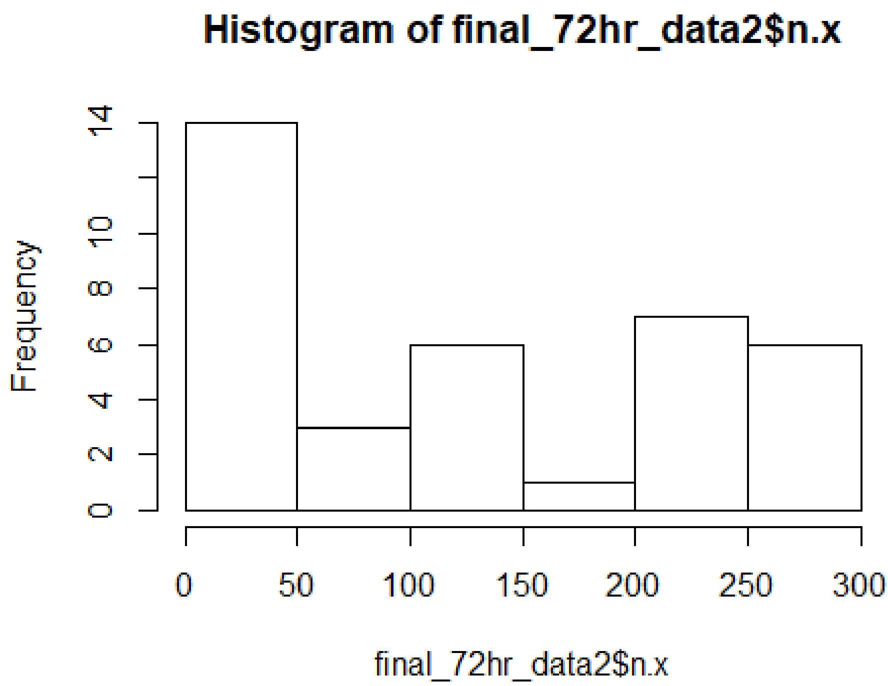
```
boxplot_seasonal,shading=500, factor.labels= F, name.on.axis=F, col =
c("#6caed1","#144c73","#8acf4e"))
```



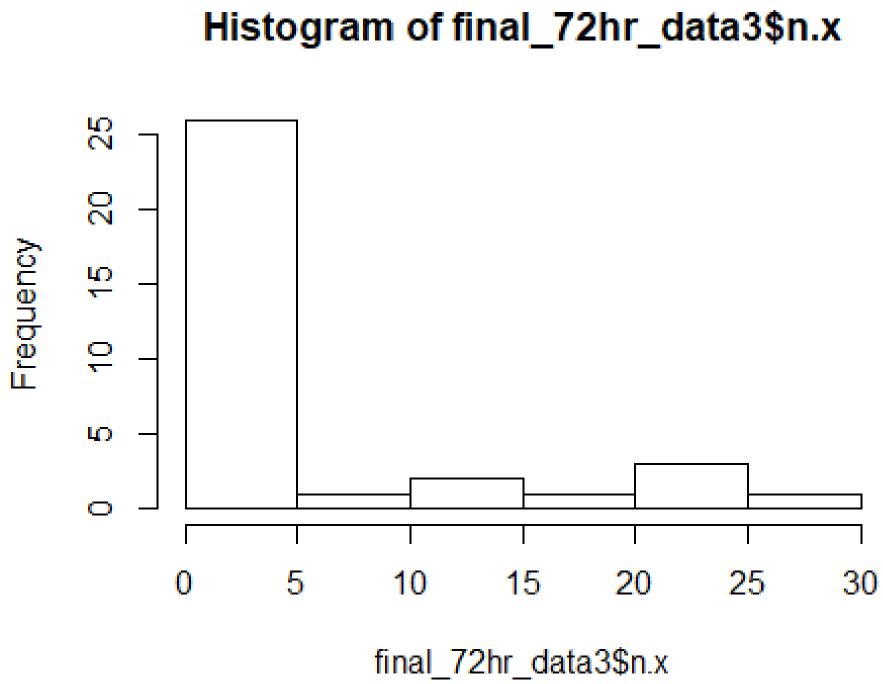

```
# produce boxplot
jpeg("E:/BL5599/high_res_plots/2D_boxplot_seasonal_index.jpg", width = 6,
height = 4, units = 'in', res = 600)
boxplotdou( Vessel_Index_72_Hourly ~ site, boxplot_seasonal,
Total_Images_With_Seal_Presence_72_Hourly  ~ site,
boxplot_seasonal,shading=500, factor.labels= F, name.on.axis=F, col =
c("#6caed1","#144c73","#8acf4e"))
dev.off()

## png
##   2

# O = outlier
# whiskers = min and max values exclude outliers
# shaded box = IQR
# intercept of whiskers = median


# double check that seal box plots for Site 1 and Site 3 correct
hist(final_72hr_data1$n.x)
```




```r
hist(final_72hr_data2$n.x)
```

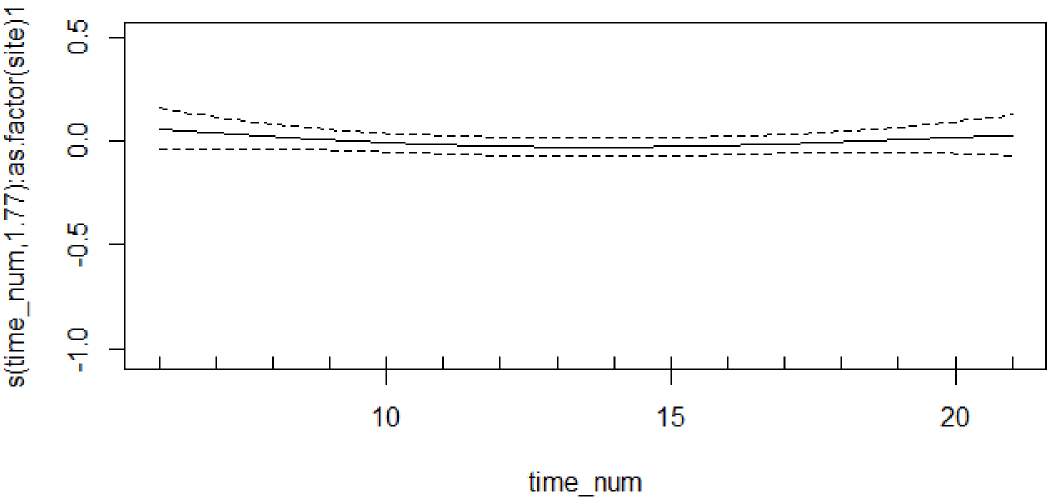


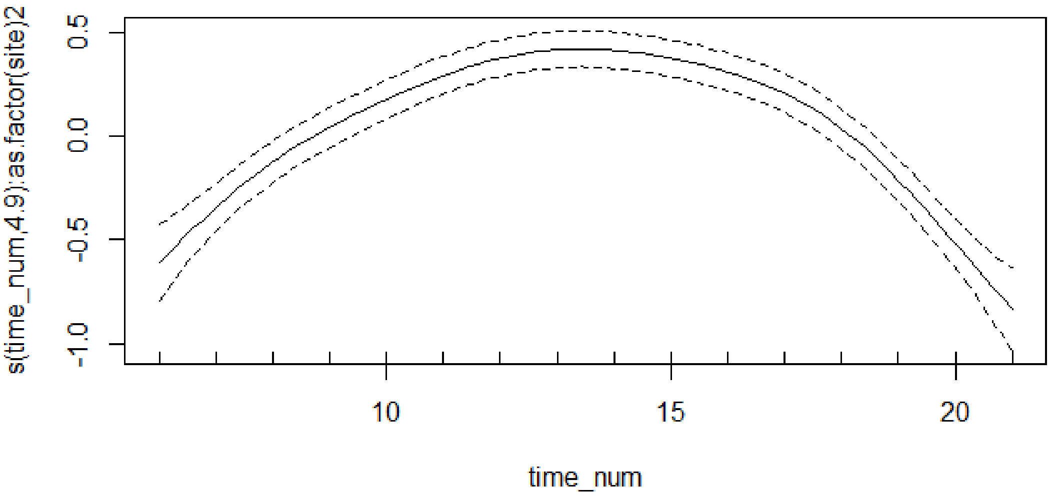

```r
hist(final_72hr_data3$n.x)
```

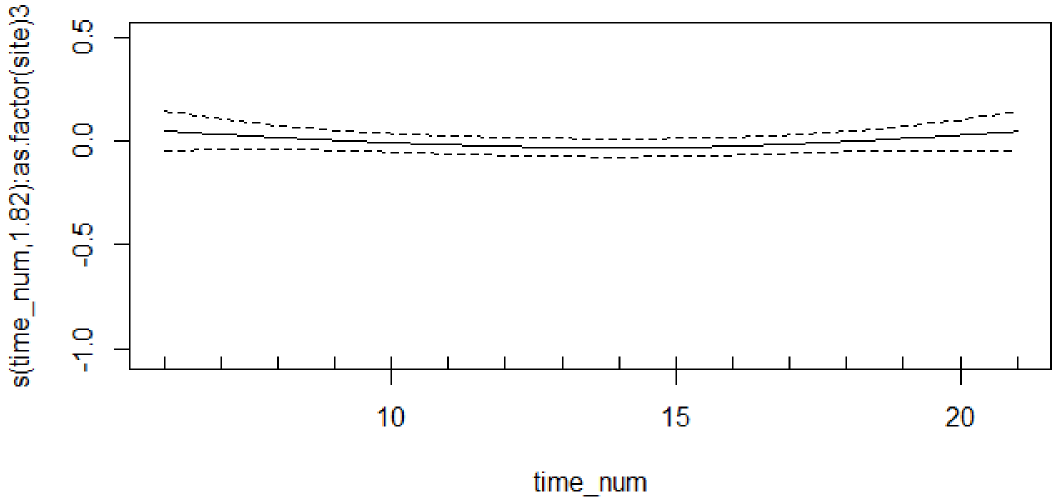


Plot Final Diel Model at Hourly Intervals

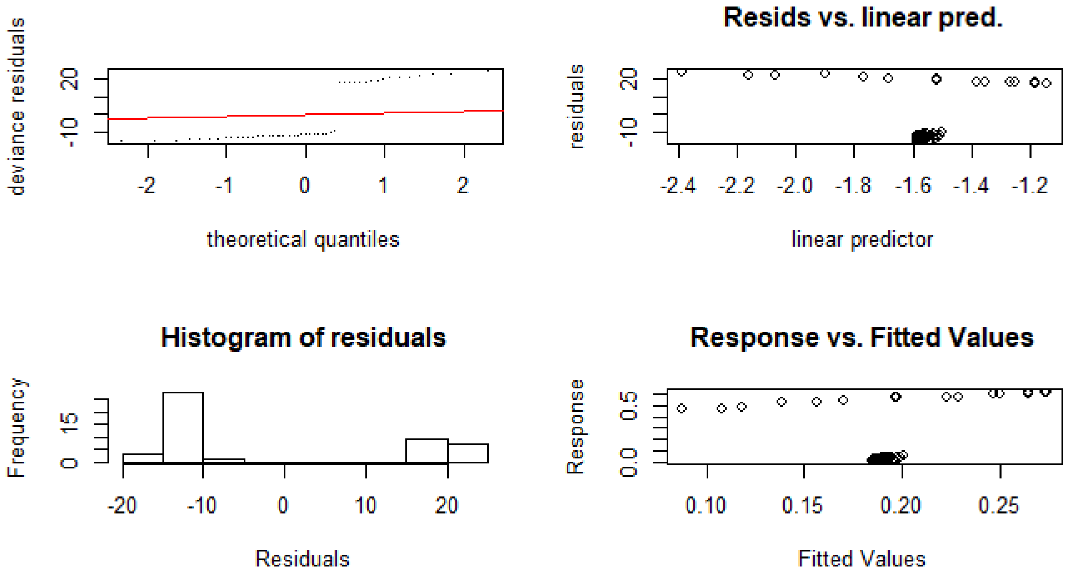
```
# load pacakges
library(mgcv)

## Loading required package: nlme

## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.

library(scales)

## Warning: package 'scales' was built under R version 3.6.3

# load data
all_data_hourly_6_22 <- read.csv('E:/BL5599/all_data_hourly_6_22.csv',
header = T)

# by time - %DE only 1.7%
hourly_binomial <- gam(cbind(successes,failures) ~ s(time_num, by =
as.factor(site)), data = all_data_hourly_6_22,
family=binomial(link="cloglog"))
plot.gam(hourly_binomial)
```


s(time_num,1.77):as.factor(site)1
0.5
0.0
-0.5
-1.0
10
15
20
time_num

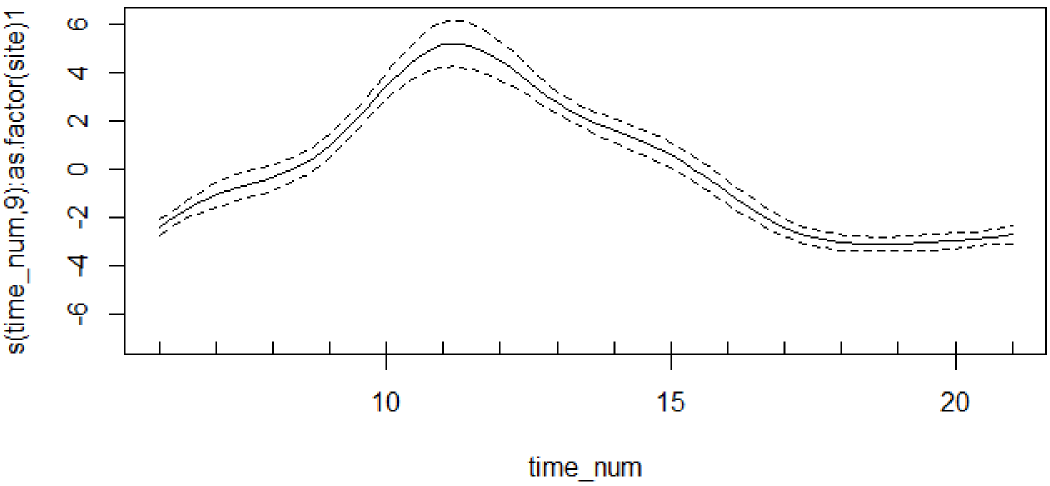

s(time_num,4.9):as.factor(site)2
0.5
0.0
-0.5
-1.0
10
15
20
time_num

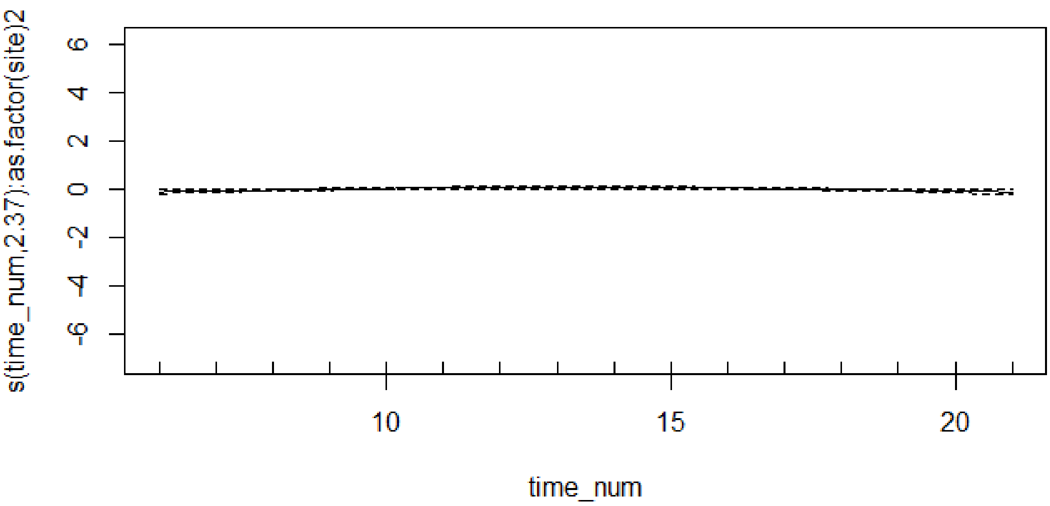


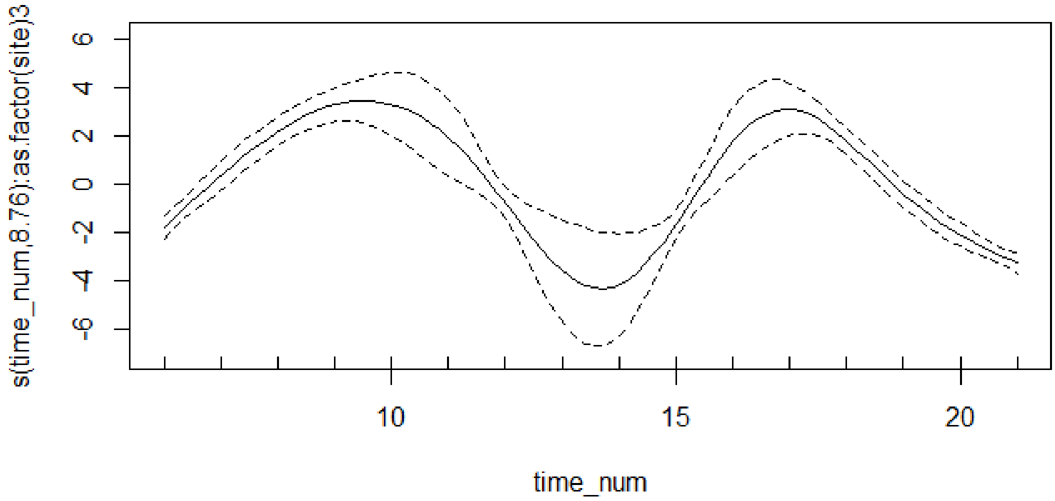

```
summary(hourly_binomial)

##
## Family: binomial
## Link function: cloglog
##
## Formula:
## cbind(successes, failures) ~ s(time_num, by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.55623    0.01341    -116   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                 edf Ref.df  Chi.sq p-value
## s(time_num):as.factor(site)1 1.770  2.205   2.367   0.375
## s(time_num):as.factor(site)2 4.897  5.985 223.168  <2e-16 ***
## s(time_num):as.factor(site)3 1.816  2.264   2.374   0.326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  -0.164   Deviance explained =  1.7%
## UBRE = 240.96  Scale est. = 1         n = 48

gam.check(hourly_binomial)
```

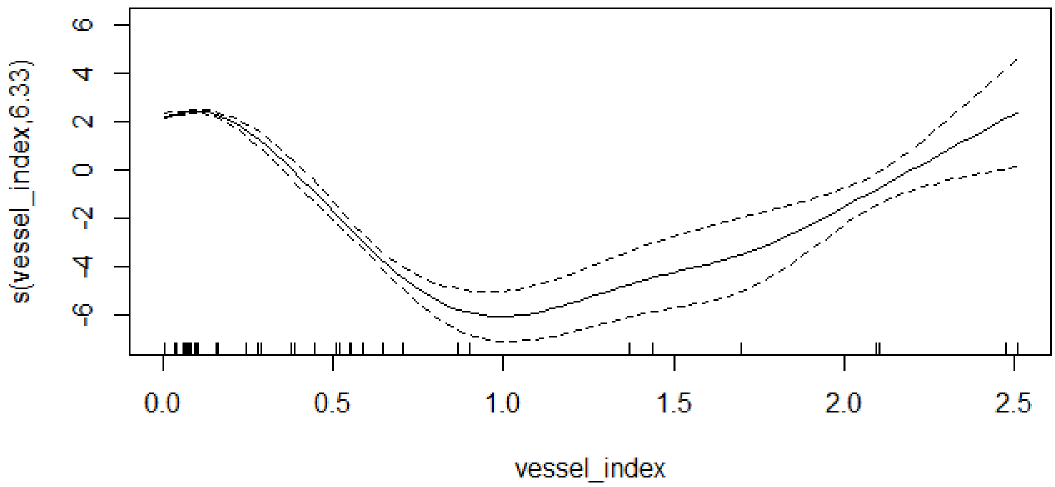



```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-3.778048e-09,4.540685e-06]
## (score 240.9572 & scale 1).
## Hessian positive definite, eigenvalue range [0.01383819,0.01785637].
## Model rank =  28 / 28
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                   k'  edf k-index p-value
## s(time_num):as.factor(site)1 9.00 1.77     1.5       1
## s(time_num):as.factor(site)2 9.00 4.90     1.5       1
## s(time_num):as.factor(site)3 9.00 1.82     1.5       1

# time by site + vessel index - %DE = 99.8%
hourly_binomial <- gam(cbind(successes,failures) ~ s(time_num, by =
as.factor(site)) + s(vessel_index), data = all_data_hourly_6_22,
family=binomial(link="cloglog"))
plot.gam(hourly_binomial)
```


s(time_num,9):as.factor(site)1
6
4
2
0
-2
-4
-6
10
15
20
time_num

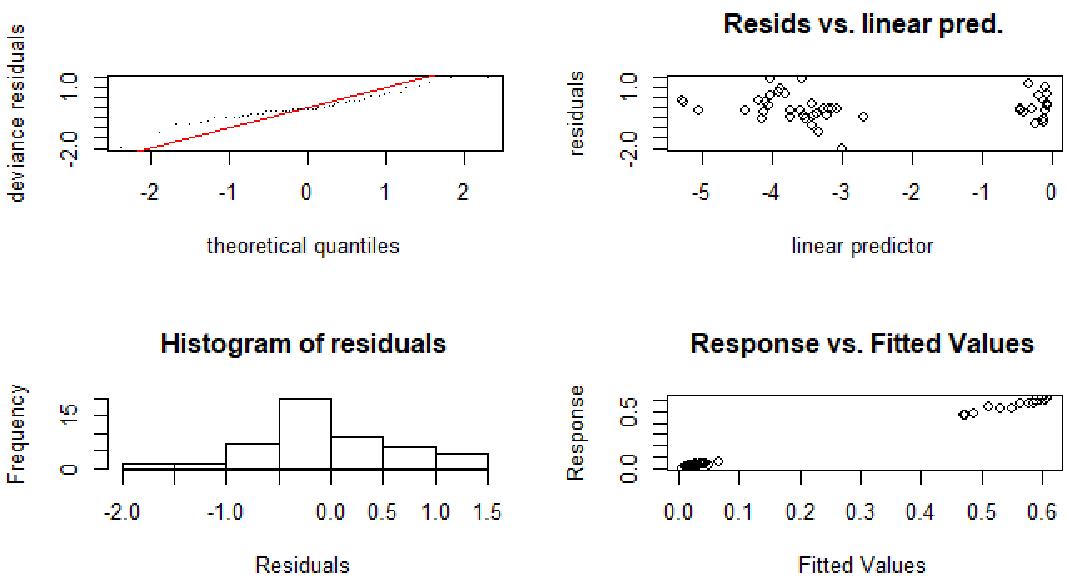

s(time_num,2.37):as.factor(site)2
6
4
2
0
-2
-4
-6
10
15
20
time_num





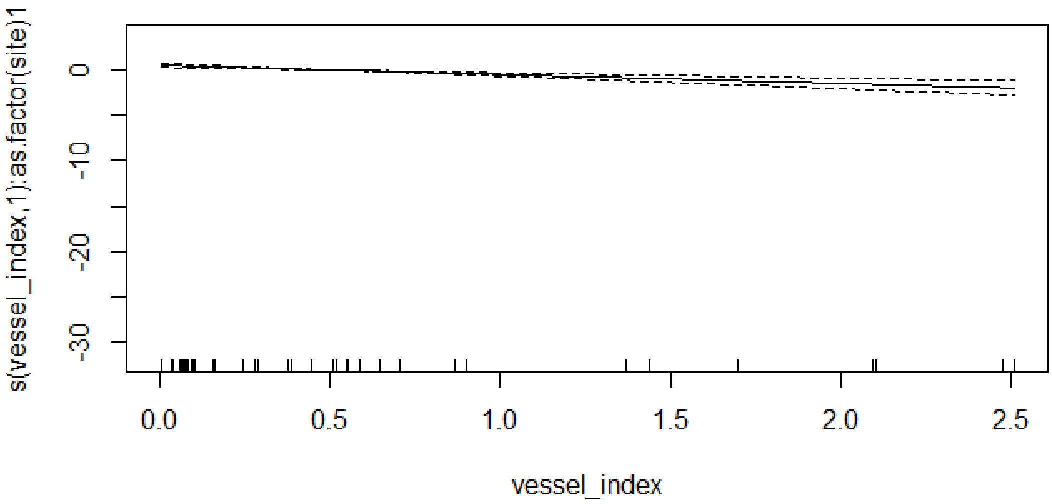

```
summary(hourly_binomial)

##
## Family: binomial
## Link function: cloglog
##
## Formula:
## cbind(successes, failures) ~ s(time_num, by = as.factor(site)) +
##     s(vessel_index)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.57709    0.03395  -75.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

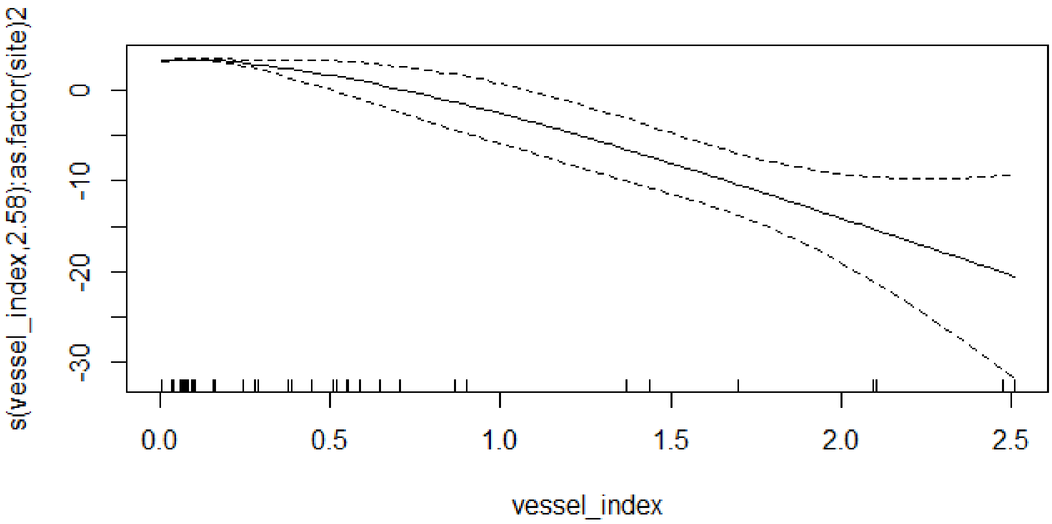
```
## Approximate significance of smooth terms:
##                                  edf Ref.df  Chi.sq p-value
## s(time_num):as.factor(site)1 9.000  9.000 1185.42 < 2e-16 ***
## s(time_num):as.factor(site)2 2.368  2.980   11.74 0.00726 **
## s(time_num):as.factor(site)3 8.761  8.906  400.75 < 2e-16 ***
## s(vessel_index)              6.334  7.369 4975.39 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.998   Deviance explained = 99.8%
## UBRE = 0.56678  Scale est. = 1         n = 48
gam.check(hourly_binomial)
```

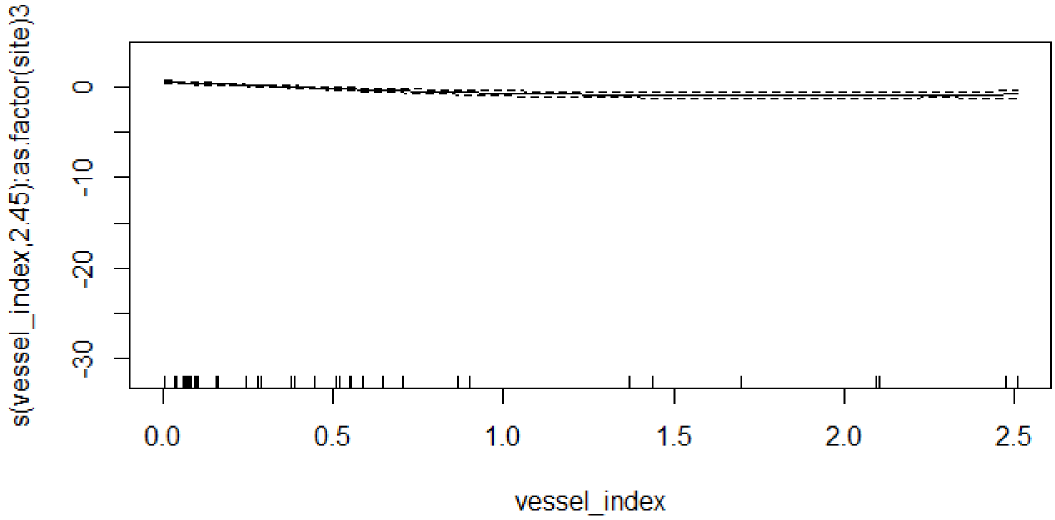


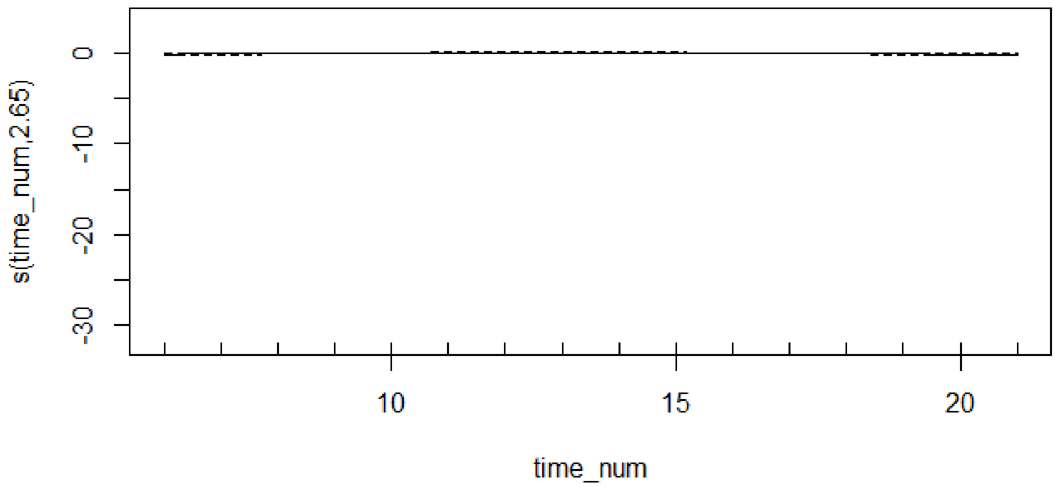

```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 15 iterations.
## Gradient range [-5.071815e-06,7.81974e-07]
## (score 0.566776 & scale 1).
## Hessian positive definite, eigenvalue range [7.981362e-07,0.05167343].
## Model rank =  37 / 37
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                k'  edf k-index p-value
## s(time_num):as.factor(site)1 9.00 9.00    1.40    1.00
## s(time_num):as.factor(site)2 9.00 2.37    1.40    1.00
## s(time_num):as.factor(site)3 9.00 8.76    1.40    1.00
## s(vessel_index)              9.00 6.33    1.12    0.76
# vessel index by site + time lowest UBRE, and best residual plots
hourly_binomial <- gam(cbind(successes,failures) ~ s(vessel_index, by =
as.factor(site)) + s(time_num), data = all_data_hourly_6_22,
```

```
family=binomial(link="cloglog"))
plot.gam(hourly_binomial)
```



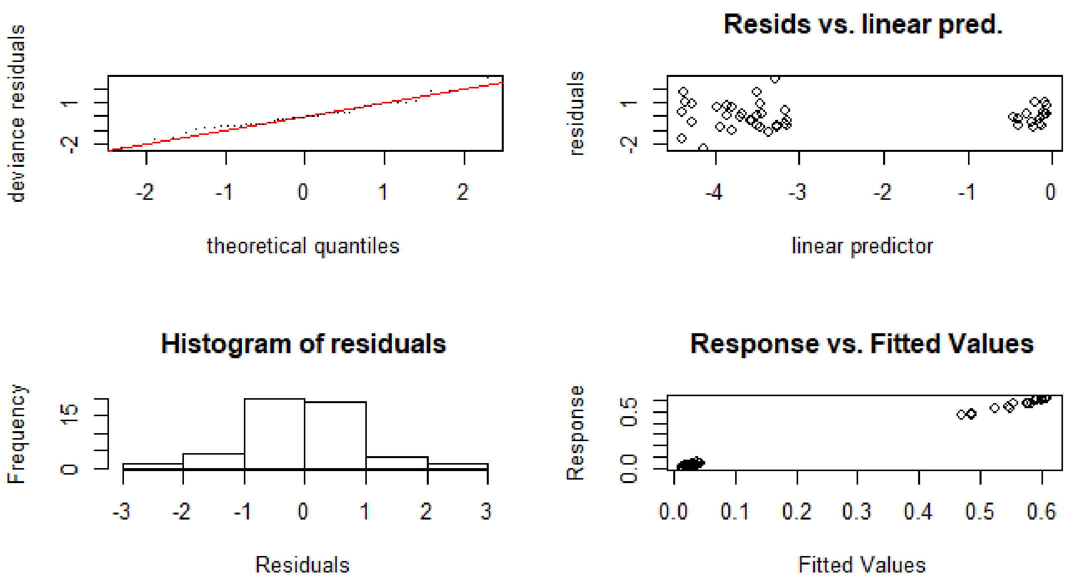






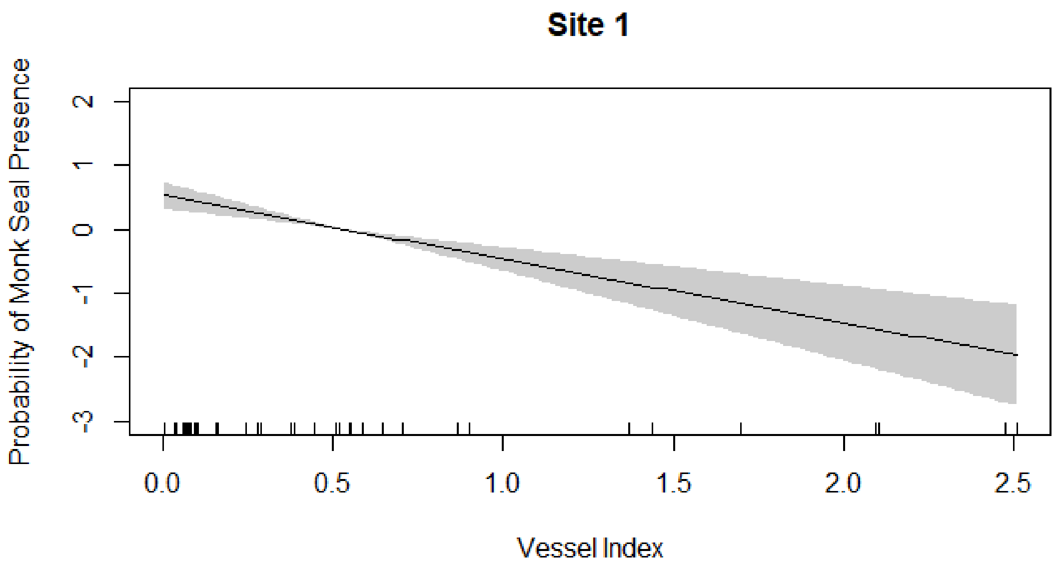

```
summary(hourly_binomial)

##
## Family: binomial
## Link function: cloglog
##
## Formula:
## cbind(successes, failures) ~ s(vessel_index, by = as.factor(site)) +
##     s(time_num)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.55545    0.05318  -66.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Approximate significance of smooth terms:
##                                       edf Ref.df  Chi.sq  p-value
## s(vessel_index):as.factor(site)1 1.000  1.000   24.97 5.83e-07 ***
## s(vessel_index):as.factor(site)2 2.581  2.798 3716.26  < 2e-16 ***
## s(vessel_index):as.factor(site)3 2.450  3.030   51.18 5.17e-11 ***
## s(time_num)                      2.654  3.332   14.24  0.00544 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.999   Deviance explained = 99.7%
## UBRE = 0.25566  Scale est. = 1         n = 48

gam.check(hourly_binomial)
```

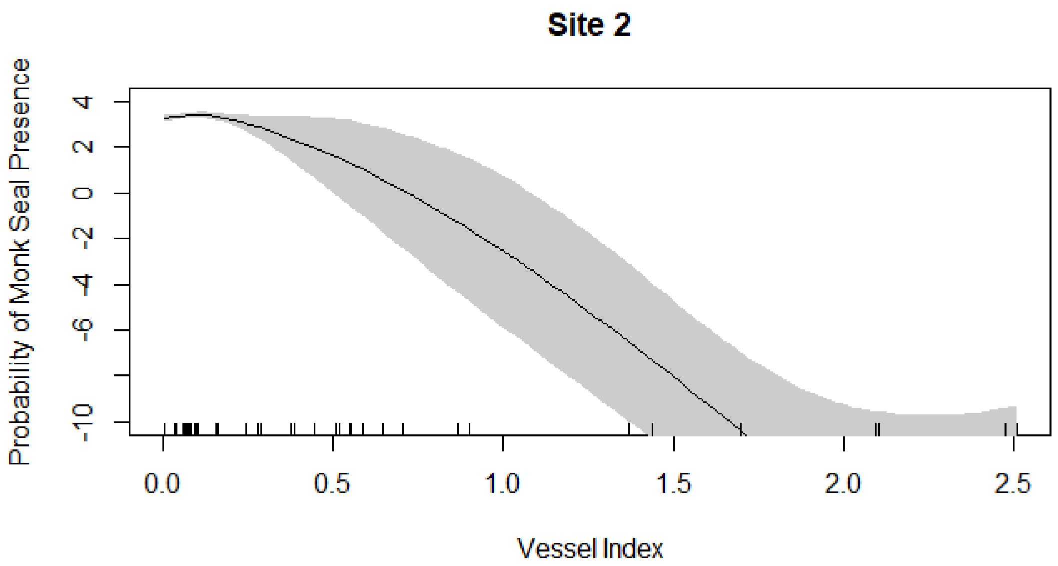



```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 10 iterations.
## Gradient range [-5.041927e-07,5.087856e-08]
## (score 0.2556646 & scale 1).
## Hessian positive definite, eigenvalue range [5.041793e-07,0.02350031].
## Model rank =  37 / 37
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                    k'  edf k-index p-value
## s(vessel_index):as.factor(site)1 9.00 1.00    1.13    0.78
## s(vessel_index):as.factor(site)2 9.00 2.58    1.13    0.80
## s(vessel_index):as.factor(site)3 9.00 2.45    1.13    0.76
## s(time_num)                      9.00 2.65    0.81    0.12

# use final model for plots and predictions

# plot hourly bernoulli model
plot.gam(hourly_binomial, select = 1, ylim=c(-3,2), shade = T,
```

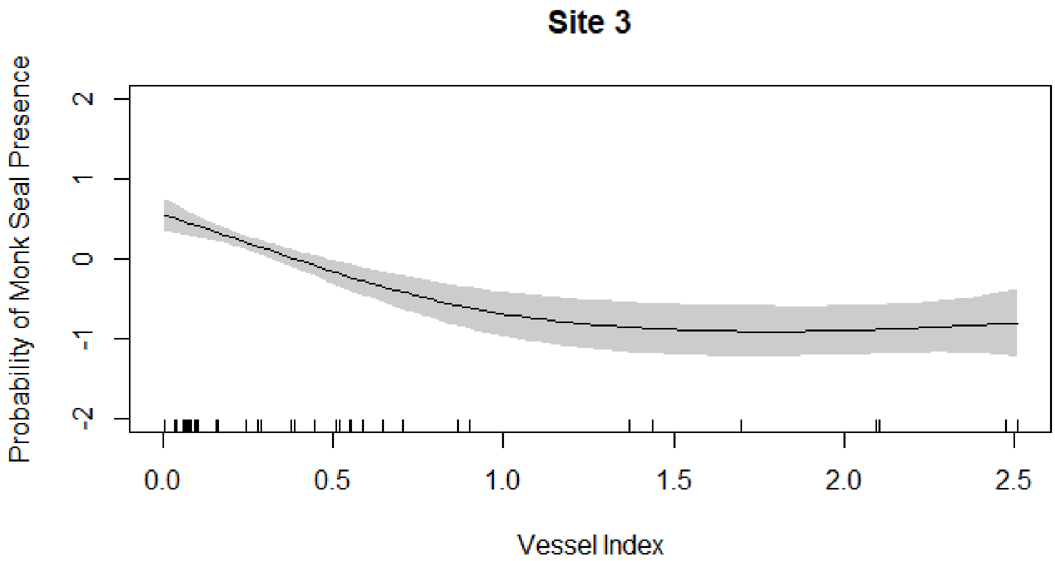
```
xlab="Vessel Index", ylab="Probability of Monk Seal Presence", main =
"Site 1")
```



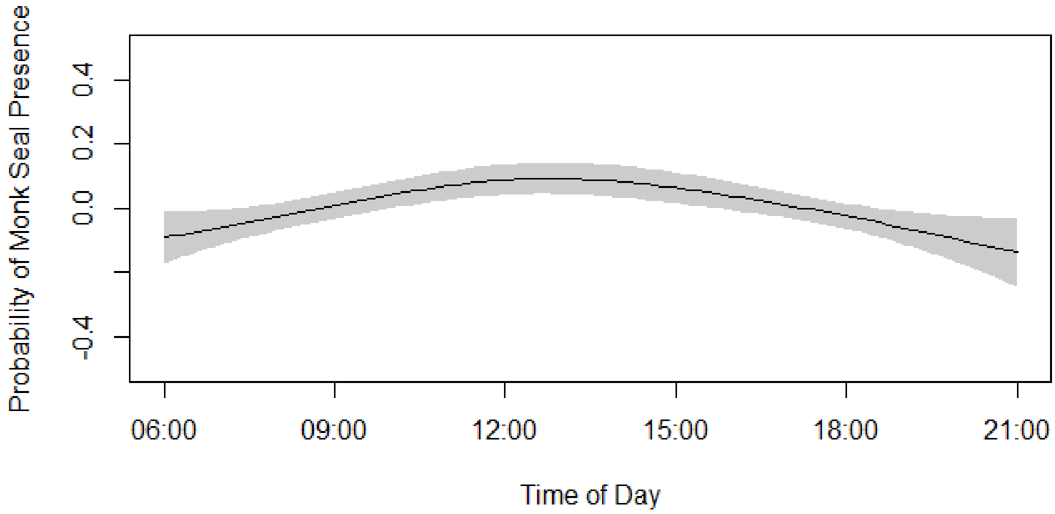

```
plot.gam(hourly_binomial, select = 2, ylim=c(-10,4), shade = T,
xlab="Vessel Index", ylab="Probability of Monk Seal Presence",main="Site
2")
```




```
plot.gam(hourly_binomial, select = 3, ylim=c(-2,2), shade = T,
xlab="Vessel Index", ylab="Probability of Monk Seal Presence", main="Site
3")
```




```
plot.gam(hourly_binomial, select = 4, ylim=c(-0.5,0.5), shade = T,
xlab="Time of Day", ylab="Probability of Monk Seal Presence", xaxt="n")
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
```

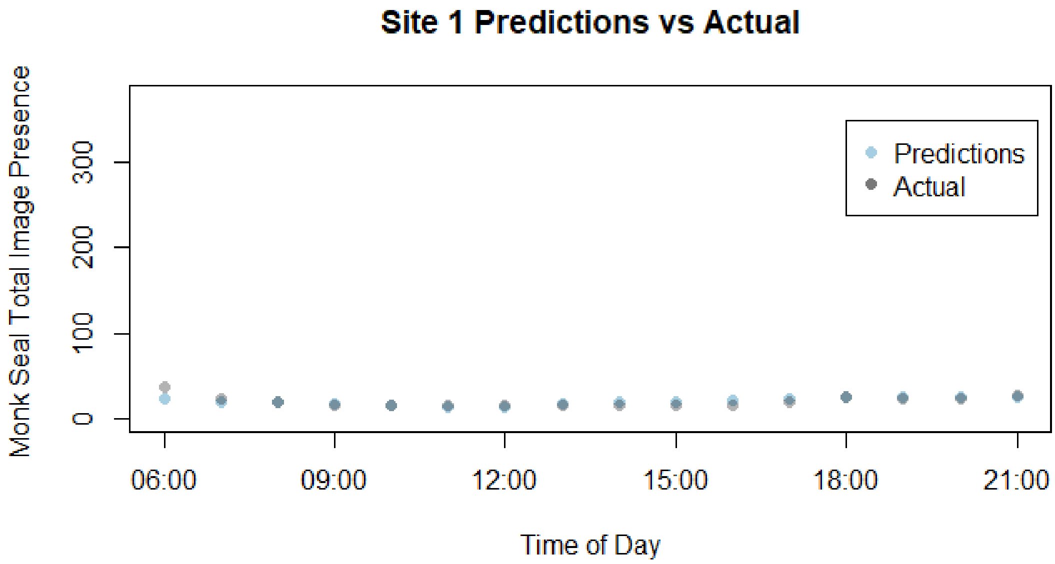



```
# save high quality plots
jpeg("E:/BL5599/high_res_plots/hourly_GAM_1.jpg", width = 6, height = 4,
units = 'in', res = 600)
plot.gam(hourly_binomial, select = 1, ylim=c(-3,2), shade = T,
xlab="Vessel Index", ylab="Probability of Monk Seal Presence", main =
"Site 1")
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()
```

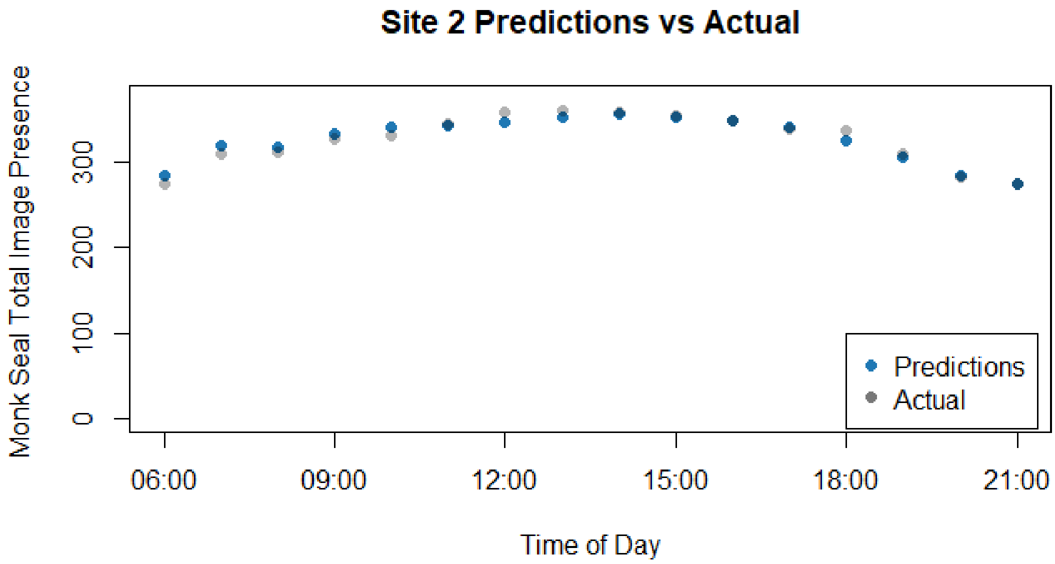
```
## png
##   2

jpeg("E:/BL5599/high_res_plots/hourly_GAM_2.jpg", width = 6, height = 4,
units = 'in', res = 600)
plot.gam(hourly_binomial, select = 2, ylim=c(-10,4), shade = T,
xlab="Vessel Index", ylab="Probability of Monk Seal Presence",main="Site
2")
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/hourly_GAM_3.jpg", width = 6, height = 4,
units = 'in', res = 600)
plot.gam(hourly_binomial, select = 3, ylim=c(-2,2), shade = T,
xlab="Vessel Index", ylab="Probability of Monk Seal Presence", main="Site
3")
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/hourly_GAM_4.jpg", width = 6, height = 4,
units = 'in', res = 600)
plot.gam(hourly_binomial, select = 4, ylim=c(-0.5,0.5), shade = T,
xlab="Time of Day", ylab="Probability of Monk Seal Presence", xaxt="n")
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()

## png
##   2

# make predictions
# subset dataset used in model into site
all_data_hourly_6_22_s1 <- subset(all_data_hourly_6_22, site == 1)
all_data_hourly_6_22_s2 <- subset(all_data_hourly_6_22, site == 2)
all_data_hourly_6_22_s3 <- subset(all_data_hourly_6_22, site == 3)

# make model predictions
hourly_predict_s1 <- predict(hourly_binomial,
all_data_hourly_6_22_s1,type='response', se.fit = T)
hourly_predict_s2 <- predict(hourly_binomial,
all_data_hourly_6_22_s2,type='response', se.fit = T)
hourly_predict_s3 <- predict(hourly_binomial,
all_data_hourly_6_22_s3,type='response', se.fit = T)

# create dataframe
hourly_predict_s1 <- data.frame(hourly_predict_s1)
hourly_predict_s2 <- data.frame(hourly_predict_s2)
```

```
hourly_predict_s3 <- data.frame(hourly_predict_s3)

# add total number of trials (eg total images taken for that time)
hourly_predict_s1$trials <- all_data_hourly_6_22_s1$trials
hourly_predict_s2$trials <- all_data_hourly_6_22_s2$trials
hourly_predict_s3$trials <- all_data_hourly_6_22_s3$trials

# add time of day number
hourly_predict_s1$time_num <- all_data_hourly_6_22_s1$time_num
hourly_predict_s2$time_num <- all_data_hourly_6_22_s2$time_num
hourly_predict_s3$time_num <- all_data_hourly_6_22_s3$time_num

# calculate total number of photos of monk seal presence to compare with
actual data
hourly_predict_s1$photos <- hourly_predict_s1$fit *
hourly_predict_s1$trials
hourly_predict_s2$photos <- hourly_predict_s2$fit *
hourly_predict_s2$trials
hourly_predict_s3$photos <- hourly_predict_s3$fit *
hourly_predict_s3$trials

# round to whole number of photos
hourly_predict_s1$photos <- round(hourly_predict_s1$photos, digits = 0)
hourly_predict_s2$photos <- round(hourly_predict_s2$photos, digits = 0)
hourly_predict_s3$photos <- round(hourly_predict_s3$photos, digits = 0)

# calculate upper bounds
hourly_predict_s1$photos_upper <-
(hourly_predict_s1$fit+hourly_predict_s1$se.fit) *
hourly_predict_s1$trials
hourly_predict_s2$photos_upper <-
(hourly_predict_s2$fit+hourly_predict_s2$se.fit) *
hourly_predict_s2$trials
hourly_predict_s3$photos_upper <-
(hourly_predict_s3$fit+hourly_predict_s3$se.fit) *
hourly_predict_s3$trials


# calculate low bounds
hourly_predict_s1$photos_lower <- (hourly_predict_s1$fit-
hourly_predict_s1$se.fit) * hourly_predict_s1$trials
hourly_predict_s2$photos_lower <-  (hourly_predict_s2$fit-
hourly_predict_s2$se.fit) * hourly_predict_s2$trials
hourly_predict_s3$photos_lower <- (hourly_predict_s3$fit-
hourly_predict_s3$se.fit) * hourly_predict_s3$trials

# find maximum number of images where seals occur for any given hour for
ylim
max(all_data_hourly_6_22$successes)

## [1] 362

# [1] 362 - round to 375
```

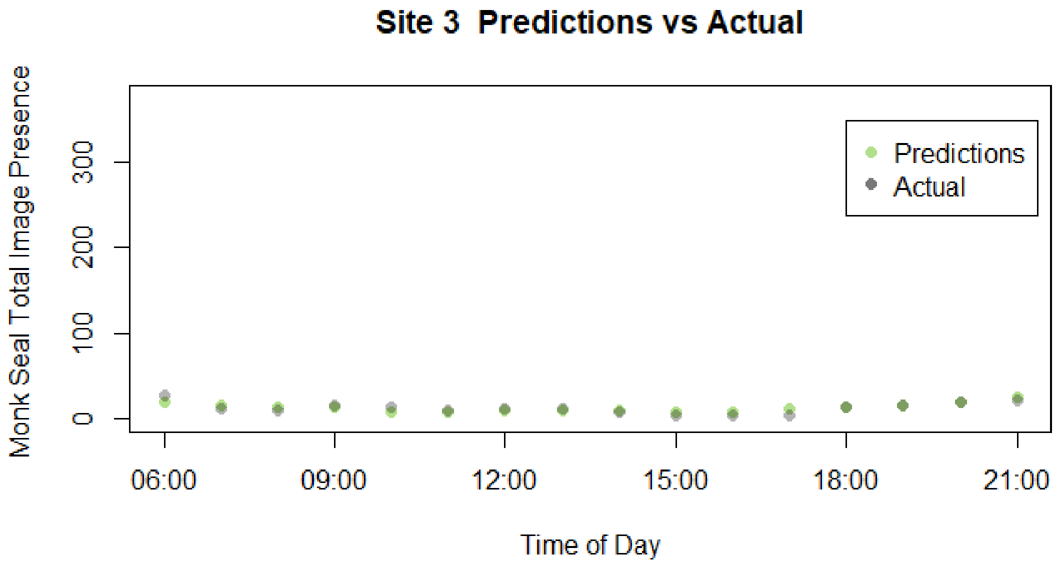
```
# plot predictions - Site 1
plot(hourly_predict_s1$time_num, hourly_predict_s1$photos,ylim=c(0,375),
col= "#A6CEE3", pch=16, main = "Site 1 Predictions vs Actual",  xlab="Time
of Day", ylab ="Monk Seal Total Image Presence ",xaxt="n")
# plot actual data
points(all_data_hourly_6_22_s1$time_num,
all_data_hourly_6_22_s1$successes, col=alpha("black",0.3), pch=16)
legend(18,350, legend=c("Predictions", "Actual"), col=c("#A6CEE3",
"grey47"), pch=16)
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
```




```
# prediction intervals tried but look extremely messy on plot over both
predictions and actual
# very narrow over points
#lines(all_data_hourly_6_22_s1$time_num, hourly_predict_s1$photos_lower,
lty = 2)
#lines(all_data_hourly_6_22_s1$time_num, hourly_predict_s1$photos_upper,
lty = 2)


# plot predictions - Site 2
plot(hourly_predict_s2$time_num, hourly_predict_s2$photos,ylim=c(0,375),
col= "#1F78B4", pch=16, main = "Site 2 Predictions vs Actual",  xlab="Time
of Day", ylab ="Monk Seal Total Image Presence ",xaxt="n")
# plot actual data
points(all_data_hourly_6_22_s2$time_num,
all_data_hourly_6_22_s2$successes, col=alpha("black",0.3), pch=16)
legend(18,100, legend=c("Predictions", "Actual"), col=c("#1F78B4",
"grey47"), pch=16)
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
```




```
# prediction intervals tried but look extremely messy on plot over both
predictions and actual
# narrow over points, not quite as narrow as site 1 and site 3
#lines(all_data_hourly_6_22_s2$time_num, hourly_predict_s2$photos_lower,
lty = 2)
#lines(all_data_hourly_6_22_s2$time_num, hourly_predict_s2$photos_upper,
lty = 2)


# plot predictions - Site 3
plot(hourly_predict_s3$time_num, hourly_predict_s3$photos,ylim=c(0,375),
col= "#B2DF8A", pch=16, main = "Site 3  Predictions vs Actual",
xlab="Time of Day", ylab ="Monk Seal Total Image Presence ",xaxt="n")
# plot actual data
points(all_data_hourly_6_22_s3$time_num,
all_data_hourly_6_22_s3$successes, col=alpha("black",0.3), pch=16)
legend(18,350, legend=c("Predictions", "Actual"), col=c("#B2DF8A",
"grey47"), pch=16)
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
```

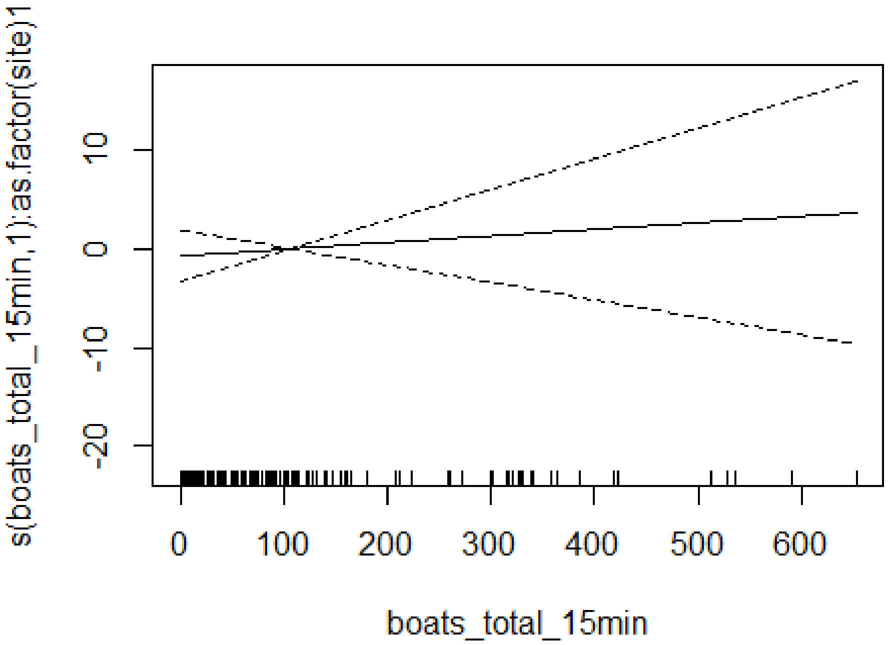


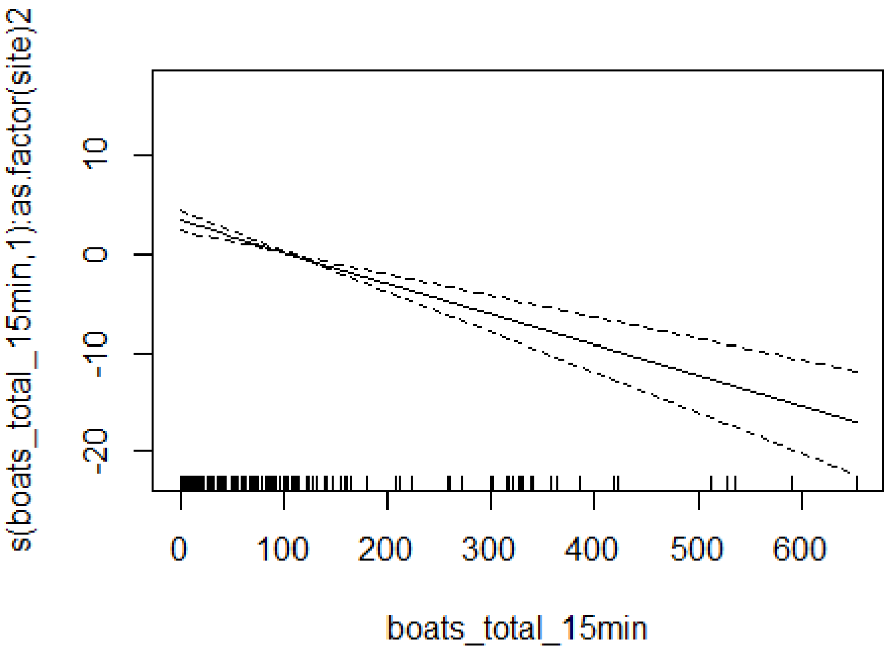

```
# prediction intervals tried but look extremely messy on plot over both
predictions and actual
# very narrow over points
# lines(all_data_hourly_6_22_s3$time_num, hourly_predict_s3$photos_lower,
lty = 2)
# lines(all_data_hourly_6_22_s3$time_num, hourly_predict_s3$photos_upper,
lty = 2)

# Site 1
jpeg("E:/BL5599/high_res_plots/hourly_predict_GAM_1.jpg", width = 6,
height = 4, units = 'in', res = 600)
plot(hourly_predict_s1$time_num, hourly_predict_s1$photos,ylim=c(0,375),
col= "#A6CEE3", pch=16, main = "Site 1 Predictions vs Actual",  xlab="Time
of Day", ylab ="Monk Seal Total Image Presence ",xaxt="n")
points(all_data_hourly_6_22_s1$time_num,
all_data_hourly_6_22_s1$successes, col=alpha("black",0.3), pch=16)
legend(17,350, legend=c("Predictions", "Actual"), col=c("#A6CEE3",
"grey47"), pch=16)
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()

## png
##   2

# Site 2
jpeg("E:/BL5599/high_res_plots/hourly_predict_GAM_2.jpg", width = 6,
height = 4, units = 'in', res = 600)
plot(hourly_predict_s2$time_num, hourly_predict_s2$photos,ylim=c(0,375),
col= "#1F78B4", pch=16, main = "Site 2 Predictions vs Actual",  xlab="Time
of Day", ylab ="Monk Seal Total Image Presence ",xaxt="n")
points(all_data_hourly_6_22_s2$time_num,
all_data_hourly_6_22_s2$successes, col=alpha("black",0.3), pch=16)
legend(17,120, legend=c("Predictions", "Actual"), col=c("#1F78B4",
"grey47"), pch=16)
```

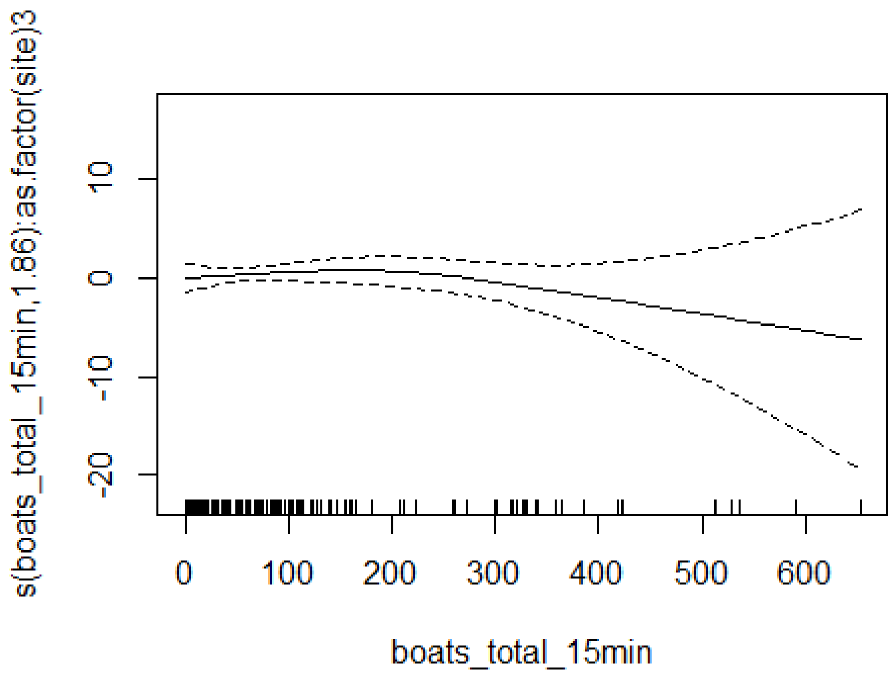
```
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()

## png
##   2

# Site 3
jpeg("E:/BL5599/high_res_plots/hourly_predict_GAM_3.jpg", width = 6,
height = 4, units = 'in', res = 600)
plot(hourly_predict_s3$time_num, hourly_predict_s3$photos,ylim=c(0,375),
col= "#B2DF8A", pch=16, main = "Site 3  Predictions vs Actual",
xlab="Time of Day", ylab ="Monk Seal Total Image Presence ",xaxt="n")
points(all_data_hourly_6_22_s3$time_num,
all_data_hourly_6_22_s3$successes, col=alpha("black",0.3), pch=16)
legend(17,350, legend=c("Predictions", "Actual"), col=c("#B2DF8A",
"grey47"), pch=16)
axis(side=1, at=c(6,9,12,15,18,21),labels =
c("06:00","09:00","12:00","15:00","18:00","21:00"), cex.axis=1)
dev.off()

## png
##   2
```

Plot initial 72hr GAMS to account for autocorrelation

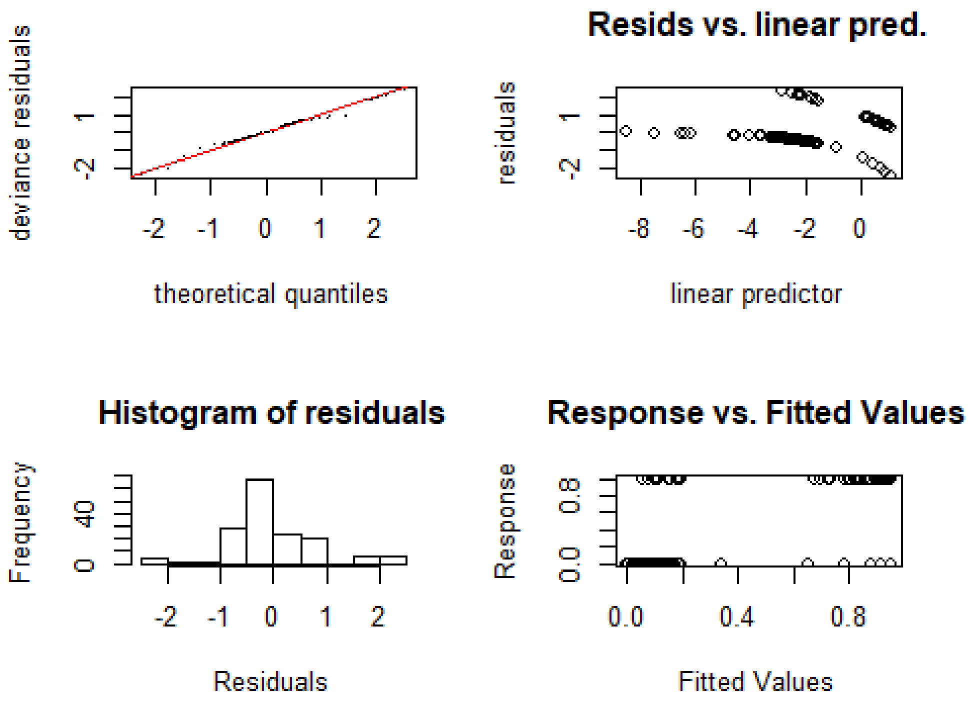
```
library(mgcv)
all_data_72 <- read.csv("E:/BL5599/vessel_seal_data_72hrs.csv", header =
T)
all_data_hourly <- read.csv("E:/BL5599/all_data_hourly.csv", header = T)

# remove May and Nov
all_data_72_Jun_Sept <- subset(all_data_72, data_72hr >= 4 & data_72hr <=
44)

# test models with new data
seals_vessels_date <- gam(presence_new ~ s(boats_total_15min,
by=as.factor(site)), data = all_data_72, family=binomial(link="cloglog"))
plot(seals_vessels_date)
```


s(boats_total_15min,1):as.factor(site)1
10
0
-10
-20
0
100
200
300
400
500
600
boats_total_15min

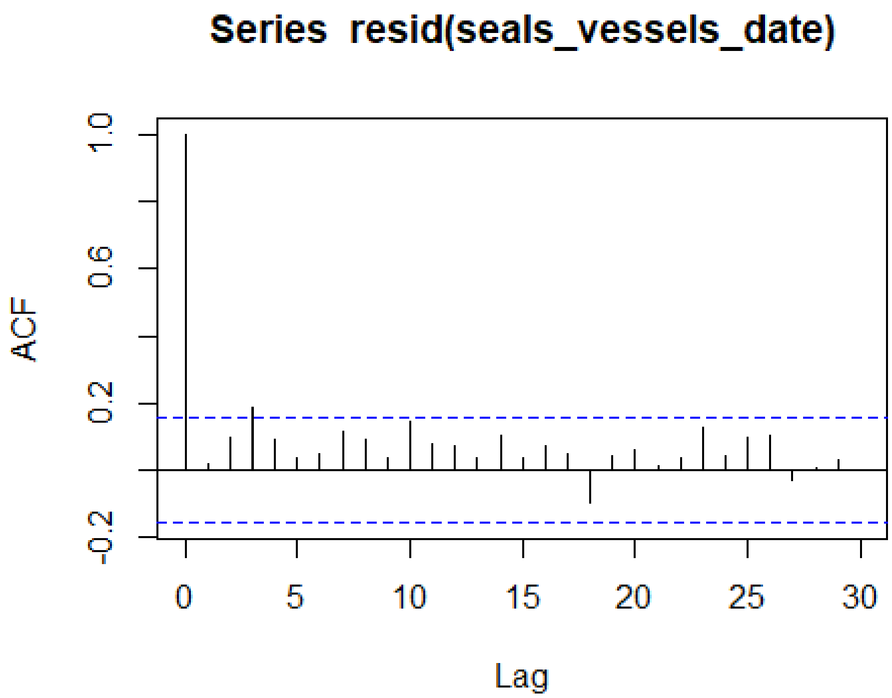

s(boats_total_15min,1):as.factor(site)2
10
0
-10
-20
0
100
200
300
400
500
600
boats_total_15min



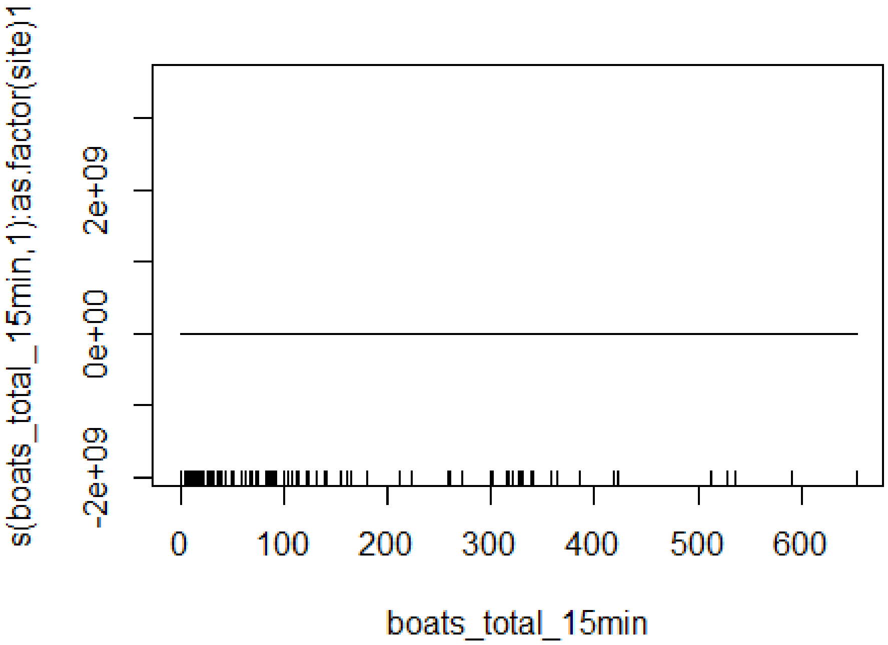

```
summary(seals_vessels_date)
##
## Family: binomial
## Link function: cloglog
##
## Formula:
## presence_new ~ s(boats_total_15min, by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.2574     0.4162  -5.423 5.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                         edf Ref.df Chi.sq  p-value
## s(boats_total_15min):as.factor(site)1 1.000  1.000  0.303    0.582
## s(boats_total_15min):as.factor(site)2 1.000  1.000 42.965 5.58e-11 ***
## s(boats_total_15min):as.factor(site)3 1.859  2.319  2.304    0.334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.593   Deviance explained =   51%
## UBRE = -0.30774  Scale est. = 1         n = 157
gam.check(seals_vessels_date)
```

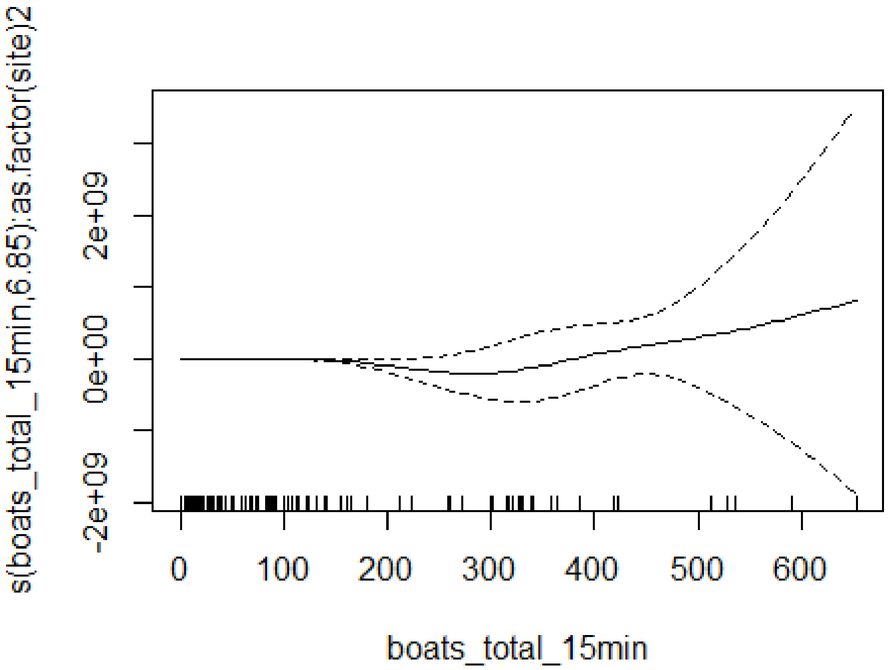


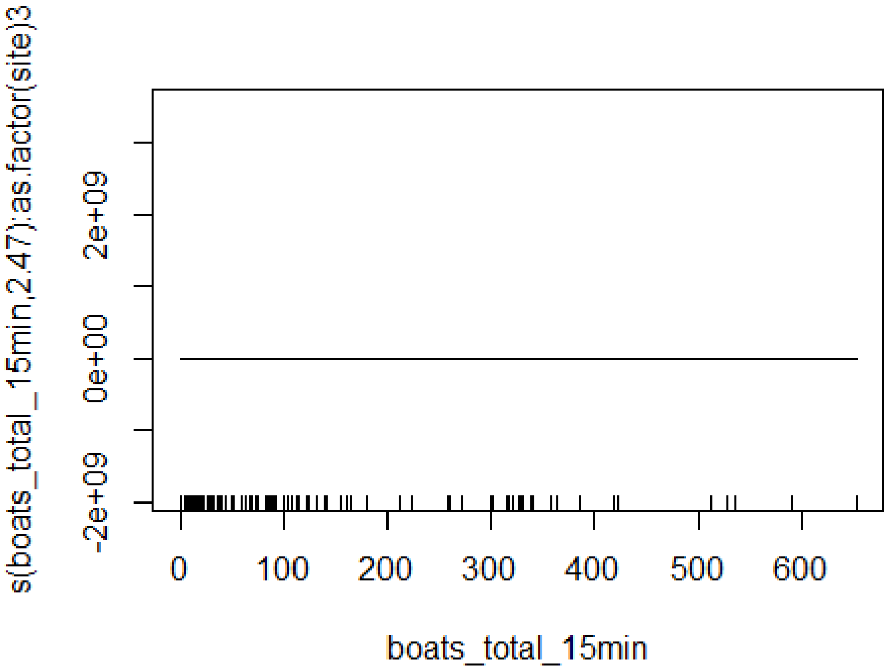

```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 9 iterations.
## Gradient range [-3.487135e-07,1.034332e-07]
## (score -0.307738 & scale 1).
## Hessian positive definite, eigenvalue range [1.139367e-07,0.004112683].
## Model rank =  28 / 28
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                           k'  edf k-index p-value
## s(boats_total_15min):as.factor(site)1 9.00 1.00     0.9   0.125
## s(boats_total_15min):as.factor(site)2 9.00 1.00     0.9   0.075 .
## s(boats_total_15min):as.factor(site)3 9.00 1.86     0.9   0.085 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
ACF  <-  acf(resid(seals_vessels_date), lag.max = 30)
```

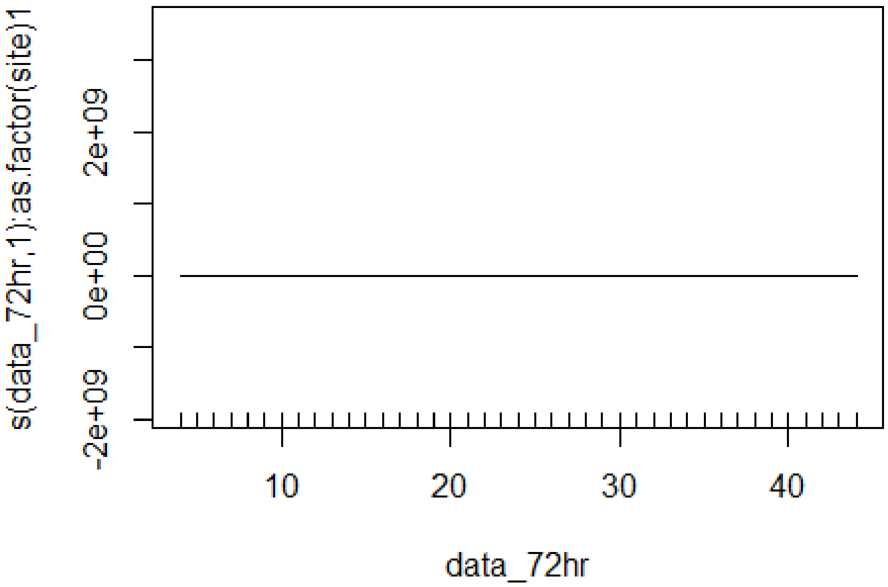


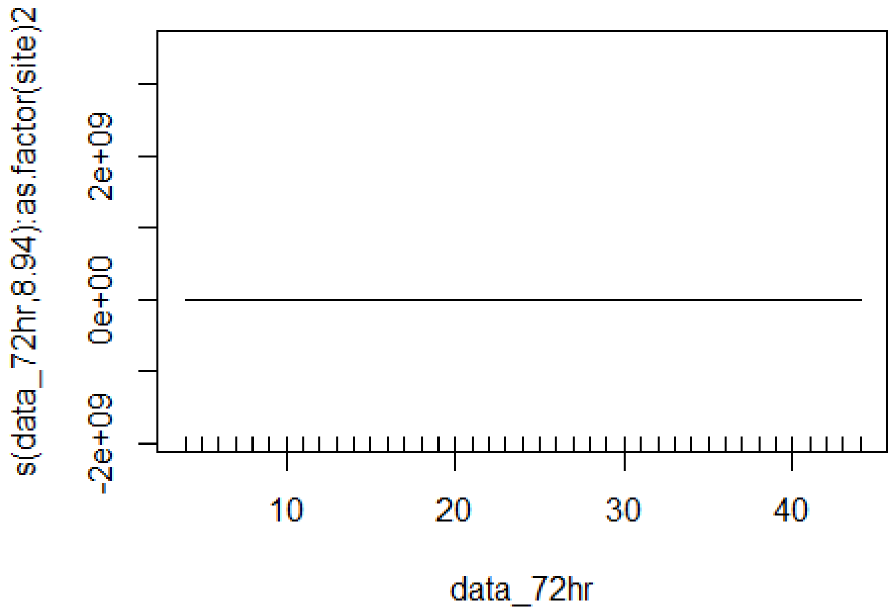

```
# include seasonal trends aswell with data_72hr variable (consecutively
numbered 72hr group)
seals_vessels_date <- gam(cbind(n.x,n.y) ~ s(boats_total_15min,
by=as.factor(site)) + s(data_72hr,by=as.factor(site)), data =
all_data_72_Jun_Sept, family=binomial(link="cloglog"))
plot(seals_vessels_date)
```

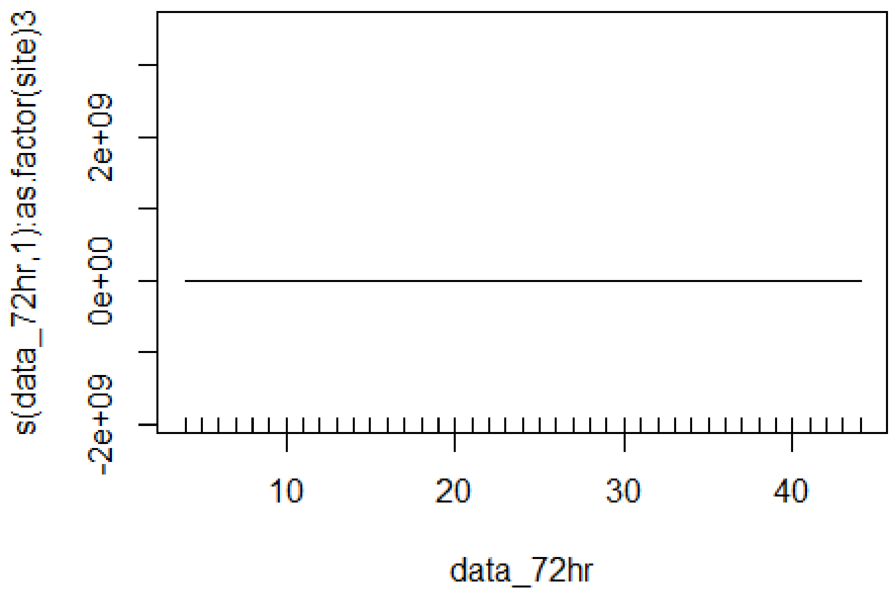

s(boats_total_15min,1):as.factor(site)1
-2e+09
0e+00
2e+09
0
100
200
300
400
500
600
boats_total_15min


s(boats_total_15min,6.85):as.factor(site)2
-2e+09
0e+00
2e+09
0
100
200
300
400
500
600
boats_total_15min


s(boats_total_15min,2.47):as.factor(site)3
-2e+09
0e+00
2e+09
0
100
200
300
400
500
600
boats_total_15min

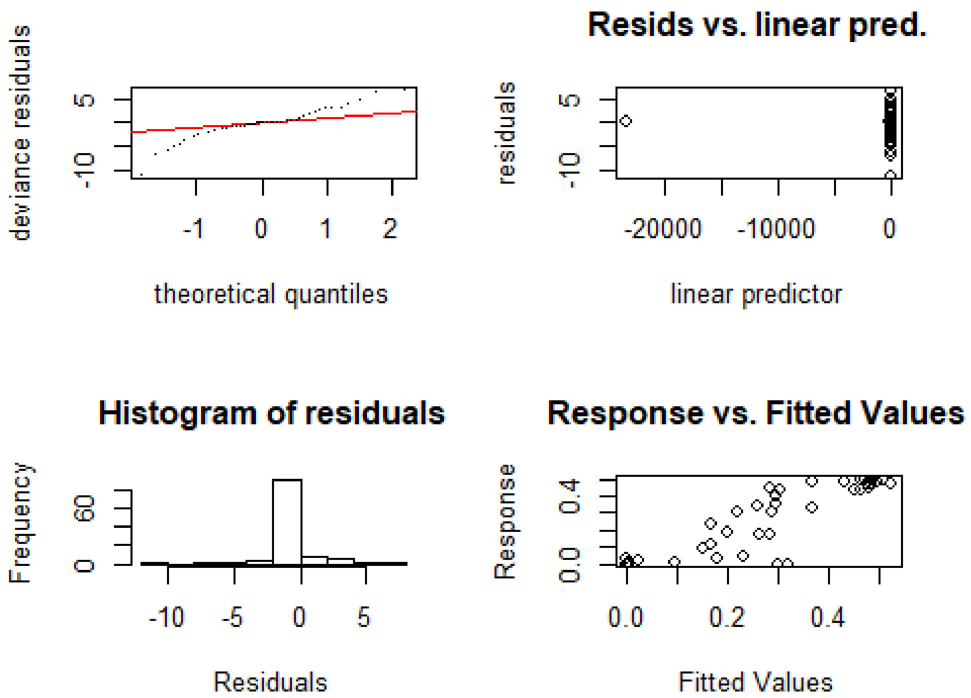

s(data_72hr,1):as.factor(site)1
-2e+09
0e+00
2e+09
10
20
30
40
data_72hr





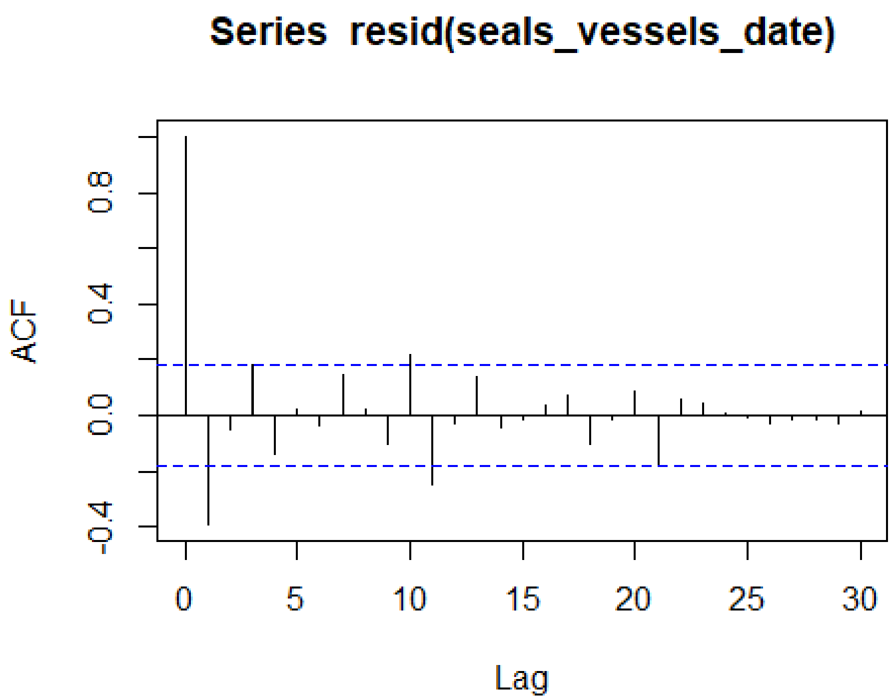

```
summary(seals_vessels_date)
##
## Family: binomial
## Link function: cloglog
```

```
##
## Formula:
## cbind(n.x, n.y) ~ s(boats_total_15min, by = as.factor(site)) +
##     s(data_72hr, by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -25.19      32.03  -0.786    0.432
##
## Approximate significance of smooth terms:
##                                        edf Ref.df  Chi.sq  p-value
## s(boats_total_15min):as.factor(site)1 1.000  1.000   0.000    1.000
## s(boats_total_15min):as.factor(site)2 6.850  6.983  59.581 1.86e-10 ***
## s(boats_total_15min):as.factor(site)3 2.473  2.906   0.814    0.800
## s(data_72hr):as.factor(site)1         1.000  1.000   0.000    1.000
## s(data_72hr):as.factor(site)2         8.941  8.998 481.378  < 2e-16 ***
## s(data_72hr):as.factor(site)3         1.000  1.000   0.421    0.516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.905   Deviance explained = 93.7%
## UBRE = 3.5269  Scale est. = 1         n = 116
gam.check(seals_vessels_date)
```

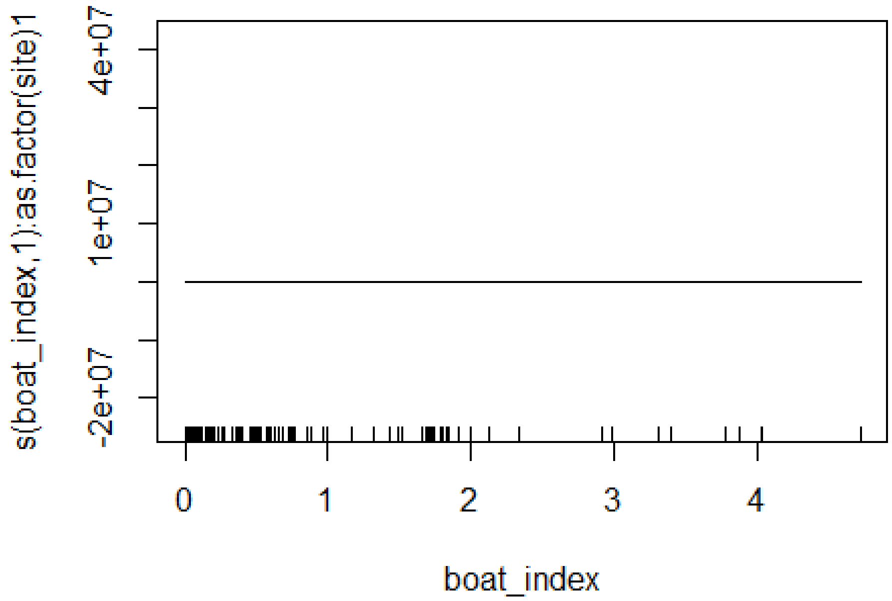


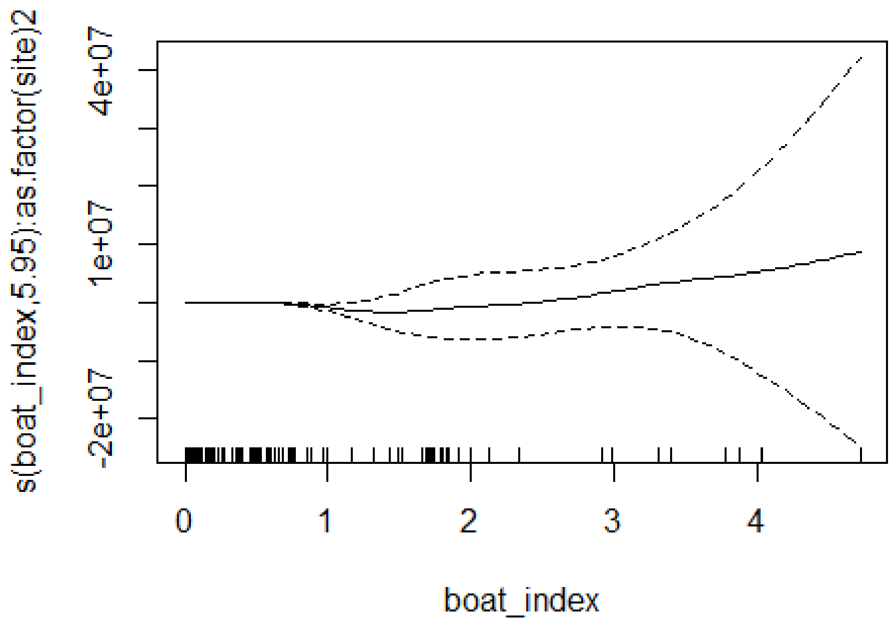

```
##
## Method: UBRE   Optimizer: outer newton
## step failed after 41 iterations.
## Gradient range [-0.0001288648,9.006486e-07]
## (score 3.52687 & scale 1).
## Hessian positive definite, eigenvalue range [1.030325e-09,27.49756].
## Model rank =  55 / 55
##
```

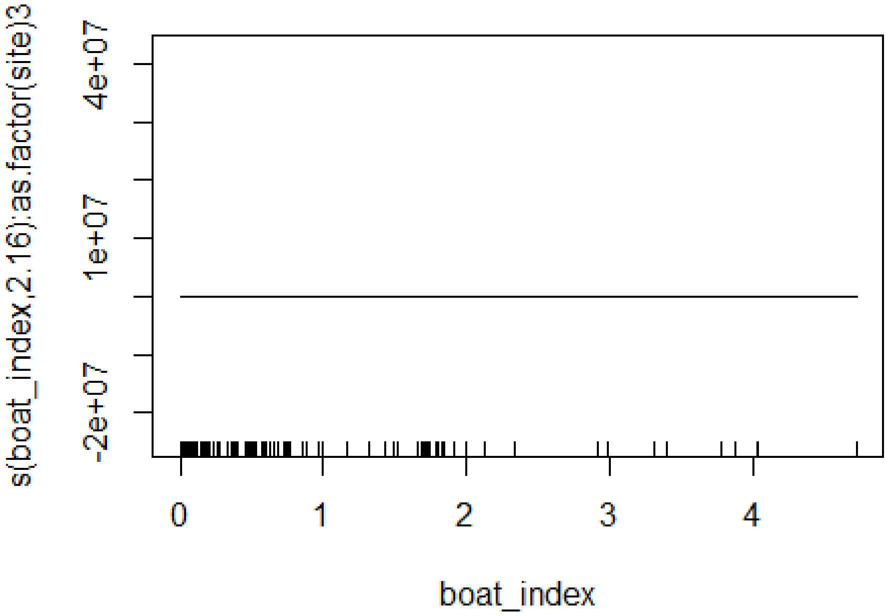
```
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                         k'   edf k-index p-value
## s(boats_total_15min):as.factor(site)1 9.00 1.00    0.90    0.12
## s(boats_total_15min):as.factor(site)2 9.00 6.85    0.90    0.15
## s(boats_total_15min):as.factor(site)3 9.00 2.47    0.90    0.14
## s(data_72hr):as.factor(site)1         9.00 1.00    1.01    0.55
## s(data_72hr):as.factor(site)2         9.00 8.94    1.01    0.54
## s(data_72hr):as.factor(site)3         9.00 1.00    1.01    0.51

# check autocorrelation
ACF  <-  acf(resid(seals_vessels_date), lag.max = 30)
```

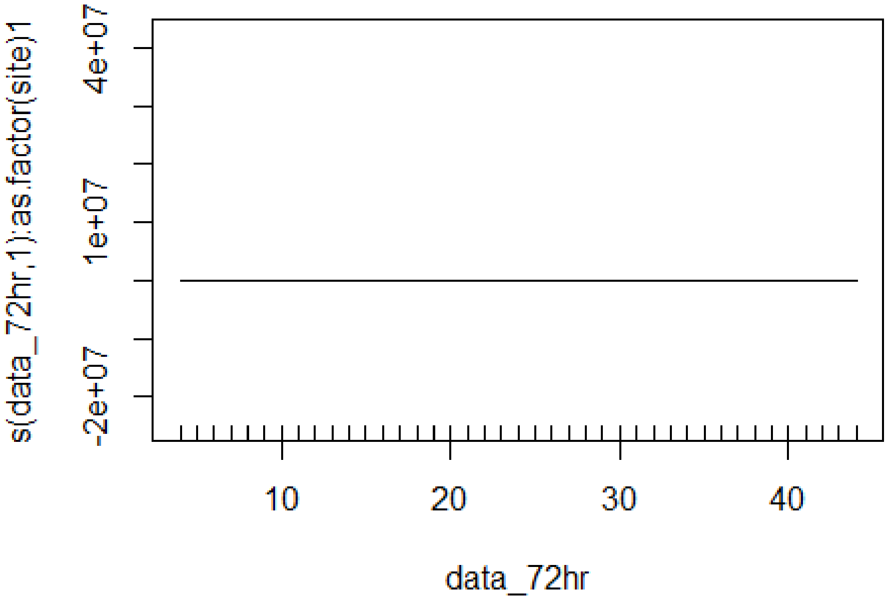


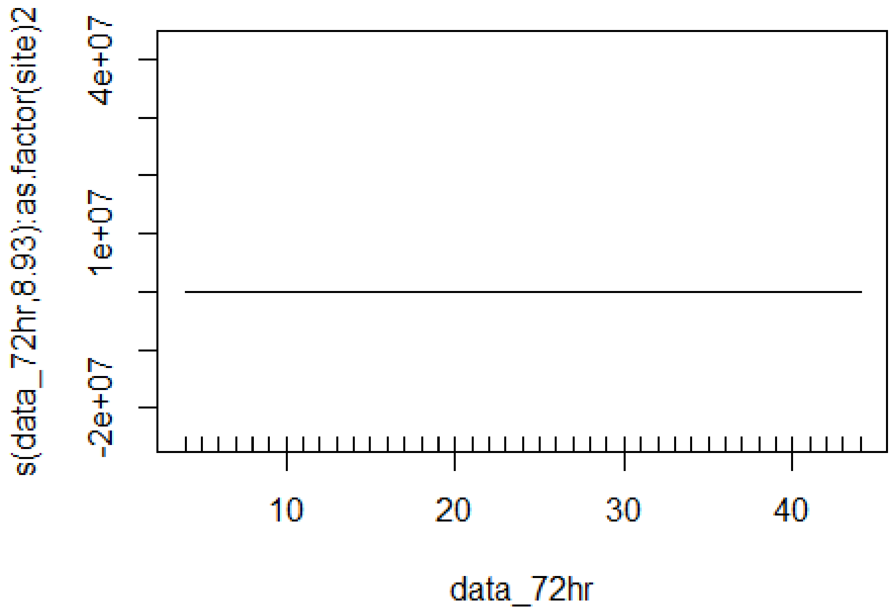

```
# lag of 1 = 72 hrs = grouping accounts for autocorrelation
# wide confidence intervals with boat total
# try model using boat_index
seals_vindex_date <- gam(cbind(n.x,n.y) ~ s(boat_index,
by=as.factor(site)) + s(data_72hr,by=as.factor(site)), data =
all_data_72_Jun_Sept, family=binomial(link="cloglog"))
plot(seals_vindex_date)
```

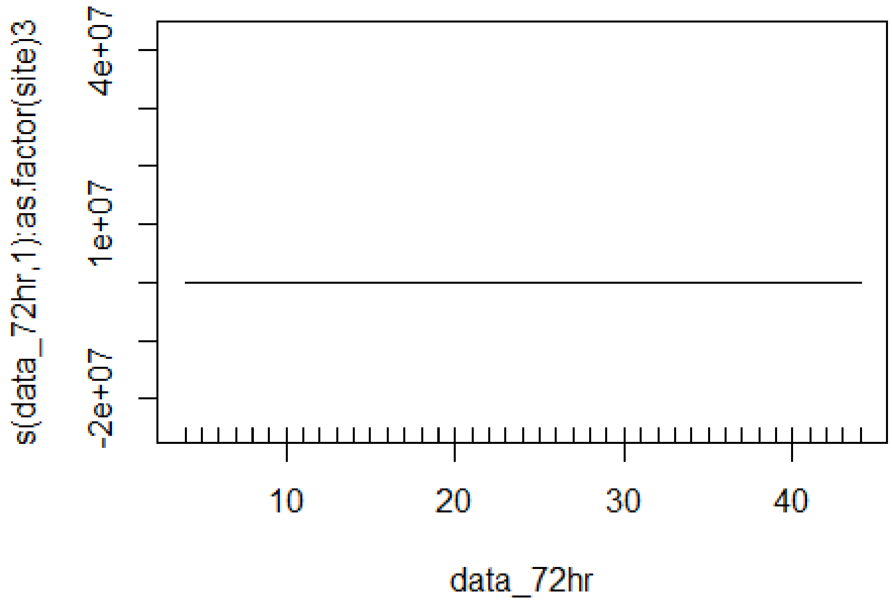

s(boat_index,1):as.factor(site)1
4e+07
1e+07
-2e+07
0
1
2
3
4
boat_index


s(boat_index,5.95):as.factor(site)2
4e+07
1e+07
-2e+07
0
1
2
3
4
boat_index


s(boat_index,2.16):as.factor(site)3
4e+07
1e+07
-2e+07
0
1
2
3
4
boat_index

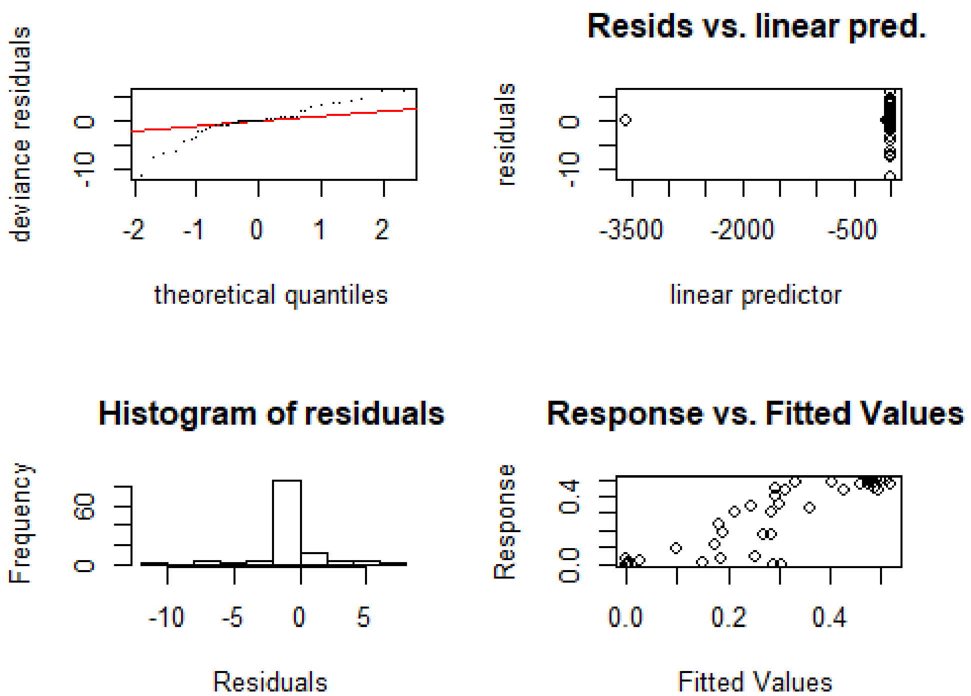

s(data_72hr,1):as.factor(site)1
4e+07
1e+07
-2e+07
10
20
30
40
data_72hr





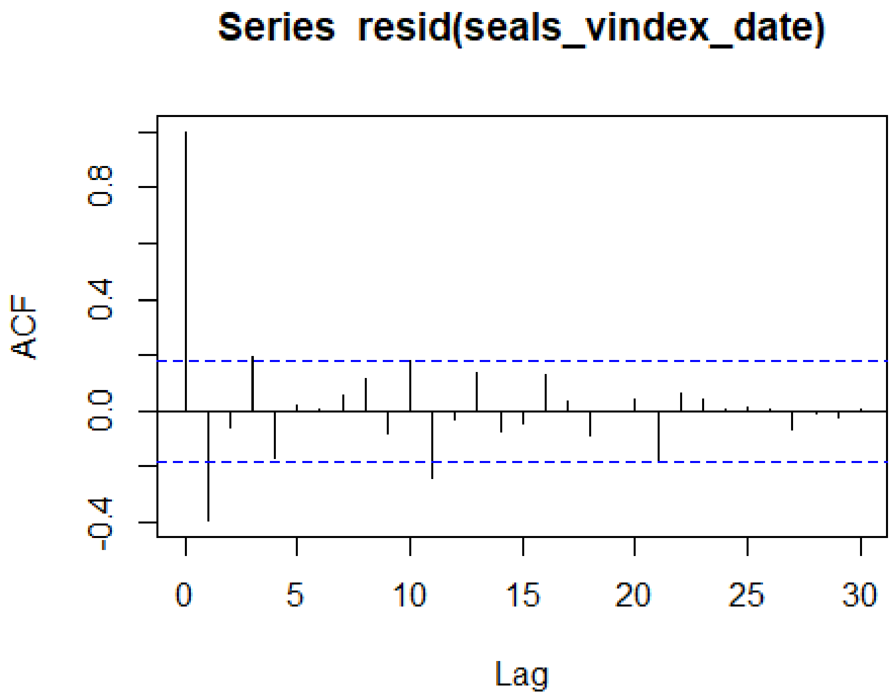

```
summary(seals_vindex_date)
##
## Family: binomial
## Link function: cloglog
```

```
##
## Formula:
## cbind(n.x, n.y) ~ s(boat_index, by = as.factor(site)) + s(data_72hr,
##     by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -13.616      5.428  -2.509   0.0121 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                  edf Ref.df  Chi.sq  p-value
## s(boat_index):as.factor(site)1 1.000  1.000   0.001    0.971
## s(boat_index):as.factor(site)2 5.950  5.998  44.164 8.75e-08 ***
## s(boat_index):as.factor(site)3 2.164  2.644   4.111    0.294
## s(data_72hr):as.factor(site)1  1.000  1.000   0.001    0.981
## s(data_72hr):as.factor(site)2  8.933  8.998 514.783  < 2e-16 ***
## s(data_72hr):as.factor(site)3  1.000  1.000   1.163    0.281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.899   Deviance explained = 93.4%
## UBRE =  3.722  Scale est. = 1         n = 116
gam.check(seals_vindex_date)
```



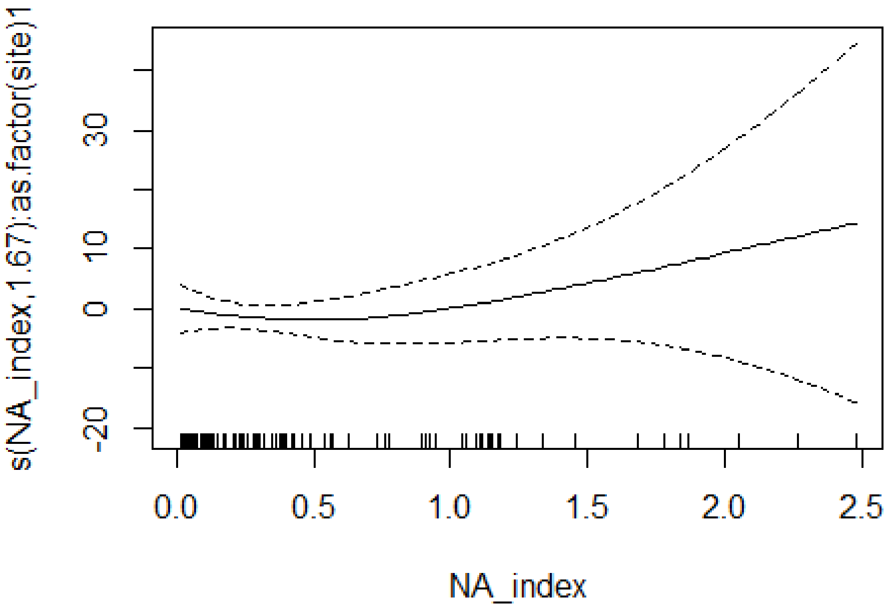

```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 30 iterations.
## Gradient range [-4.025897e-07,3.860808e-07]
## (score 3.722001 & scale 1).
## Hessian positive definite, eigenvalue range [2.37518e-07,0.004120901].
## Model rank =  55 / 55
##
```

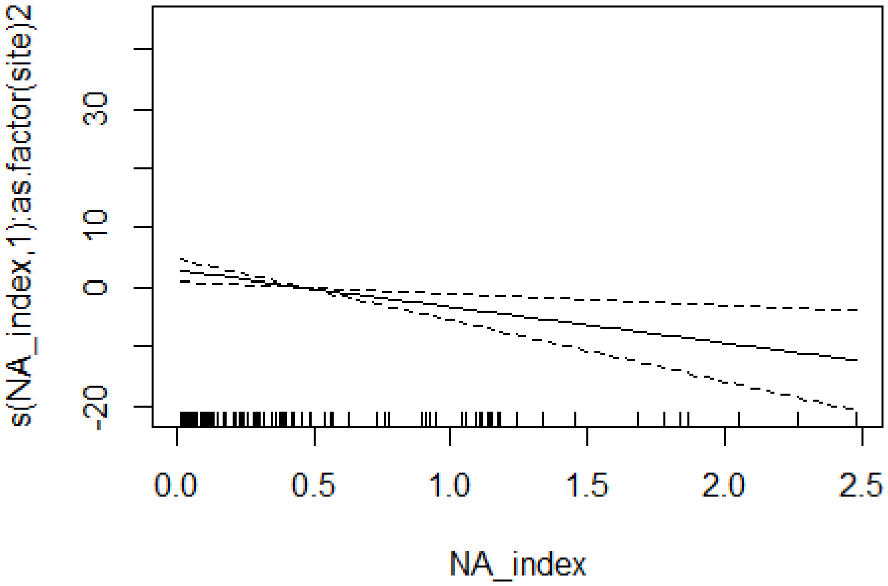
```
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                     k'   edf  k-index  p-value
## s(boat_index):as.factor(site)1  9.00  1.00     0.98     0.49
## s(boat_index):as.factor(site)2  9.00  5.95     0.98     0.39
## s(boat_index):as.factor(site)3  9.00  2.16     0.98     0.41
## s(data_72hr):as.factor(site)1   9.00  1.00     0.99     0.44
## s(data_72hr):as.factor(site)2   9.00  8.93     0.99     0.44
## s(data_72hr):as.factor(site)3   9.00  1.00     0.99     0.36

# check autocorrelation
ACF  <-  acf(resid(seals_vindex_date), lag.max = 30)
```

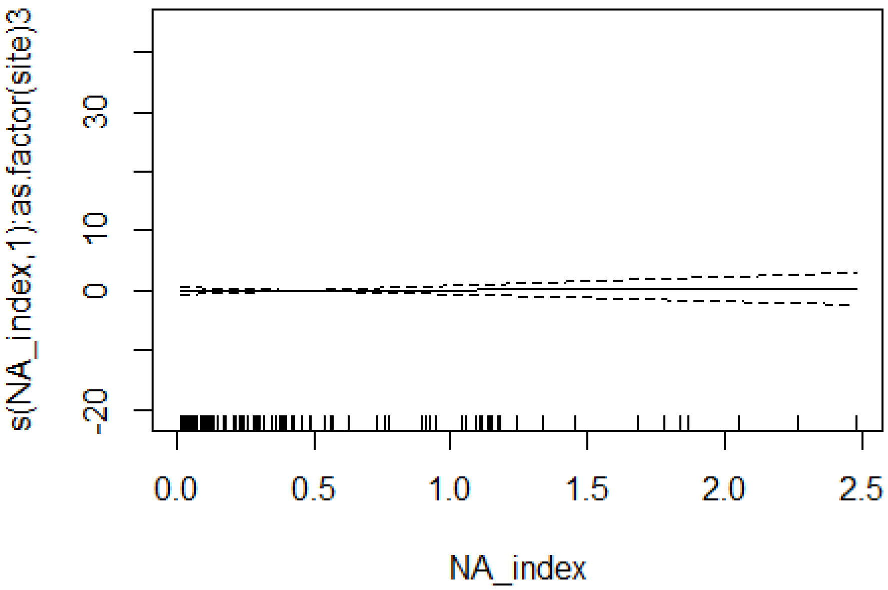


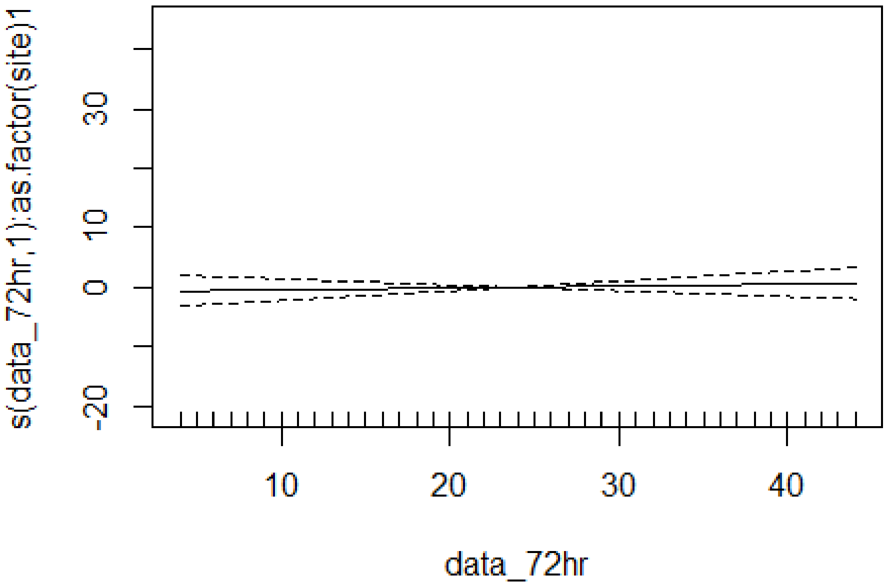

```
# still outlier where vessel index > 4
# manually checked data - only 28 photos within 72hr period
# typically number of images in 72hr period = 288

# create new vessel index where if external camera is operational <50% of
72hrs, vessel index = NA
288 * 0.5
## [1] 144
# [1] 144 photos = 50%
all_data_72_Jun_Sept$NA_50index <- ifelse(all_data_72_Jun_Sept$n.y < 144,
"NA", all_data_72_Jun_Sept$boat_index)

str(all_data_72_Jun_Sept)
# NA_index column = chr -> convert to num
all_data_72_Jun_Sept$NA_50index <-
as.numeric(all_data_72_Jun_Sept$NA_50index)

# remove rows where there is NA value
```

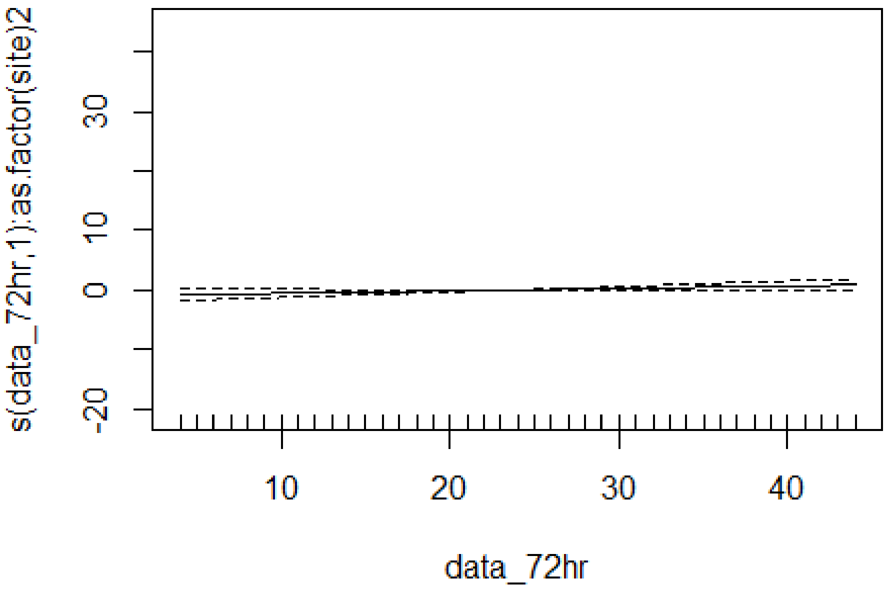
```
library(tidyr)
all_data_72_Jun_Sept <- all_data_72_Jun_Sept %>% drop_na(NA_50index)

# use new index excluding 72hr periods with low vessel data coverage
# try model using boat_index
seals_NA50index_date <- gam(presence_new ~ s(NA_index, by=
as.factor(site)) + s(data_72hr, by= as.factor(site)), data =
final_72hr_data, family=binomial(link="cloglog"))
plot(seals_NA50index_date)
```

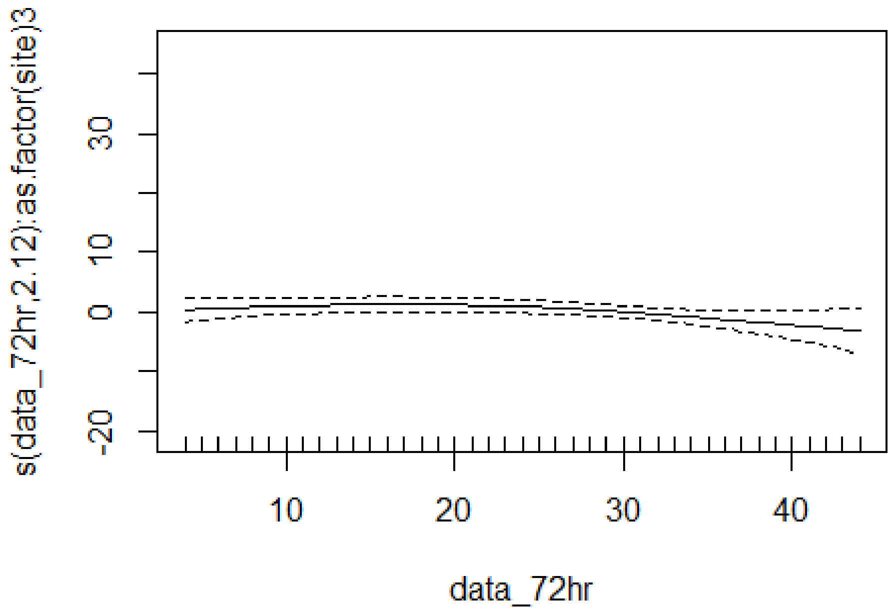



s(NA_index,1):as.factor(site)2
30
10
0
-20
0.0
0.5
1.0
1.5
2.0
2.5
NA_index

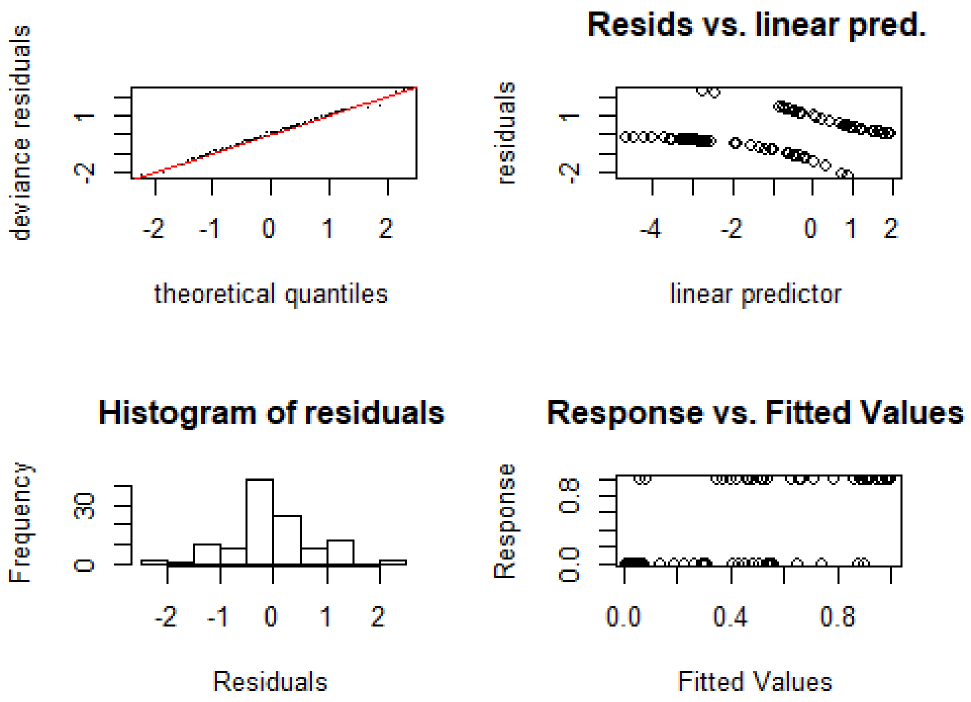

s(NA_index,1):as.factor(site)3
30
10
0
-20
0.0
0.5
1.0
1.5
2.0
2.5
NA_index


s(data_72hr,1):as.factor(site)1
-20
0
10
30
10
20
30
40
data_72hr

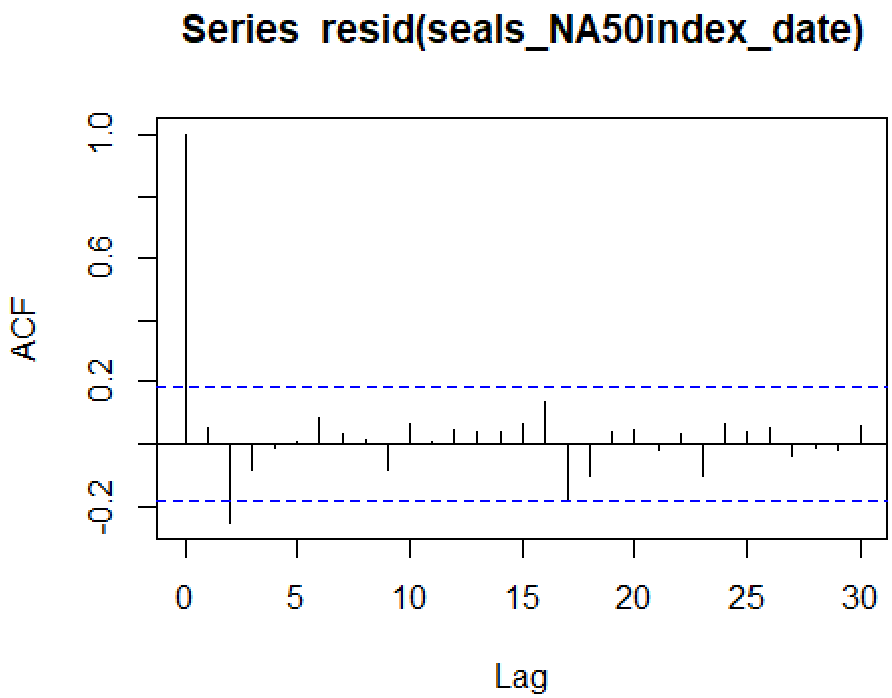

s(data_72hr,1):as.factor(site)2
-20
0
10
30
10
20
30
40
data_72hr



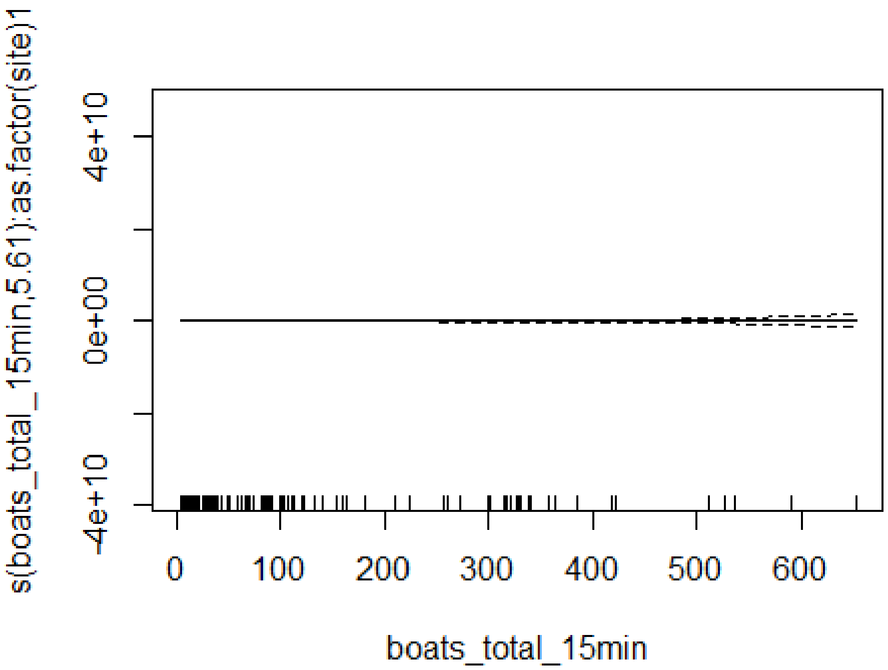

```
summary(seals_NA50index_date)

##
## Family: binomial
## Link function: cloglog
##
## Formula:
## presence_new ~ s(NA_index, by = as.factor(site)) + s(data_72hr,
##     by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.534      0.786  -1.952   0.0509 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                 edf Ref.df Chi.sq p-value
## s(NA_index):as.factor(site)1  1.674  1.978  2.725 0.28855
## s(NA_index):as.factor(site)2  1.000  1.000  8.616 0.00333 **
## s(NA_index):as.factor(site)3  1.000  1.000  0.049 0.82565
## s(data_72hr):as.factor(site)1 1.000  1.000  0.245 0.62073
## s(data_72hr):as.factor(site)2 1.000  1.000  2.773 0.09588 .
## s(data_72hr):as.factor(site)3 2.118  2.674  3.808 0.21132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.552   Deviance explained =   53%
## UBRE = -0.20398  Scale est. = 1         n = 112

gam.check(seals_NA50index_date)
```

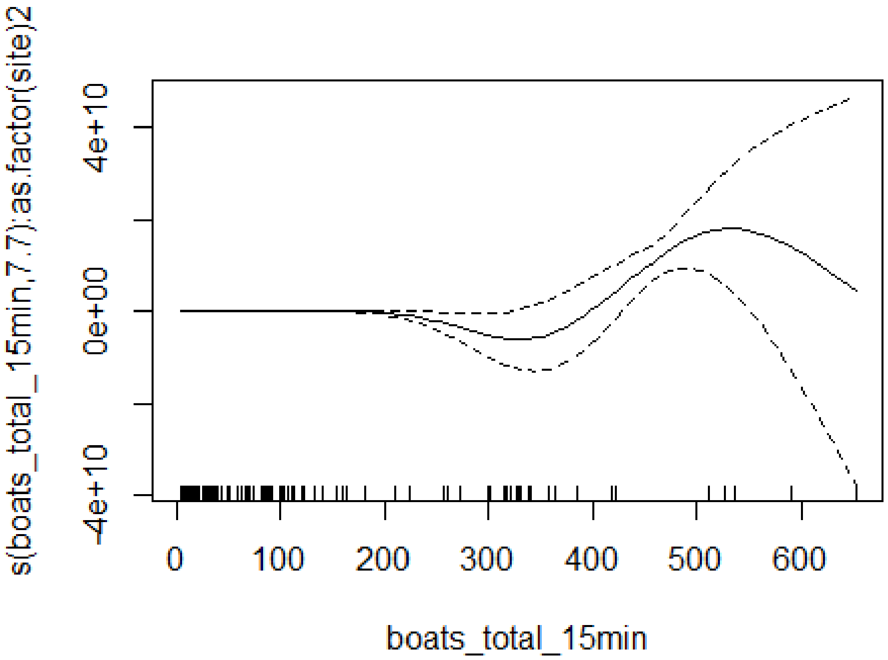


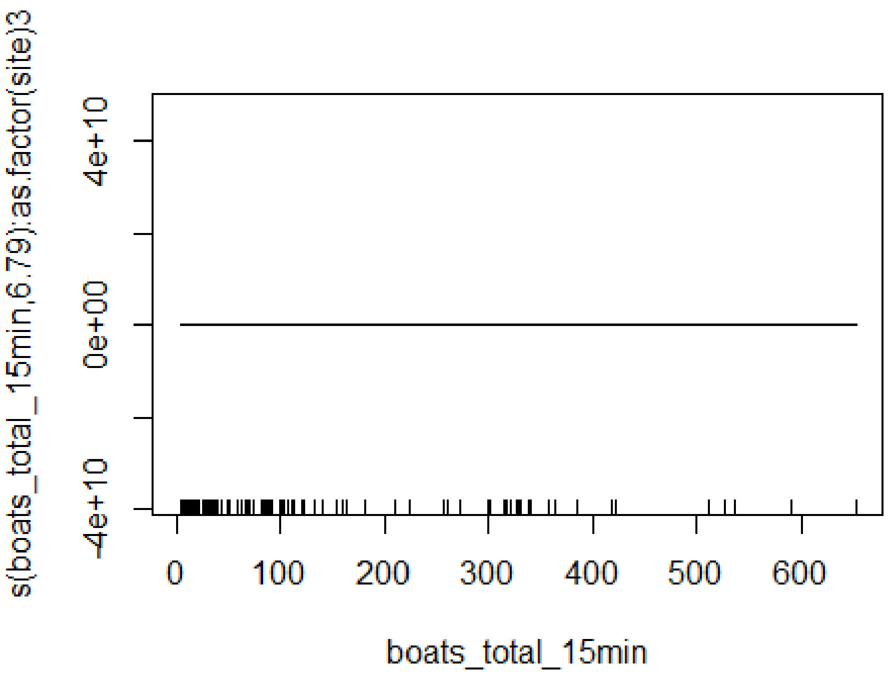

```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 10 iterations.
## Gradient range [-1.872232e-07,6.047795e-08]
## (score -0.203983 & scale 1).
## Hessian positive definite, eigenvalue range [2.481975e-08,0.005405299].
## Model rank =  55 / 55
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                    k'   edf k-index p-value
## s(NA_index):as.factor(site)1  9.00 1.67    1.03    0.59
## s(NA_index):as.factor(site)2  9.00 1.00    1.03    0.64
## s(NA_index):as.factor(site)3  9.00 1.00    1.03    0.61
## s(data_72hr):as.factor(site)1 9.00 1.00    1.03    0.54
## s(data_72hr):as.factor(site)2 9.00 1.00    1.03    0.66
## s(data_72hr):as.factor(site)3 9.00 2.12    1.03    0.62

# check autocorrelation
ACF  <-  acf(resid(seals_NA50index_date), lag.max = 30)
```

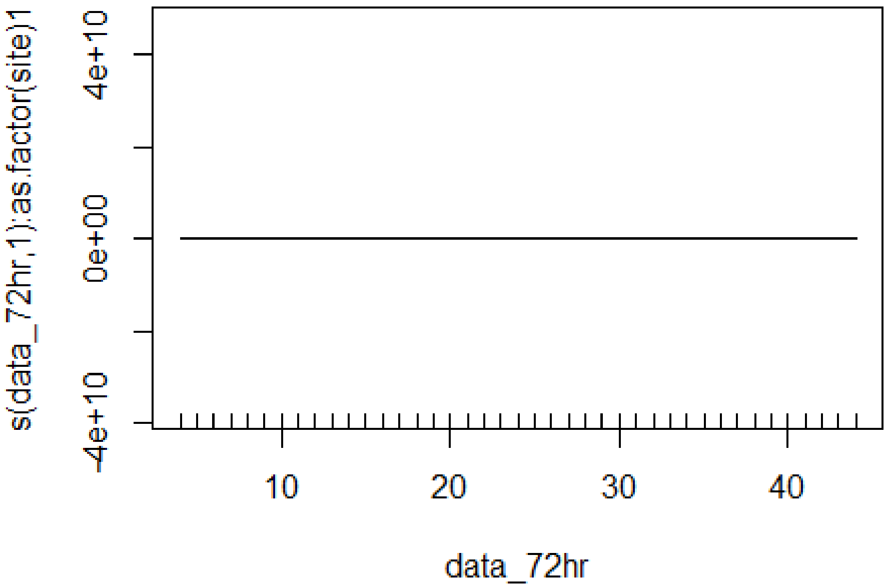


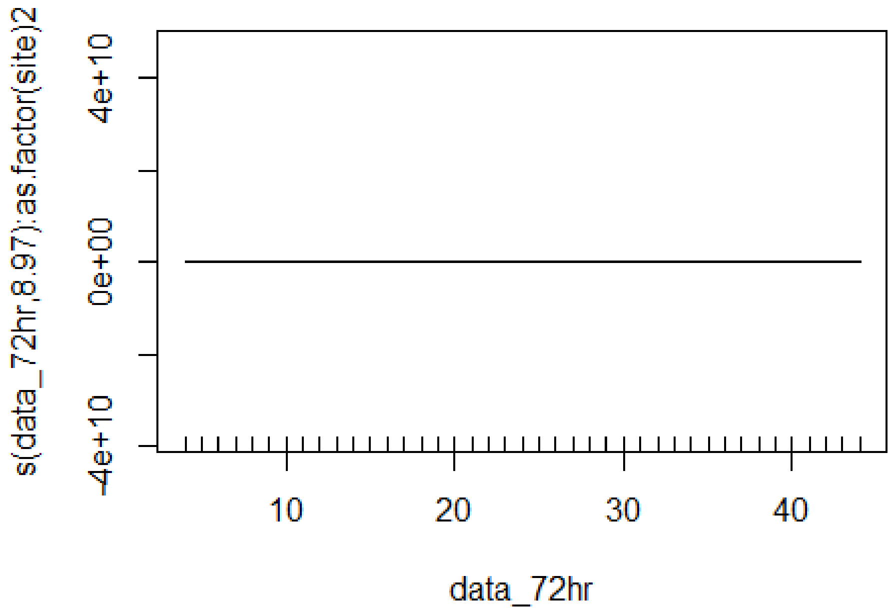

```
# save final dataframe for model as csv file
final_72hr_data <- all_data_72_Jun_Sept
write.csv(final_72hr_data, 'E:/BL5599/final_72hr_data_copy.csv', row.names
=F)

# include seasonal trends aswell with data_72hr variable (consecutively
numbered 72hr group)
seals_vessels_date <- gam(cbind(n.x,seal_fail) ~ s(boats_total_15min, by=
as.factor(site)) + s(data_72hr, by= as.factor(site)), data =
final_72hr_data, family = binomial(link = "cloglog"))
plot(seals_vessels_date)
```

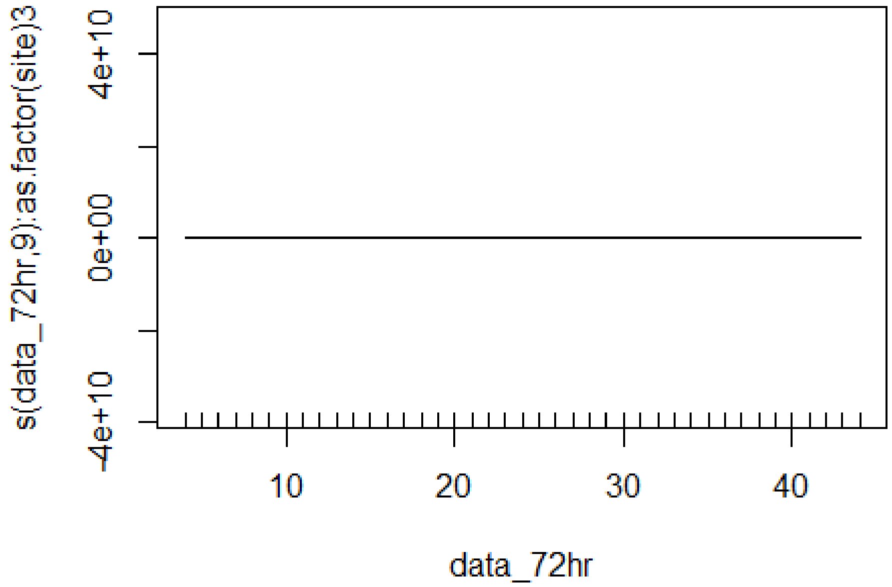

s(boats_total_15min,5.61):as.factor(site)1
4e+10
0e+00
-4e+10
0
100
200
300
400
500
600
boats_total_15min


s(boats_total_15min,7.7):as.factor(site)2
4e+10
0e+00
-4e+10
0
100
200
300
400
500
600
boats_total_15min


s(boats_total_15min,6.79):as.factor(site)3
4e+10
0e+00
-4e+10
0
100
200
300
400
500
600
boats_total_15min

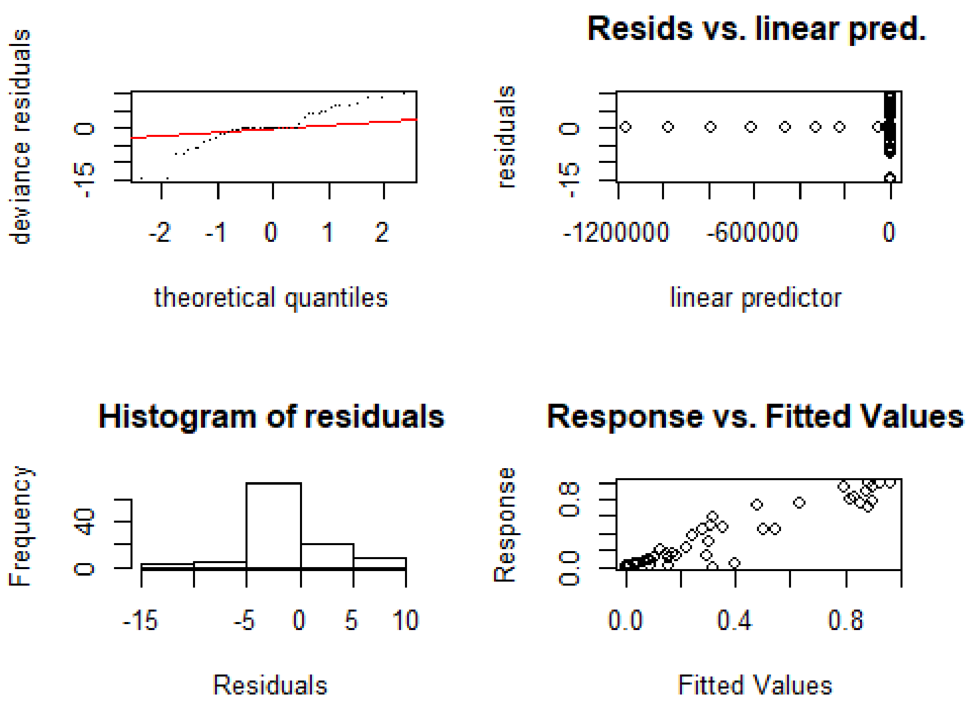

s(data_72hr,1):as.factor(site)1
4e+10
0e+00
-4e+10
10
20
30
40
data_72hr





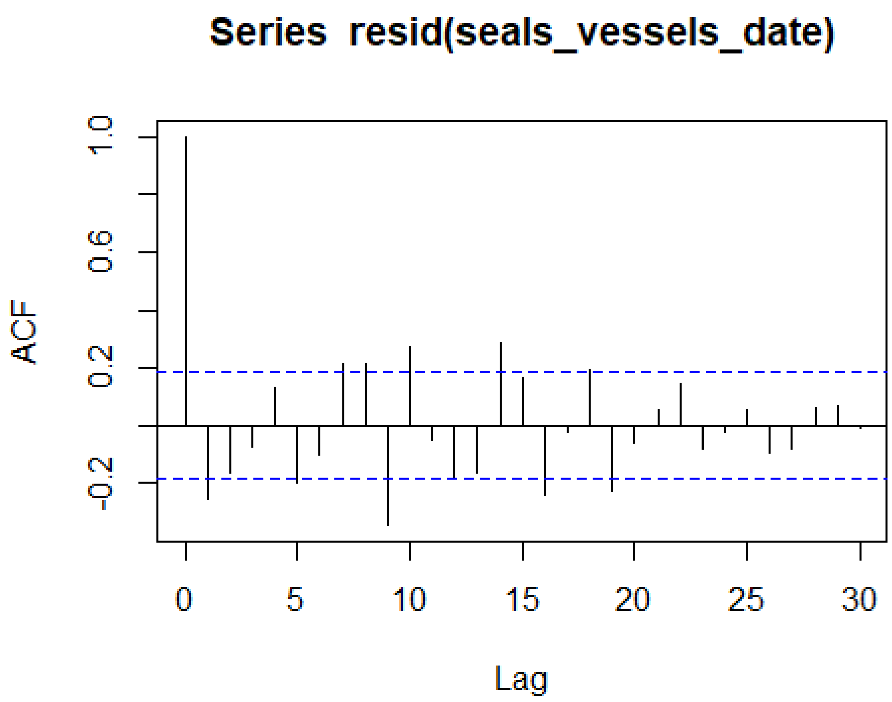

```
summary(seals_vessels_date)

##
## Family: binomial
## Link function: cloglog
##
## Formula:
## cbind(n.x, seal_fail) ~ s(boats_total_15min, by = as.factor(site)) +
```

```
##     s(data_72hr, by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -127120     253231  -0.502    0.616
##
## Approximate significance of smooth terms:
##                                        edf Ref.df   Chi.sq p-value
## s(boats_total_15min):as.factor(site)1 5.605  5.852    0.252   1.000
## s(boats_total_15min):as.factor(site)2 7.702  4.000    2.132   0.711
## s(boats_total_15min):as.factor(site)3 6.787  7.051    0.299   1.000
## s(data_72hr):as.factor(site)1         1.000  1.000    0.000   0.999
## s(data_72hr):as.factor(site)2         8.970  8.999 2398.373  <2e-16 ***
## s(data_72hr):as.factor(site)3         9.000  9.000    7.015   0.635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.891   Deviance explained = 92.6%
## UBRE = 11.253  Scale est. = 1         n = 112

gam.check(seals_vessels_date)
```



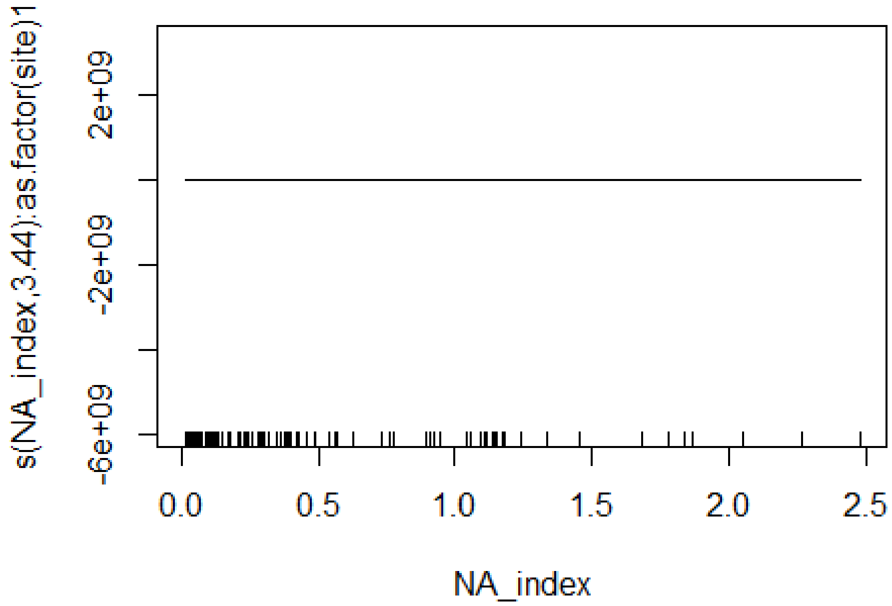

```
##
## Method: UBRE   Optimizer: outer newton
## step failed after 86 iterations.
## Gradient range [-0.0001568243,1.522383]
## (score 11.25278 & scale 1).
## eigenvalue range [-141524.5,354.4641].
## Model rank =  55 / 55
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
```

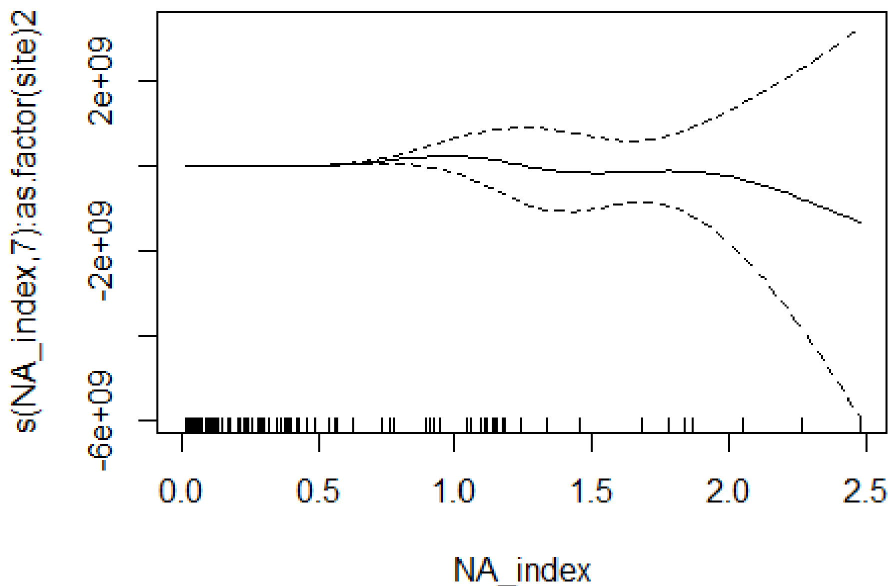
```
## indicate that k is too low, especially if edf is close to k'.
##
##                                          k'   edf k-index p-value
## s(boats_total_15min):as.factor(site)1 9.00 5.61    0.87   0.060 .
## s(boats_total_15min):as.factor(site)2 9.00 7.70    0.87   0.040 *
## s(boats_total_15min):as.factor(site)3 9.00 6.79    0.87   0.065 .
## s(data_72hr):as.factor(site)1         9.00 1.00    1.01   0.540
## s(data_72hr):as.factor(site)2         9.00 8.97    1.01   0.490
## s(data_72hr):as.factor(site)3         9.00 9.00    1.01   0.485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


# check autocorrelation
ACF  <-  acf(resid(seals_vessels_date), lag.max = 30)
```

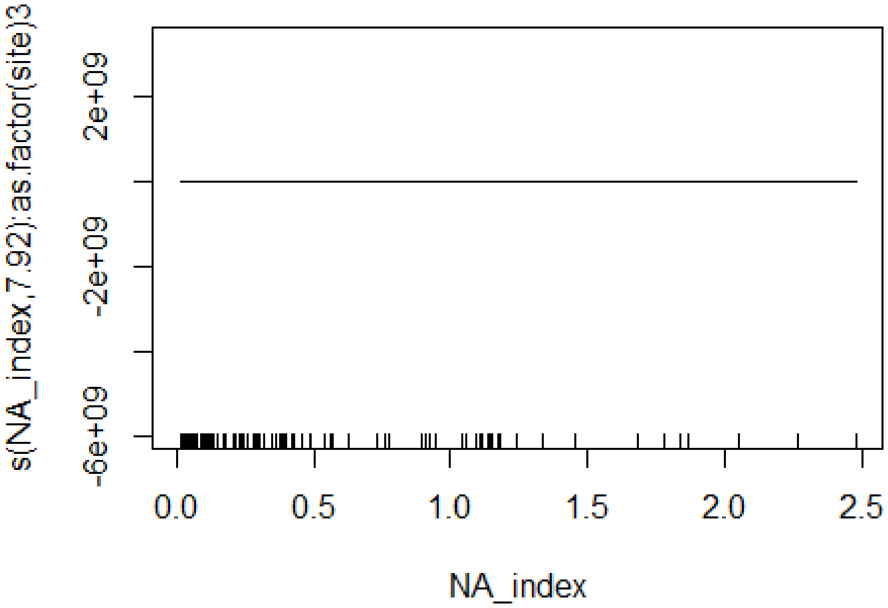


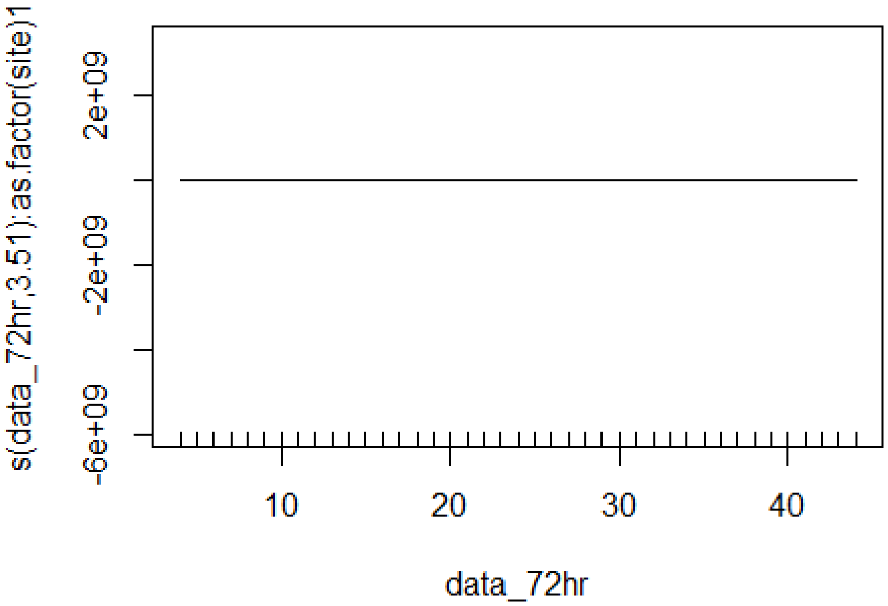

```
# lag of 1 = 72 hrs = grouping accounts for autocorrelation

# repeat using success/failures of seals and vessel index over 72hrs and
site
model_72hr_seal_vessel <- gam(cbind(n.x, seal_fail) + s(NA_index, by=
as.factor(site)) + s(data_72hr, by= as.factor(site)), data =
final_72hr_data, family = binomial(link = "cloglog"))
summary(model_72hr_seal_vessel)

##
## Family: binomial
## Link function: cloglog
```

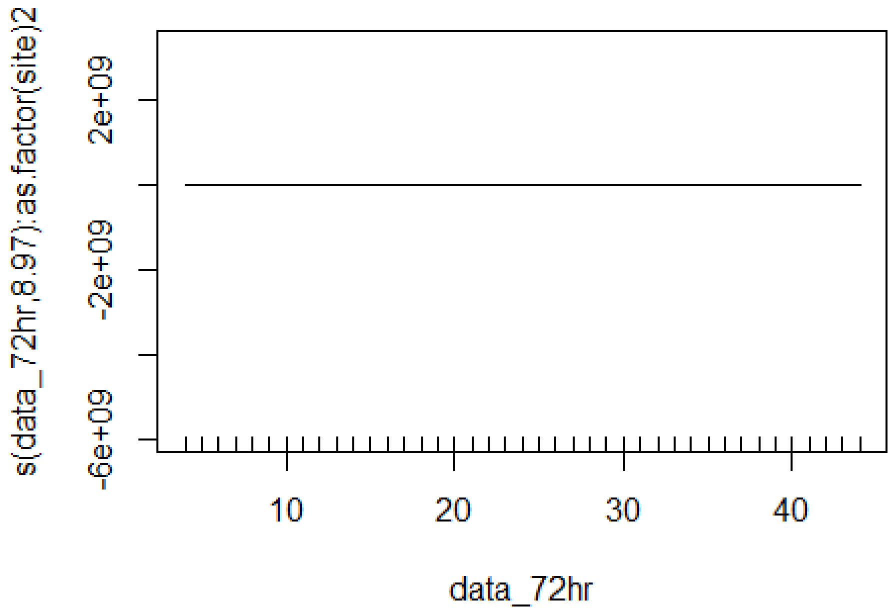
```
##
## Formula:
## cbind(n.x, n.y) ~ s(NA_index, by = as.factor(site)) + s(data_72hr,
##  by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -77.72      54.26  -1.432    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                 edf Ref.df  Chi.sq  p-value
## s(NA_index):as.factor(site)1  3.438  3.471   0.00        1
## s(NA_index):as.factor(site)2  7.004  7.000 111.73  < 2e-16 ***
## s(NA_index):as.factor(site)3  7.920  8.025  66.37 2.67e-11 ***
## s(data_72hr):as.factor(site)1 3.507  3.547   0.00        1
## s(data_72hr):as.factor(site)2 8.968  8.999 860.80  < 2e-16 ***
## s(data_72hr):as.factor(site)3 6.082  6.408  37.66 1.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.895   Deviance explained = 93.6%
## UBRE = 6.0386  Scale est. = 1         n = 112

plot(model_72hr_seal_vessel)
```

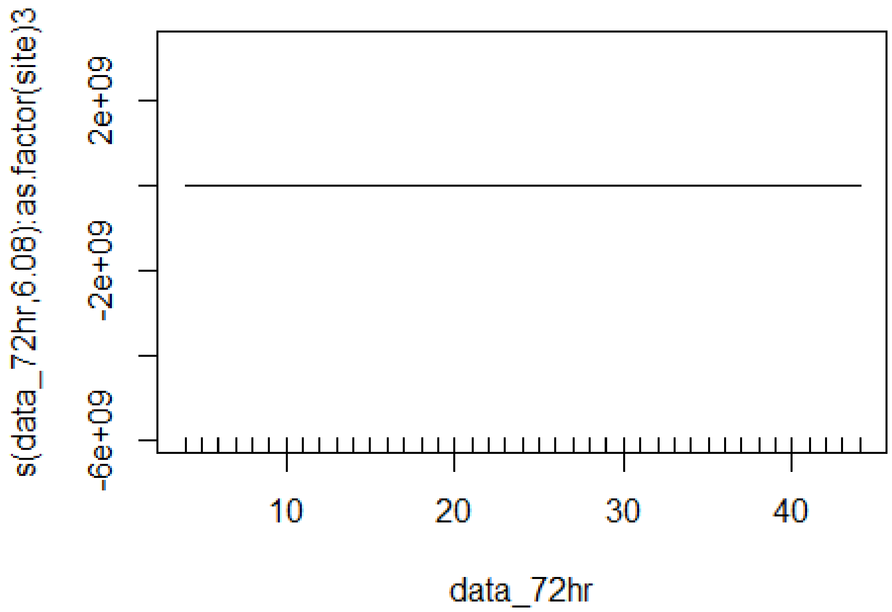

s(NA_index,3.44):as.factor(site)1
2e+09
-2e+09
-6e+09
0.0
0.5
1.0
1.5
2.0
2.5
NA_index


s(NA_index,7):as.factor(site)2
2e+09
-2e+09
-6e+09
0.0
0.5
1.0
1.5
2.0
2.5
NA_index


s(NA_index,7.92):as.factor(site)3
2e+09
-2e+09
-6e+09
0.0
0.5
1.0
1.5
2.0
2.5
NA_index

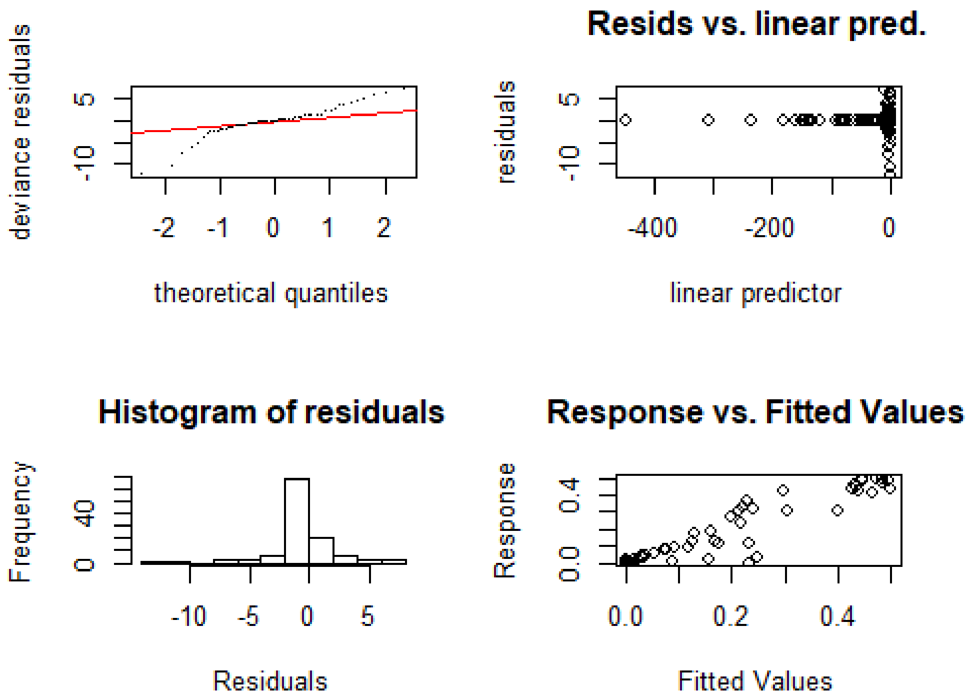

s(data_72hr,3.51):as.factor(site)1
2e+09
-2e+09
-6e+09
10
20
30
40
data_72hr






```
summary(model_72hr_seal_vessel)

##
## Family: binomial
## Link function: cloglog
##
## Formula:
## cbind(n.x, n.y) ~ s(NA_index, by = as.factor(site)) + s(data_72hr,
```

```
##     by = as.factor(site))
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -77.72      54.26  -1.432    0.152
##
## Approximate significance of smooth terms:
##                                  edf Ref.df Chi.sq  p-value
## s(NA_index):as.factor(site)1  3.438  3.471   0.00        1
## s(NA_index):as.factor(site)2  7.004  7.000 111.73  < 2e-16 ***
## s(NA_index):as.factor(site)3  7.920  8.025  66.37 2.67e-11 ***
## s(data_72hr):as.factor(site)1 3.507  3.547   0.00        1
## s(data_72hr):as.factor(site)2 8.968  8.999 860.80  < 2e-16 ***
## s(data_72hr):as.factor(site)3 6.082  6.408  37.66 1.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.895   Deviance explained = 93.6%
## UBRE = 6.0386  Scale est. = 1         n = 112

gam.check(model_72hr_seal_vessel)
```

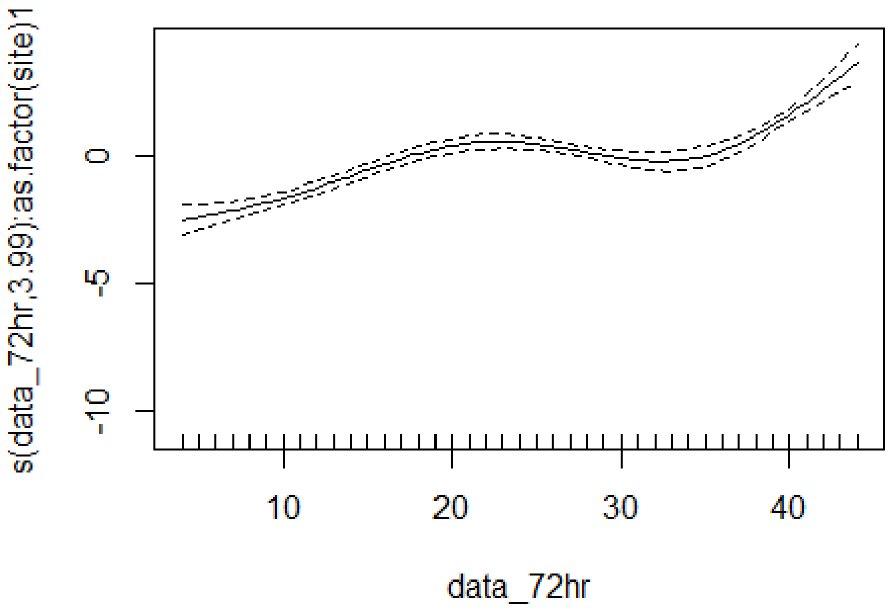



```
##
## Method: UBRE   Optimizer: outer newton
## step failed after 50 iterations.
## Gradient range [-0.002869359,0.01077663]
## (score 6.038614 & scale 1).
## eigenvalue range [-2965.434,0.4399269].
## Model rank =  55 / 55
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
```

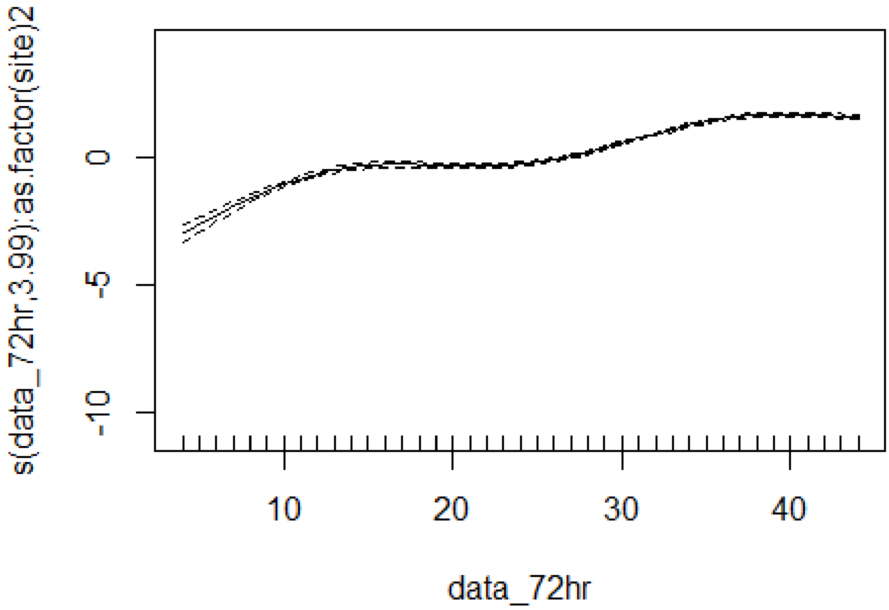
```
## indicate that k is too low, especially if edf is close to k'.
##
##                                   k'   edf k-index p-value
## s(NA_index):as.factor(site)1  9.00 3.44    0.91    0.13
## s(NA_index):as.factor(site)2  9.00 7.00    0.91    0.16
## s(NA_index):as.factor(site)3  9.00 7.92    0.91    0.15
## s(data_72hr):as.factor(site)1 9.00 3.51    1.02    0.57
## s(data_72hr):as.factor(site)2 9.00 8.97    1.02    0.56
## s(data_72hr):as.factor(site)3 9.00 6.08    1.02    0.57



# Confidence intervals on plots are really wide
```

FINAL FINAL 72HR GAM

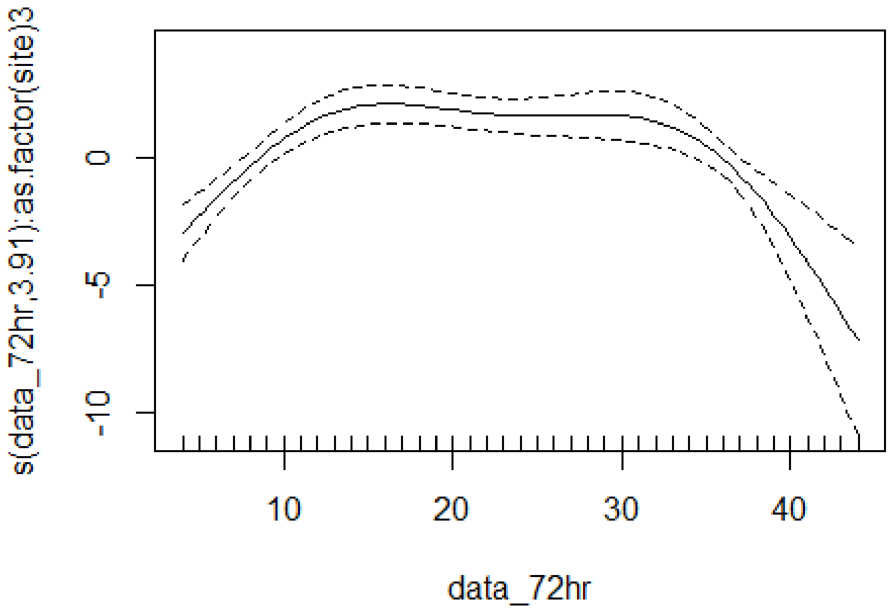
```
# load packages
library(mgcv)

library(scales)

# load dataset
final_72hr_data <-
read.csv('E:/BL5599/vessel_seal_data_72hrs_21_07_21.csv',header=T)

seal_vessel_72hr_model <- gam(cbind(n.x, seal_fail) ~ s(data_72hr,
by=as.factor(site), k=5) + s(NA_index), data=final_72hr_data,
family=binomial(link="cloglog"))

# plot 72 hourly model
plot(seal_vessel_72hr_model)
```

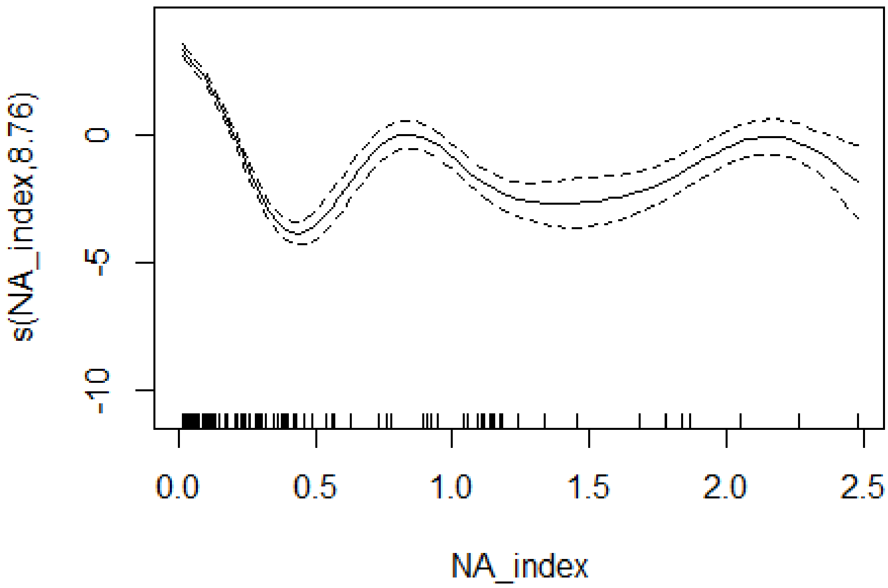


s(data_72hr,3.99):as.factor(site)1
0
-5
-10
10
20
30
40
data_72hr



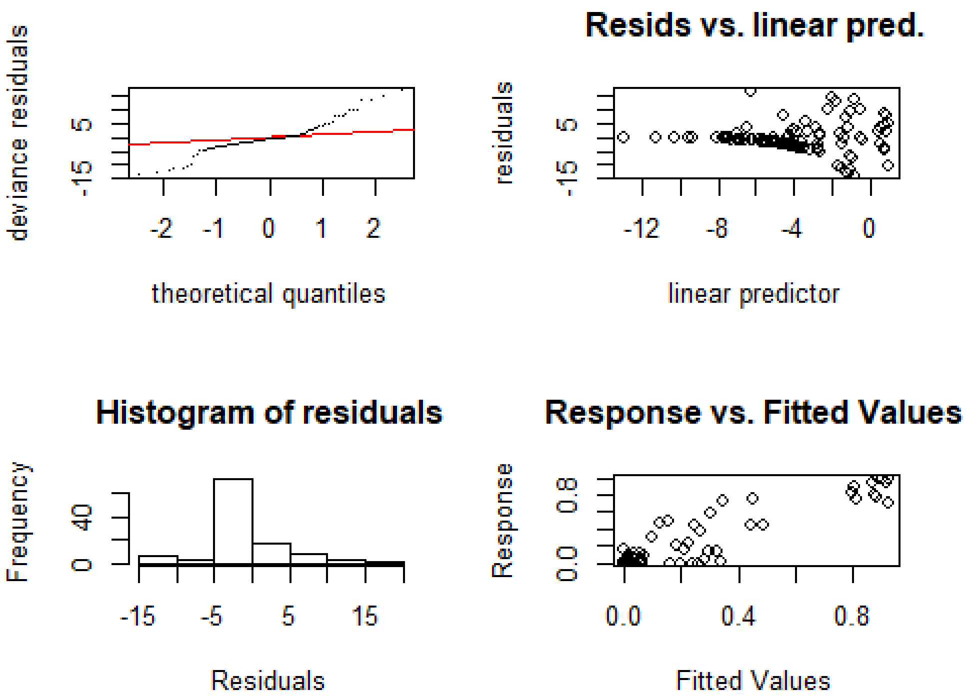
s(data_72hr,3.99):as.factor(site)2
0
-5
-10
10
20
30
40
data_72hr






```
summary(seal_vessel_72hr_model)

## Family: binomial
## Link function: cloglog
##
## Formula:
## cbind(n.x, seal_fail) ~ s(data_72hr, by = as.factor(site), k = 5) +
##     s(NA_index)
##
```

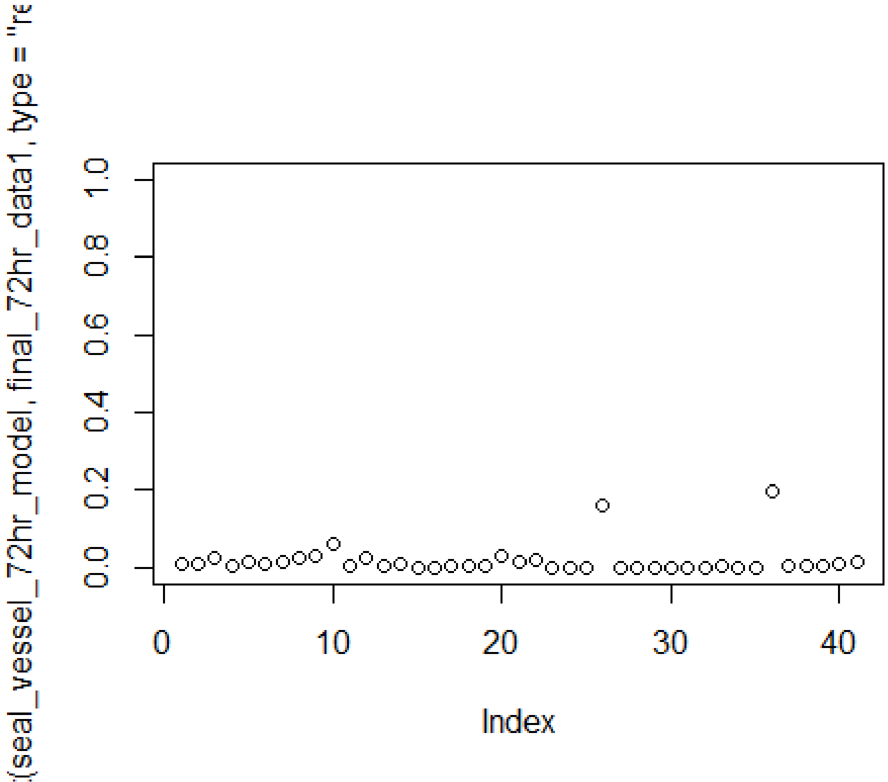
```
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.9726     0.1162  -34.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                  edf Ref.df  Chi.sq  p-value
## s(data_72hr):as.factor(site)1 3.990  4.000  280.04  < 2e-16 ***
## s(data_72hr):as.factor(site)2 3.992  4.000 2647.28  < 2e-16 ***
## s(data_72hr):as.factor(site)3 3.905  3.993   50.53 2.91e-10 ***
## s(NA_index)                   8.762  8.982 1420.94  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.835   Deviance explained = 82.8%
## UBRE = 26.434  Scale est. = 1         n = 112

# check assumptions
gam.check(seal_vessel_72hr_model)
```




```
## Method: UBRE   Optimizer: outer newton
## full convergence after 10 iterations.
## Gradient range [2.024928e-08,2.715347e-05]
## (score 26.4342 & scale 1).
## Hessian positive definite, eigenvalue range [2.067716e-05,0.003815353].
## Model rank =  22 / 22
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                  k'   edf k-index p-value
## s(data_72hr):as.factor(site)1 4.00 3.99    1.06    0.79
```

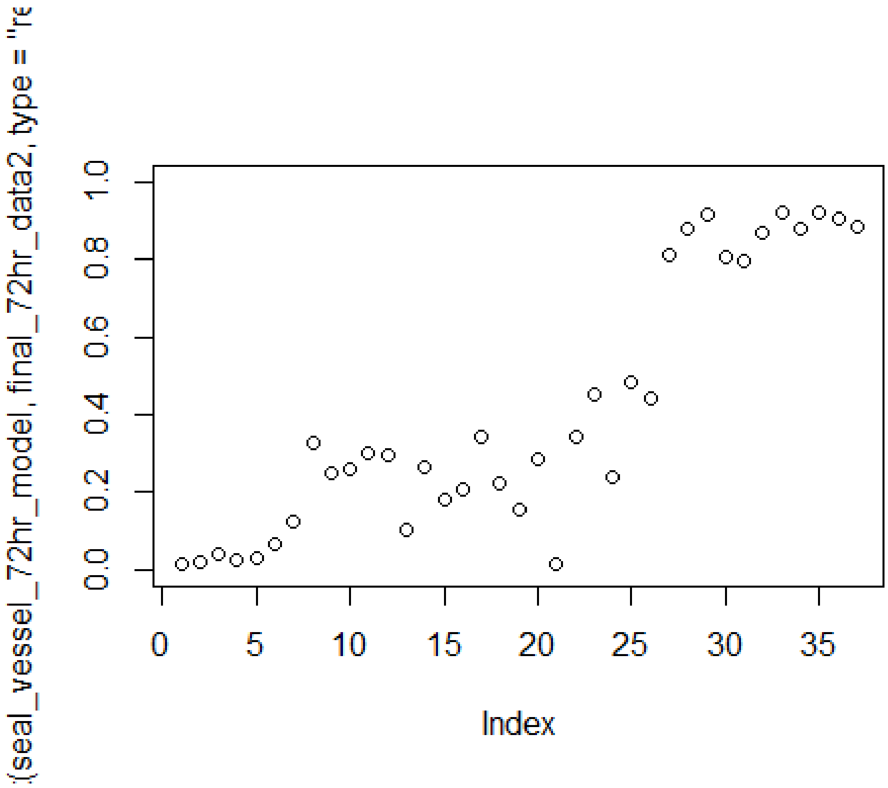
```
## s(data_72hr):as.factor(site)2 4.00 3.99    1.06    0.80
## s(data_72hr):as.factor(site)3 4.00 3.91    1.06    0.76
## s(NA_index)                   9.00 8.76    0.86    0.07 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# final plots
jpeg("E:/BL5599/high_res_plots/final_72hr_GAM1.jpeg", width = 6, height =
4, units = 'in', res = 600)
plot.gam(seal_vessel_72hr_model, select = 1, ylim=c(-3,3), shade = T,
xlab="Date", ylab="Probability of Monk Seal Presence", main = "Site 1",
xaxt="n", cex.lab=1.2, cex.axis=1.2, shade.col='#A6CEE3')
abline(h=0, col="red", lty=2, lwd=1)
label_x <- c("June", "July", "August","September","October")
axis(1, at = c("4", "14", "24","34","44"), labels = label_x,cex.axis=1.2)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/final_72hr_GAM2.jpeg", width = 6, height =
4, units = 'in', res = 600)
plot.gam(seal_vessel_72hr_model, select = 2, ylim=c(-4,2), shade = T,
xlab="Date", ylab="Probability of Monk Seal Presence",main="Site 2",
xaxt="n", cex.lab=1.2, cex.axis=1.2, shade.col='#1F78B4')
abline(h=0, col="red", lty=2, lwd=1)
label_x <- c("June", "July", "August","September","October")
axis(1, at = c("4", "14", "24","34","44"), labels = label_x, cex.axis=1.2)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/final_72hr_GAM3.jpeg", width = 6, height =
4, units = 'in', res = 600)
plot.gam(seal_vessel_72hr_model, select = 3, ylim=c(-14,4), shade = T,
xlab="Date", ylab="Probability of Monk Seal Presence", main="Site 3",
xaxt="n", cex.lab=1.2, cex.axis=1.2, shade.col='#B2DF8A')
abline(h=0, col="red", lty=2, lwd=1)
label_x <- c("June", "July", "August","September","October")
axis(1, at = c("4", "14", "24","34","44"), labels = label_x, cex.axis=1.2)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/final_72hr_GAM4.jpeg", width = 6, height =
4, units = 'in', res = 600)
plot.gam(seal_vessel_72hr_model, select = 4, ylim=c(-5,4), shade = T,
xlab="Vessel Index", ylab="Probability of Monk Seal Presence",
cex.lab=1.2,  cex.axis=1.2)
abline(h=0, col="red", lty=2, lwd=1)
dev.off()

## png
##   2
```

```r
# subset data by site to make predictions
#  new data frame not used as not enough data/confidence in mirador to
accurately predict vessel index at each site by date and 72hr period

final_72hr_data1 <- subset(final_72hr_data, site == '1')
final_72hr_data2 <- subset(final_72hr_data, site == '2')
final_72hr_data3 <- subset(final_72hr_data, site == '3')

# plot predictions
```

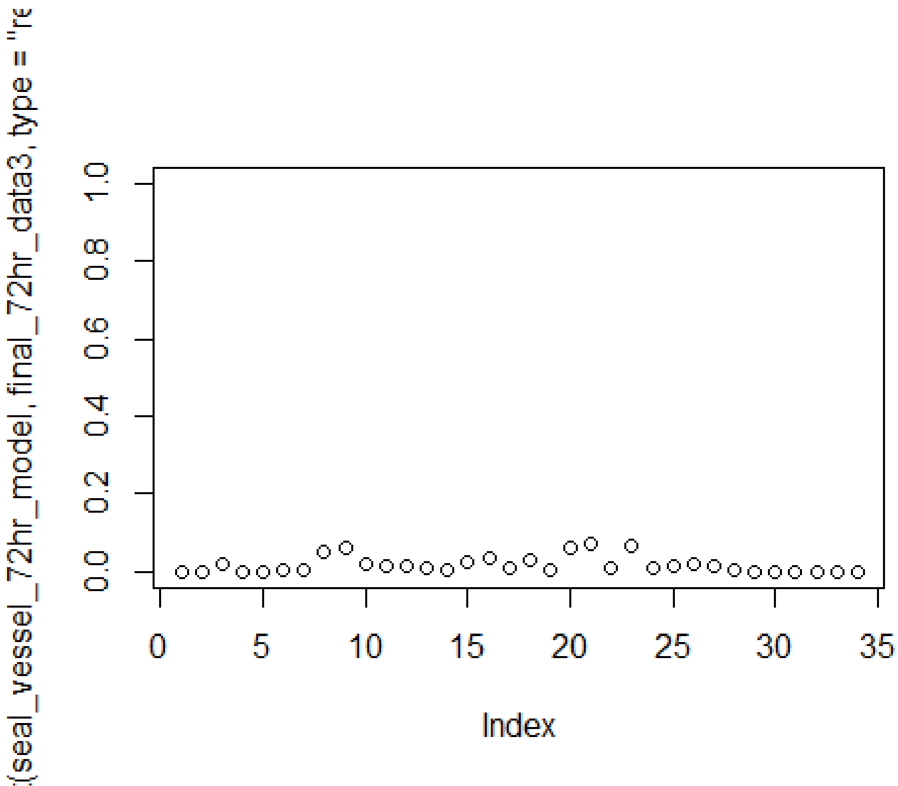



```r
plot(predict(seal_vessel_72hr_model,final_72hr_data1,type='response'),ylim
=c(0,1))
```

```r
plot(predict(seal_vessel_72hr_model,final_72hr_data2,type='response'),ylim
=c(0,1))
```




```r
plot(predict(seal_vessel_72hr_model,final_72hr_data3,type='response'),ylim
=c(0,1))
```



```
# save model predictions
predict_72hr_s1 <- predict(seal_vessel_72hr_model,
final_72hr_data1,type='response', se.fit = T)
predict_72hr_s2 <- predict(seal_vessel_72hr_model,
final_72hr_data2,type='response', se.fit = T)
predict_72hr_s3 <- predict(seal_vessel_72hr_model,
final_72hr_data3,type='response', se.fit = T)

# create dataframe
predict_72hr_s1 <- data.frame(predict_72hr_s1)
predict_72hr_s2 <- data.frame(predict_72hr_s2)
predict_72hr_s3 <- data.frame(predict_72hr_s3)
# 2 variables - fit and se.fit

# add total number of trials (eg total images taken for that time)
predict_72hr_s1$trials <- final_72hr_data1$n.y
predict_72hr_s2$trials <- final_72hr_data2$n.y
predict_72hr_s3$trials <- final_72hr_data3$n.y

# add time of day number
predict_72hr_s1$time_num <- final_72hr_data1$data_72hr
predict_72hr_s2$time_num <- final_72hr_data2$data_72hr
predict_72hr_s3$time_num <- final_72hr_data3$data_72hr

# calculate total number of photos of monk seal presence to compare with
actual data
predict_72hr_s1$photos <- predict_72hr_s1$fit * predict_72hr_s1$trials
predict_72hr_s2$photos <- predict_72hr_s2$fit * predict_72hr_s2$trials
predict_72hr_s3$photos <- predict_72hr_s3$fit * predict_72hr_s3$trials

# round to whole number of photos
predict_72hr_s1$photos <- round(predict_72hr_s1$photos, digits = 0)
predict_72hr_s2$photos <- round(predict_72hr_s2$photos, digits = 0)
predict_72hr_s3$photos <- round(predict_72hr_s3$photos, digits = 0)


jpeg("E:/BL5599/high_res_plots/72hr_GAM_predict1.jpeg", width = 6, height
= 4, units = 'in', res = 600)
plot(predict_72hr_s1$time_num, predict_72hr_s1$photos, ylim=c(0,
288),pch=16, col = "#A6CEE3",xaxt="n", cex.lab =1.2, main="Site 1", xlab =
"Date", ylab = "Total Images with Monk Seal Presence",cex=1.2,
cex.axis=1.2)
points(final_72hr_data1$data_72hr, final_72hr_data1$n.x,pch=16,
col=alpha("black",0.3),cex=1.2)
label_x <- c("June", "July", "August","September","October")
axis(1, at = c("4", "14", "24","34","44"), labels = label_x, cex.axis=1.2)
legend(33.5,280, legend=c("Predictions", "Actual"), col=c("#A6CEE3",
"grey47"), pch=16,cex=1)
dev.off()

## png
##   2
```

```
jpeg("E:/BL5599/high_res_plots/72hr_GAM_predict2.jpeg", width = 6, height
= 4, units = 'in', res = 600)
plot(predict_72hr_s2$time_num, predict_72hr_s2$photos, ylim=c(0,
288),pch=16, col = "#1F78B4",xaxt="n", cex.lab =1.2,main="Site 2", xlab =
"Date", ylab = "Total Images with Monk Seal Presence",cex=1.2,
cex.axis=1.2)
points(final_72hr_data2$data_72hr, final_72hr_data2$n.x,pch=16,
col=alpha("black",0.3),cex=1.2)
label_x <- c("June", "July", "August","September","October")
axis(1, at = c("4", "14", "24","34","44"), labels = label_x, cex.axis=1.2)
legend(33.5,90, legend=c("Predictions", "Actual"), col=c("#1F78B4",
"grey47"), pch=16,cex=1)
dev.off()

## png
##   2

jpeg("E:/BL5599/high_res_plots/72hr_GAM_predict3.jpeg", width = 6, height
= 4, units = 'in', res = 600)
plot(predict_72hr_s3$time_num, predict_72hr_s3$photos, ylim=c(0,
288),pch=16, col = "#B2DF8A",xaxt="n", cex.lab =1.2, main="Site 3", xlab =
"Date", ylab = "Total Images with Monk Seal Presence",cex=1.2,
cex.axis=1.2)
points(final_72hr_data3$data_72hr, final_72hr_data3$n.x,pch=16,
col=alpha("black",0.3),cex=1.2)
label_x <- c("June", "July", "August","September","October")
axis(1, at = c("4", "14", "24","34","44"), labels = label_x, cex.axis=1.2)
legend(33.5,280, legend=c("Predictions", "Actual"), col=c("#B2DF8A",
"grey47"), pch=16,cex=1)
dev.off()

## png
##   2
```