Appendix 1

This appendix contains supplemental information for work presented in Chapter 2.

Penguin breeding phenology

Adélie penguin breeding phenology data used in Chapter 2 were collected from seven breeding sites around the Antarctic. As these data were collected from several (separate) long-term studies, it was necessary to standardize all data to a similar metric of breeding phenology. The population mean clutch initiation date (CID – the mean day at which the first egg is laid in a nest) was used. Note when referring to year *t,* this represents the austral summer spanning years *t* and *t+1*.

Admiralty Bay – CID data were available for individual nests from 1985-2012. Nests were checked daily to determine the date of clutch initiation, using methodologies outlined by Hinke et al. (2012). Mean CIDs were calculated from individual nest data.

Humble Island – CID data were available for individual nests from 1991-2010, though years 1997, 1998, 2002, 2004, and 2005 were excluded from this analysis due to concerns with data integrity. These reflect years with heavy sea-ice in the region, which made sampling over the course of that season more intermittent. Nests were checked daily, when possible, to determine the date of clutch initiation. As data may be missing non-randomly from this time series, I take caution in interpreting trends from this site. Mean CIDs were calculated from individual nest data.

Petermann Island – CID data were available for individual nests from 2005-2007. Nests were checked daily to determine the date of clutch initiation, using methodologies outline by Lynch et al. (2010). Due to the short duration of this time series, these data were only used to calculate mean CID for Point Géologie (see below).

Point Géologie – While data were available since 1952 at Point Géologie, only data since 1979, the year in which satellite-derived sea-ice data are first available, were used.

Date of first CID (the first penguin of the colony to lay) was recorded from 1979-2012. Mean CIDs were extrapolated from first CID using the relationship between the first CID and mean CID for a given first CID. To do so, I used individual nest data from Admiralty Bay, Humble Island, and Petermann Island. Means and standard deviations were calculated at these sites for individual nest CIDs. Nest CIDs that were both greater (i.e. later) than their respective colony-wide means and more than 3.5 standard deviations away from them were excluded here. This was done to prevent the few nests in the right-hand tail of the breeding distribution from impacting the mean and standard deviation of lay dates within a year – breeding distributions within a season were slightly right-skewed. In this way, late outliers would not impact the relationship between first CID and the mean CID. All nest CIDs less (i.e. earlier) than the colony-wide mean were included, as these are relevant in modeling the relationship between mean and first CID.

The mean CIDs and the standard deviations were then used to simulate 100 normally distributed datasets. Minimum and mean values were then calculated for each one of these simulated datasets. In order to determine the relationship between the minimum and mean value for a given minimum value, a linear model was used.

$$y \sim N(\alpha + \beta X, \sigma^2) \qquad \text{(A1-1)}$$

where y is the mean value of the simulated data, X is the minimum value, and α and β are the intercept and slope, respectively. Coefficient estimates from this model were then used to extrapolate mean CID from minimum CID ($r^2 = 0.90$). This regression was conducted to account for any effect of mean CID on intra-annual variation in a given year. Any inaccuracies in the relationship to the true mean CID are unlikely to affect this analysis, as all values were standardized within sites and I would expect significant shifts in first lay date to track mean lay date.

Cape Crozier – Peak hatch date data were recorded from 1996-2010. Peak hatch date was assumed to be analogous to mean hatch date in this case. Mean CID was determined by subtracting the mean length of Adélie penguin incubation period (39 days; Trathan and Ballard 2013) from the mean hatch date. Any inaccuracies here in the relationship to the true mean CID are unlikely to affect this analysis, as all values were standardized within sites and significant shifts in mean hatch date would be expected to track mean CID.

Cape Royds – Peak hatch date data were recorded from 1997-2010. Data was processed in the same way as at Cape Crozier.

Cape Bird – Peak hatch date data were recorded from 1996-2010. Data was processed in the same way as at Cape Crozier and Cape Royds.

Béchervaise Island – Mean CIDs were calculated from individual nest CID data collected from 1990-2003. Nests were checked daily to determine the date of clutch initiation, using methodologies outlined by Smiley and Emmerson (2016).

Penguin breeding success

Data on Adélie penguin breeding success, defined here as the number of chicks to reach the crèche stage (period prior to fledging, characterized by distinctive grouping of penguin chicks) per breeding pair, were collected at each of the study sites. Similar protocols were used to collect all field data.

Admiralty Bay – Breeding success data were collected from 1986-2012. See Hinke et al. (2012) for a summary of methods used to collect breeding success data in the field.

Humble Island – Breeding success data were collected from 1991-2003. CEMP Standard Methods (CCAMLR 2014) were used to collect data on breeding success, except that the number of chicks raised per pair was monitored across several penguin colonies, as opposed to a single one. A different, random sample of nests was used in each season.

Point Géologie – Breeding success data were collected from 1992-2002, and from 2004-2013. See Jenouvrier et al. (2006) for a summary of methods used to collect breeding success data in the field.

Cape Crozier – Breeding success data were collected from 1996-2010. See Dugger et al. (2014) Dugger et al. (2014) for a summary of methods used to collect breeding success data in the field.

Cape Royds – Breeding success data were collected from 1996-2010. See Dugger et al. (2014) for a summary of methods used to collect breeding success data in the field.

Cape Bird – Breeding success data were collected from 1996-2010. See Dugger et al. (2014) for a summary of methods used to collect breeding success data in the field.

Béchervaise Island – Breeding success data were collected from 1990 – 2003. CEMP Standard Methods (CCAMLR 2014) were used to collect data on breeding success.

Phenology of the environment

Metrics for both phytoplankton-bloom onset (sea-ice adjusted light availability) were calculated for 25 km x 25 km pixels surrounding each site. Values were calculated for each pixel within the specified radius (Figure A1-1). The mean value of pixels within that specified area was then taken as the value at that site in that year.

Sea-ice adjusted light availability was calculated according to the methodology outlined in Li et al. (2016). A 10-hour day-length threshold was used as this more accurately represents bloom timing at Admiralty Bay than the 8-hour threshold used in Li et al. (2016), while also capturing bloom timing at other locations used in this study (Li et al. 2016). Bias between sea-ice adjusted light availability (bloom proxy) and actual bloom date among sites can be ignored as bloom timing was normalized across years and within sites.

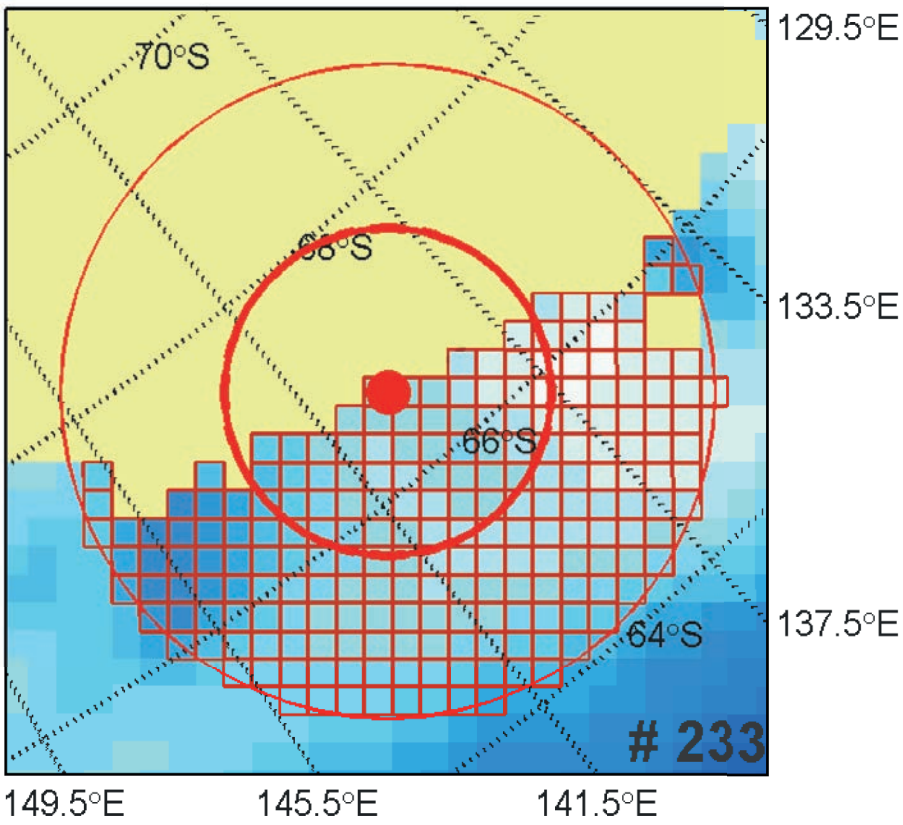

Figure A1-1: 25 km x 25 km pixels surrounding Point Géologie, with 150 km and 300 km radius boundaries depicted. These demarcations were made to calculate phytoplankton-bloom phenology metrics at each site used in this study.

Model for statistical analysis of trends in phenology and Mismatch Index

$$y_i \sim N(\mu_{ij}, {\sigma_j}^2) \qquad \text{(A1-2)}$$

$$\mu_{ij} = \alpha_j + \beta_j * Year_i$$

$$\alpha_j \sim N(\mu_\alpha, {\sigma_\alpha}^2)$$

$$\beta_j \sim N(\mu_\beta, {\sigma_\beta}^2)$$

$$1/{\sigma_j}^2 \sim Gamma(0.01,0.01)$$

$$\mu_\alpha \sim N(0,0.001)$$

$$\mu_\beta \sim N(0,0.001)$$

$$1/{\sigma_\alpha}^2 \sim Gamma(0.01,0.01)$$

$$1/{\sigma_\beta}^2 \sim Gamma(0.01,0.01)$$

where the response variable (y) is modeled as normally distributed with a mean μ_{ij} that is a linear function of year (i) with location (j)-specific slope and intercept. The coefficients of the linear model for μ_{ij} (α_j and β_j) are modeled as normally distributed with mean μ_α and μ_β, and precision $1/{\sigma_\alpha}^2$ and $1/{\sigma_\beta}^2$, respectively. Hyper-parameters were given uninformative normal priors, while coefficient precisions were given uninformative gamma priors. Note the second parameter specified for the normal distributions is the precision, or inverse variance, rather than variance as per formatting in JAGS (Plummer 2003).

Model for quantile regression

$$y = \mu + \varepsilon \tag{A1-3}$$

$$\mu = \alpha + \beta_1 * X + \beta_2 * X^2$$

$$\varepsilon \sim ALD(location = 0, scale = \sigma, quantile = \tau)$$

$$\alpha \sim N(0,0.01)$$

$$\beta_1 \sim N(0,0.01)$$

$$\beta_2 \sim N(0,0.01)$$

$$\sigma \sim invGamma(0.01,0.01)$$

where *y* is the response variable (breeding success), modeled as a quadratic. The errors are distributed according to an asymmetric Laplace distribution. The quadratic model parameters (α, β_1, β_2) were each given uninformative normal priors, while the scale parameter for the asymmetric Laplace distribution (σ) was given an uninformative inverse gamma prior.

Appendix 2

This appendix contains supplemental results for analyses presented in Chapter 2.

Covariates do not predict mean breeding success

Table A2-1: Breeding success (BS) as a function of penguin breeding phenology (CID), Bloom Mismatch Index (BMI), and Sea-ice Mismatch Index (SMI). Model is a quadratic regression fit in a frequentist framework (so as to produce an interpretable r^2 value). Low r^2 values indicate that the predictor values do not predict the response variable with any reliability.

Model	r^2
BS ~ CID	0.078
BS ~ BMI	0.042
BS ~ SMI	0.035

Parameter estimates for hierarchical regressions

Parameter estimates derived from Bayesian analyses (posterior mean values) can be interpreted in a similar manner to point estimates derived from frequentist analyses. Bayesian posteriors represent the uncertainty around the estimate of a particular parameter. Larger 'credible intervals' (the range of values in which there is an X percent certainty that the X% interval contains the true parameter value) result from wider posterior distributions and denote 'less confidence' in the estimated value for the parameter.

Table A2-2: Parameter estimates (posterior mean and 95% credible intervals) for penguin breeding phenology as a function of time, modeled in a hierarchical Bayesian framework (Equation 2-3). $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Site	Parameter	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	α	0.064	-0.681	0.793	1.001
AB	β	-0.005	-0.051	0.041	1.001
HI	α	0.269	-1.105	2.135	1.001
HI	β	-0.008	-0.058	0.03	1.001
PG	α	-0.118	-1.708	1.144	1.001
PG	β	0.002	-0.018	0.026	1.001
CC	α	-0.079	-2.604	1.832	1.001
CC	β	9.538×10^{-4}	-0.021	0.029	1.001
CR	α	-0.195	-3.241	1.663	1.001
CR	β	0.002	-0.016	0.031	1.001
CB	α	-0.033	-2.53	2.032	1.001
CB	β	2.98×10^{-4}	-0.017	0.021	1.001
BE	β	0.073	-0.016	0.031	1.001
BE	β	-5.91×10^{-4}	-0.018	0.016	1.001

Table A2-3: Parameter estimates (posterior mean and 95% credible intervals) for Bloom Mismatch Index as a function of time, modeled in a hierarchical Bayesian framework (Equation 2-3). $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Site	Parameter	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	α	-0.258	-1.037	0.517	1.001
AB	β	0.021	-5.466	6.372	1.001
HI	α	-3.065	-5.376	-0.079	1.001
HI	β	0.018	-0.03	0.066	1.001
PG	α	1.483	-0.71	3.717	1.001
PG	β	0.083	0.005	0.144	1.001
CC	α	-0.316	-5.246	4.791	1.001
CC	β	-0.023	-0.057	0.01	1.001
CR	α	-0.672	-6.198	4.633	1.001
CR	β	0.003	-0.053	0.058	1.001
CB	α	-0.206	-5.548	5.551	1.001
CB	β	0.006	-0.044	0.059	1.001
BE	β	0.021	-0.044	0.059	1.001
BE	β	-1.874×10^{-4}	-0.047	0.041	1.001

Table A2-4: Parameter estimates (posterior mean and 95% credible intervals) for Sea-ice Mismatch Index as a function of time, modeled in a hierarchical Bayesian framework (Equation 2-3). $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Site	Parameter	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	α	-0.397	-1.164	0.368	1.001
AB	β	-0.478	-6.413	5.135	1.001
HI	α	-2.264	-4.813	0.31	1.001
HI	β	0.028	-0.019	0.076	1.001
PG	α	1.974	-0.449	4.188	1.001
PG	β	0.061	-0.007	0.129	1.001
CC	α	-1.096	-6.605	3.442	1.001
CC	β	-0.031	-0.064	0.006	1.001
CR	α	-1.619	-8.351	3.138	1.001
CR	β	0.012	-0.038	0.073	1.001
CB	α	-0.861	-6.79	4.24	1.001
CB	β	0.015	-0.03	0.079	1.001
BE	β	-0.478	-0.03	0.079	1.001
BE	β	0.004	-0.038	0.048	1.001

Table A2-5: Parameter estimates (posterior mean and 95% credible intervals) for differences in the β (slope) parameter for penguin breeding phenology. This metric was used to determine if a parameter was changing at any one site more rapidly than another. $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Site 1	Site 2	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	HI	0.002	−0.044	0.039	1.001
AB	PG	0.011	−0.016	0.045	1.001
AB	CC	0.011	−0.02	0.046	1.001
AB	CR	0.011	−0.02	0.047	1.001
AB	CB	0.013	−0.014	0.047	1.001
AB	BE	0.013	−0.012	0.048	1.001
HI	PG	0.009	−0.015	0.049	1.001
HI	CC	0.009	−0.016	0.046	1.001
HI	CR	0.01	−0.015	0.047	1.001
HI	CB	0.011	−0.01	0.048	1.001
HI	BE	0.011	−0.009	0.049	1.001
PG	CC	5.568×10^{-5}	−0.023	0.018	1.001
PG	CR	7.382×10^{-4}	−0.022	0.018	1.001
PG	CB	0.002	−0.016	0.019	1.001
PG	BE	0.003	−0.014	0.02	1.001
CC	CR	6.825×10^{-4}	−0.018	0.02	1.001
CC	CB	0.002	−0.014	0.022	1.001
CC	BE	0.003	−0.012	0.023	1.001
CR	CB	0.001	−0.014	0.021	1.001
CR	BE	0.002	−0.012	0.022	1.001
CB	BE	5.843×10^{-4}	−0.013	0.016	1.001

Table A2-6: Parameter estimates (posterior mean and 95% credible intervals) for differences in the β (slope) parameter for Bloom Mismatch Index. This metric was used to determine if a parameter was changing at any one site more rapidly than another. $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Site 1	Site 2	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	HI	-0.059	-0.127	0.019	1.001
AB	PG	-2.763×10^{-5}	-0.044	0.05	1.001
AB	CC	0.004	-0.049	0.053	1.001
AB	CR	0.005	-0.046	0.053	1.001
AB	CB	0.008	-0.039	0.055	1.001
AB	BE	0.009	-0.037	0.056	1.001
HI	PG	0.059	-0.008	0.122	1.001
HI	CC	0.063	-0.003	0.127	1.001
HI	CR	0.064	-0.001	0.127	1.001
HI	CB	0.067	2.865×10^{-4}	0.129	1.001
HI	BE	0.068	7.942×10^{-4}	0.13	1.001
PG	CC	0.004	-0.044	0.043	1.001
PG	CR	0.005	-0.041	0.044	1.001
PG	CB	0.008	-0.033	0.045	1.001
PG	BE	0.009	-0.031	0.045	1.001
CC	CR	0.001	-0.044	0.048	1.001
CC	CB	0.004	-0.037	0.05	1.001
CC	BE	0.005	-0.035	0.05	1.001
CR	CB	0.003	-0.038	0.047	1.001
CR	BE	0.004	-0.036	0.047	1.001
CB	BE	0.001	-0.037	0.04	1.001

Table A2-7: Parameter estimates (posterior mean and 95% credible intervals) for differences in the β (slope) parameter for Sea-ice Mismatch Index. This metric was used to determine if a parameter was changing at any one site more rapidly than another. $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Site 1	Site 2	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	HI	-0.031	-0.105	0.037	1.001
AB	PG	-3.543×10^{-4}	-0.048	0.053	1.001
AB	CC	0.009	-0.051	0.061	1.001
AB	CR	0.009	-0.054	0.061	1.001
AB	CB	0.014	-0.037	0.063	1.001
AB	BE	0.016	-0.032	0.065	1.001
HI	PG	0.031	-0.031	0.099	1.001
HI	CC	0.04	-0.02	0.108	1.001
HI	CR	0.04	-0.02	0.106	1.001
HI	CB	0.045	-0.011	0.112	1.001
HI	BE	0.047	-0.009	0.114	1.001
PG	CC	0.009	-0.044	0.051	1.001
PG	CR	0.009	-0.047	0.051	1.001
PG	CB	0.014	-0.029	0.052	1.001
PG	BE	0.016	-0.024	0.053	1.001
CC	CR	-2.683×10^{-4}	-0.053	0.049	1.001
CC	CB	0.005	-0.038	0.055	1.001
CC	BE	0.007	-0.033	0.057	1.001
CR	CB	0.005	-0.036	0.058	1.001
CR	BE	0.007	-0.031	0.06	1.001
CB	BE	0.002	-0.035	0.042	1.001

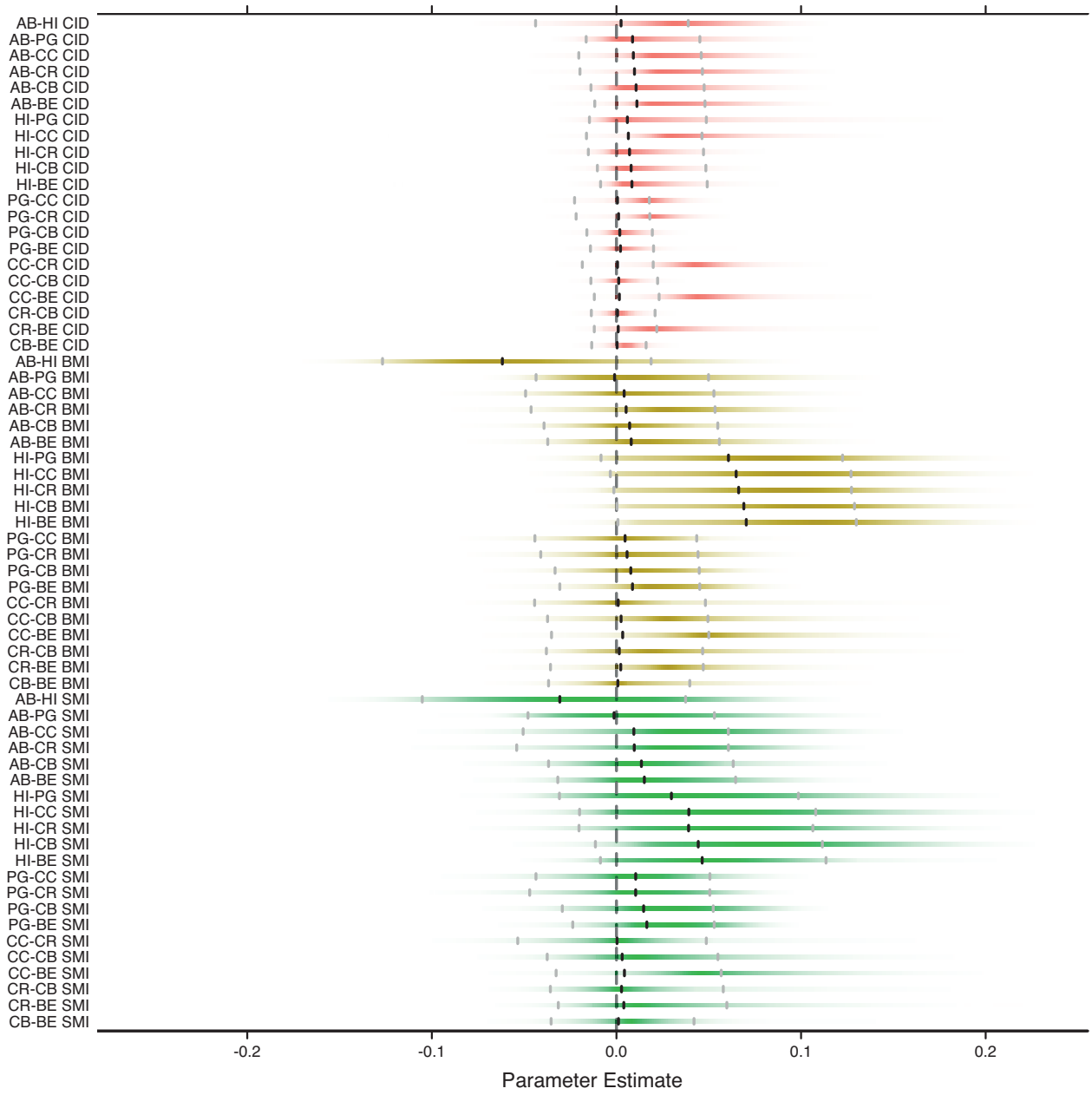

Figure A2-1: Posterior distributions for differences in the β (slope) parameter between any given site and all others for penguin breeding phenology (CID - red), Bloom Mismatch Index (BMI - yellow), and Sea-ice Mismatch Index (SMI - green). Posterior means are indicated by the black tick. 95% credible intervals are indicated by the gray ticks. Letter codes represent breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie, CC = Cape Crozier, CR = Cape Royds, CB = Cape Bird, BE = Béchervaise Island.

Parameter estimates for quantile regression

Table A2-8: Parameter estimates (posterior mean and 95% credible intervals) for model of breeding success (BS) as a function of penguin breeding phenology (CID), Bloom Mismatch Index (BMI), and Sea-ice Mismatch Index (SMI) obtained from a quadratic 85th quantile regression implemented in a Bayesian framework (Equation 2-2).

Model	Coefficient term	Estimate	Lower 95% CI	Upper 95% CI
BS ~ CID	α	0.884	0.738	1.04
BS ~ CID	β_1 (linear)	-0.321	-0.462	-0.214
BS ~ CID	β_2 (quadratic)	0.002	-0.076	0.088
BS ~ BMI	α	1.156	1.004	1.292
BS ~ BMI	β_1	0.088	-0.059	0.236
BS ~ BMI	β_2	-0.173	-0.266	-0.076
BS ~ SMI	α	1.171	1.001	1.308
BS ~ SMI	β_1	-0.079	-0.221	0.082
BS ~ SMI	β_2	-0.225	-0.32	-0.123

Appendix 3

This appendix contains supplemental information and supplemental analyses for work presented in Chapter 2.

Intro

Trends in penguin breeding phenology were previously analyzed at Admiralty Bay (1991-2006) and Humble Island (1991-2000) by Lynch et al. (2012a) and at Point Géologie (1952-2005) by Barbraud and Weimerskirch (2006). Lynch et al. (2012a) concluded that penguin breeding phenology was advancing (occurring earlier over time) over 16 years, from 1991-2006, at Admiralty Bay and over 10 years, from 1991-2000, at Humble Island. Barbraud and Weimerskirch (2006) concluded that penguin breeding phenology was becoming increasingly delayed (breeding occurring later over time) over 54 years, from 1952-2005. This suggests a dichotomy between phenological trends on the Western Antarctic Peninsula (Admiralty Bay and Humble Island) and Eastern Antarctic (Point Géologie).

I found no evidence to support the notion of a shift in Adélie penguin breeding phenology or a dichotomy between the Western Antarctic Peninsula and Eastern Antarctica. However, the length of the time series used in these previous studies differs from those used here. Lynch et al. (2012a) also incorporated species other than the Adélie penguin in their previous analysis.

Table A3-1: Outline of time periods used in this study and previous ones at Point Géologie, Admiralty Bay, and Humble Island. Trends found are also included. +: delaying trend, 0: no trend, -: advancing trend.

Site	Study	Time Period	Trend Found
Admiralty Bay	Lynch et al. 2012a	1991 - 2006	-
Admiralty Bay	This study	1986 - 2012	0
Humble Island	Lynch et al. 2012a	1991 - 2000	-
Humble Island	This study	1991 - 2010	0
Point Géologie	Barbraud and Weimerskirch 2006	1952 - 2005	+
Point Géologie	This study	1979 - 2012	0

Methods

Time series reanalyses

To resolve this discrepancy in phenological trends, I reanalyzed these time series. Temporal trends for Adélie penguin clutch initiation dates (CIDs) were modeled individually using a Bayesian approach. Phenology was modeled as normally distributed with a mean μ_i that is a linear function of year (i). The coefficients of the linear model for μ_i were given uninformative normal priors. The precision $(1/\sigma^2)$ was given an uninformative gamma prior.

$$
\begin{aligned}
y_i &\sim N(\mu_i, \sigma^2) \\
\mu_i &= \alpha + \beta * Year_i \\
\alpha &\sim N(0, 0.001) \\
\beta &\sim N(0, 0.001) \\
1/\sigma_2 &\sim Gamma(0.01, 0.01)
\end{aligned}
\tag{A3-1}
$$

Models were fitted using the R package 'R2jags' (Su and Yajima 2015), to interface with JAGS (Plummer 2003) in the R statistical environment (R Development Core Team 2016). Inferences were derived from 10,000 samples, following a 'burn-in' period of 40,000 draws using 3 chains. Model convergence was assessed through a visual analysis of posterior chains,

in addition to the use of the Gelman-Rubin convergence diagnostic (Brooks and Gelman 1998). All models unambiguously converged.

CID data (x_{ij}) were normalized across years (i) for each distinct time period used in previous analyses (j), to create a standardized variable (S) that allows for more meaningful inter-site comparisons:

$$S_{ij} = \frac{x_{ij} - \bar{x_j}}{sd(x_j)} \tag{A3-2}$$

Resampling of time series

To determine the effect on the length and the particular period used in these studies, I re-sampled each time series over various lengths of time, from 3 years to the entire length of the time series used in the primary analysis of this study. For each length of time, 1000 random segments (consisting of consecutive years) were sampled from the complete time series. A frequentist linear model was fitted to each sample to determine whether a phenological trend was apparent. The proportion of randomly sampled segments determined to be statistically significant at the $\alpha = 0.05$ level were then recorded for each length of time.

Results

Estimated trends in penguin breeding phenology at Admiralty Bay (Figure A3-1, Table A3-2) were similar for both periods of time analyzed. Estimated slopes for both models were near zero (Table A3-2, Figure A3-4).

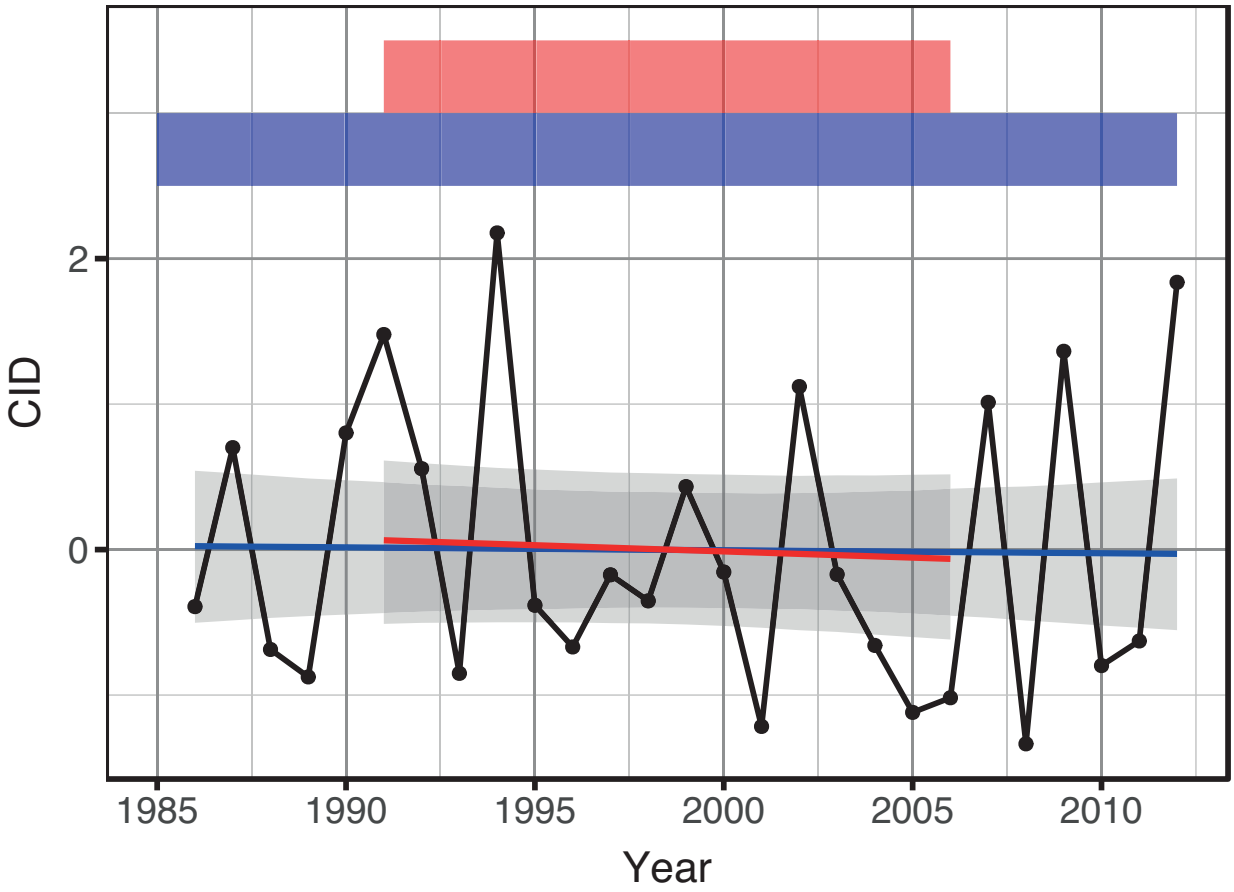

Figure A3-1: Adélie penguin breeding phenology over time at Admiralty Bay (Figure 2-1). Standardized clutch initiation date (CID) is shown on the y-axis, year on the x-axis. The red bar represents the period of time analyzed by Lynch et al. (2012a). The blue bar represents the period of time analyzed in this study. Red and blue lines represent model fits for 1991-2006 (used by Lynch et al. [2012a]) and 1986-2012 (used by this study), respectively.

Estimated trends in penguin breeding phenology at Humble Island (Figure A3-2, Table A3-2) were also similar for both periods of time analyzed. Estimated slopes for both models were near zero (Table A3-2, Figure A3-4).

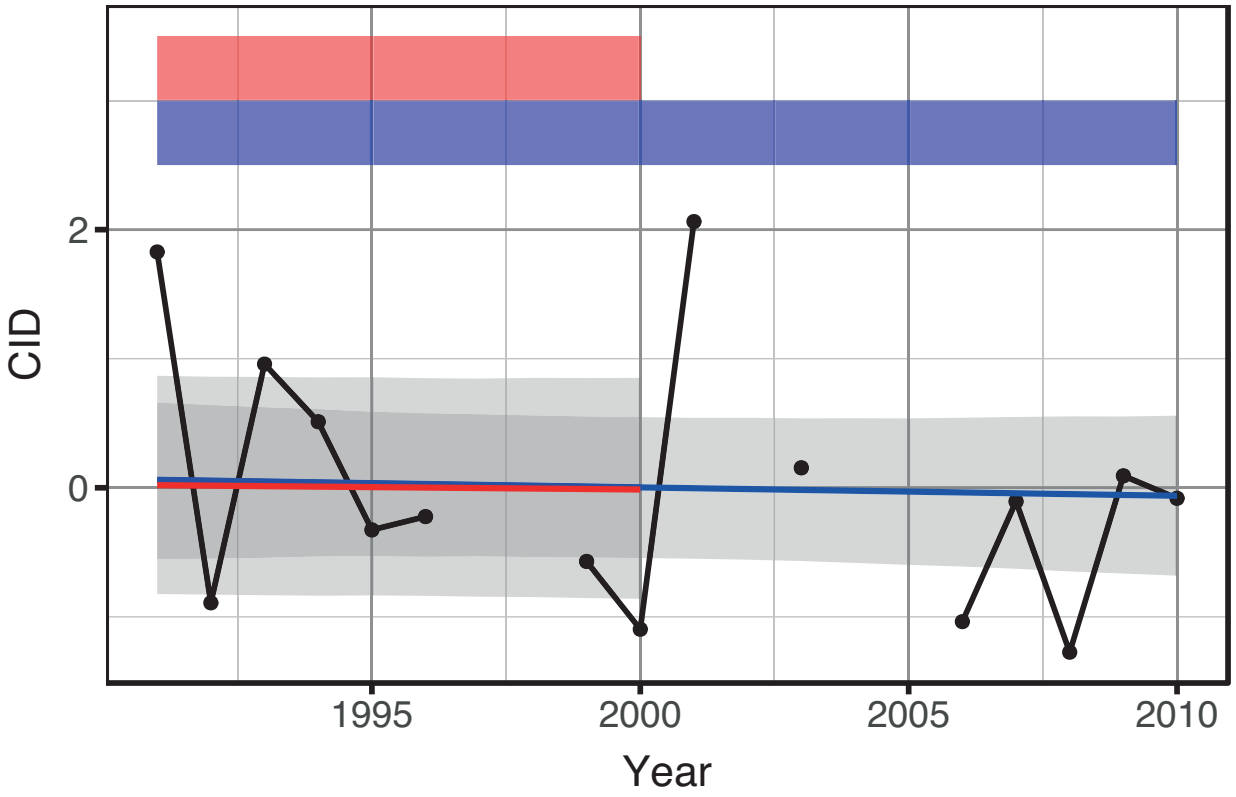

Figure A3-2: Adélie penguin breeding phenology over time at Humble Island (Fig. 1). Standardized clutch initiation date (CID) is shown on the y-axis, year on the x-axis. The red bar represents the period of time analyzed by Lynch et al. (2012a). The blue bar represents the period of time analyzed in this study. Red and blue lines represent model fits for 1991-2000 (used by Lynch et al. [2012a]) and 1991-2010 (used by this study), respectively.

Estimated trends in penguin breeding phenology at Point Géologie (Figure A3-3, Table A3-2) were different between the two periods of time analyzed. The time period 1952-2005, used by Barbraud and Weimerskirch (2006), showed a stronger trend than did the time period 1979-2012, used in this study (Table A3-2, Figure A3-4).

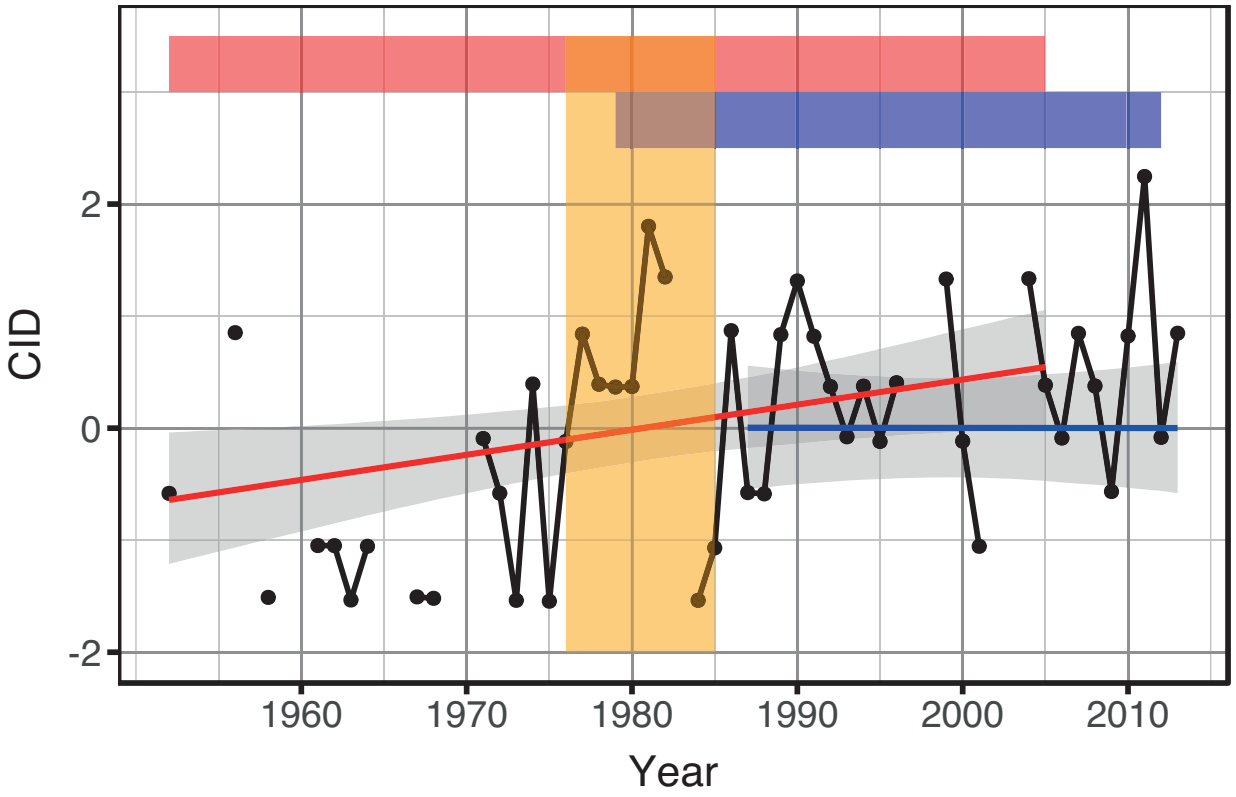

Figure A3-3: Adélie penguin breeding phenology over time at Point Géologie (Figure 2-1). Standardized clutch initiation date (CID) is shown on the y-axis, year on the x-axis. The red bar represents the period of time analyzed by Barbraud and Weimerskirch (2006). The blue bar represents the period of time analyzed in this study. The orange bar denotes period of regime shift at this site (Jenouvrier et al. 2005). Red and blue lines represent model fits for 1952-2005 (used by Barbraud and Weimerskirch [2006]) and 1979-2012 (used by this study), respectively.

Table A3-2: Parameter estimates (posterior mean and 95% credible intervals) for penguin breeding phenology as a function of time. $\hat{R}$ represents the Gelman-Rubin convergence diagnostic. Values near 1 signify convergence for that particular parameter. Letter codes represent breeding breeding sites: AB = Admiralty Bay, HI = Humble Island, PG = Point Géologie. B & W = Barbraud and Weimerskirch 2006

Site	Study	Parameter	Mean	Lower 95% CI	Upper 95% CI	$\hat{R}$
AB	Lynch et al. 2012a	α	17.04	-44.43	78.74	1
AB	Lynch et al. 2012a	β	-0.01	-0.04	0.02	1
AB	This study	α	4.06	-49.51	55.88	1
AB	This study	β	0	-0.03	0.02	1
HI	Lynch et al. 2012a	α	7.39	-54.59	69.53	1
HI	Lynch et al. 2012a	β	0	-0.03	0.03	1
HI	This study	α	13.03	-46.63	71.12	1
HI	This study	β	-0.01	-0.04	0.02	1
PG	B & W 2006	α	-44.33	-79.02	-8.1	1
PG	B & W 2006	β	0.02	0	0.04	1
PG	This study	α	0.24	-54	53.94	1
PG	This study	β	-1.19×10^{-4}	-0.03	0.03	1

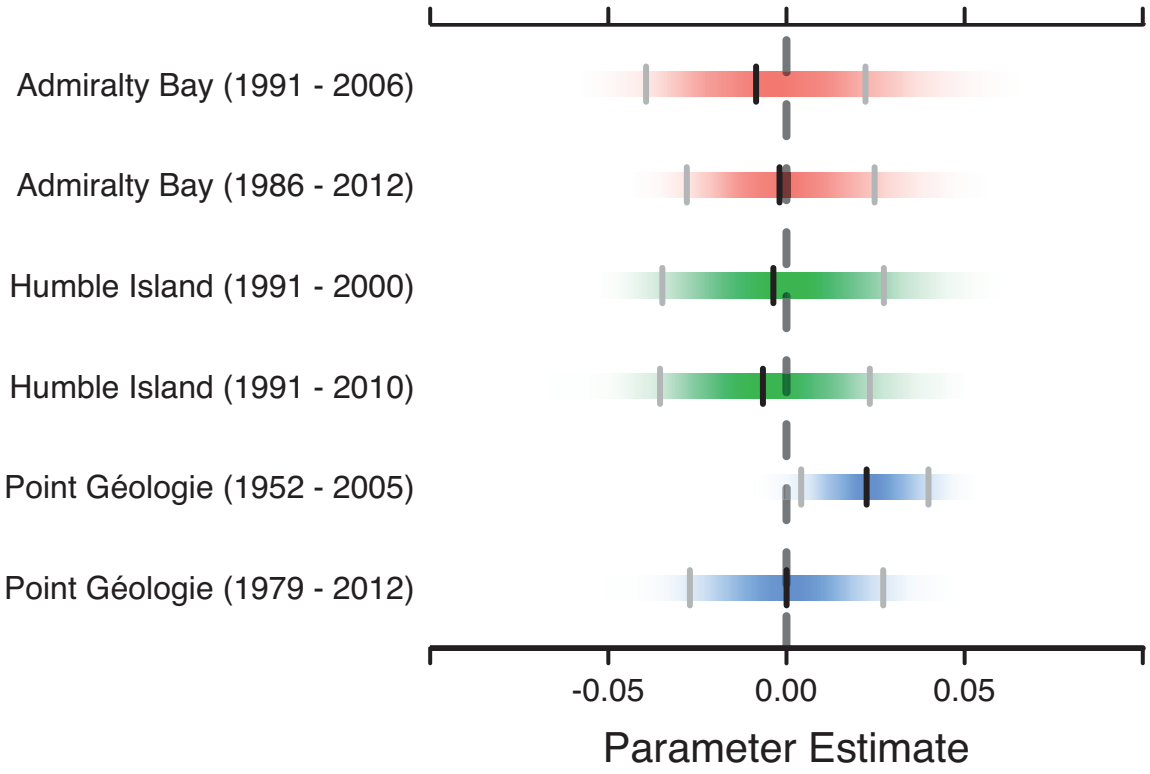

Figure A3-4: Posterior estimates for slope (β) parameters at Admiralty Bay, Humble Island, and Point Géologie. Time periods correspond to those used in previous studies by Lynch et al. (2012a) and Barbraud and Weimerskirch (2006), as well as this study.

Time series reanalyses

Time series reanalyses using subsets of the time series showed that 'statistically significant' trends in Adélie penguin breeding phenology may be found when using particular subsets (Figure A3-5).

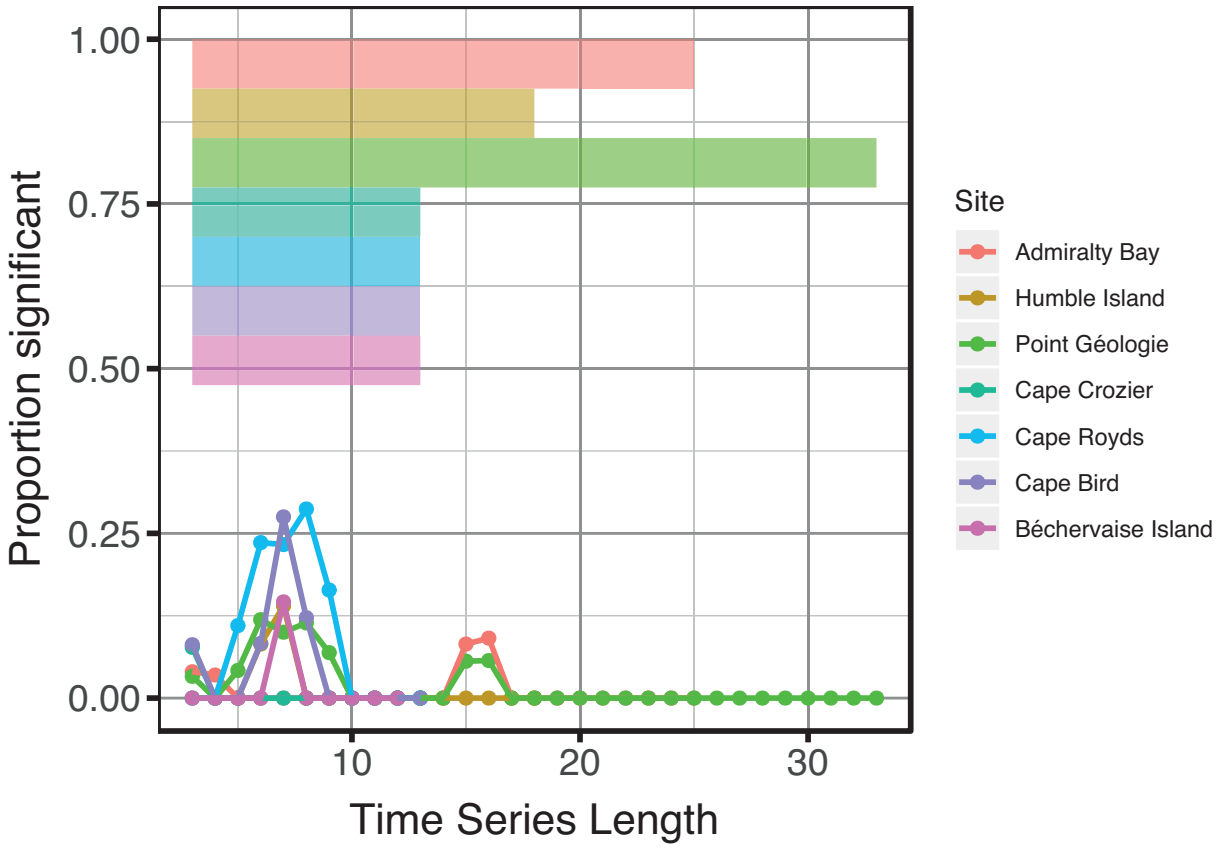

Figure A3-5: Proportion of time series subsets of a particular length that were found to have a 'statistically significant' trend in Adélie penguin breeding phenology (slope, β, parameter $p < 0.05$). Colored bars represent the length of time series at each site used in the primary analysis for this study.

Discussion

While this analysis of the time series suggested no trend in Adélie penguin breeding phenology at either Admiralty Bay or Humble Island, Lynch et al. (2012a) suggested that a trend was apparent. This is surprising considering that I reanalyzed data over the same period of time used in this previous study. However, Lynch et al. (2012a) used a hierarchical model, incorporating data from not only Adélie penguins, but also chinstrap penguins (*Pygoscelis antarctica*) and gentoo penguins (*Pygoscelis papua*). Lynch et al. (2012a) also explicitly modeled individual nest breeding phenology, where I consider here mean population breeding phenology. As Hinke et al. (2012) point out, individual Adélie penguins exhibit some flexibility around a more rigid population mean. The inclusion of both individual level data and these other species likely resulted in the differing results between this study and Lynch et al. (2012a).

Estimated trends at Point Géologie differ substantially. I attribute this difference to a regime shift at Point Géologie during the 1970s/1980s (Jenouvrier et al. 2005; Figure A3-3). This discrete change in phenology is distinct from long-term trends in phenology and better explains the shift towards later breeding phenology between the 1950s and present at this site.

Evaluation of trends for random time series subsets at each site showed that 'statistically significant' trends can be found for some subsets of these time series, despite no trend being seen using the complete datasets. This is likely due to the low signal to noise ratio present in these time series. It should also be noted that a statistically significant trend does not equate to a biologically significant trend, particularly when large interannual variation persists in the system. Ultimately, I suggest that penguin breeding phenology is not changing at a biologically significant rate. Any small trend that does exist is largely masked by large interannual variation.

Appendix 4

This appendix contains supplemental information and supplemental analyses for work presented in Chapter 2.

While I initially used a 250 km radius around each colony to calculate the sea-ice adjusted light (as a proxy for phytoplankton bloom timing) and sea-ice retreat, I also analyzed these metrics at a 150 km radius around each colony. These values were used to calculate Bloom Mismatch Index and Sea-ice Mismatch Index values for 150 km as well as the initial 250 km. A quadratic 85th quantile regression implemented in a Bayesian framework (Equation 2-2), was used to model breeding success as a function of Bloom Mismatch Index and Sea-ice Mismatch Index at each of these scales. Data from all sites were used to conduct the analyses.

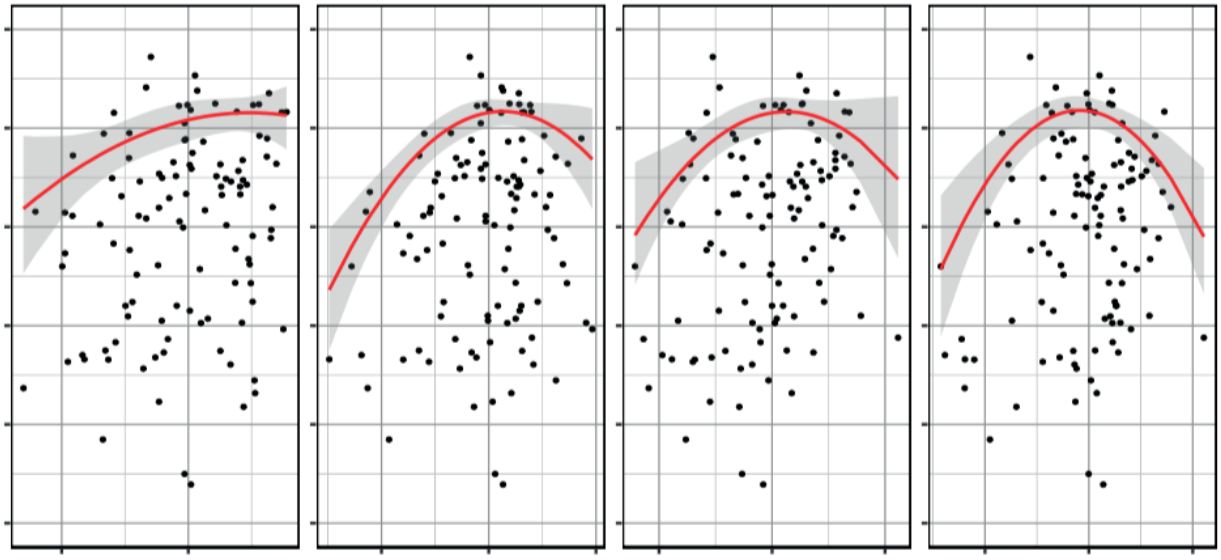

Figure A4-1: Breeding success as a function of: a) Bloom Mismatch Index using a 150 km radius, b) Bloom Mismatch Index using a 250 km radius, c) Sea-ice Mismatch Index using a 150 km radius, and d) Sea-ice Mismatch Index using a 250 km radius. Model fit for 85th quantile regression shown in red with credible intervals (95%) shown in grey. All measures are normalized. Data points from all sites are shown.

Table A4-1: Parameter estimates (posterior mean and 95% credible intervals) for model of breeding success (BS) as a function of Bloom Mismatch Index (BMI) and Sea-ice Mismatch Index (SMI) obtained from a quantile regression.

Model	Scale	Coef term	Estimate	Lower 95% CI	Upper 95% CI
BS ~ BMI	150km	β_1	0.15	$-2.8 * 10^{-4}$	0.29
BS ~ BMI	150km	β_2	-0.08	-0.19	0.05
BS ~ BMI	250km	β_1	0.09	-0.06	0.24
BS ~ BMI	250km	β_2	-0.17	-0.27	-0.08
BS ~ SMI	150km	β_1	0.09	-0.07	0.24
BS ~ SMI	150km	β_2	-0.17	-0.3	-0.02
BS ~ SMI	250km	β_1	-0.08	-0.22	0.08
BS ~ SMI	250km	β_2	-0.22	-0.32	-0.12

Results show that the mismatch indices are not good predictors of mean breeding success at any scale, but that the 'necessary but not sufficient' pattern, in which the indices set an upper limit on breeding success is more apparent at 250 km, than at 150 km. As these metrics were derived from satellite sensors, I ultimately believe that larger spatial scales more accurately

capture measures of environmental phenology - values will be integrated over more pixels. The sensitivity of the mean to the value of any one pixel will be reduced using a larger radius in this way, and incorporate more pixels potentially used by penguins as determined using their maximum foraging range.

Appendix 5

This appendix contains supplemental information and supplemental analyses for work presented in Chapter 2.

Intro

One proposed explanation for the evolution of coloniality in seabirds is the avoidance of predators (Darling 1938, Emlen and Demong 1975). A nest surrounded on all sides by other nests will experience less of an 'edge effect', making it easier to effectively defend against aerial predators, ultimately resulting in higher breeding success (Young 1994). It might be assumed that breeding synchrony among individuals is important to take full advantage of this strategy. I analyzed individual clutch initiation date (CID) and associated breeding success data to assess the importance of intra-colony synchrony in reproductive success.

Methods

Effect of breeding synchrony on breeding success

Data on clutch initiation at individual nests were collected at one site, Admiralty Bay, which permitted an analysis of the effect of within-site (intra-annual) breeding synchrony on population level breeding success. Variance in clutch initiation date was calculated for each year in which data were available (1986-2012). Nesting dates that fell outside of 3.5 standard deviations from the mean in each year were excluded from calculations of breeding success and CID variance to avoid bias in the variance component attributable to these extreme outliers. A linear model was used to examine the effect of breeding synchrony on breeding success:

$$\begin{aligned} y &\sim N(\alpha + \beta * X, \sigma^2) \\ \alpha &\sim N(0, 0.001) \\ \beta &\sim N(0, 0.001) \\ 1/\sigma_2 &\sim Gamma(0.01, 0.01) \end{aligned} \tag{A5-1}$$

where y is standardized breeding success, X is intra-annual variance in breeding phenology, α and β are the intercept and slope, respectively, and $1/\sigma^2$ is the or precision, or inverse-variance. Both α and β were given uninformative normal priors, while $1/\sigma^2$ was given an uninformative gamma prior.

Results

A decrease in breeding success is associated with a decrease in breeding synchrony (increase in CID variance; Figure A5-1).

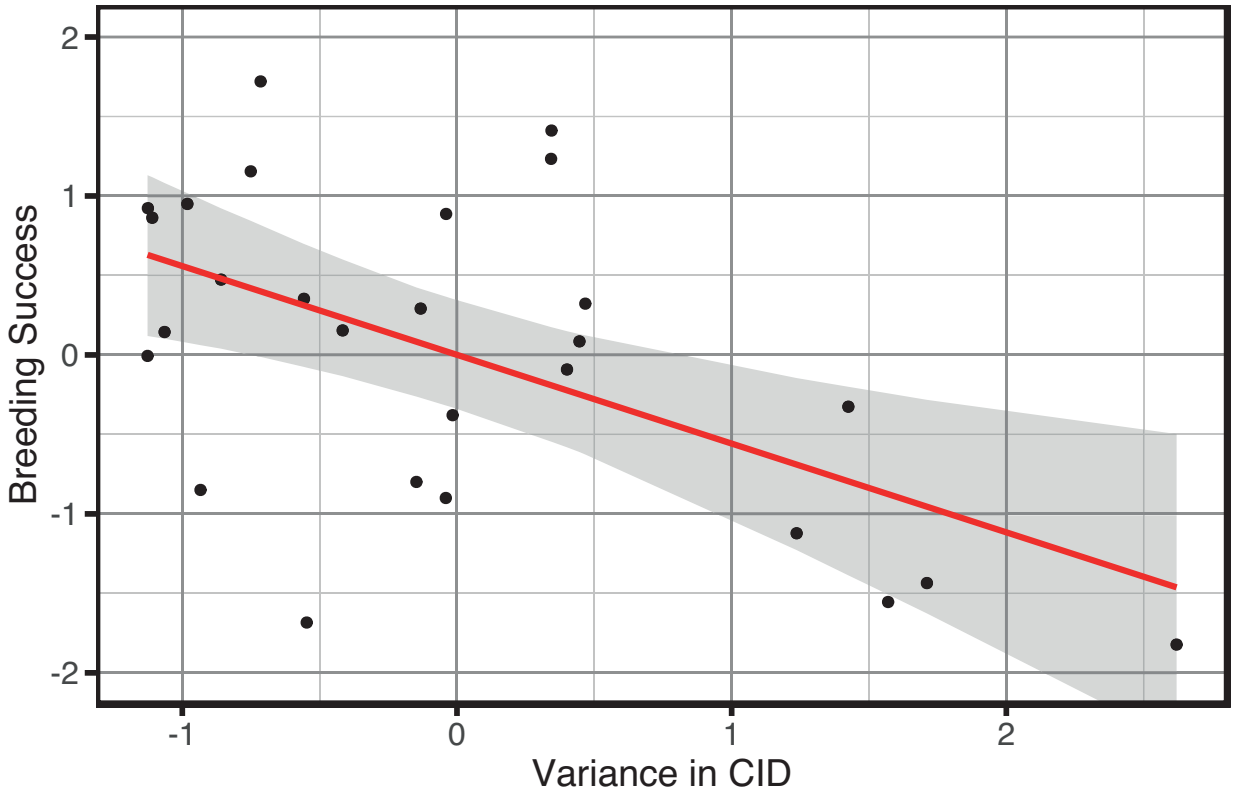

Figure A5-1: Breeding success as a function of intra-annual variance in CID at Admiralty Bay. Measures are standardized. Fit for linear model shown in red with credible intervals (95%) shown in gray.

Table A5-1: Parameter estimates (posterior mean and 95% credible intervals) for linear regression (Equation A5-1).

Coefficient term	Estimate	Lower 95% CI	Upper 95% CI
α	6.548×10^{-4}	-0.339	0.343
β	-0.558	-0.9	-0.211

Discussion

These results suggest that breeding synchrony is important for highly successful breeding for Adélie penguins at this location. This is likely due to reduced predation experienced when all individuals initiate breeding at a similar time. Previous work has shown that predation by avian predators, such as brown skuas (*Catharacta lonnbergi*), kelp gulls (*Larus dominicanus*), and giant petrels (*Macronectes giganteus*), can significantly reduce breeding success of Adélie penguins (Emslie et al. 1995). Penguin nests on colony edges were found to be more vulnerable to predation due to lack of neighboring nests that facilitate predator resistance. While I show here synchrony to be important, rather than position in the colony, I suggest a similar mechanism is at play.

Appendix 6

This appendix contains supplemental information, supplemental analyses, and R code to reproduce all analyses presented in Chapter 3.

Initial set up

Packages

```
#install packages if they don't exist - then load them
if('pacman' %in% rownames(installed.packages()) == FALSE)
{
  install.packages('pacman')
}

pacman::p_load(ggplot2,
               reshape2,
               dplyr,
               rjags,
               moments,
               MCMCvis)
```

Inter-annual variation in CID

Load data

```
setwd('Data')

captive_data <- read.csv('Captive_CID.csv', header = TRUE)
wild_data <- read.csv('Wild_CID.csv', header = TRUE)
```

Function to calculate intra-annual median and var

```
srt.fun <- function(IN)
{
  #IN <- SD_lay
  yrs <- unique(IN$YEAR)

  OUT <- c()
  for(i in min(yrs):max(yrs))
  {
    #i <- 1993
    temp <- filter(IN, YEAR == i)
    t_md <- median(temp$J_CID)
    t_var <- var(temp$J_CID)
    temp2 <- data.frame(YEAR = i, MEDIAN = t_md, VAR = t_var)
    OUT <- rbind(OUT, temp2)
  }
  return(OUT)
}
```

Function to calculate SE of variance

$\sigma_{s^2} = s^2 * \sqrt{2/(n-1)}$

```
se_var <- function(data)
{
  OUT <- var(data)*sqrt(2/(length(data)-1))
  return(OUT)
}
```

Captive inter-annual variance - $var(y_i.)$

```
captive_md_sd <- srt.fun(captive_data)

(c_med <- var(captive_md_sd$MEDIAN))
```

```
## [1] 15.83288
```

Captive standard error of variance

```
se_var(captive_md_sd$MEDIAN)
```

```
## [1] 4.668862
```

Wild inter-annual variance - $var(y_i.)$

```
wild_md_sd <- srt.fun(wild_data)

(w_med <- var(wild_md_sd$MEDIAN))
```

```
## [1] 13.49003
```

Wild standard error of variance

```
se_var(wild_md_sd$MEDIAN)
```

```
## [1] 3.741461
```

Inter-annual variation in CID from literature

Inter-annual variation in CID in the Adélie penguin population are of a similar magnitude to those seen in other bird species.

```
setwd('Data')

lit_data <- read.csv('Lit_data.csv', header = TRUE)
```

Both et al. 2009 (Figure 2) - Blue tit, Great tit, Pied flycatcher, Coal tit, Sparrowhawk

```
#detrend because I am interested in the interannual variance - trends would conflate this estimate
fit_BT_B_2009 <- lm(lit_data$BT_B_2009 ~ lit_data$YEAR)
res_BT_B_2009 <- residuals(fit_BT_B_2009)
v_BT_B_2009 <- var(res_BT_B_2009)

fit_GT_B_2009 <- lm(lit_data$GT_B_2009 ~ lit_data$YEAR)
res_GT_B_2009 <- residuals(fit_GT_B_2009)
v_GT_B_2009 <- var(res_GT_B_2009)

fit_PF_B_2009 <- lm(lit_data$PF_B_2009 ~ lit_data$YEAR)
res_PF_B_2009 <- residuals(fit_PF_B_2009)
v_PF_B_2009 <- var(res_PF_B_2009)

fit_CT_B_2009 <- lm(lit_data$CT_B_2009 ~ lit_data$YEAR)
res_CT_B_2009 <- residuals(fit_CT_B_2009)
```

```
v_CT_B_2009 <- var(res_CT_B_2009)

#no need to detrend - no significant change
v_SH_B_2009 <- var(lit_data$SH_B_2009, na.rm = TRUE)
```

Valtonen et al. 2017 (Figure 2) - Common redstart, Great tit, Pied flycatcher

```
#no need to detrend - no significant change
v_CR_V_2017 <- var(lit_data$CR_V_2017, na.rm = TRUE)

v_GT_V_2017 <- var(lit_data$GT_V_2017, na.rm = TRUE)

v_PF_V_2017 <- var(lit_data$PF_V_2017, na.rm = TRUE)
```

Table A6-1: Inter-annual variance estimates for breeding phenology from published literature

Species	Publication	Metric	Inter-annual Variance
Blue tit	Both et al. 2009	Hatch date	11.94
Great tit	Both et al. 2009	Hatch date	12.82
Pied flycatcher	Both et al. 2009	Hatch date	4.22
Coal tit	Both et al. 2009	Hatch date	25.11
Sparrowhawk	Both et al. 2009	Hatch date	9.74
Common redstart	Valtonen et al. 2017	Clutch initiation date	10.65
Great tit	Valtonen et al. 2017	Clutch initiation date	15.62
Pied flycatcher	Valtonen et al. 2017	Clutch initiation date	10.26
Adélie penguin (captive)	This publication	Clutch initiation date	15.83
Adélie penguin (wild)	This publication	Clutch initiation date	13.49

Adélie penguins exhibit degrees of inter-annual variance in phenology comparable to other species.

Plot median CID over time - Captive

```
plt_t <- data.frame(YEAR = 1992:2015, MD_CID = captive_md_sd$MEDIAN)
plt <- melt(plt_t, id = 'YEAR')


#CAPTIVE PLOT
ggplot(plt, aes(YEAR, value)) +
  geom_line(size = 1.2, col = 'red') +
  theme_bw() +
  ggtitle('Captive penguin breeding phenology') +
  xlab('Year') +
  ylab('CID (days from Sep 30)') +
  coord_cartesian(xlim = c(1990, 2015)) +
  coord_cartesian(ylim = c(25, 45)) +
  scale_x_continuous(breaks = seq(1990, 2015, by = 5)) +
  scale_y_continuous(breaks = c(25, 30, 35, 40, 45)) +
  theme(
    axis.text = element_text(size = 12), #axis label size
    axis.title = element_text(size = 14),
```

```
    panel.grid.major = element_line(color = 'gray40'), #lower # is darker
    panel.grid.minor = element_line(color = 'gray95'),
    panel.background = element_blank(),
    panel.border = element_rect(fill = NA, color= 'black', size = 1.5),
    axis.ticks.length= unit(0.15, 'cm'), #length of axis tick
    axis.ticks = element_line(size = 1)
  )
```

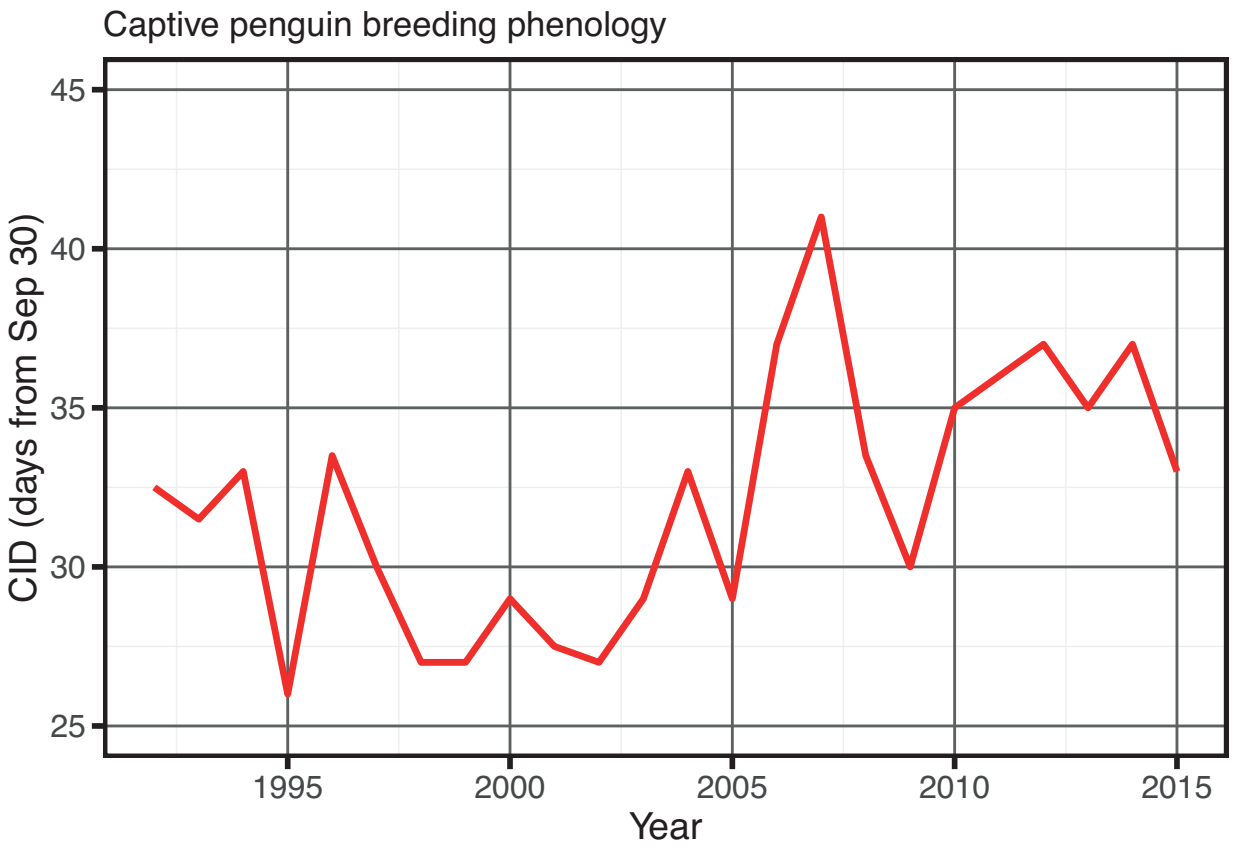

Figure A6-1: Mean colony breeding phenology (CID) in each year for the captive population.

Plot median CID over time - Wild

```
plt_t2 <- data.frame(YEAR = 1986:2012, MD_CID = wild_md_sd$MEDIAN)
plt2 <- melt(plt_t2, id = 'YEAR')


#WILD PLOT
ggplot(plt2, aes(YEAR, value)) +
  geom_line(size = 1.2, col = 'blue') +
  theme_bw() +
  ggtitle('Wild penguin breeding phenology') +
  xlab('Year') +
  ylab('CID (days from Sep 30)') +
  coord_cartesian(xlim = c(1984, 2012)) +
  coord_cartesian(ylim = c(25, 45)) +
```

```
  scale_x_continuous(breaks = seq(1985, 2012, by = 5)) +
  scale_y_continuous(breaks = c(25, 30, 35, 40, 45)) +
  theme(
    axis.text = element_text(size = 12), #axis label size
    axis.title = element_text(size = 14),
    panel.grid.major = element_line(color = 'gray40'), #lower # is darker
    panel.grid.minor = element_line(color = 'gray95'),
    panel.background = element_blank(),
    panel.border = element_rect(fill = NA, color= 'black', size = 1.5),
    axis.ticks.length= unit(0.15, 'cm'), #length of axis tick
    axis.ticks = element_line(size = 1)
  )
```

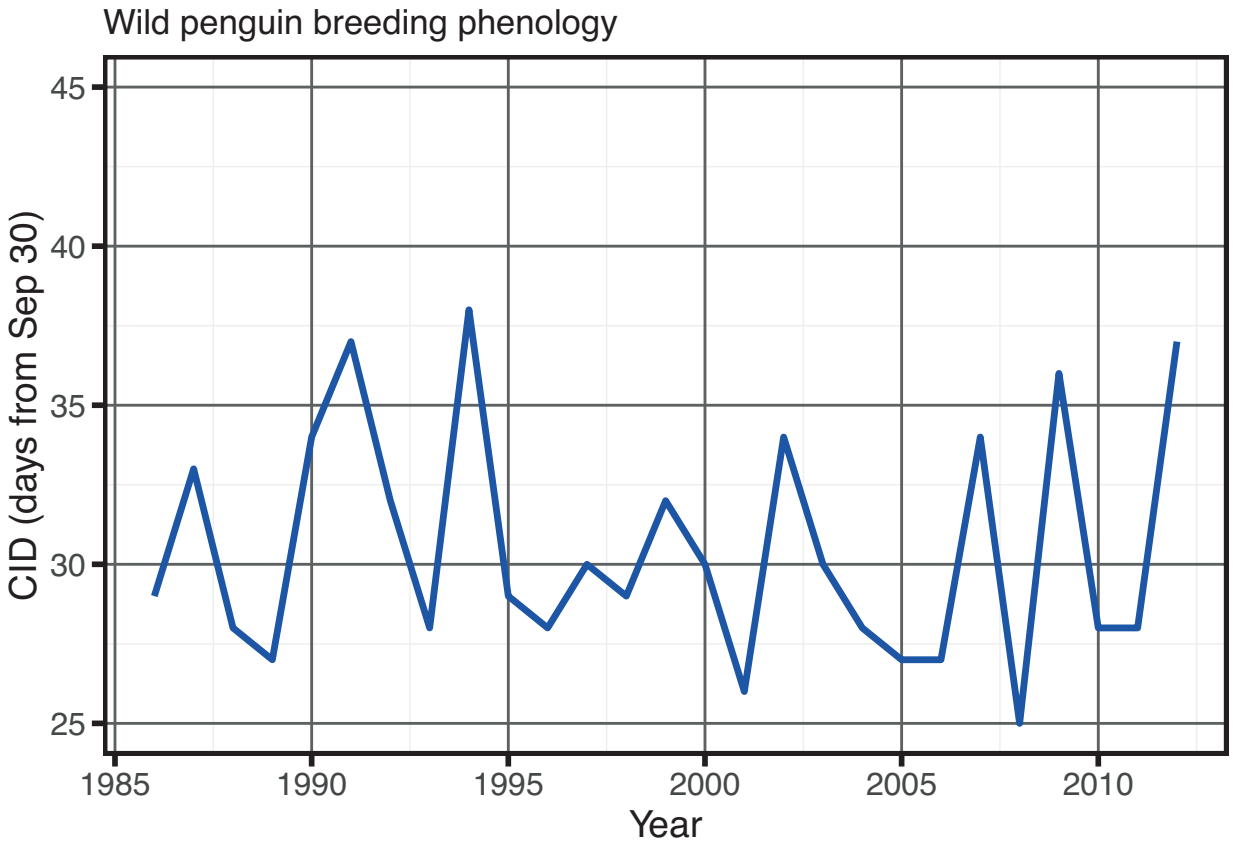

Figure A6-2: Mean colony breeding phenology (CID) in each year for the wild population.

Determine which female is the first to lay in each year (including ties for first)

```
FEM <- c()
for (i in 1992:2015)
{
  #i <- 1997
  temp <- filter(captive_data, YEAR == i)
  pos <- which(temp$J_CID == min(temp$J_CID))
  t_data <- temp[pos,]
```

```
  FEM <- rbind(FEM, t_data)
}
```

16 different 'leaders' in 24 years

```
length(unique(FEM$FEMALE_ID))
```

```
## [1] 16
```

Intra-annual variation in CID

```
#t-test to determine if intra-annual variation differs between captive and wild popualtions
t.test(captive_md_sd$VAR, wild_md_sd$VAR)
```

```
##
##  Welch Two Sample t-test
##
## data:  captive_md_sd$VAR and wild_md_sd$VAR
## t = 4.9893, df = 28.488, p-value = 2.733e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  19.51181 46.65717
## sample estimates:
## mean of x mean of y
##  50.75654  17.67205
```

Number of breeders

```
len_fun <- function(IN)
{
  yrs <- range(IN$YEAR)

  LEN <- c()
  for(i in yrs[1]:yrs[2])
  {
    #i <- 1992
    temp <- filter(IN, YEAR == i)
    tl <- dim(temp)[1]
    tb <- c(i, tl)
    LEN <- rbind(LEN, tb)
  }

  return(LEN)
}

#range
range(len_fun(captive_data)[,2])
```

```
## [1] 12 37
```

Hierarchical model - captive population

JAGS model

$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma * AGE_{ij} + \epsilon_{ij} \tag{A6-1}$$

$$\alpha_i \sim N(0, \sigma^2_{year})$$

$$\beta_j \sim N(0, \sigma^2_{individual})$$

$$\epsilon_{ij} \sim N(0, \sigma^2_{model})$$

```
setwd('Data')

AGE_mat <- read.csv('AGE_mat.csv', header= TRUE)
CID_mat <- read.csv('CID_mat.csv', header= TRUE)

DATA <- list(
  y = CID_mat,
  yr = 1:NCOL(CID_mat),
  ind = 1:NROW(CID_mat),
  age = AGE_mat,
  N = NCOL(CID_mat), #columns are year in matrix
  M = NROW(CID_mat)) #rows are individuals
```

```
#----------------#
#model

#alpha = YEAR - random
#beta = INDIVIDUAL - random
#gamma = AGE - fixed

setwd('JAGS')

{
sink("captive.jags")

cat("
    model {

    for(t in 1:N)
    {
    for(i in 1:M)
    {
    y[i,t] ~ dnorm(mu.g[i,t], tau)
    mu.g[i,t] <- mu + alpha[yr[t]] + beta[ind[i]] + gamma*age[i,t]
    }
    }

    #priors
```

```
    #year
    for(t in 1:N)
    {
    alpha[t] ~ dnorm(0, tau.year)
    }

    #individual
    for(i in 1:M)
    {
    beta[i] ~ dnorm(0, tau.ind)
    }

    #mu, gamma, and tau
    mu ~ dnorm(0, 0.001)
    gamma ~ dnorm(0, 0.001)
    tau ~ dgamma(0.01, 0.01)
    var.model <- 1/tau #convert from precision to variance

    #hyperparameters
    tau.year ~ dgamma(0.01, 0.01)
    var.year <- 1/tau.year
    tau.ind ~ dgamma(0.01, 0.01)
    var.ind <- 1/tau.ind

    }",fill = TRUE)

sink()
}
```

Run model

```
#----------------------#
#Starting values


Inits <- function() {list(alpha = rep(rnorm(1),
                                      ncol(CID_mat)),
                          beta = rep(rnorm(1),
                                     nrow(CID_mat)),
                          tau = rgamma(1, 1),
                          mu = rnorm(1),
                          gamma = rnorm(1),
                          tau.year = rgamma(1, 1),
                          tau.ind = rgamma(1, 1))}


#----------------------#
#Parameters to track

Pars <- c('alpha', 'beta', 'gamma', 'var.year', 'var.ind', 'var.model')



# Inputs for MCMC ------------------------------------------------------
```

```
n_adapt <- 5000  # number for initial adapt
n_burn <- 40000 # number burnin
n_draw <- 10000  # number of final draws to make
n_thin <- 2    # thinning rate
n_chain <- 3  # number of chains

Rhat_max <- 1.1 # max allowable Rhat (close to 1 = convergence)
n_max <- 1e7 # max allowable iterations


#----------------------#
#Run model

jm = jags.model(data = DATA,
                file = "JAGS/captive.jags",
                inits = Inits,
                n.chains = 3,
                n.adapt = n_adapt)

update(jm, n.iter = n_burn)

out = coda.samples(jm,
                   n.iter = n_draw,
                   variable.names = Pars,
                   thin = n_thin)


#extra draws to ensure convergence
n_total <- n_burn + n_draw
n_extra <- 0
while(max(MCMCsummary(out)[,6]) > Rhat_max &
      n_total < n_max)
{

  out <- update(out,
                n.iter = n_draw,
                n.chains = n_chain,
                n.thin = n_thin)

  n_extra <- n_extra + n_draw
  n_total <- n_total + n_draw
}

n_final <- n_draw/n_thin
```

Inferences were derived from 5000 samples drawn following an adaptation period of 5000 draws, and a burn-in period of 4×10^4 draws using 3 chains and a thinning rate of 2.

Year effect (alpha)

```
MCMCplot(out,
         params = 'alpha',
         rank = FALSE,
```

```
         labels = NULL,
         horiz = FALSE,
         ref_ovl = FALSE,
         ylim = c(-15, 15))
```

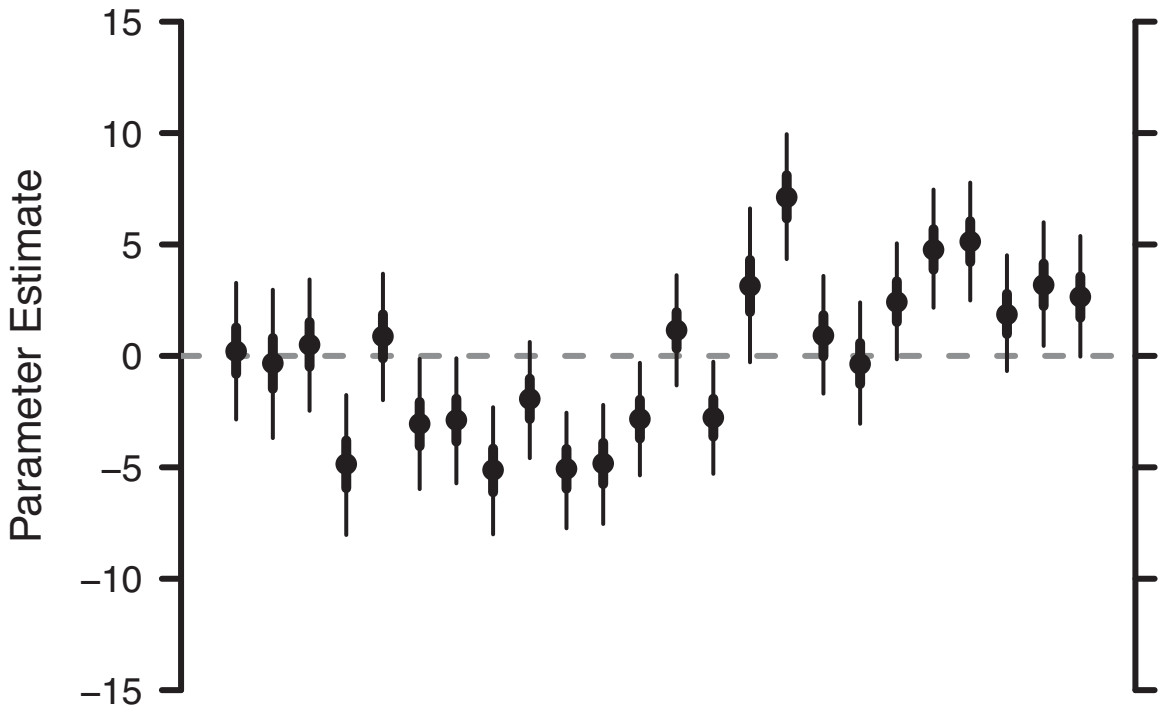

Figure A6-3: Posterior estimates for the captive population year effect – α. Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals.

```
MCMCsummary(out,
            params = 'alpha')
```

```
##                mean       sd        2.5%        50%      97.5% Rhat
## alpha[1]   0.2229872 1.555508 -2.85435508  0.2080336  3.2769796    1
## alpha[2]  -0.3465951 1.691202 -3.68849174 -0.3289091  2.9687629    1
## alpha[3]   0.5124202 1.505971 -2.46319895  0.5030116  3.4310656    1
## alpha[4]  -4.8690395 1.593993 -8.03708036 -4.8546227 -1.7560671    1
## alpha[5]   0.8712890 1.460211 -1.98846655  0.8769365  3.6895502    1
## alpha[6]  -3.0545023 1.483188 -5.97651778 -3.0485988 -0.1313759    1
## alpha[7]  -2.8894778 1.435873 -5.72009610 -2.8789334 -0.1059880    1
## alpha[8]  -5.1392574 1.453386 -8.00794437 -5.1178607 -2.3025231    1
## alpha[9]  -1.9345799 1.330041 -4.58907211 -1.9306750  0.6262959    1
## alpha[10] -5.0770343 1.320141 -7.73712156 -5.0656895 -2.5493876    1
## alpha[11] -4.8286578 1.355513 -7.54292669 -4.8272646 -2.1982776    1
## alpha[12] -2.8415181 1.297315 -5.36158886 -2.8298798 -0.3105335    1
## alpha[13]  1.1476250 1.247279 -1.32466694  1.1538558  3.6228242    1
## alpha[14] -2.7722022 1.285679 -5.29183913 -2.7693316 -0.2623508    1
## alpha[15]  3.1464795 1.753427 -0.28781185  3.1413817  6.6193415    1
## alpha[16]  7.1311937 1.435904  4.34379544  7.1223370  9.9515428    1
## alpha[17]  0.9158833 1.349519 -1.69288975  0.9154327  3.5909334    1
## alpha[18] -0.3411178 1.375022 -3.04581522 -0.3641231  2.4023101    1
```

```
## alpha[19]  2.4310636 1.343781 -0.15352534  2.4201725  5.0528796    1
## alpha[20]  4.7798712 1.347878  2.16187835  4.7662082  7.4642792    1
## alpha[21]  5.1341071 1.357752  2.48671537  5.1322236  7.7786430    1
## alpha[22]  1.8863430 1.334595 -0.67955304  1.8580493  4.5173090    1
## alpha[23]  3.1945544 1.409347  0.44822061  3.1859226  5.9969884    1
## alpha[24]  2.6561089 1.377829 -0.03339546  2.6507148  5.3805759    1
```

Individual effect (beta)

```
MCMCplot(out,
         params = 'beta',
         rank = TRUE,
         labels = NULL,
         horiz = FALSE,
         thick_sz = 2,
         thin_sz = 1,
         med_sz = .6,
         ref_ovl = FALSE,
         ylim = c(-15, 15))
```

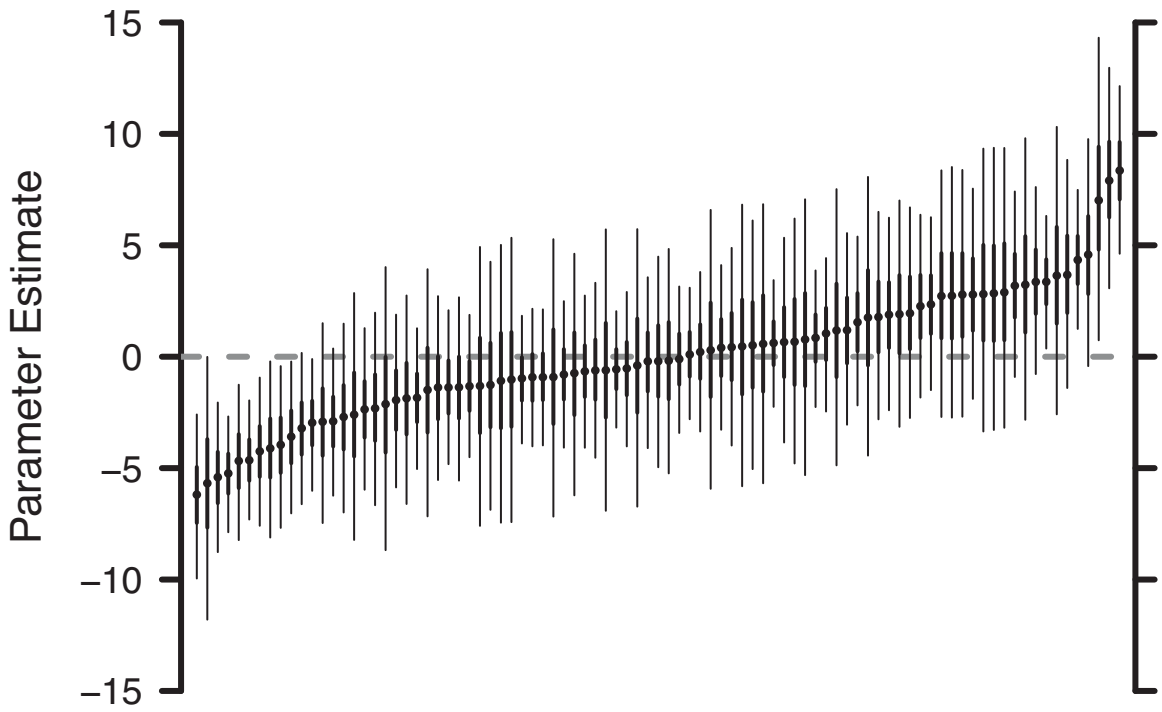

Figure A6-4: Posterior estimates for the captive population individual effect – β. Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals.

```
MCMCsummary(out,
            params = 'beta')
```

```
##                mean       sd        2.5%         50%       97.5% Rhat
## beta[1]   2.2583955 2.085555  -1.8320575  2.26708081  6.37066686    1
## beta[2]  -2.7250290 2.167336  -6.9895051 -2.70369617  1.48302140    1
## beta[3]  -0.4002533 3.169689  -6.7276908 -0.39220644  5.72389925    1
## beta[4]  -2.9882923 1.494097  -6.0174670 -2.96252255 -0.10502828    1
```

```
## beta[5]   1.7670165 3.178883  -4.4374756  1.75629544  8.06755485    1
## beta[6]   4.5886398 2.610636  -0.4212271  4.58015104  9.76640621    1
## beta[7]   8.3569945 1.913623   4.6218751  8.35517157 12.14412474    1
## beta[8]  -2.1797849 3.233453  -8.6801285 -2.12617495  4.02145152    1
## beta[9]  -0.2234771 1.959542  -4.1027364 -0.20713775  3.56414531    1
## beta[10] -6.2151796 1.869553  -9.9534300 -6.19045006 -2.59230455    1
## beta[11] -0.9276130 1.569853  -4.0222665 -0.91880643  2.15113014    1
## beta[12] -1.5178082 2.828045  -7.1637194 -1.49229992  3.92539145    1
## beta[13]  0.8329233 1.564820  -2.2586869  0.84604381  3.87093325    1
## beta[14] -0.5349528 1.769653  -4.0246160 -0.52703830  2.90510796    1
## beta[15] -0.9089226 1.550279  -3.9790444 -0.91797111  2.12376141    1
## beta[16]  1.0219663 1.760101  -2.4595199  1.04357633  4.42546765    1
## beta[17] -1.8504041 1.611346  -5.0410944 -1.84159559  1.27527043    1
## beta[18] -2.9398046 2.263537  -7.4627973 -2.91944192  1.50427920    1
## beta[19] -0.6530116 1.718633  -4.0844029 -0.65864746  2.74475676    1
## beta[20]  2.7901801 2.418873  -1.8910276  2.78840014  7.54207369    1
## beta[21] -3.6004604 1.743850  -7.0300972 -3.58825436 -0.21680250    1
## beta[22]  7.1707865 3.448346   0.7388647  7.01704747 14.31357925    1
## beta[23]  3.6993812 2.587622  -1.4043766  3.67417113  8.83417038    1
## beta[24]  1.2049764 3.159633  -4.8748147  1.17726031  7.52014473    1
## beta[25] -4.6397581 1.373324  -7.3065967 -4.64566594 -1.96147331    1
## beta[26] -0.9937676 1.456335  -3.8935117 -0.97328459  1.83566001    1
## beta[27]  2.8894911 3.220269  -3.2865640  2.84429590  9.37437667    1
## beta[28] -1.1088856 3.192663  -7.4472965 -1.08354611  5.01766424    1
## beta[29] -0.7530655 2.778601  -6.2214530 -0.73905692  4.62035628    1
## beta[30] -1.0175834 3.218741  -7.4226423 -1.02488259  5.33690144    1
## beta[31] -0.5678616 1.325348  -3.1820603 -0.56573325  2.04166503    1
## beta[32]  0.6644054 2.347114  -3.8491819  0.65559275  5.33738603    1
## beta[33]  0.6797036 2.828425  -4.7897282  0.67065148  6.19996873    1
## beta[34] -3.2222650 1.735916  -6.6179652 -3.21175152  0.16826253    1
## beta[35] -1.9484799 1.984626  -5.8652442 -1.94701112  1.88717960    1
## beta[36] -3.9731918 1.840116  -7.6812654 -3.95369885 -0.42302813    1
## beta[37] -4.2587900 1.694274  -7.5866057 -4.24606404 -0.93717834    1
## beta[38] -5.4156117 1.705867  -8.7692471 -5.40688784 -2.05664790    1
## beta[39] -0.8963493 3.161583  -7.1742799 -0.91370625  5.27296702    1
## beta[40] -5.7239086 2.973287 -11.7996941 -5.67981159 -0.01106878    1
## beta[41] -0.1083765 1.670588  -3.4213851 -0.10486736  3.14945588    1
## beta[42] -1.3195190 1.622114  -4.5092673 -1.32343112  1.87745556    1
## beta[43] -4.1153871 2.005664  -8.1148136 -4.11291908 -0.21124077    1
## beta[44]  1.7981222 2.365051  -2.8098948  1.78037881  6.50042457    1
## beta[45] -5.2489900 1.331365  -7.8758179 -5.23667657 -2.68243716    1
## beta[46]  0.3081331 3.171325  -5.9272581  0.29094848  6.58721230    1
## beta[47]  0.1086346 1.486760  -2.8055002  0.09970197  3.09500251    1
## beta[48]  2.8658005 3.223614  -3.3549194  2.81047074  9.33765496    1
## beta[49]  3.6837934 3.282063  -2.5848899  3.64099588 10.31418989    1
## beta[50]  0.2176423 1.817374  -3.3563500  0.21071346  3.80075972    1
## beta[51]  4.3475995 1.590649   1.2485148  4.34164224  7.48568242    1
## beta[52]  0.6006352 1.452991  -2.2488752  0.61492872  3.43561257    1
## beta[53] -2.3046906 2.210849  -6.6636522 -2.31716475  1.97446499    1
## beta[54] -1.3127403 3.196501  -7.5898814 -1.31179965  4.92477753    1
## beta[55]  2.7462270 2.824200  -2.7039817  2.72307299  8.36185886    1
## beta[56] -1.3854522 2.088616  -5.5589572 -1.37320319  2.66473427    1
## beta[57] -2.9198623 1.674453  -6.2360023 -2.89836081  0.36711117    1
## beta[58]  2.7939754 2.818192  -2.6913606  2.78674060  8.37946953    1
```

```
## beta[59]  3.3613055 1.507195   0.3667331  3.36204010  6.31828657    1
## beta[60]  2.3553260 1.968187  -1.4985140  2.34851205  6.25906020    1
## beta[61]  2.9365671 3.231799  -3.1883128  2.88927934  9.36718803    1
## beta[62] -1.3924404 2.101839  -5.5327186 -1.37938835  2.71722004    1
## beta[63] -1.2766366 2.835204  -6.8712151 -1.26586430  4.26051204    1
## beta[64]  0.4104081 1.902254  -3.2774075  0.40336347  4.11134569    1
## beta[65]  1.9051570 2.577789  -3.1455288  1.90672151  7.00931706    1
## beta[66] -2.6068824 2.803368  -8.2224917 -2.60161062  2.85676773    1
## beta[67] -1.3690210 1.772221  -4.8221474 -1.37871601  2.08467218    1
## beta[68]  0.4534130 3.182078  -5.8129459  0.45947213  6.82386282    1
## beta[69]  3.2897036 3.221241  -2.8296522  3.22983646  9.80430016    1
## beta[70]  7.9587820 2.529881   3.0708398  7.90369608 12.96596571    1
## beta[71]  0.4418459 2.252550  -3.9848740  0.42775630  4.88321342    1
## beta[72]  3.3617462 2.138596  -0.7784425  3.35866415  7.61434253    1
## beta[73] -0.1750495 2.569887  -5.2349210 -0.17801950  4.83543343    1
## beta[74] -1.8971834 2.384975  -6.6089495 -1.86799098  2.74672844    1
## beta[75] -0.6098815 3.184388  -6.9140680 -0.60781999  5.71406185    1
## beta[76]  0.5199924 2.852038  -5.0431378  0.51192086  6.11309229    1
## beta[77] -0.2044660 2.388232  -4.9592098 -0.20277908  4.49044764    1
## beta[78]  1.9579394 2.411324  -2.7486993  1.94983648  6.70045011    1
## beta[79]  2.7623710 2.873817  -2.7329657  2.73392438  8.51290013    1
## beta[80] -4.6926031 1.784707  -8.2295393 -4.67686473 -1.25785892    1
## beta[81] -0.8019634 1.675763  -4.0808591 -0.80054228  2.49124049    1
## beta[82]  1.8813161 2.212718  -2.4015669  1.88553191  6.23902979    1
## beta[83]  1.1868766 2.184994  -3.0512294  1.19150011  5.54772995    1
## beta[84]  0.7898171 3.134276  -5.3139449  0.77598315  7.06323745    1
## beta[85]  3.2007227 2.123495  -0.9087302  3.18544193  7.41406897    1
## beta[86] -2.3551584 1.860330  -5.9684592 -2.36385855  1.28158725    1
## beta[87]  0.5805557 3.192332  -5.6792298  0.57516152  6.84393330    1
## beta[88]  1.5635749 1.926779  -2.1835840  1.54727836  5.39001761    1
## beta[89] -0.5932761 2.001270  -4.5338659 -0.61544462  3.31637093    1
```

Age effect (gamma)

```
MCMCplot(out,
         params = 'gamma',
         rank = TRUE,
         labels = NULL,
         horiz = FALSE,
         med_sz = 2,
         ref_ovl = FALSE,
         ylim = c(-15, 15))
```

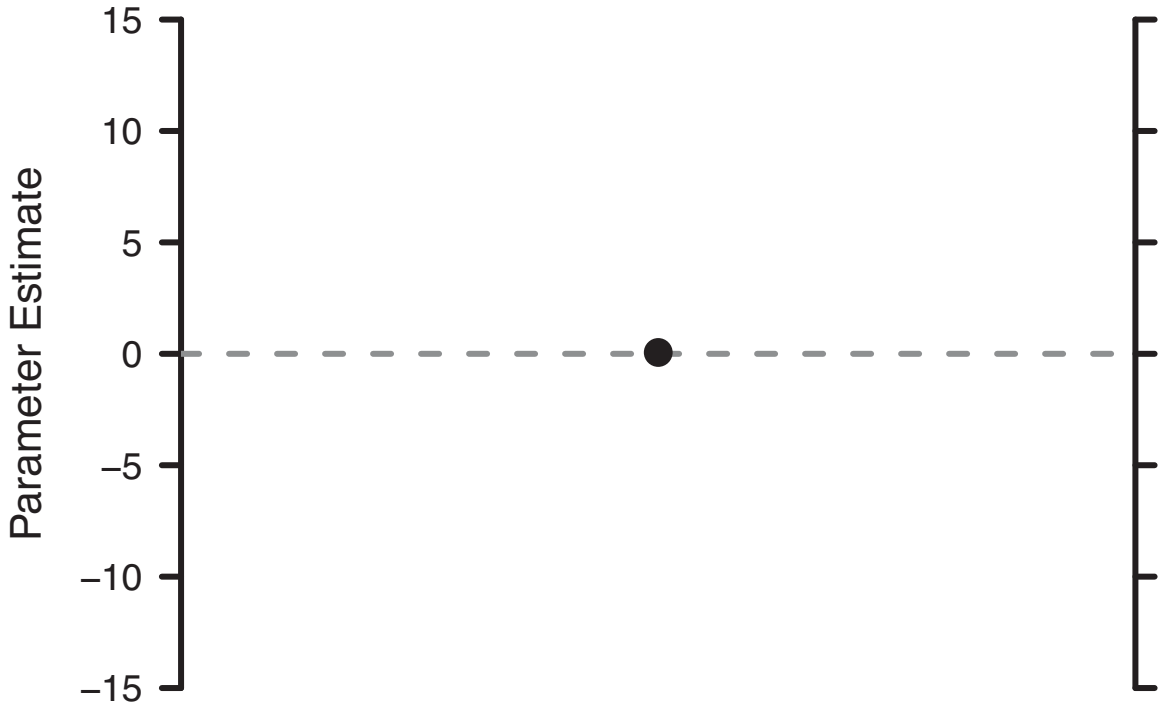

Figure A6-5: Posterior estimates for the captive population age effect – γ parameters. Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals. Error bars are obscured by the point itself.

```
MCMCsummary(out,
            params = 'gamma')
```

```
##              mean         sd        2.5%        50%     97.5% Rhat
## gamma 0.06144969 0.04444429 -0.02242663 0.06027295 0.1530932    1
```

Variance estimates (inverse precision)

```
MCMCsummary(out,
            params = c('var.year', 'var.ind', 'var.model'))
```

```
##                mean       sd      2.5%      50%    97.5% Rhat
## var.year  14.74483 5.485685  7.129838 13.74286 28.23142    1
## var.ind   14.18147 3.790029  8.009549 13.79093 22.82989    1
## var.model 34.99582 2.260097 30.798545 34.88104 39.69440    1
```

###Skewness test and plots

Captive

```
#scale and aggregate data

sc_agg_fun <- function(IN)
{
  yrs <- range(IN$YEAR)

  OUT <- c()
```

```
  for(i in yrs[1]:yrs[2])
  {
    temp <- filter(IN, YEAR == i)
    s_CID <- scale(temp$J_CID, scale = TRUE)

    t.out <- cbind(temp, s_CID)
    OUT <- rbind(OUT, t.out)
  }
  return(OUT)
}


cap_sk <- sc_agg_fun(captive_data)
```

Skew determined using D'Agostino test

```
skew_captive <- agostino.test(cap_sk$s_CID)$statistic[1]
agostino.test(cap_sk$s_CID)
```

```
##
##  D'Agostino skewness test
##
## data:  cap_sk$s_CID
## skew = 0.54072, z = 5.24950, p-value = 1.525e-07
## alternative hypothesis: data have a skewness
```

Standard error skew

```
#standard error of skewness function
ses <- function(n)
{
  sqrt((6*n*(n-1))/((n-2)*(n+1)*(n+3)))
}

length_captive <- dim(cap_sk)[1]

ses(length_captive)
```

```
## [1] 0.09774553
```

Wild

```
wild_sk <- sc_agg_fun(wild_data)
```

Skew

```
skew_wild <- agostino.test(wild_sk$s_CID)$statistic[1]
agostino.test(wild_sk$s_CID)
```

```
##
##  D'Agostino skewness test
##
## data:  wild_sk$s_CID
## skew = 0.79228, z = 16.09200, p-value < 2.2e-16
## alternative hypothesis: data have a skewness
```

Standard error skew

```
length_wild <- dim(wild_sk)[1]

ses(length_wild)
```

```
## [1] 0.04376183
```

Simulate normal breeding distribution given true CID mean and sd - Plot

Captive

```
m_cap <- mean(cap_sk$s_CID, na.rm = TRUE)
sd_cap <- sd(cap_sk$s_CID, na.rm = TRUE)
cap_rd <- rnorm(100000, mean = m_cap, sd = sd_cap)

hist(cap_sk$s_CID, prob = TRUE,
     main = 'Breeding distribution - Captive',
     xlab = 'CID', ylab= 'Density', col = 'grey90',
     xlim = c(-2.5, 3),
     breaks = 15)
lines(density(cap_rd), col = 'blue', lwd = 5)
lines(density(cap_sk$s_CID, na.rm = TRUE), col = 'red', lwd = 5)
```

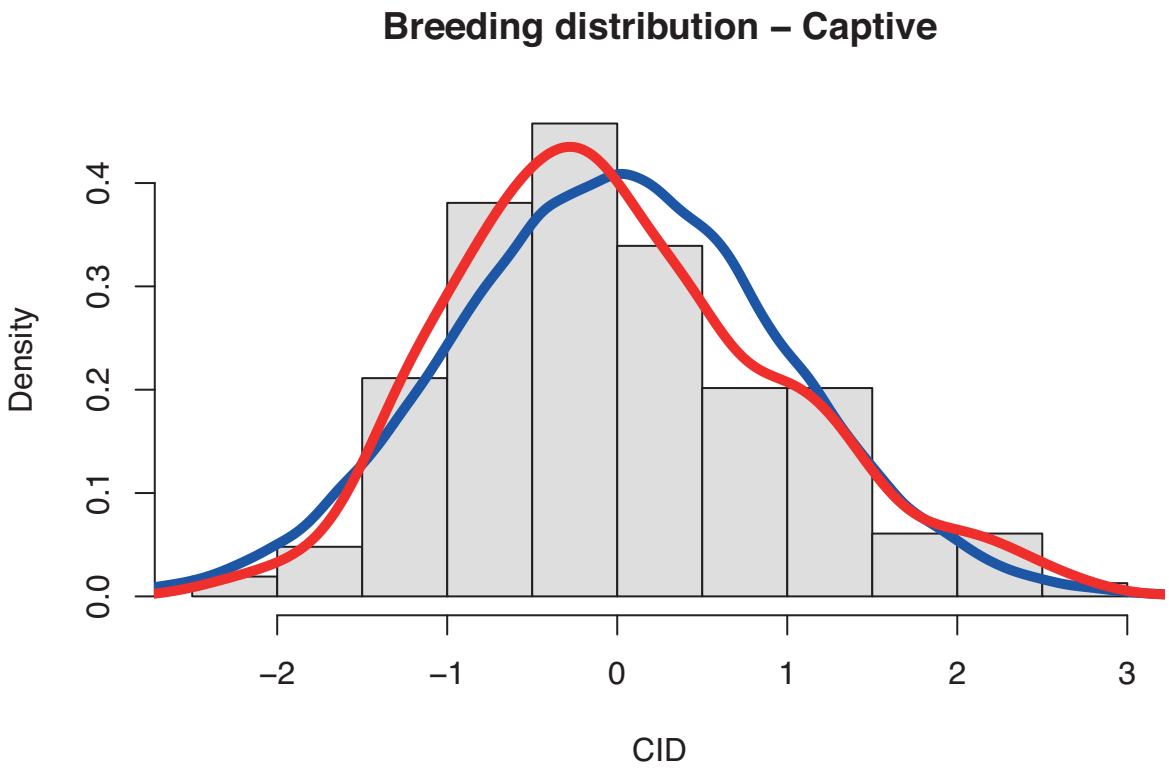

Figure A6-6: Distribution of CIDs for the captive population. Data were standardized and aggregated across years. Colored lines depict the kernel density estimates on the distribution. The red line represents the actual distribution of CID, while the blue line represents the normal distribution generated using the actual mean and variance of the CID distribution.

Wild

```r
m_wild <- mean(wild_sk$s_CID, na.rm=TRUE)
sd_wild <- sd(wild_sk$s_CID, na.rm=TRUE)
wild_rd <- rnorm(100000, mean= m_wild, sd= sd_wild)

hist(wild_sk$s_CID, prob=TRUE,
     main='Breeding distribution - Wild',
     xlab= 'CID', ylab= 'Density', col = 'grey90',
     xlim = c(-2.5, 3),
     breaks = 25)
lines(density(wild_rd), col='blue', lwd=5)
lines(density(wild_sk$s_CID, na.rm=TRUE), col='red', lwd=5)
```

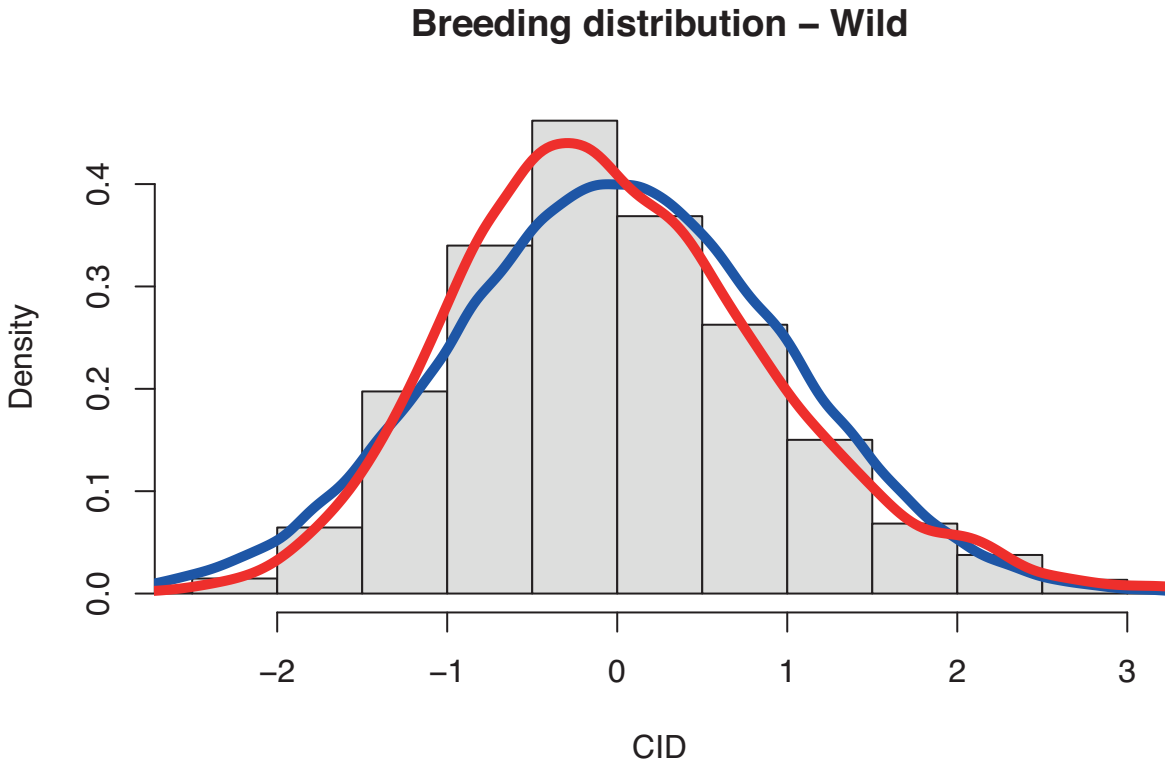

Figure A6-7: Distribution of CIDs for the wild population. Data were standardized and aggregated across years. Colored lines depict the kernel density estimates on the distribution. The red line represents the actual distribution of CID, while the blue line represents the normal distribution generated using the actual mean and variance of the CID distribution.

First CID as a predictor for median CID

Fit a linear model to examine the predictive power of first CID for median CID in a given year.

```r
#remove first values when calculating median for that year

#CAPTIVE
YRS_C <- unique(captive_data$YEAR)

OUT_C <- c()
for (i in 1:length(YRS_C))
{
```

```
  #i <- 1
  temp <- filter(captive_data, YEAR == YRS_C[i])
  min_temp <- min(temp$J_CID)
  min_ind <- which(temp$J_CID == min(temp$J_CID))
  med_temp <- median(temp$J_CID[-min_ind])

  t_OUT_C <- c(YRS_C[i], min_temp, med_temp)
  OUT_C <- rbind(OUT_C, t_OUT_C)
}


fit_C <- summary(lm(OUT_C[,3] ~ OUT_C[,2]))


#WILD
YRS_W <- unique(wild_data$YEAR)

OUT_W <- c()
for (i in 1:length(YRS_W))
{
  #i <- 1
  temp <- filter(wild_data, YEAR == YRS_W[i])
  min_temp <- min(temp$J_CID)
  min_ind <- which(temp$J_CID == min(temp$J_CID))
  med_temp <- median(temp$J_CID[-min_ind])

  t_OUT_W <- c(YRS_W[i], min_temp, med_temp)
  OUT_W <- rbind(OUT_W, t_OUT_W)
}


fit_W <- summary(lm(OUT_W[,3] ~ OUT_W[,2]))
```

Captive model $r^2 = 0.7$

Wild model $r^2 = 0.7$

Explanatory power is high and very similar between the captive and wild populations.

Skew estimates from literature

Data on skew of phenological data from Thomson 1980, Sparks et al. 2005, Wilson 2013 from plant and bird species. Data is composed of various phenological measures, including first flowering for plants and first arrival for birds (which differs from CID used in this study).

```
setwd('Data')

lit_skew <- read.csv('Lit_skew.csv', header = TRUE)

mn_skew_lit  <- mean(lit_skew$Skew)
```

mean skew = 0.43

Histogram of skew from literature

```
hist(lit_skew$Skew,
     main = 'Red = Captive; Blue = Wild',
     xlab = 'Estimated Skew')
abline(v = skew_captive, col = 'red', lwd = 5)
abline(v = skew_wild, col = 'blue', lwd = 5)
```

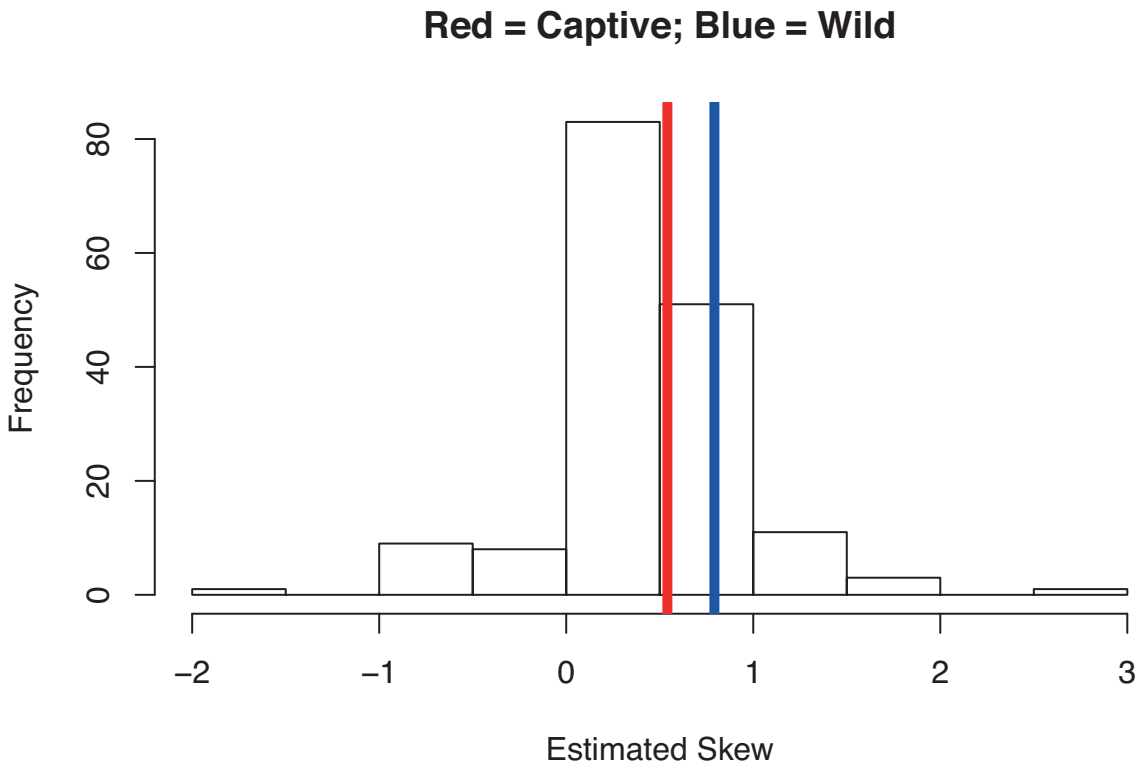

Figure A6-8: Histogram of skew values for breeding phenology taken from the published literature. The red and blue lines represent the skew calculated for the captive and wild populations, respectively.

```
per_captive <- length(which(lit_skew$Skew > skew_captive))/NROW(lit_skew)
per_wild <- length(which(lit_skew$Skew > skew_wild))/NROW(lit_skew)
```

Percent literature values greater than captive population skew = 37%

Percent literature values greater than wild population skew = 17%

Skew for Adélie penguin CID is slightly higher than the skew estimated for most populations.

Simulation of right skew breeding distribution

Simple model to show that a right skew distribution can be produced when breeding is accelerated proportional to the number of pairs that have recently initiated breeding.

```
#simulate a normal distirbution (breeding in the absence of conspecifics)
set.seed(1)
time <- round(rnorm(1000, 50, 10))
time <- sort(time)
#no significant skew
agostino.test(time)
```

```
##
##  D'Agostino skewness test
##
## data:  time
## skew = -0.01578, z = -0.20520, p-value = 0.8374
## alternative hypothesis: data have a skewness
```

```
#simulate 'contagion effect' of breeding
for (i in min(time):max(time))
{
   time[time == i] <- time[time == i] - sum(as.numeric((time > (i - 3)) & (time < i)))/4
}

hist(time, prob = TRUE)
lines(density(time), col = 'blue', lwd = 5)
```

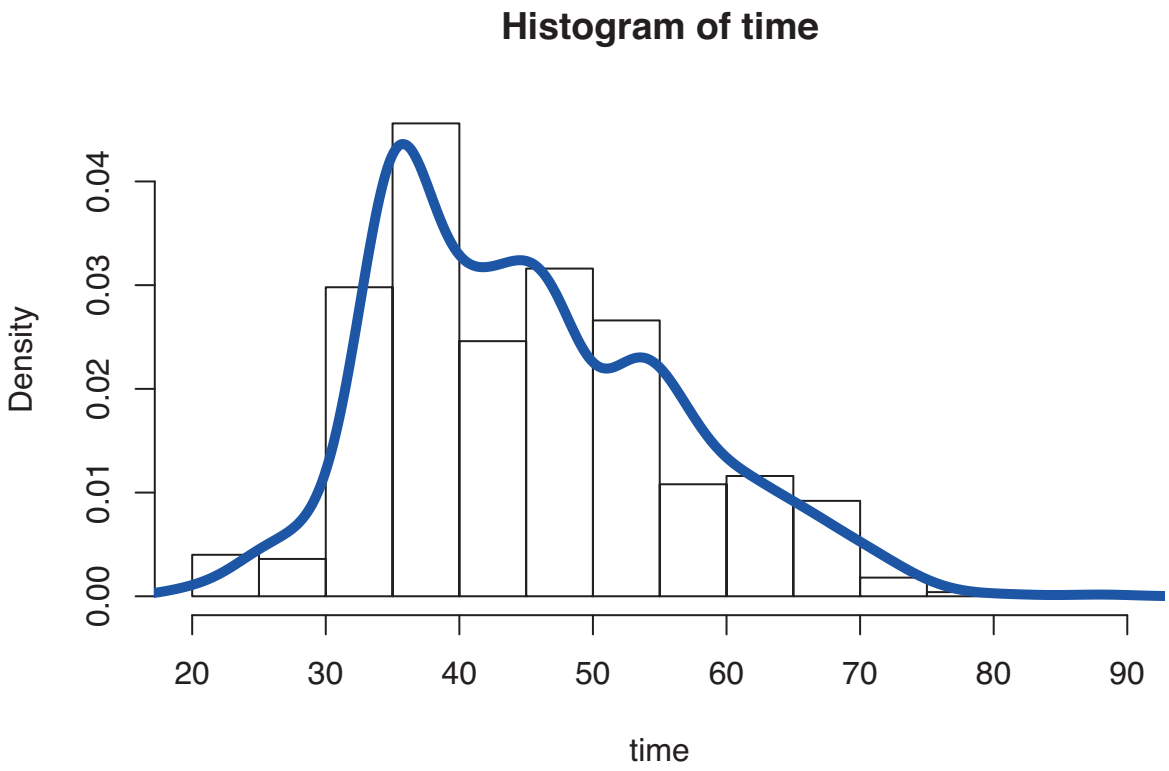

Figure A6-9: Histogram of temporal breeding distribution, as determined from data simulated from a simple 'contagion' model.

```
agostino.test(time)
```

```
##
##  D'Agostino skewness test
##
## data:  time
## skew = 0.55355, z = 6.74630, p-value = 1.516e-11
## alternative hypothesis: data have a skewness
```

Appendix 7

This appendix contains supplemental analyses for work presented in Chapter 3.

The effect of breeding in year t-1 on breeding in year t

I conducted a post-hoc analysis on the captive population data to investigate temporal autocorrelation in the posterior estimates for the year effect (Figure 3-2a). Plotting clutch initiation date (CID) in year t against CID in year t-1, the effect of this temporal inertia on CID is apparent (Figure A7-1).

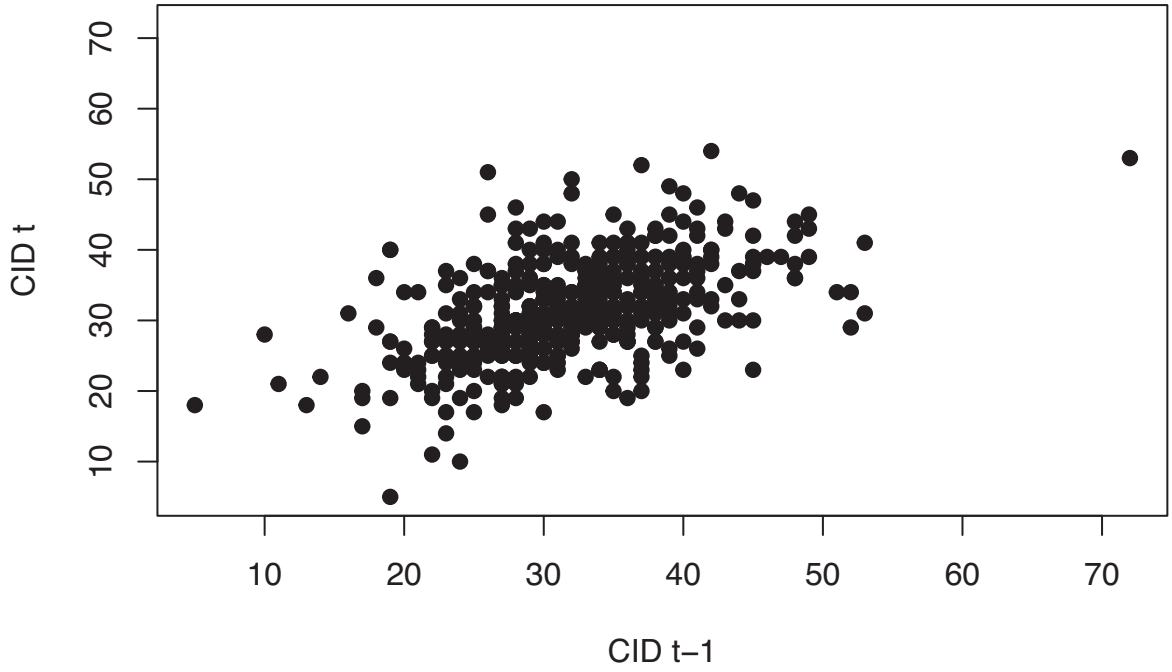

Figure A7-1: CID for all individuals in the captive population in year t plotted against CID in year t-1 ($r^2 = 0.27$).

To further investigate this pattern, I fit a hierarchical Bayesian model, similar to Equation 3-1, but with an added autoregressive term, to account for possible temporal autocorrelation.

$$y_{ij} = \mu + \alpha_i + \pi * y_{i-1j} + \beta_j + \gamma * AGE_{ij} + \epsilon_{ij} \tag{A7-1}$$

$$\alpha_i \sim N(0, \sigma^2_{year})$$

$$\pi \sim N(0, 1000)$$

$$\beta_j \sim N(0, \sigma^2_{individual})$$

$$\gamma \sim N(0, 1000)$$

$$\epsilon_{ij} \sim N(0, \sigma^2_{model})$$

where y_{ij} represents CID, μ represents the intercept, α represents the year effect, β represents the individual effect, π represents the autoregressive term, γ represents the effect of age, AGE represents the age of the female penguin, and ϵ represents the error term.

The model was fit using the R package 'R2jags' (Su and Yajima 2015), to interface with JAGS (Plummer 2003) in the R statistical environment (R Development Core Team 2016). Broad Gamma priors were used for all precision ($\tau = \frac{1}{\sigma^2}$) parameters (shape = 0.01, rate = 0.01). Inferences were derived from 20,000 samples drawn following a 'burn-in' period of 30,000 draws, using a thinning rate of 2 and 3 chains. Model convergence was assessed through a visual analysis of the posterior chains, in addition to the use of the Gelman-Rubin convergence diagnostic (Brooks and Gelman 1998). All models unambiguously converged. Parameter estimates plots were generated using the 'MCMCvis' package (Youngflesh 2017) in the R statistical environment.

RESULTS

Even when accounting for temporal autocorrelation, the year effect appears strong. This suggests that even when controlling for the effect of previous year, the stochastic component to CID across years is still apparent. Inclusion of the autoregressive term does not substantially impact the posterior results of the other parameters. Individual effects are similar between the two models, with estimates closer to 0 when including the autoregressive term. The effect of age is also similar between the two models (Fig A7-2, A7-3).

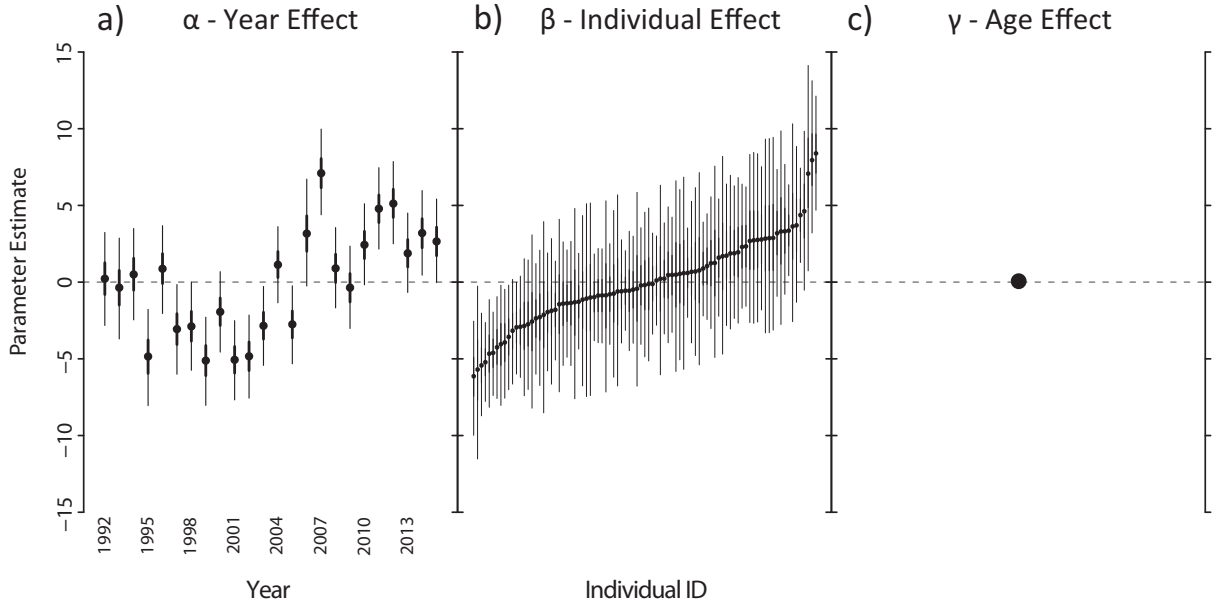

Figure A7-2: Posterior estimates for the captive population parameters: (a) year effect – α; (b) individual effect – β; and (c) age effect – γ parameters for the original model, that does not include the auto-regressive term (see Eq 1). Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals. Error bars for the γ parameter are not visible.

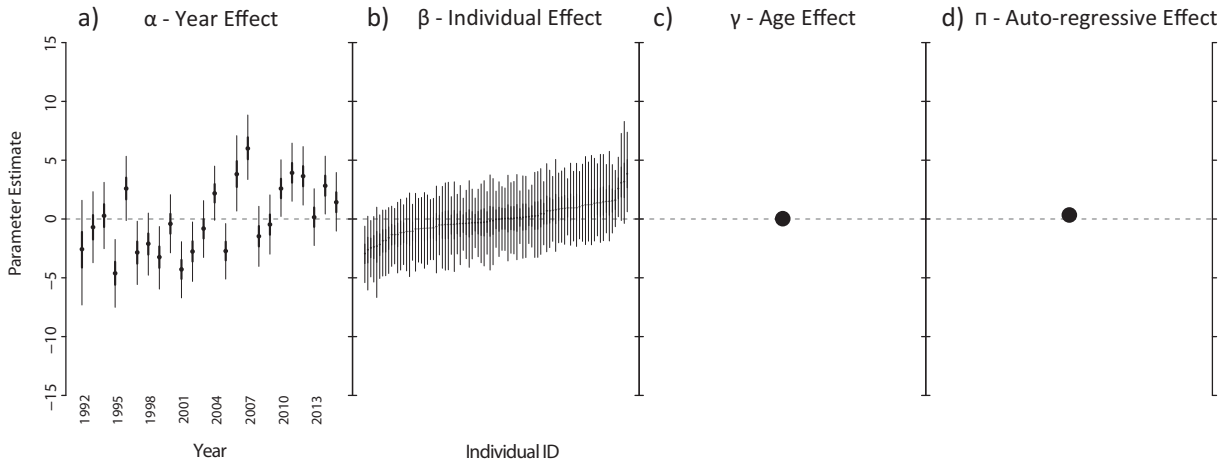

Figure A7-3: Posterior estimates for the captive population parameters: (a) year effect – α; (b) individual effect – β; (c) age effect – γ; and (d) autoregressive effect - π parameters for the model that includes the auto-regressive term (see Eq S1). Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals. Error bars for the γ and π parameters are not visible.

My primary goal was to determine whether autocorrelation in the CID of individual penguins across years could explain the year effect apparent in the original model (Equation 3-1). I show that temporal inertia in CID does exist, though it does not explain the year-to-year stochasticity in CID. The reason for this temporal autocorrelation may be due to physiological factors but is beyond the scope of this study. Model results show a strong year effect in CID is still apparent when accounting for this potentially confounding factor.

The effect of breeding population size on CID

I was also interested to know if the number of breeders in a particular year impacted the breeding phenology of the captive penguins. To further investigate this pattern, I fit a hierarchical Bayesian model, similar to Equation 3-1, but with an added term for the number of breeders in each year.

$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma * AGE_{ij} + \zeta * NB_i + \epsilon_{ij} \tag{A7-2}$$

$$\alpha_i \sim N(0, \sigma^2_{year})$$

$$\beta_j \sim N(0, \sigma^2_{individual})$$

$$\gamma \sim N(0, 1000)$$

$$\zeta \sim N(0, 1000)$$

$$\epsilon_{ij} \sim N(0, \sigma^2_{model})$$

where y_{ij} represents CID, μ represents the intercept, α represents the year effect, β represents the individual effect, γ represents the effect of age, AGE represents the age of the female penguin, ζ represents the effect of the number of breeders, NB represents the number of breeders in each year, and ϵ represents the error term.

The model was fit using the same methodology denoted above.

RESULTS

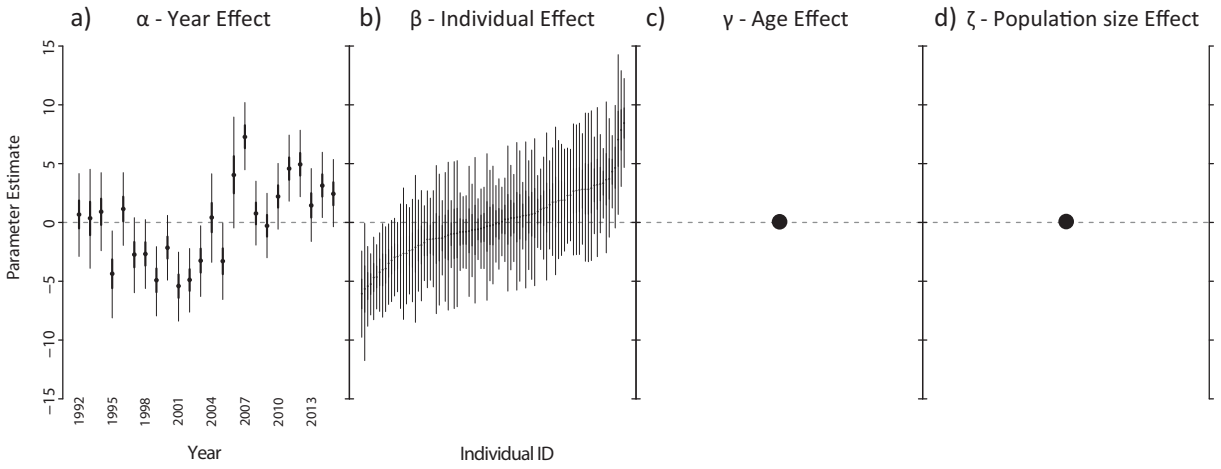

Figure A7-4: Posterior estimates for the captive population parameters: (a) year effect – α; (b) individual effect – β; (c) age effect – γ; and (d) effect of population size - ζ (see Equation A7-2). Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals. Error bars for the γ and ζ parameters are not visible.

It does not appear that population size has a strong effect on captive penguin CID. The parameter estimate for ζ is small and a year-effect is still apparent.

The effect of the timing of nesting material availability on breeding phenology

I initially hypothesized that the timing of nesting material availability may also impact CID. Adélie penguin rely on the availability of stones to build nests before laying eggs during the breeding season. I compared the 'waiting time' of a wild Adélie penguin population at Cape Crozier (77.45°S, 169.20°E) (i.e., the time between the first arrival at the colony and the first clutch initiation date [data presented in Table 3.2 of Ainley et al. (1983)]), to the 'waiting time' of the captive population (i.e., the time between when nesting materials are first made available to the penguins [available for 2007-2015] and the first clutch initiation date).

Mean time between the first arrival at the colony and the first egg laid in a wild population was found to be 13.7 days (Ainley et al. 1983).

Mean time between when nesting materials first became available and the first egg laid in this captive population was found to be 41.3 days.

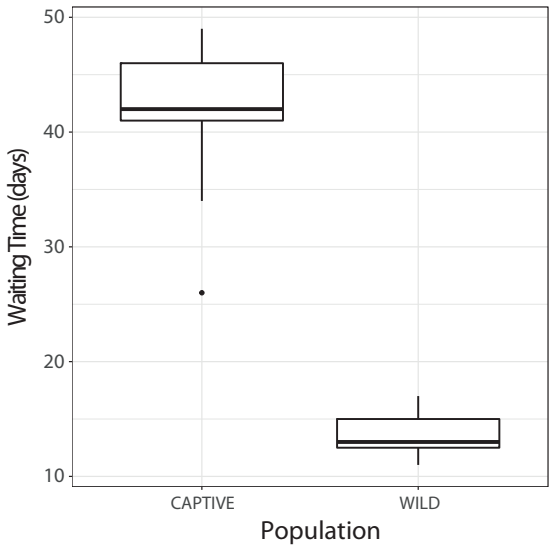

Figure A7-5: Barchart showing distribution of waiting times for both the captive Adélie penguin population (time between first availability of nesting materials and first CID) and the wild Adélie penguin population observed by Ainley et al. (1983) (time between first arrival at the colony and first CID). This analysis was conducted using the mean CID in the captive population as opposed to the min CID with the same results.

The mean waiting time is more than 3 times higher in captivity than the wild. This suggests that availability of nesting materials is not a barrier for breeding for Adélie penguins in captivity.

Potential effects of 2005 exhibit renovation on phenology

CID appears slightly delayed following the captive exhibit renovation in 2005. To investigate whether this impacted the results, I standardized median CID for 1992-2004 and from 2006-2015 (removing 2005 to account for any impact that the renovation may have had in that year), and calculated the variance for the time series.

To further investigate the effect that any potential step-change due to the 2005 renovation may have had on CID, I fit a hierarchical Bayesian model, similar to Equation 3-1, but with an added term for pre- and post-renovation identity.

$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma * AGE_{ij} + \kappa_l * RID_i + \epsilon_{ij} \tag{A7-3}$$

$$\alpha_i \sim N(0, \sigma^2_{year})$$

$$\beta_j \sim N(0, \sigma^2_{individual})$$

$$\gamma \sim N(0, 1000)$$

$$\kappa_l \sim N(0, 1000)$$

$$\epsilon_{ij} \sim N(0, \sigma^2_{model})$$

where y_{ij} represents CID, μ represents the intercept, α represents the year effect, β represents the individual effect, γ represents the effect of age, AGE represents the age of the female penguin, κ represents the effect of pre/post renovation identity, RID represents whether a particular year belongs in the pre or post renovation period, and ϵ represents the error term.

The model was fit using the same methodology denoted above.

RESULTS

σ^2_{RID} = 7.59

$\sigma^2_{AdmiraltyBay}$ = 13.5

When standardizing the pre and post renovation periods, the overall variance of the captive time series decreases, leading to a slightly lower variance than that seen at Admiralty Bay. This is well within the range of variance in CID found at other wild Adélie penguin colonies in Antarctica (Youngflesh et al. 2017), however, confirming that variance in CID in this captive population is as great as that seen in the wild.

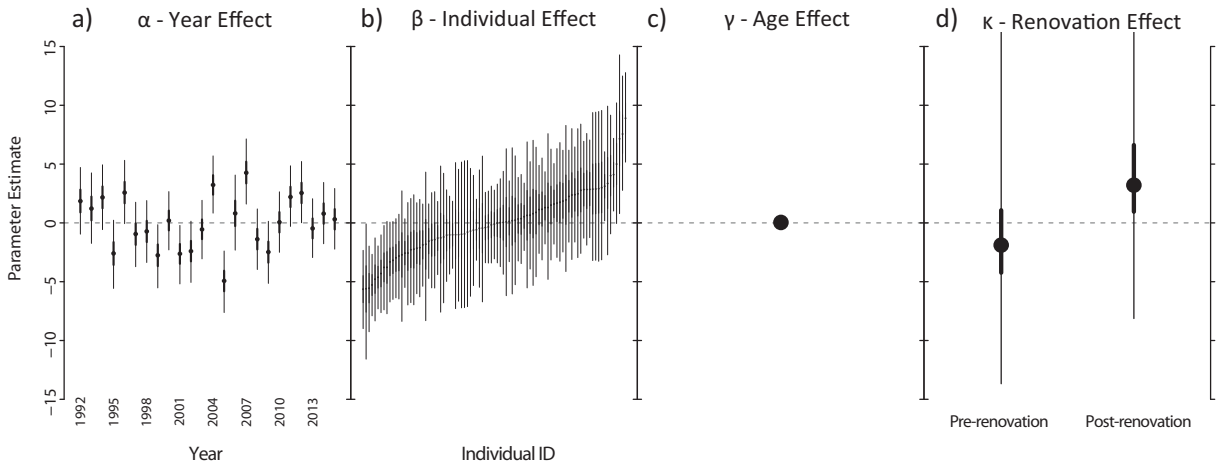

Figure A7-6: Posterior estimates for the captive population parameters: (a) year effect – α; (b) individual effect – β; (c) age effect – γ; and (d) effect of period id (pre/post renovation) - κ (see Equation A7-3). Black circles represent posterior medians. Thicker lines represent 50% credible intervals while thinner lines represent 95% credible intervals. Error bars for the γ parameter are not visible.

None of the factors examined changed the overall conclusions of the study. Year effects are slightly weaker when adding the κ term compared to the original model (Figure A7-2), however are still prominent. Parameter estimates for κ are relatively close to zero, with large credible intervals. Even when accounting for any potential effect that the renovation may have had on CID, interannual variability in CID appears to be the norm.

Appendix 8

This appendix contains supplemental information, and R code to reproduce all analyses presented in Chapter 4.

Initial set up

```
#install packages if they don't exist - then load them
if('pacman' %in% rownames(installed.packages()) == FALSE)
{
  install.packages('pacman')
}

pacman::p_load(dplyr,
               rjags,
               ggplot2,
               gridExtra,
               parallel,
               MCMCvis,
               boot)
```

Long-term trends in Abundance and Breeding Success

Load data

```
setwd('Data')

data <- read.csv('pub_data.csv', header=TRUE)

ADPE <- filter(data, SPECIES == 'ADPE')
EMPE <- filter(data, SPECIES == 'EMPE')
SOFU <- filter(data, SPECIES == 'SOFU')
CAPE <- filter(data, SPECIES == 'CAPE')
SNPE <- filter(data, SPECIES == 'SNPE')
SPSK <- filter(data, SPECIES == 'SPSK')
```

Data availability

Table A8-1: Abundance and Breeding Success data availability

Species	Measure	Years
Adélie penguin	Abundance	1984, 1988-1991, 1993-2016
Adélie penguin	Breeding Success	1993-2016
Emperor penguin	Abundance	1980-2016
Emperor penguin	Breeding Success	1980-2016
Southern fulmar	Abundance	1980-2016
Southern fulmar	Breeding Success	1981-2016
Cape petrel	Abundance	1985, 1989-1992, 1994, 1996, 1998-2016
Cape petrel	Breeding Success	1992, 1994, 1996, 1998-2016
Snow petrel	Abundance	1980-2016
Snow petrel	Breeding Success	1980-2016
South polar skua	Abundance	1981, 1984-2016
South polar skua	Breeding Success	1985-1986, 1989, 1992-2016

Model - used for Abundance ~ Time and Breeding Success ~ Time

$$
\begin{aligned}
y_{ij} &\sim N(\mu_{ij}, \sigma_j^2) \\
\mu_{ij} &= \alpha_i + \beta_j * YEAR_i \\
\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} &\sim N\left(\begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix}, \Sigma\right)
\end{aligned}
\tag{A8-1}
$$

Abundance ~ Time

```
#all log(abundance) data
all_ABUN <- c(log(ADPE$ABUN), log(EMPE$ABUN), log(SOFU$ABUN),
              log(CAPE$ABUN), log(SNPE$ABUN), log(SPSK$ABUN))


#prepare data for JAGS model
ABUN_DATA <- list(
  y = all_ABUN, #response
  N = length(all_ABUN), #length data
  M = 6,
  id = c(rep(1, NROW(ADPE)), rep(2, NROW(EMPE)),
         rep(3, NROW(SOFU)), rep(4, NROW(CAPE)),
         rep(5, NROW(SNPE)), rep(6, NROW(SPSK))),
  YEAR = rep(1:length(ADPE$YEAR), 6))

DATA <- ABUN_DATA
```

```
# JAGS Model
{
sink("ABUN_trend.jags")

cat("
    model {

    #N = number of data points
```

```
    #M = number of species
    #y = response
    #alpha = intercept
    #beta = slope
    #id = each species has unique id

    for (i in 1:N)
    {
    y[i] ~ dnorm(mu[i], tau.y[id[i]])
    mu[i] <- alpha[id[i]] + beta[id[i]]*YEAR[i]
    }

    #priors

    for(j in 1:M)
    {
    tau.y[j] <- pow(sigma.y[j], -2)
    sigma.y[j] ~ dunif(0, 100)

    alpha[j] <- B[j,1]
    beta[j] <- B[j,2]
    B[j, 1:2] ~ dmnorm(B.hat[j,], Tau.B[,])
    B.hat[j,1] <- mu.a
    B.hat[j,2] <- mu.b
    }

    mu.a ~ dnorm(0, 0.01)
    mu.b ~ dnorm(0, 0.01)

    Tau.B[1:2, 1:2] <- inverse(Sigma.B[,])
    Sigma.B[1, 1] <- pow(sigma.a, 2)
    sigma.a ~ dunif(0, 200)
    Sigma.B[2, 2] <- pow(sigma.b, 2)
    sigma.b ~ dunif(0, 10)
    Sigma.B[1, 2] <- rho * sigma.a * sigma.b
    Sigma.B[2, 1] <- Sigma.B[1, 2]
    rho ~ dunif(-1, 1)

    }",fill = TRUE)

sink()
}

# Starting values -------------------------------------------------------

Inits_1 <- list(B = array(rnorm(2 * DATA$M), c(DATA$M, 2)),
                mu.a = 1,
                mu.b = 1,
                sigma.a = 1,
                sigma.b = 1,
                rho = 0.5,
                sigma.y = runif(DATA$M, 0 , 100),
                .RNG.name = "base::Mersenne-Twister",
```

```
                .RNG.seed = 1)

Inits_2 <- list(B = array(rnorm(2 * DATA$M), c(DATA$M, 2)),
                mu.a = 1,
                mu.b = 1,
                sigma.a = 1,
                sigma.b = 1,
                rho = 0.5,
                sigma.y = runif(DATA$M, 0 , 100),
                .RNG.name = "base::Wichmann-Hill",
                .RNG.seed = 2)

Inits_3 <- list(B = array(rnorm(2 * DATA$M), c(DATA$M, 2)),
                mu.a = 1,
                mu.b = 1,
                sigma.a = 1,
                sigma.b = 1,
                rho = 0.5,
                sigma.y = runif(DATA$M, 0 , 100),
                .RNG.name = "base::Marsaglia-Multicarry",
                .RNG.seed = 3)

F_Inits <- list(Inits_1, Inits_2, Inits_3)



# Parameters to track -------------------------------------------------


Pars <- c('alpha',
          'beta',
          'mu.a',
          'mu.b',
          'sigma.y',
          'sigma.a',
          'sigma.b',
          'rho',
          'mu'
)


# Inputs for MCMC -----------------------------------------------------

JAGS_FILE <- 'ABUN_trend.jags'
n_adapt <- 8000  # number for initial adapt
n_burn <- 30000000 # number burnin
n_draw <- 100000000  # number of final draws to make
n_thin <- 500    # thinning rate
n_chain <- 3  # number of chains

# Run model (parallel) ------------------------------------------------------------

#number of chains
```

```
cl <- parallel::makeCluster(n_chain)

pid <- NA
for(i in 1:n_chain)
{
  pidNum <- capture.output(cl[[i]])
  start <- regexpr("pid", pidNum)[[1]]
  end <- nchar(pidNum)
  pid[i] <- substr(pidNum, (start + 4), end)
}

parallel::clusterExport(cl,
                        c('DATA',
                          'n_adapt',
                          'n_burn',
                          'n_draw',
                          'n_thin',
                          'Pars',
                          'pid',
                          'F_Inits',
                          'JAGS_FILE'
                        ))

out.1 <- parallel::clusterEvalQ(cl,
                                {
                                  require(rjags)
                                  processNum <- which(pid==Sys.getpid())
                                  m.inits <- F_Inits[[processNum]]

                                  jm = jags.model(data = DATA,
                                                  file = paste0(JAGS_FILE),
                                                  inits = m.inits,
                                                  n.chains = 1,
                                                  n.adapt = n_adapt)

                                  update(jm,
                                         n.iter = n_burn)

                                  samples = coda.samples(jm,
                                                         n.iter = n_draw,
                                                         variable.names = Pars,
                                                         thin = n_thin)
                                  return(samples)
                                })


out_ABUN <- coda::mcmc.list(out.1[[1]][[1]],
                        out.1[[2]][[1]],
                        out.1[[3]][[1]])

invisible(file.remove(JAGS_FILE))
```

Posterior summary

```
MCMCsummary(out_ABUN, params = c('alpha', 'beta'), round = 3)
```

```
##             mean    sd   2.5%    50% 97.5% Rhat
## alpha[1]  8.926 0.090  8.751  8.925 9.105    1
## alpha[2]  7.905 0.045  7.816  7.905 7.992    1
## alpha[3]  3.248 0.201  2.852  3.248 3.646    1
## alpha[4]  6.118 0.124  5.868  6.119 6.358    1
## alpha[5]  5.159 0.151  4.862  5.159 5.458    1
## alpha[6]  3.263 0.065  3.137  3.263 3.392    1
## beta[1]   0.025 0.004  0.018  0.025 0.032    1
## beta[2]   0.006 0.002  0.002  0.006 0.010    1
## beta[3]   0.014 0.009 -0.004  0.014 0.031    1
## beta[4]  -0.003 0.005 -0.013 -0.003 0.007    1
## beta[5]   0.011 0.007 -0.002  0.011 0.025    1
## beta[6]   0.029 0.003  0.024  0.029 0.035    1
```

Breeding Success ~ Time

```
#all breeding success data
all_BS <- c(ADPE$BS, EMPE$BS, SOFU$BS,
            CAPE$BS, SNPE$BS, SPSK$BS)

#prepare data for JAGS model
BS_DATA <- list(
  y = all_BS, #response
  N = length(all_BS), #length data
  M = 6,
  id = c(rep(1, NROW(ADPE)), rep(2, NROW(EMPE)),
         rep(3, NROW(SOFU)), rep(4, NROW(CAPE)),
         rep(5, NROW(SNPE)), rep(6, NROW(SPSK))),
  YEAR = rep(1:length(ADPE$YEAR), 6))


DATA <- BS_DATA
```

```
#JAGS model
{
sink("BS_trend.jags")

cat("
    model {

    for (i in 1:N)
    {
    y[i] ~ dnorm(mu[i], tau.y[id[i]])
    mu[i] <- alpha[id[i]] + beta[id[i]]*YEAR[i]
    }

    #priors

    for(j in 1:M)
    {
    tau.y[j] <- pow(sigma.y[j], -2)
```

```
    sigma.y[j] ~ dunif(0, 100)

    alpha[j] <- B[j,1]
    beta[j] <- B[j,2]
    B[j, 1:2] ~ dmnorm(B.hat[j,], Tau.B[,])
    B.hat[j,1] <- mu.a
    B.hat[j,2] <- mu.b
    }

    mu.a ~ dnorm(1, 0.01)
    mu.b ~ dnorm(1, 0.01)

    Tau.B[1:2, 1:2] <- inverse(Sigma.B[,])
    Sigma.B[1, 1] <- pow(sigma.a, 2)
    sigma.a ~ dunif(0, 200)
    Sigma.B[2, 2] <- pow(sigma.b, 2)
    sigma.b ~ dunif(0, 10)
    Sigma.B[1, 2] <- rho * sigma.a * sigma.b
    Sigma.B[2, 1] <- Sigma.B[1, 2]
    rho ~ dunif(-1, 1)

    }",fill = TRUE)

sink()
}

# Starting values -------------------------------------------------------


Inits_1 <- list(B = array(rnorm(2 * DATA$M), c(DATA$M, 2)),
                mu.a = 1,
                mu.b = 1,
                sigma.a = 1,
                sigma.b = 1,
                rho = 0.5,
                sigma.y = runif(DATA$M, 0 , 100),
                .RNG.name = "base::Mersenne-Twister",
                .RNG.seed = 1)

Inits_2 <- list(B = array(rnorm(2 * DATA$M), c(DATA$M, 2)),
                mu.a = 1,
                mu.b = 1,
                sigma.a = 1,
                sigma.b = 1,
                rho = 0.5,
                sigma.y = runif(DATA$M, 0 , 100),
                .RNG.name = "base::Wichmann-Hill",
                .RNG.seed = 2)

Inits_3 <- list(B = array(rnorm(2 * DATA$M), c(DATA$M, 2)),
                mu.a = 1,
                mu.b = 1,
                sigma.a = 1,
```

```
                sigma.b = 1,
                rho = 0.5,
                sigma.y = runif(DATA$M, 0 , 100),
                .RNG.name = "base::Marsaglia-Multicarry",
                .RNG.seed = 3)

F_Inits <- list(Inits_1, Inits_2, Inits_3)


# Parameters to track -------------------------------------------------


Pars <- c('alpha',
          'beta',
          'mu.a',
          'mu.b',
          'sigma.y',
          'sigma.a',
          'sigma.b',
          'rho',
          'mu'
)


# Inputs for MCMC -----------------------------------------------------

JAGS_FILE <- 'BS_trend.jags'
n_adapt <- 8000  # number for initial adapt
n_burn <- 30000000 # number burnin
n_draw <- 100000000  # number of final draws to make
n_thin <- 500    # thinning rate
n_chain <- 3  # number of chains


# Run model (parallel) ------------------------------------------------------------

#number of chains
cl <- parallel::makeCluster(n_chain)

pid <- NA
for(i in 1:n_chain)
{
  pidNum <- capture.output(cl[[i]])
  start <- regexpr("pid", pidNum)[[1]]
  end <- nchar(pidNum)
  pid[i] <- substr(pidNum, (start + 4), end)
}

parallel::clusterExport(cl,
                        c('DATA',
                          'n_adapt',
                          'n_burn',
                          'n_draw',
```

```
                            'n_thin',
                            'Pars',
                            'pid',
                            'F_Inits',
                            'JAGS_FILE'
                          ))


out.1 <- parallel::clusterEvalQ(cl,
                                {
                                  require(rjags)
                                  processNum <- which(pid==Sys.getpid())
                                  m.inits <- F_Inits[[processNum]]

                                  jm = jags.model(data = DATA,
                                                  file = paste0(JAGS_FILE),
                                                  inits = m.inits,
                                                  n.chains = 1,
                                                  n.adapt = n_adapt)

                                  update(jm,
                                         n.iter = n_burn)

                                  samples = coda.samples(jm,
                                                         n.iter = n_draw,
                                                         variable.names = Pars,
                                                         thin = n_thin)
                                  return(samples)
                                })


out_BS <- coda::mcmc.list(out.1[[1]][[1]],
                          out.1[[2]][[1]],
                          out.1[[3]][[1]])

invisible(file.remove(JAGS_FILE))
```

Posterior summary

```
MCMCsummary(out_BS, params = c('alpha', 'beta'), round = 3)
```

```
##            mean    sd   2.5%    50%  97.5% Rhat
## alpha[1]  0.830 0.144  0.606  0.807  1.180    1
## alpha[2]  0.592 0.086  0.412  0.597  0.745    1
## alpha[3]  0.735 0.053  0.630  0.735  0.837    1
## alpha[4]  0.783 0.091  0.613  0.778  0.977    1
## alpha[5]  0.685 0.060  0.569  0.684  0.807    1
## alpha[6]  0.841 0.104  0.666  0.830  1.070    1
## beta[1]  -0.006 0.005 -0.019 -0.005  0.002    1
## beta[2]  -0.002 0.004 -0.008 -0.003  0.006    1
## beta[3]  -0.003 0.002 -0.007 -0.003  0.002    1
## beta[4]  -0.004 0.003 -0.011 -0.004  0.002    1
## beta[5]  -0.007 0.003 -0.013 -0.007 -0.002    1
```

```
## beta[6]  -0.006 0.004 -0.015 -0.006  0.000    1
```

Plots of Abundance and Breeding Success over time

```
#function to plot trends over time
plt.fun <- function(IN, i, TITLE = '', LABEL = '')
{
  gg_color_hue <- function(n) {
    hues = seq(15, 375, length=n+1)
    hcl(h=hues, l=65, c=100)[1:n]
  }

  cols <- gg_color_hue(6)

    TPLOT <- ggplot(IN, aes(YEAR, MN), color= cols[i]) +
      geom_ribbon(aes(ymin = LCI, ymax = UCI), fill=cols[i], alpha=0.25) +
      geom_line(colour= cols[i], size=2.5, linetype='dashed') +
      geom_line(data = IN, aes(YEAR, VAR), size=2.5, color= cols[i]) +
      geom_point(data = IN, aes(YEAR, VAR), size=4, color= cols[i]) +
      xlab("") +
      ylab("") +
      ggtitle(paste0(TITLE)) +
      theme_bw() +
      xlab('Year') +
      ylab(paste0(LABEL))

  return(TPLOT)
}



#ABUN_DATA

#calculate median and 2.5% and 97.5% CI
ABUN_med <- MCMCsummary(out_ABUN, params = 'mu', round = 3, Rhat = FALSE)[,4]
ABUN_LCI <- MCMCsummary(out_ABUN, params = 'mu', round = 3, Rhat = FALSE)[,3]
ABUN_UCI <- MCMCsummary(out_ABUN, params = 'mu', round = 3, Rhat = FALSE)[,5]

ABUN_tplot <- data.frame(YEAR = 1980:2016, SITE = factor(ABUN_DATA$id), VAR = ABUN_DATA$y,
                         MN = ABUN_med, LCI = ABUN_LCI, UCI = ABUN_UCI)

#create plots
i <- 1
ABUN_pst <- filter(ABUN_tplot, SITE == i)
ABUN_plt1 <- plt.fun(ABUN_pst, i, TITLE = 'Adélie penguin Abundance', LABEL = 'log(Abundance)')

i <- 2
ABUN_pst <- filter(ABUN_tplot, SITE == i)
ABUN_plt2 <- plt.fun(ABUN_pst, i, TITLE = 'Emperor penguin Abundance', LABEL = 'log(Abundance)')

i <- 3
ABUN_pst <- filter(ABUN_tplot, SITE == i)
ABUN_plt3 <- plt.fun(ABUN_pst, i, TITLE = 'Southern fulmar Abundance', LABEL = 'log(Abundance)')
```

```
i <- 4
ABUN_pst <- filter(ABUN_tplot, SITE == i)
ABUN_plt4 <- plt.fun(ABUN_pst, i, TITLE = 'Cape petrel Abundance', LABEL = 'log(Abundance)')

i <- 5
ABUN_pst <- filter(ABUN_tplot, SITE == i)
ABUN_plt5 <- plt.fun(ABUN_pst, i, TITLE = 'Snow petrel Abundance', LABEL = 'log(Abundance)')

i <- 6
ABUN_pst <- filter(ABUN_tplot, SITE == i)
ABUN_plt6 <- plt.fun(ABUN_pst, i, TITLE = 'South polar skua Abundance', LABEL = 'log(Abundance)')




#BS_DATA

#calculate median and 2.5% and 97.5% CI
BS_med <- MCMCsummary(out_BS, params = 'mu', round = 3, Rhat = FALSE)[,4]
BS_LCI <- MCMCsummary(out_BS, params = 'mu', round = 3, Rhat = FALSE)[,3]
BS_UCI <- MCMCsummary(out_BS, params = 'mu', round = 3, Rhat = FALSE)[,5]

BS_tplot <- data.frame(YEAR = 1980:2016, SITE = factor(BS_DATA$id), VAR = BS_DATA$y,
                       MN = BS_med, LCI = BS_LCI, UCI = BS_UCI)

#create plots
i <- 1
BS_pst <- filter(BS_tplot, SITE == i)
BS_plt1 <- plt.fun(BS_pst, i, TITLE = 'Adélie penguin Breeding Success', LABEL = 'Breeding Success (chi

i <- 2
BS_pst <- filter(BS_tplot, SITE == i)
BS_plt2 <- plt.fun(BS_pst, i, TITLE = 'Emperor penguin Breeding Success', LABEL = 'Breeding Success (ch

i <- 3
BS_pst <- filter(BS_tplot, SITE == i)
BS_plt3 <- plt.fun(BS_pst, i, TITLE = 'Southern fulmar Breeding Success', LABEL = 'Breeding Success (ch

i <- 4
BS_pst <- filter(BS_tplot, SITE == i)
BS_plt4 <- plt.fun(BS_pst, i, TITLE = 'Cape petrel Breeding Success', LABEL = 'Breeding Success (chicks/

i <- 5
BS_pst <- filter(BS_tplot, SITE == i)
BS_plt5 <- plt.fun(BS_pst, i, TITLE = 'Snow petrel Breeding Success', LABEL = 'Breeding Success (chicks/

i <- 6
BS_pst <- filter(BS_tplot, SITE == i)
BS_plt6 <- plt.fun(BS_pst, i, TITLE = 'South polar skua Breeding Success', LABEL = 'Breeding Success (c
```

```
#plot both abundance and breeding success together


#ADPE
grid.arrange(ABUN_plt1, BS_plt1, ncol=1)
```

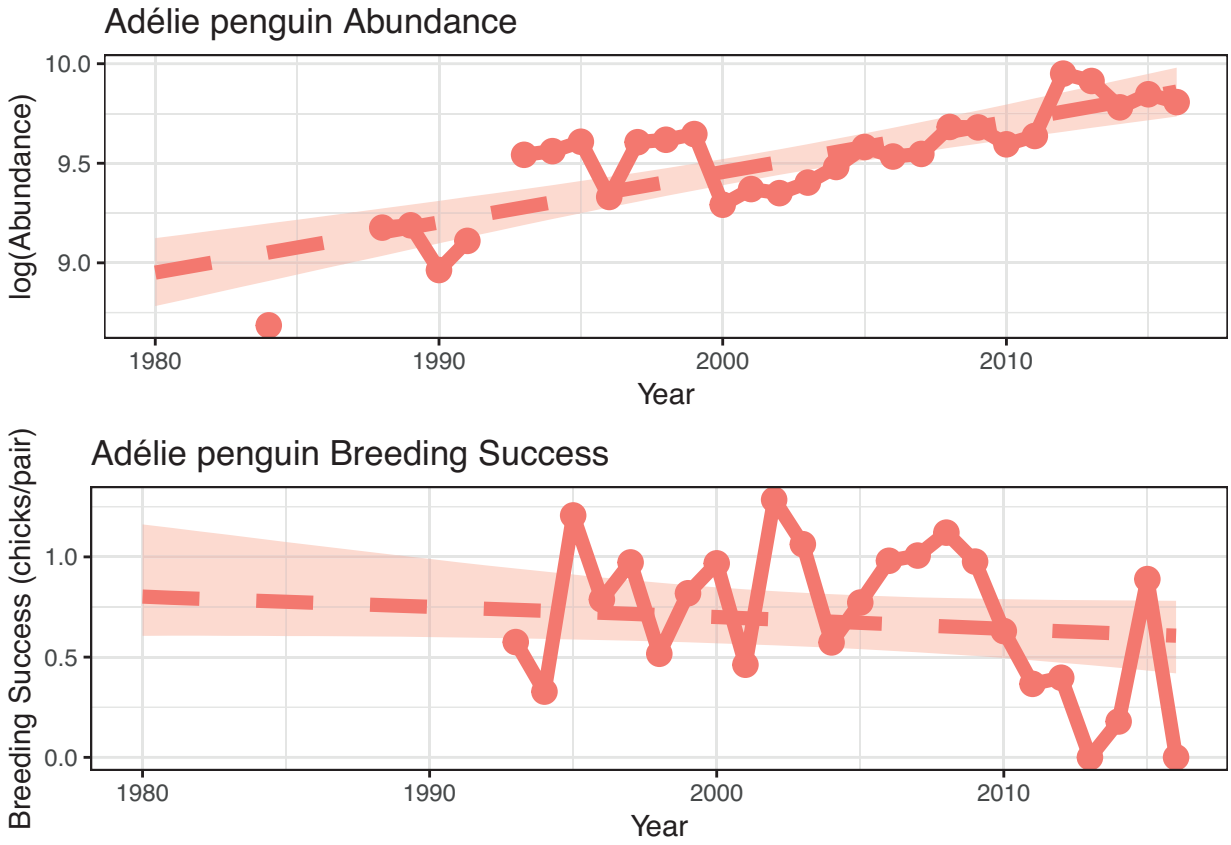

Figure A8-1: Abundance and Breeding Success for Adélie penguin

```
#EMPE
grid.arrange(ABUN_plt2, BS_plt2, ncol=1)
```

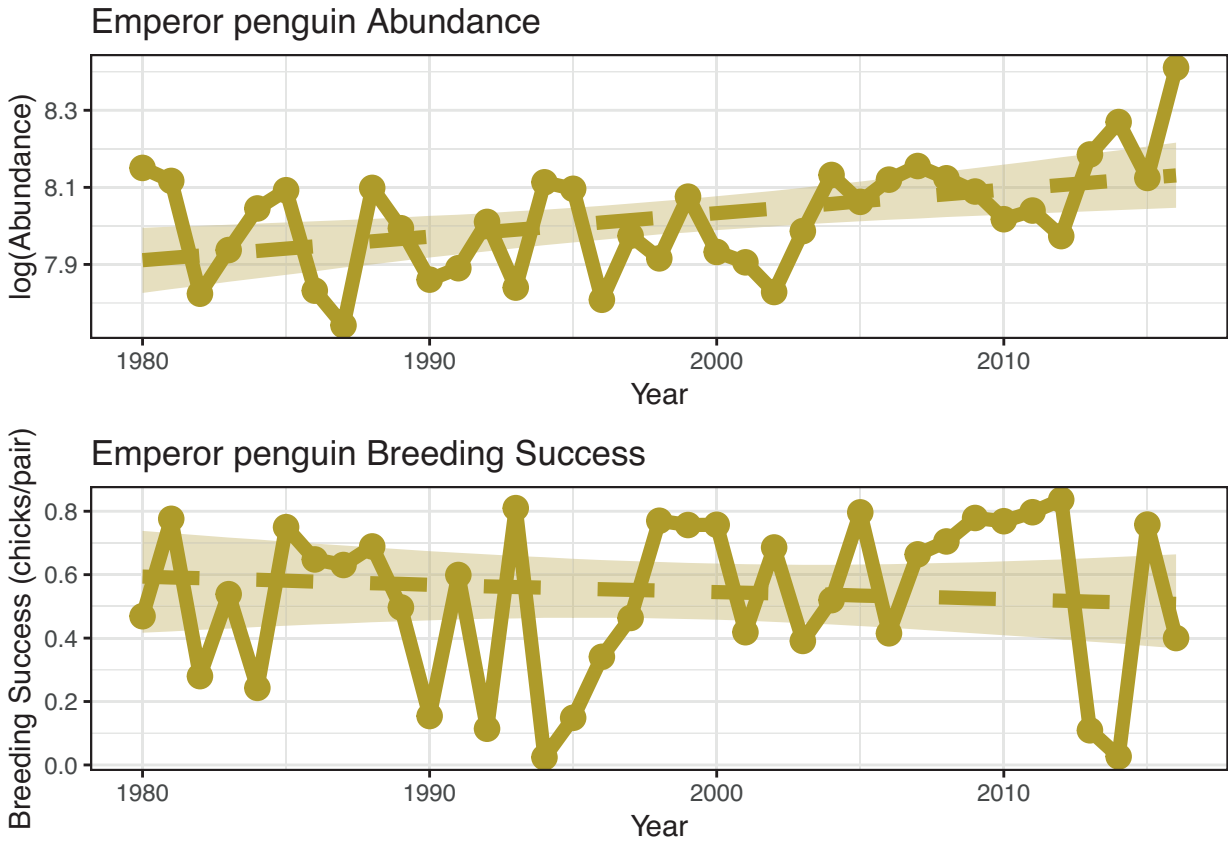

Figure A8-2: Abundance and Breeding Success for Emperor penguin

```
#SOFU
grid.arrange(ABUN_plt3, BS_plt3, ncol=1)
```

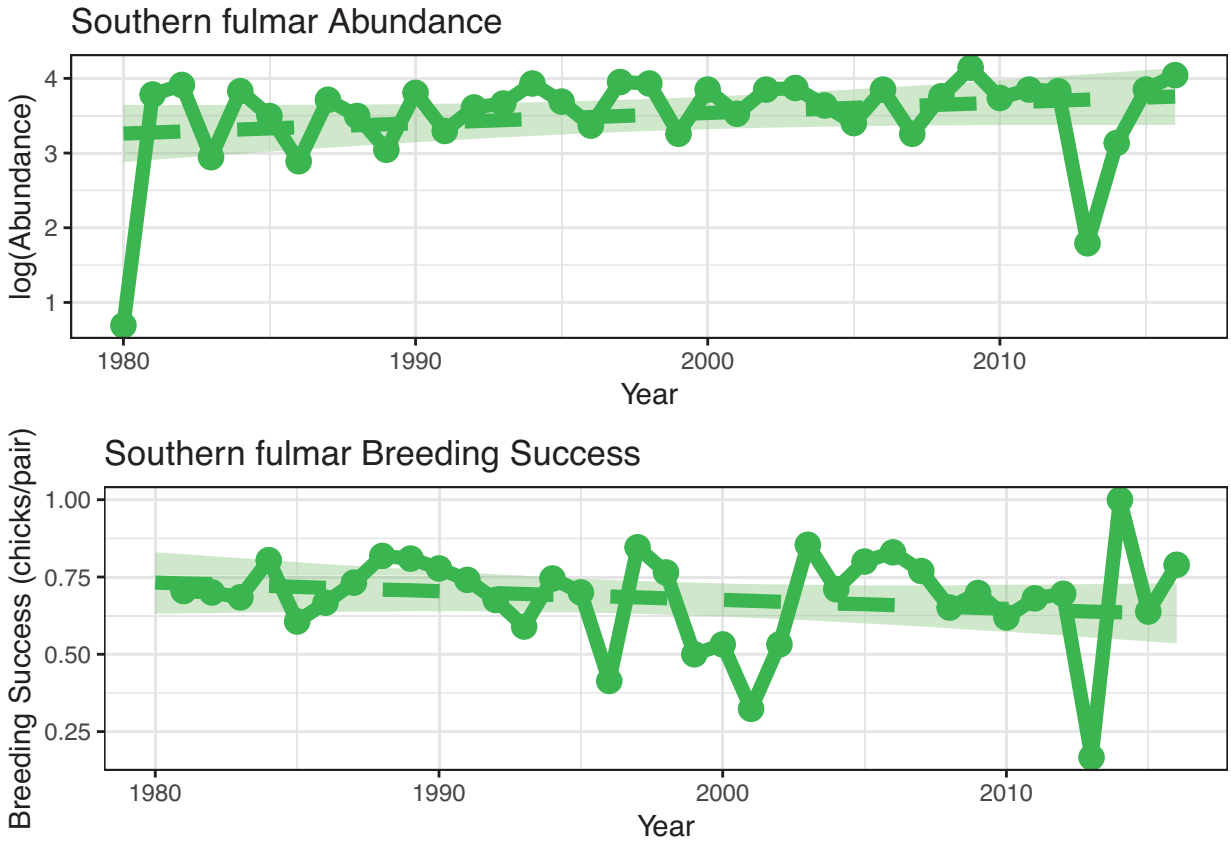

Figure A8-3: Abundance and Breeding Success for southern fulmar

```
#CAPE
grid.arrange(ABUN_plt4, BS_plt4, ncol=1)
```

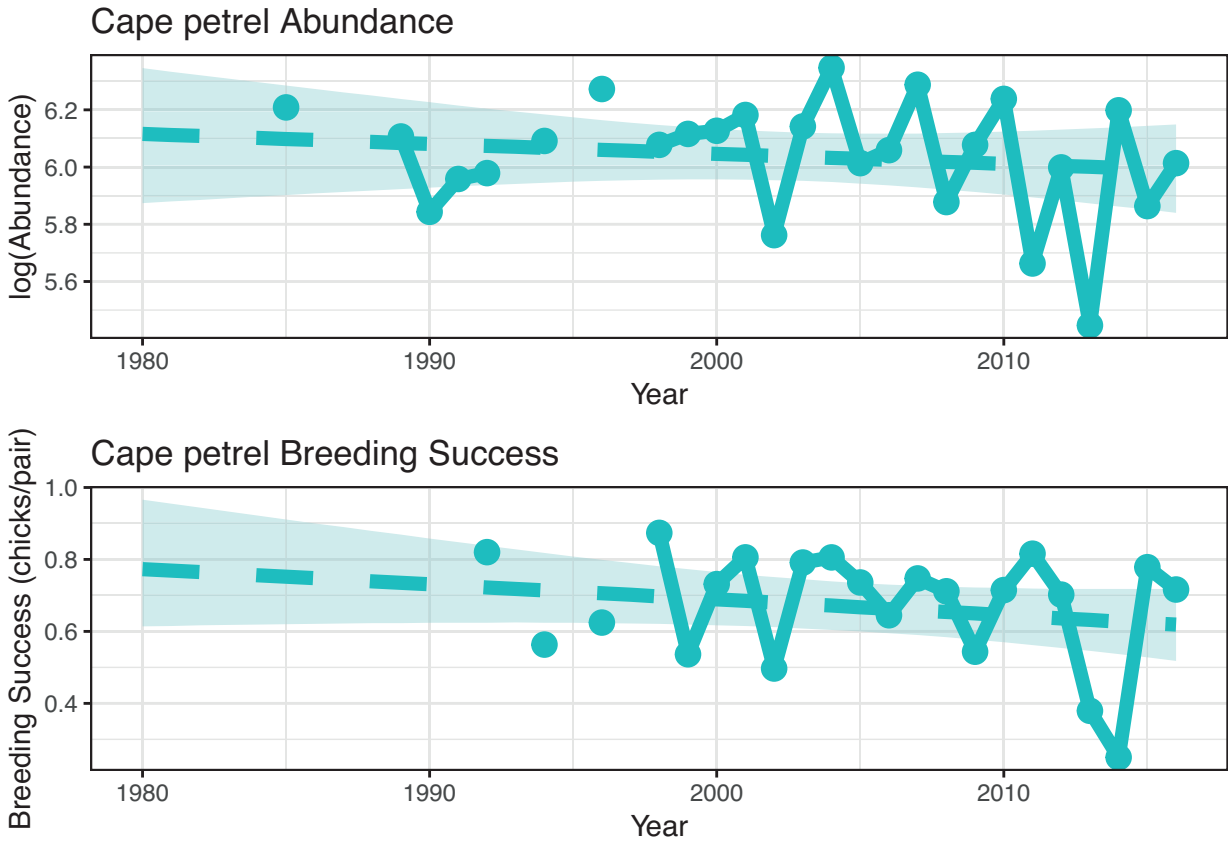

Figure A8-4: Abundance and Breeding Success for cape petrel

```
#SNPE
grid.arrange(ABUN_plt5, BS_plt5, ncol=1)
```

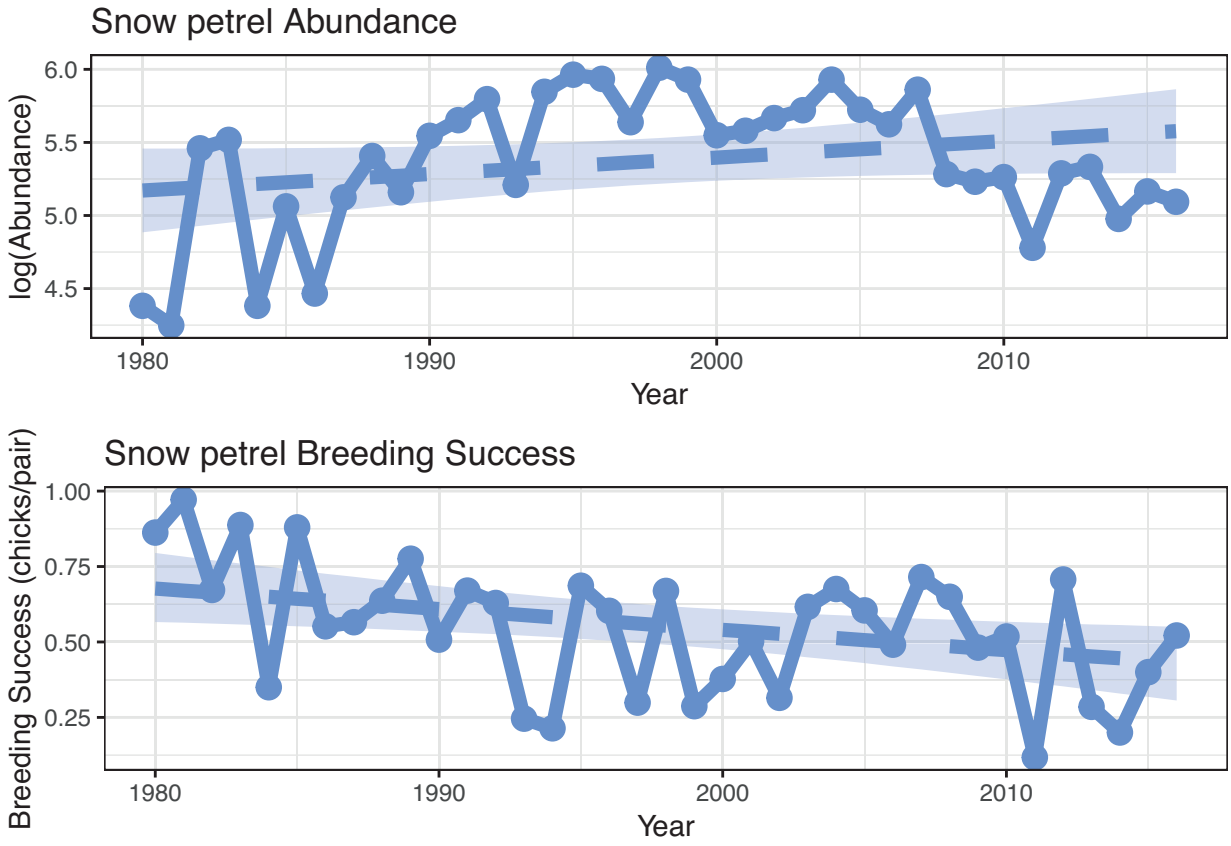

Figure A8-5: Abundance and Breeding Success for snow petrel

```
#SPSK
grid.arrange(ABUN_plt6, BS_plt6, ncol=1)
```

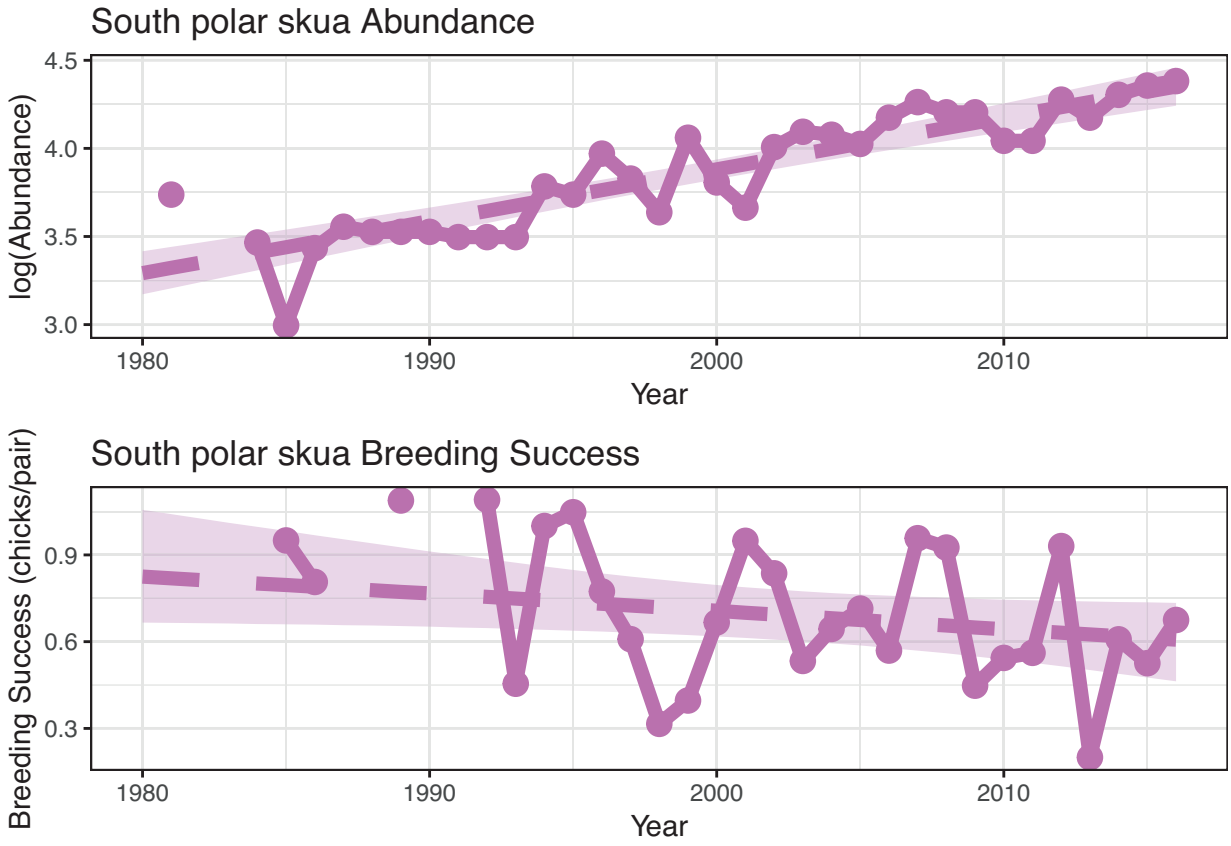

Figure A8-6: Abundance and Breeding Success for south polar skua

Slope estimates for Abundance and Breeding Success models

```
MCMCplot(out_ABUN,
        params = 'beta',
        horiz = FALSE,
        ylim = c(-0.04, 0.06),
        labels = c('ADPE', 'EMPE', 'SOFU', 'CAPE', 'SNPE', 'SPSK'),
        main = 'Abundance',
        main_text_sz = 1.5,
        ref_ovl = FALSE)
```

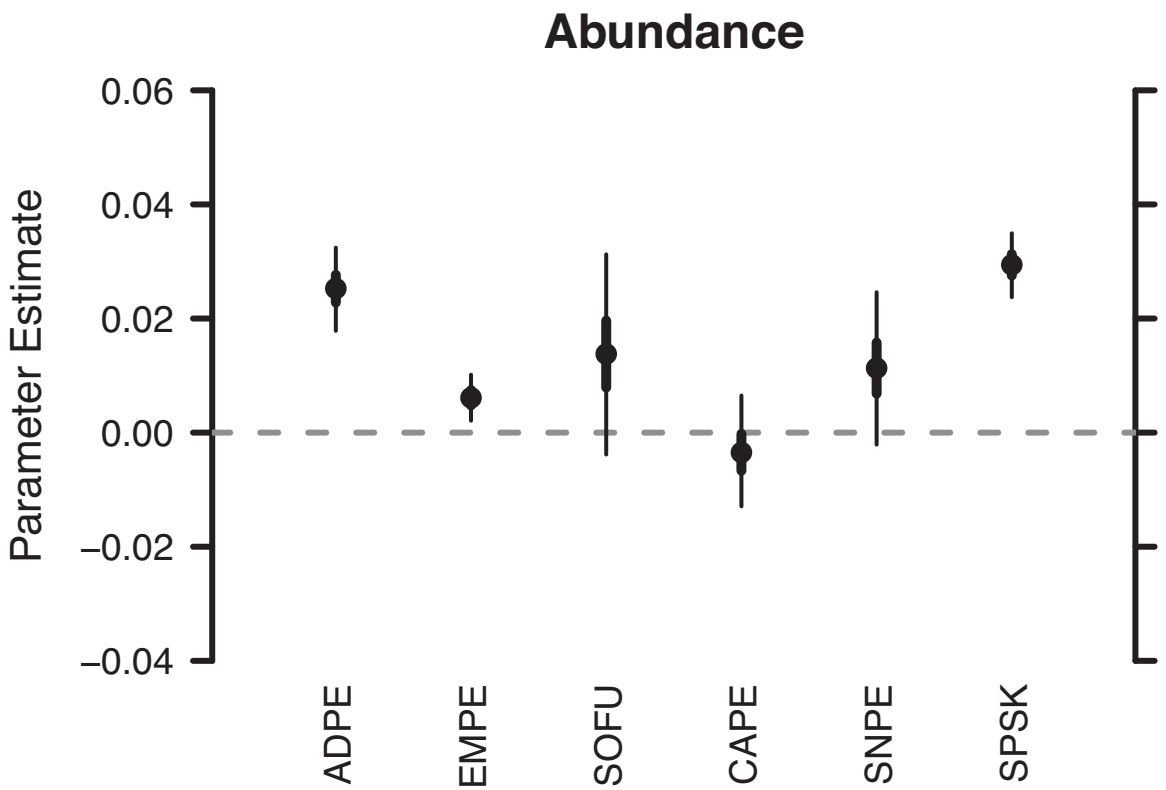

Figure A8-7: Posterior estimates for β parameters in Abundance model. ADPE – Adélie penguin; EMPE – emperor penguin; SOFU – southern fulmar; CAPE – cape petrel; SNPE – snow petrel; SPSK – south polar skua

```
MCMCplot(out_BS,
        params = 'beta',
        horiz = FALSE,
        ylim = c(-0.04, 0.06),
        labels = c('ADPE', 'EMPE', 'SOFU', 'CAPE', 'SNPE', 'SPSK'),
        main = 'Breeding Success',
        main_text_sz = 1.5,
        ref_ovl = FALSE)
```

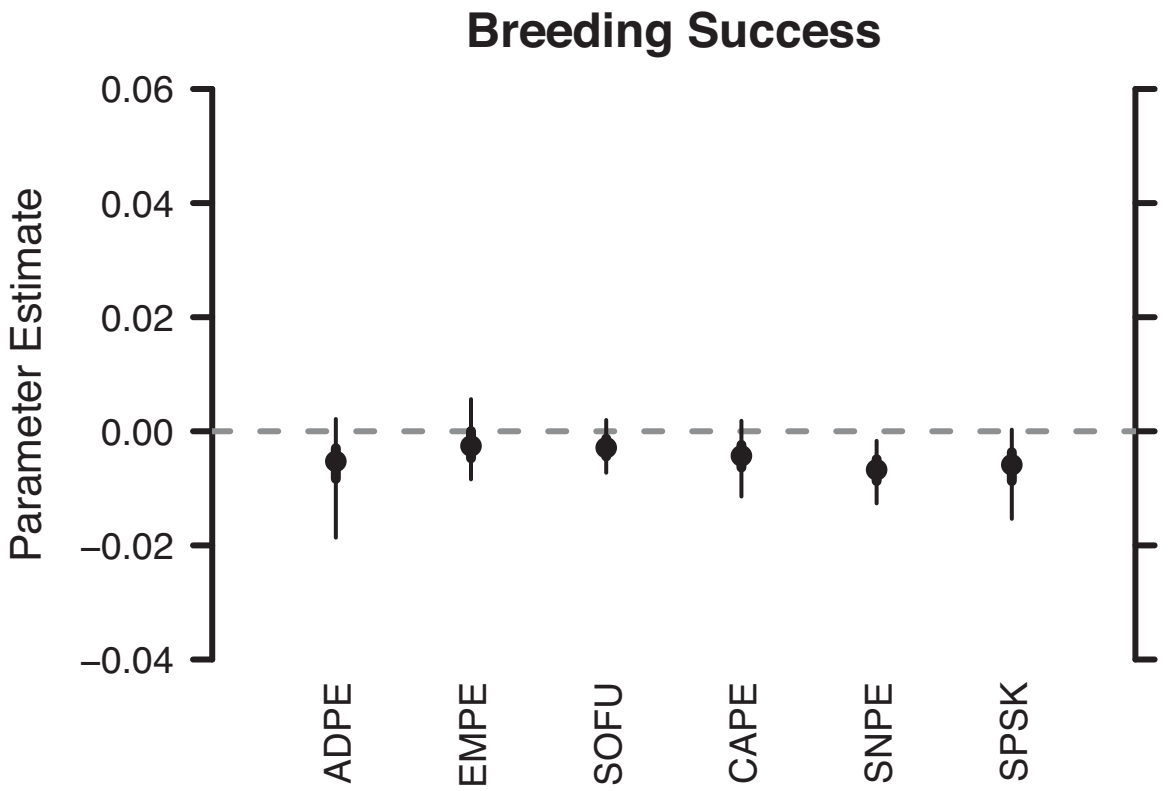

Figure A8-8: Posterior estimates for β parameters in Breeding Success model. ADPE – Adélie penguin; EMPE – emperor penguin; SOFU – southern fulmar; CAPE – cape petrel; SNPE – snow petrel; SPSK – south polar skua

Relationship between Breeding Success and Abundance

Detrend data

```
#function that detrends - can use residuals() if no NA - this accomodates NAs
detrend_fun <- function(INPUT)
{
  tfit <- summary(lm(INPUT ~ c(1:length(INPUT))))
  alpha <- tfit$coefficients[1,1]
  beta <- tfit$coefficients[2,1]

  TOUT <- c()
  for (i in 1:length(INPUT))
  {
    temp <- alpha + beta*i
    temp2 <- INPUT[i] - temp
    TOUT <- c(TOUT, temp2)
  }

  return(TOUT)
}


#NO DETREND
```

```
ADPE_ndt <- data.frame(YEAR = ADPE$YEAR, BS = ADPE$BS, L_ABUN = log(ADPE$ABUN))
EMPE_ndt <- data.frame(YEAR = EMPE$YEAR, BS = EMPE$BS, L_ABUN = log(EMPE$ABUN))
SOFU_ndt <- data.frame(YEAR = SOFU$YEAR, BS = SOFU$BS, L_ABUN = log(SOFU$ABUN))
CAPE_ndt <- data.frame(YEAR = CAPE$YEAR, BS = CAPE$BS, L_ABUN = log(CAPE$ABUN))
SNPE_ndt <- data.frame(YEAR = SNPE$YEAR, BS = SNPE$BS, L_ABUN = log(SNPE$ABUN))
SPSK_ndt <- data.frame(YEAR = SPSK$YEAR, BS = SPSK$BS, L_ABUN = log(SPSK$ABUN))

#DETREND
ADPE_dt <- data.frame(YEAR = ADPE$YEAR, apply(ADPE_ndt[,-1], 2, detrend_fun))
EMPE_dt <- data.frame(YEAR = EMPE$YEAR, apply(EMPE_ndt[,-1], 2, detrend_fun))
SOFU_dt <- data.frame(YEAR = SOFU$YEAR, apply(SOFU_ndt[,-1], 2, detrend_fun))
CAPE_dt <- data.frame(YEAR = CAPE$YEAR, apply(CAPE_ndt[,-1], 2, detrend_fun))
SNPE_dt <- data.frame(YEAR = SNPE$YEAR, apply(SNPE_ndt[,-1], 2, detrend_fun))
SPSK_dt <- data.frame(YEAR = SPSK$YEAR, apply(SPSK_ndt[,-1], 2, detrend_fun))
```

Calculate correlation coefficients on lags

```
#function to calculate correlation coefficients
boot.fun <- function(data, i)
{
  cor(data[i, 1], data[i, 2], use = 'pairwise.complete.obs')
}


#returns confidence interval on correlation coefficient
m.fun <- function(ONE, TWO)
{
  cc_temp <- boot.fun(cbind(ONE, TWO))
  boot_temp <- boot(cbind(ONE,TWO), boot.fun, 10000)
  ci_temp <- suppressWarnings(boot.ci(boot_temp))

  lci <- ci_temp$normal[,2] #lower CI
  uci <- ci_temp$normal[,3] #upper CI

  TB <- cbind(cc_temp, lci, uci)
  colnames(TB) <- c('CC', 'LCI', 'UCI')

  return(TB)
}


#calculates cross correlation coefficients and CI for 0:LM lags
#IN_1 is ABUN, IN_2 is BS, LM is lag
cross_cor_boot <- function(IN_1, IN_2, LM)
{
  #0 lag
  OUT <- cbind(Lag = 0, m.fun(IN_1, IN_2))

  #1 to LM lag
  LEN <- length(IN_1)
  for (i in 1:LM)
  {
    temp <- cbind(Lag = i, m.fun(IN_1[-c(1:i)], IN_2[-c((LEN - (i-1)):LEN)]))
```

```
    OUT <- rbind(OUT, temp)
  }
  return(OUT)
}
```

```
#run functions
ADPE_cc_ABUN_BS <- cross_cor_boot(ADPE_dt$L_ABUN, ADPE_dt$BS, 8)
EMPE_cc_ABUN_BS <- cross_cor_boot(EMPE_dt$L_ABUN, EMPE_dt$BS, 8)
SOFU_cc_ABUN_BS <- cross_cor_boot(SOFU_dt$L_ABUN, SOFU_dt$BS, 8)
CAPE_cc_ABUN_BS <- cross_cor_boot(CAPE_dt$L_ABUN, CAPE_dt$BS, 8)
SNPE_cc_ABUN_BS <- cross_cor_boot(SNPE_dt$L_ABUN, SNPE_dt$BS, 8)
SPSK_cc_ABUN_BS <- cross_cor_boot(SPSK_dt$L_ABUN, SPSK_dt$BS, 8)
```

Create correlation plots

```
#function to create plots
cor_plt_fun <- function(INPUT, TITLE)
{

  IN <- INPUT[,-1]
  X <- 1
  LL <- NROW(IN) * X

  plot(1:LL, rep(0, LL), ylim = c(-1,1), xlim = c(0.5,(LL+0.3)), type = "n",
       ann = TRUE, xaxt = 'n', yaxt = "n",
       bty = "n", ylab = 'Correlation coefficient', xlab = NA,
       main = paste0(TITLE))

  colfunc <- colorRampPalette(c("red", 'white', "blue"))
  cols <- colfunc(201)
  THICK <- 1.6

  #plot correlations
  for (i in 1:NROW(IN))
  {
    d <- i * X

    col_ind <- 100 + round((IN[i,1]*100), digits = 0)
    COLOR <- cols[col_ind]

    rect(d-0.33, IN[i,2], d+0.33, IN[i,3], col = COLOR, lwd = THICK)

    #CC vertical
    matlines(c(d-0.33, d+0.33), c(IN[i, 1], IN[i, 1]),
             type = 'l', lty = 1, lwd = THICK, col = 'black')
  }

  #0 tick line
  abline(h=0, lty = 2, lwd = 4, col = 'grey43')

  #y axis
  axis(2, lwd.ticks = THICK, lwd = THICK)
```

```
  #x axis
  tick_pos <- (1:NROW(IN)) * X
  lbs <- c(rep(NA, NROW(IN)))
  for (i in 1:NROW(IN))
  {
    lbs[i] <- paste0('Lag ', i-1)
  }

  axis(1, lwd.ticks = THICK,
       lwd = THICK,
       at = tick_pos,
       labels = lbs,
       las = 2,
       col = NA)
}
```

```
#ADPE
cor_plt_fun(ADPE_cc_ABUN_BS, 'Adélie penguin')
```

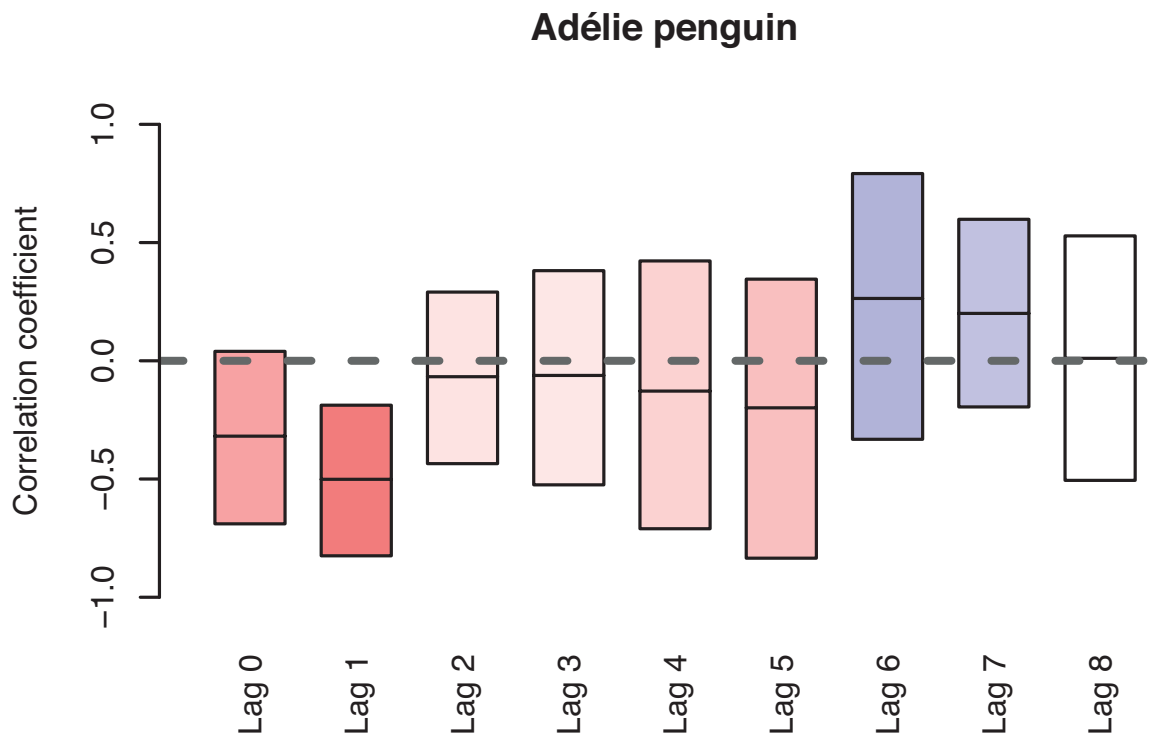

Figure A8-9: Cross correlation plots between Abundance and Breeding Success - Adélie penguin

```
#EMPE
cor_plt_fun(EMPE_cc_ABUN_BS, 'Emperor penguin')
```

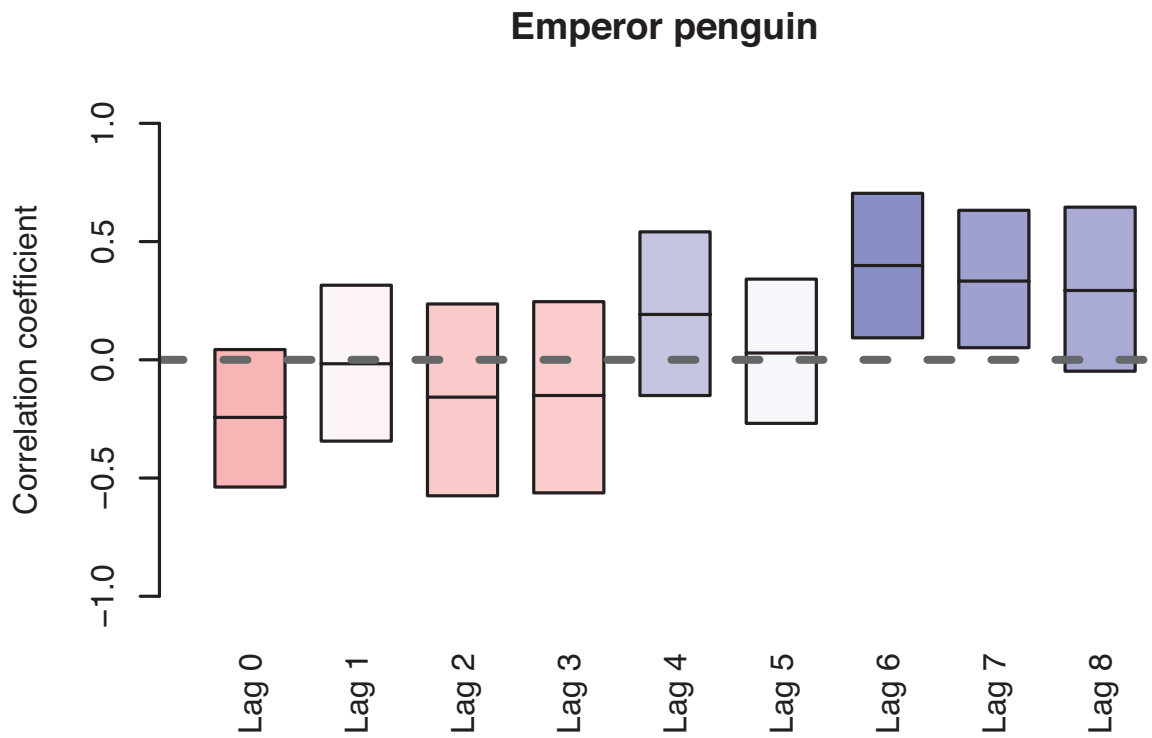

Figure A8-10: Cross correlation plots between Abundance and Breeding Success - emperor penguin

```
#SOFU
cor_plt_fun(SOFU_cc_ABUN_BS, 'Southern fulmar')
```

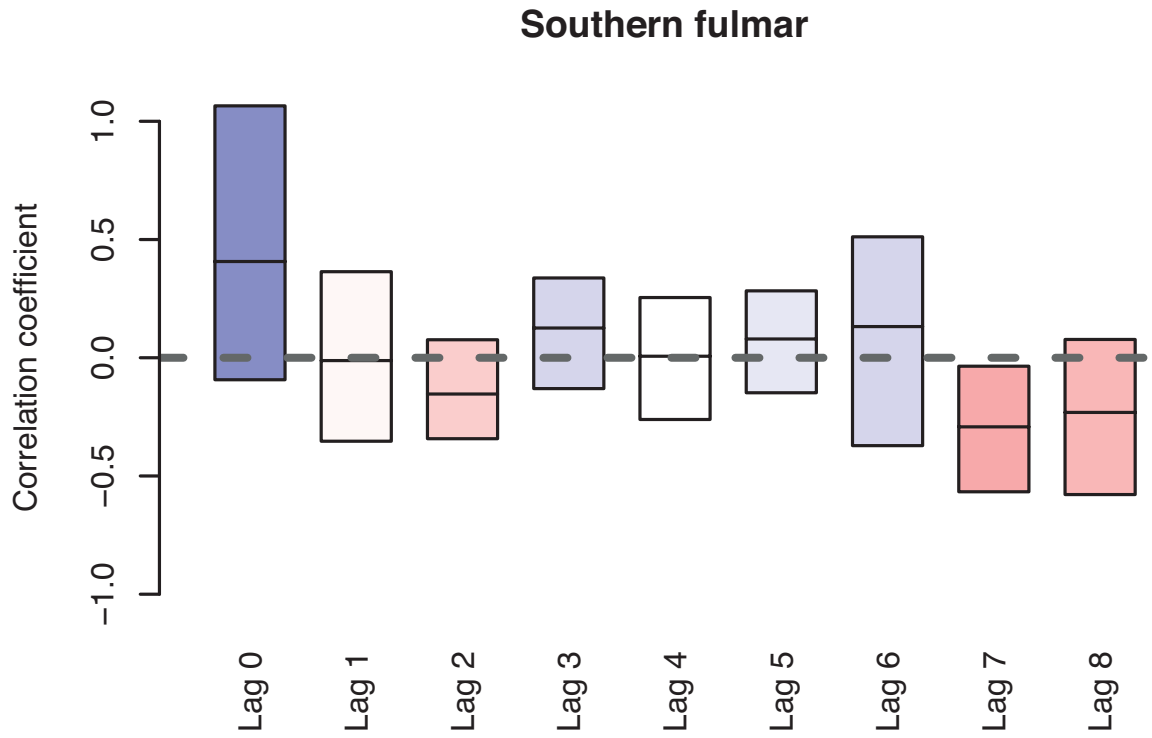

Figure A8-11: Cross correlation plots between Abundance and Breeding Success - southern fulmar

```
#CAPE
cor_plt_fun(CAPE_cc_ABUN_BS, 'Cape petrel')
```

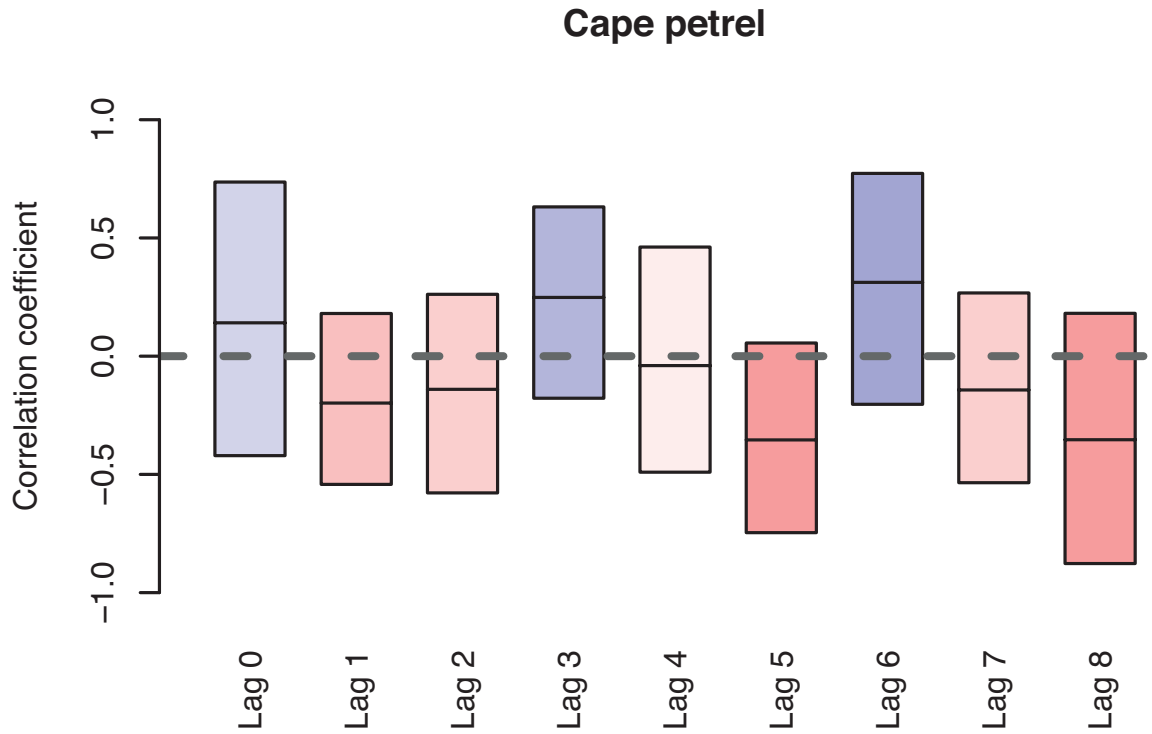

Figure A8-12: Cross correlation plots between Abundance and Breeding Success - cape petrel

```
#SNPE
cor_plt_fun(SNPE_cc_ABUN_BS, 'Snow petrel')
```

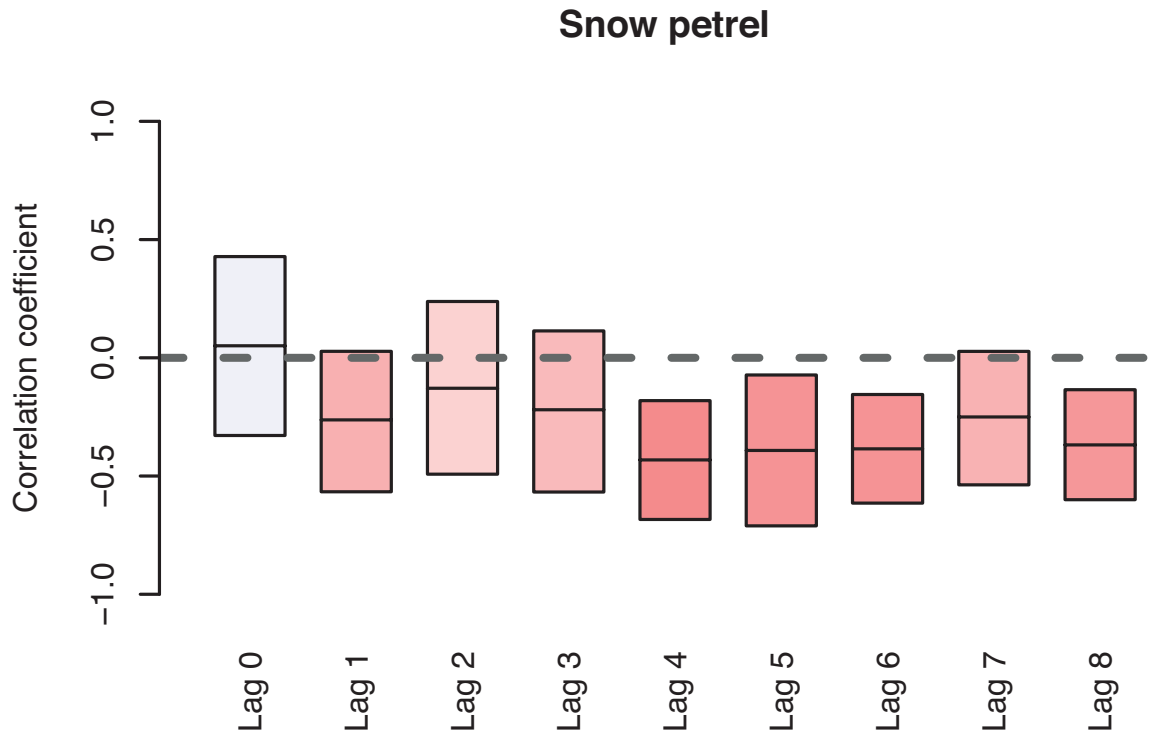

Figure A8-13: Cross correlation plots between Abundance and Breeding Success - snow petrel

```
#SPSK
cor_plt_fun(SPSK_cc_ABUN_BS, 'South polar skua')
```

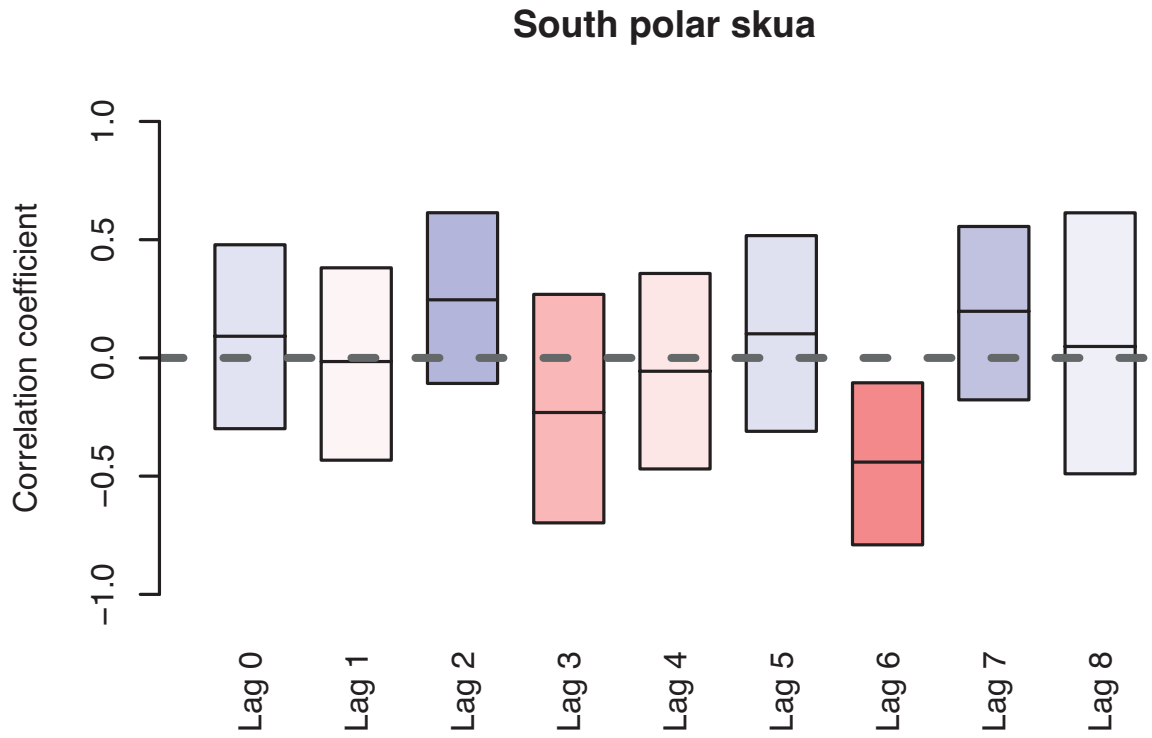

Figure A8-14: Cross correlation plots between Abundance and Breeding Success - south polar skua

Community synchrony in breeding productivity

$$
\begin{aligned}
F_{ij} &\sim Binom(E_{ij}, p_{ij}) \\
logit(p_{ij}) &= \mu_j + \delta_i + \epsilon_{ij} \\
\delta_i &\sim N(0, \sigma^2_{\delta}) \\
\epsilon_{ij} &\sim N(0, \sigma^2_{\epsilon_j})
\end{aligned}
\tag{A8-2}
$$

Model all years

```
#sort for complete years only
BS_mrg <- data.frame(YEAR = ADPE$YEAR,
                     ADPE_BS = ADPE$BS,
                     EMPE_BS = EMPE$BS,
                     SOFU_BS = SOFU$BS,
                     CAPE_BS = CAPE$BS,
                     SNPE_BS = SNPE$BS,
                     SPSK_BS = SPSK$BS)

#which rows have no NAs (data for every species)
BS_ind <- which(table(which(!is.na(BS_mrg), arr.ind=TRUE)[,1]) == 7)

ADPE_f <- ADPE[BS_ind, ]
EMPE_f <- EMPE[BS_ind, ]
SOFU_f <- SOFU[BS_ind, ]
CAPE_f <- CAPE[BS_ind, ]
SNPE_f <- SNPE[BS_ind, ]
SPSK_f <- SPSK[BS_ind, ]


#Data for JAGS model

LM_DATA <- list(
  E1 = ADPE_f$ABUN * 2, #two eggs
  E2 = EMPE_f$ABUN, #one egg
  E3 = SOFU_f$ABUN, #one egg
  E4 = CAPE_f$ABUN, #one egg
  E5 = SNPE_f$ABUN, #one egg
  E6 = SPSK_f$ABUN * 2, #two eggs
  F1 = ADPE_f$CHICKS,
  F2 = EMPE_f$CHICKS,
  F3 = SOFU_f$CHICKS,
  F4 = CAPE_f$CHICKS,
  F5 = SNPE_f$CHICKS,
  F6 = SPSK_f$CHICKS,
  N = NROW(ADPE_f))

DATA <- LM_DATA
```

```
{
sink("LM_model.jags")

cat("

    model {

    # Likelihood -------------------------------------------------------------

    for (i in 1:N)
    {
    F1[i] ~ dbin(rho[1,i], E1[i])
    F2[i] ~ dbin(rho[2,i], E2[i])
    F3[i] ~ dbin(rho[3,i], E3[i])
    F4[i] ~ dbin(rho[4,i], E4[i])
    F5[i] ~ dbin(rho[5,i], E5[i])
    F6[i] ~ dbin(rho[6,i], E6[i])
    # where Es[i] = # eggs laid for that species * number of breeding pairs in year i
    }


    # Priors and regressions -------------------------------------------------
    for (i in 1:N)
    {
    # Logistic regressions on productivity (with random terms)
    #bs are offsets for each species

    logit(rho[1,i]) <- b0[1] + d[i] + e[1,i]
    logit(rho[2,i]) <- b0[2] + d[i] + e[2,i]
    logit(rho[3,i]) <- b0[3] + d[i] + e[3,i]
    logit(rho[4,i]) <- b0[4] + d[i] + e[4,i]
    logit(rho[5,i]) <- b0[5] + d[i] + e[5,i]
    logit(rho[6,i]) <- b0[6] + d[i] + e[6,i]

    #priors
    d[i] ~ dnorm(0, tau.d)

    for (j in 1:6)
    {
    e[j, i] ~ dnorm(0, tau.e[j])
    }
  }

    tau.d  <- 1/var.d
    var.d <- sd.d * sd.d
    sd.d ~ dunif(0, 3)

    for (j in 1:6)
    {
      #priors for species effects
      b0[j] ~ dunif(-5,5)

      tau.e[j] <- 1/var.e[j]
```

```
      var.e[j] <- sd.e[j] * sd.e[j]
      sd.e[j] ~ dunif(0,3)

      # Synchrony indices
      I[j] <- var.d/(var.e[j] + var.d)

    }

    # mean synchrony index
    I_bar <- mean(I)

    #for mean productivity (2 eggs for ADPE and SPSK)
    mn_pro[1] <- mean(rho[1,]) * 2
    mn_pro[2] <- mean(rho[2,])
    mn_pro[3] <- mean(rho[3,])
    mn_pro[4] <- mean(rho[4,])
    mn_pro[5] <- mean(rho[5,])
    mn_pro[6] <- mean(rho[6,]) * 2

    }",fill = TRUE)

sink()
}


# Starting values -------------------------------------------------------


Inits_1 <- list(b0 = rep(0, 6),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Mersenne-Twister",
                .RNG.seed = 1)

Inits_2 <- list(b0 = rep(0, 6),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Wichmann-Hill",
                .RNG.seed = 2)

Inits_3 <- list(b0 = rep(0, 6),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Marsaglia-Multicarry",
                .RNG.seed = 3)

F_Inits <- list(Inits_1, Inits_2, Inits_3)
```

```
# Parameters to track -------------------------------------------------

Pars <- c('b0',
          'd',
          'var.d',
          'var.e',
          'mn_pro',
          'I',
          'I_bar')


# Inputs for MCMC -----------------------------------------------------------


JAGS_FILE <- 'LM_model.jags'
n_adapt <- 8000  # number for initial adapt
n_burn <- 30000000 # number burnin
n_draw <- 100000000  # number of final draws to make
n_thin <- 500    # thinning rate
n_chain <- 3  # number of chains


# Run model (parallel) ------------------------------------------------------------------

#number of chains
cl <- parallel::makeCluster(n_chain)

pid <- NA
for(i in 1:n_chain)
{
  pidNum <- capture.output(cl[[i]])
  start <- regexpr("pid", pidNum)[[1]]
  end <- nchar(pidNum)
  pid[i] <- substr(pidNum, (start + 4), end)
}

parallel::clusterExport(cl,
                        c('DATA',
                          'n_adapt',
                          'n_burn',
                          'n_draw',
                          'n_thin',
                          'Pars',
                          'pid',
                          'F_Inits',
                          'JAGS_FILE'
                        ))


out.1 <- parallel::clusterEvalQ(cl,
                                {
                                  require(rjags)
                                  processNum <- which(pid==Sys.getpid())
```

```
                                    m.inits <- F_Inits[[processNum]]

                                    jm = jags.model(data = DATA,
                                                    file = paste0(JAGS_FILE),
                                                    inits = m.inits,
                                                    n.chains = 1,
                                                    n.adapt = n_adapt)

                                    update(jm,
                                           n.iter = n_burn)

                                    samples = coda.samples(jm,
                                                           n.iter = n_draw,
                                                           variable.names = Pars,
                                                           thin = n_thin)
                                    return(samples)
                                  })


out_LM <- coda::mcmc.list(out.1[[1]][[1]],
                          out.1[[2]][[1]],
                          out.1[[3]][[1]])
```

Posterior summary

```
MCMCsummary(out_LM, round = 3)
```

```
##             mean    sd   2.5%    50%  97.5% Rhat
## I[1]       0.023 0.015  0.002  0.020  0.061    1
## I[2]       0.068 0.049  0.006  0.057  0.193    1
## I[3]       0.207 0.122  0.025  0.187  0.495    1
## I[4]       0.294 0.165  0.030  0.274  0.663    1
## I[5]       0.252 0.148  0.023  0.231  0.590    1
## I[6]       0.403 0.186  0.057  0.396  0.784    1
## I_bar      0.208 0.095  0.027  0.205  0.401    1
## b0[1]     -1.559 0.593 -2.731 -1.559 -0.407    1
## b0[2]      0.109 0.357 -0.598  0.110  0.813    1
## b0[3]      0.739 0.221  0.303  0.738  1.181    1
## b0[4]      0.745 0.168  0.413  0.745  1.078    1
## b0[5]     -0.132 0.184 -0.497 -0.131  0.230    1
## b0[6]     -0.764 0.148 -1.059 -0.763 -0.472    1
## d[1]      -0.113 0.279 -0.697 -0.098  0.416    1
## d[2]       0.005 0.247 -0.489  0.005  0.496    1
## d[3]       0.179 0.289 -0.360  0.163  0.787    1
## d[4]      -0.339 0.264 -0.881 -0.330  0.143    1
## d[5]      -0.005 0.244 -0.491 -0.005  0.482    1
## d[6]       0.137 0.259 -0.362  0.128  0.664    1
## d[7]      -0.110 0.261 -0.649 -0.099  0.388    1
## d[8]       0.181 0.258 -0.301  0.168  0.718    1
## d[9]       0.224 0.258 -0.251  0.211  0.764    1
## d[10]      0.230 0.252 -0.237  0.219  0.753    1
## d[11]      0.017 0.243 -0.464  0.015  0.503    1
## d[12]      0.387 0.269 -0.100  0.379  0.936    1
```

```
## d[13]      0.269 0.255 -0.204  0.260  0.794    1
## d[14]     -0.137 0.248 -0.639 -0.129  0.343    1
## d[15]      0.014 0.244 -0.464  0.011  0.507    1
## d[16]     -0.095 0.260 -0.627 -0.086  0.412    1
## d[17]      0.319 0.261 -0.159  0.309  0.852    1
## d[18]     -0.812 0.352 -1.495 -0.816 -0.095    1
## d[19]     -0.405 0.321 -1.103 -0.380  0.141    1
## d[20]      0.005 0.246 -0.479  0.002  0.503    1
## d[21]      0.039 0.241 -0.439  0.036  0.523    1
## mn_pro[1]  0.672 0.001  0.670  0.672  0.675    1
## mn_pro[2]  0.558 0.002  0.555  0.558  0.561    1
## mn_pro[3]  0.660 0.017  0.627  0.660  0.692    1
## mn_pro[4]  0.665 0.005  0.656  0.665  0.675    1
## mn_pro[5]  0.475 0.007  0.462  0.475  0.488    1
## mn_pro[6]  0.656 0.019  0.619  0.656  0.692    1
## var.d      0.166 0.108  0.017  0.145  0.434    1
## var.e[1]   7.354 1.148  4.806  7.538  8.931    1
## var.e[2]   2.508 0.939  1.249  2.318  4.872    1
## var.e[3]   0.703 0.395  0.213  0.616  1.703    1
## var.e[4]   0.415 0.193  0.153  0.379  0.891    1
## var.e[5]   0.525 0.238  0.206  0.479  1.116    1
## var.e[6]   0.250 0.145  0.056  0.220  0.612    1
```

δ - community-wide year effect

```
yrs <- ADPE_f$YEAR
MCMCplot(out_LM,
         params = 'd',
         horiz = FALSE,
         labels = yrs,
         ylim = c(-2, 2),
         ref_ovl = FALSE,
         main = 'delta - ALL YEARS')
```

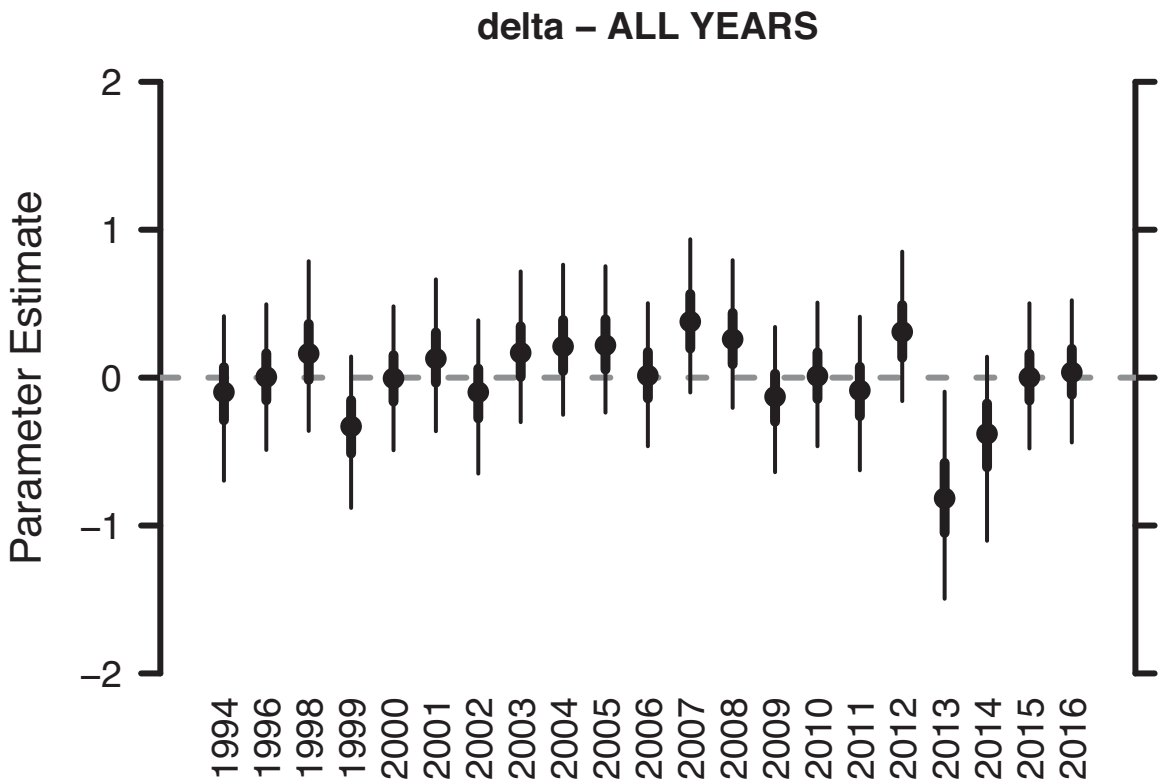

Figure A8-15: Parameter estimates for δ parameters - year effect for all years (1994, 1996, 1998-2016)

I - synchrony indices

```
MCMCplot(out_LM, params = 'I',
         excl = 'I_bar',
         ISB = FALSE,
         ref = NULL,
         labels = c('ADPE', 'EMPE',
                    'SOFU', 'CAPE',
                    'SNPE', 'SPSK'),
         xlim = c(0, 1),
         ref_ovl = FALSE,
         main = 'I - ALL YEARS')
```

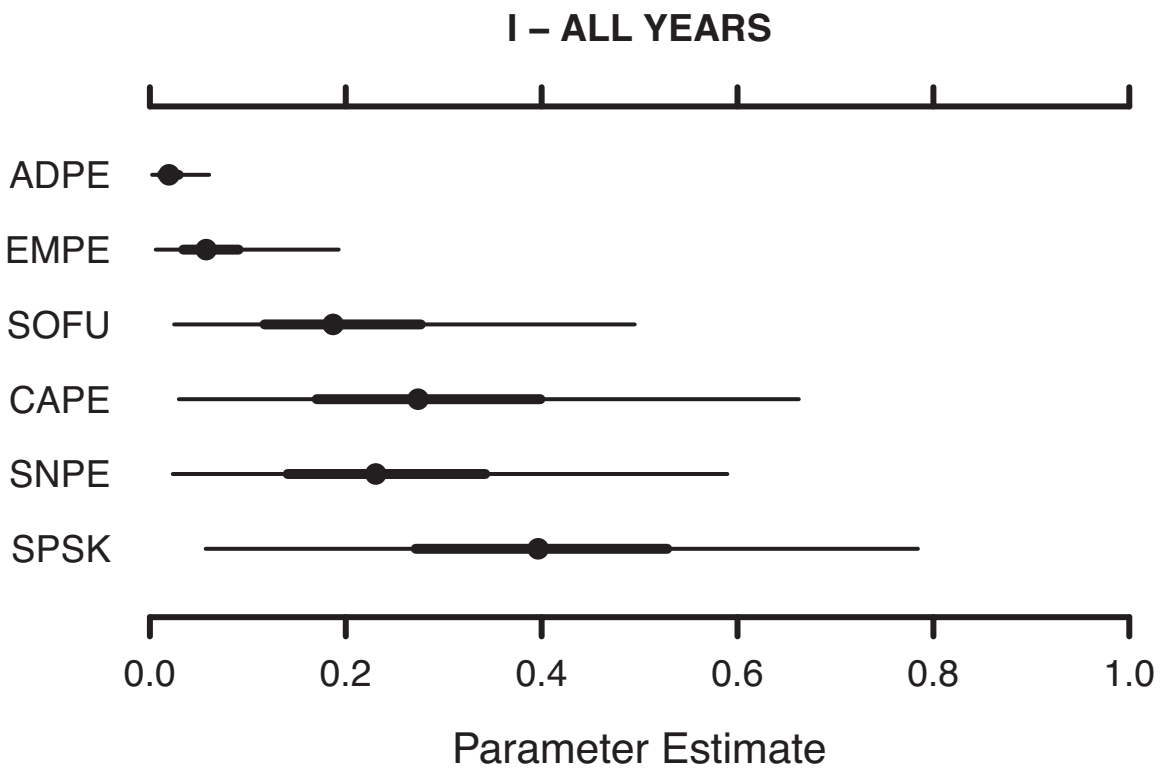

Figure A8-16: Parameter estimates for *I* parameters - synchrony index for all years (1994, 1996, 1998-2016). ADPE – Adélie penguin; EMPE – emperor penguin; SOFU – southern fulmar; CAPE – cape petrel; SNPE – snow petrel; SPSK – south polar skua.

Model without identified extreme year (2013)

```
#exclude 2013
ADPE_ne <- filter(data, SPECIES == 'ADPE' & YEAR != 2013)
EMPE_ne <- filter(data, SPECIES == 'EMPE' & YEAR != 2013)
SOFU_ne <- filter(data, SPECIES == 'SOFU' & YEAR != 2013)
CAPE_ne <- filter(data, SPECIES == 'CAPE' & YEAR != 2013)
SNPE_ne <- filter(data, SPECIES == 'SNPE' & YEAR != 2013)
SPSK_ne <- filter(data, SPECIES == 'SPSK' & YEAR != 2013)


#sort for complete years only
BS_mrg_ne <- data.frame(YEAR = ADPE_ne$YEAR,
                        ADPE_BS = ADPE_ne$BS,
                        EMPE_BS = EMPE_ne$BS,
                        SOFU_BS = SOFU_ne$BS,
                        CAPE_BS = CAPE_ne$BS,
                        SNPE_BS = SNPE_ne$BS,
                        SPSK_BS = SPSK_ne$BS)

#which rows have no NAs (data for every species)
BS_ind_ne <- which(table(which(!is.na(BS_mrg_ne), arr.ind=TRUE)[,1]) == 7)
```

```
ADPE_f_ne <- ADPE_ne[BS_ind_ne, ]
EMPE_f_ne <- EMPE_ne[BS_ind_ne, ]
SOFU_f_ne <- SOFU_ne[BS_ind_ne, ]
CAPE_f_ne <- CAPE_ne[BS_ind_ne, ]
SNPE_f_ne <- SNPE_ne[BS_ind_ne, ]
SPSK_f_ne <- SPSK_ne[BS_ind_ne, ]


#Data for JAGS model

LM_DATA_ne <- list(
  E1 = ADPE_f_ne$ABUN * 2, #two eggs
  E2 = EMPE_f_ne$ABUN, #one egg
  E3 = SOFU_f_ne$ABUN, #one egg
  E4 = CAPE_f_ne$ABUN, #one egg
  E5 = SNPE_f_ne$ABUN, #one egg
  E6 = SPSK_f_ne$ABUN * 2, #two eggs
  F1 = ADPE_f_ne$CHICKS,
  F2 = EMPE_f_ne$CHICKS,
  F3 = SOFU_f_ne$CHICKS,
  F4 = CAPE_f_ne$CHICKS,
  F5 = SNPE_f_ne$CHICKS,
  F6 = SPSK_f_ne$CHICKS,
  N = NROW(ADPE_f_ne))


DATA <- LM_DATA_ne
```

```
#SAME MODEL USED IN PRIOR MODEL

# Starting values ------------------------------------------------------


Inits_1 <- list(b0 = rep(0, 6),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Mersenne-Twister",
                .RNG.seed = 1)

Inits_2 <- list(b0 = rep(0, 6),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Wichmann-Hill",
                .RNG.seed = 2)

Inits_3 <- list(b0 = rep(0, 6),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Marsaglia-Multicarry",
                .RNG.seed = 3)
```

```
F_Inits <- list(Inits_1, Inits_2, Inits_3)


# Inputs for MCMC ---------------------------------------------------------


JAGS_FILE <- 'LM_model.jags'
n_adapt <- 8000  # number for initial adapt
n_burn <- 30000000 # number burnin
n_draw <- 100000000  # number of final draws to make
n_thin <- 500    # thinning rate
n_chain <- 3  # number of chains


# Run model (parallel) ----------------------------------------------------------------

#number of chains
cl <- parallel::makeCluster(n_chain)

pid <- NA
for(i in 1:n_chain)
{
  pidNum <- capture.output(cl[[i]])
  start <- regexpr("pid", pidNum)[[1]]
  end <- nchar(pidNum)
  pid[i] <- substr(pidNum, (start + 4), end)
}

parallel::clusterExport(cl,
                        c('DATA',
                          'n_adapt',
                          'n_burn',
                          'n_draw',
                          'n_thin',
                          'Pars',
                          'pid',
                          'F_Inits',
                          'JAGS_FILE'
                        ))


out.1 <- parallel::clusterEvalQ(cl,
                                {
                                  require(rjags)
                                  processNum <- which(pid==Sys.getpid())
                                  m.inits <- F_Inits[[processNum]]

                                  jm = jags.model(data = DATA,
                                                  file = paste0(JAGS_FILE),
                                                  inits = m.inits,
                                                  n.chains = 1,
                                                  n.adapt = n_adapt)
```

```
                                update(jm,
                                       n.iter = n_burn)

                                samples = coda.samples(jm,
                                                       n.iter = n_draw,
                                                       variable.names = Pars,
                                                       thin = n_thin)
                                return(samples)
                              })


out_LM_ne <- coda::mcmc.list(out.1[[1]][[1]],
                        out.1[[2]][[1]],
                        out.1[[3]][[1]])

invisible(file.remove(JAGS_FILE))
```

Posterior summary

```
MCMCsummary(out_LM_ne, round = 3)
```

```
##            mean    sd   2.5%    50%  97.5% Rhat
## I[1]      0.007 0.009  0.000  0.004  0.032    1
## I[2]      0.015 0.020  0.000  0.008  0.070    1
## I[3]      0.063 0.074  0.000  0.038  0.265    1
## I[4]      0.074 0.090  0.000  0.042  0.324    1
## I[5]      0.058 0.075  0.000  0.031  0.269    1
## I[6]      0.156 0.160  0.000  0.105  0.569    1
## I_bar     0.062 0.065  0.000  0.041  0.229    1
## b0[1]    -1.078 0.530 -2.138 -1.072 -0.039    1
## b0[2]     0.219 0.366 -0.508  0.220  0.944    1
## b0[3]     0.816 0.196  0.436  0.812  1.215    1
## b0[4]     0.807 0.165  0.480  0.806  1.135    1
## b0[5]    -0.094 0.193 -0.479 -0.093  0.289    1
## b0[6]    -0.693 0.117 -0.928 -0.692 -0.462    1
## d[1]     -0.030 0.160 -0.407 -0.011  0.277    1
## d[2]     -0.015 0.148 -0.344 -0.005  0.292    1
## d[3]      0.025 0.161 -0.294  0.008  0.397    1
## d[4]     -0.144 0.193 -0.623 -0.091  0.117    1
## d[5]     -0.017 0.147 -0.347 -0.006  0.285    1
## d[6]      0.033 0.155 -0.267  0.013  0.395    1
## d[7]     -0.039 0.155 -0.404 -0.016  0.252    1
## d[8]      0.043 0.154 -0.247  0.018  0.408    1
## d[9]      0.057 0.156 -0.221  0.026  0.434    1
## d[10]     0.069 0.157 -0.201  0.035  0.451    1
## d[11]    -0.008 0.147 -0.331 -0.002  0.300    1
## d[12]     0.134 0.186 -0.124  0.084  0.593    1
## d[13]     0.091 0.166 -0.173  0.051  0.497    1
## d[14]    -0.070 0.159 -0.454 -0.036  0.203    1
## d[15]    -0.019 0.147 -0.348 -0.007  0.283    1
## d[16]    -0.049 0.156 -0.418 -0.022  0.238    1
## d[17]     0.107 0.172 -0.153  0.062  0.532    1
## d[18]    -0.140 0.203 -0.657 -0.081  0.131    1
```

```
## d[19]    -0.022 0.148 -0.354 -0.008  0.280    1
## d[20]    -0.010 0.146 -0.331 -0.003  0.294    1
## var.d     0.037 0.045  0.000  0.021  0.159    1
## var.e[1]  5.539 1.574  2.858  5.405  8.648    1
## var.e[2]  2.646 0.990  1.328  2.444  5.168    1
## var.e[3]  0.591 0.328  0.188  0.518  1.423    1
## var.e[4]  0.499 0.206  0.225  0.458  1.011    1
## var.e[5]  0.689 0.290  0.304  0.632  1.410    1
## var.e[6]  0.197 0.105  0.060  0.176  0.459    1
```

δ - community-wide year effect

```
yrs <- ADPE_f_ne$YEAR
MCMCplot(out_LM_ne,
         params = 'd',
         horiz = FALSE,
         labels = yrs,
         ylim = c(-2, 2),
         ref_ovl = FALSE,
         main = 'delta - NO EXTREME YEAR')
```

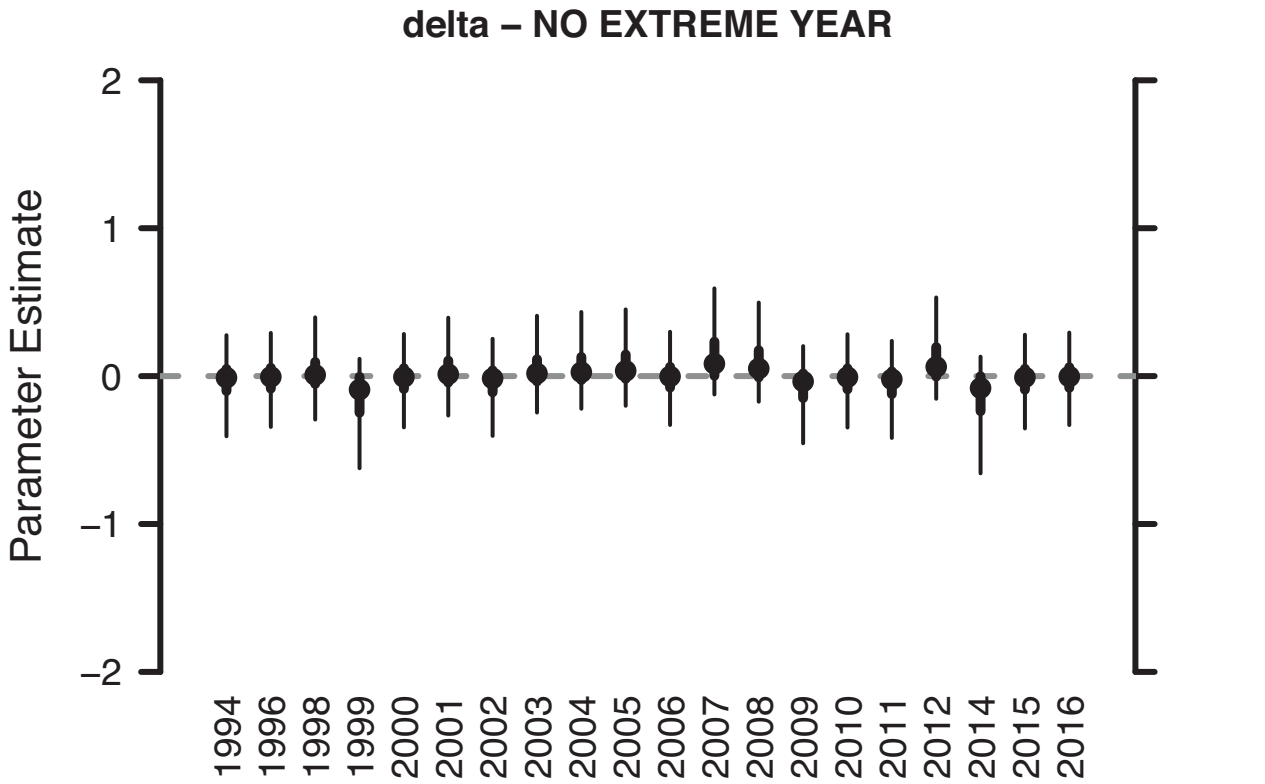

Figure A8-17: Parameter estimates for δ parameters - year effect for all years except 2013 (1994, 1996, 1998-2012, 2014-2016)

I - synchrony indices

```
MCMCplot(out_LM_ne, params = 'I',
         excl = 'I_bar',
```

```
        ISB = FALSE,
        ref = NULL,
        labels = c('ADPE', 'EMPE',
                   'SOFU', 'CAPE',
                   'SNPE', 'SPSK'),
        xlim = c(0, 1),
        ref_ovl = FALSE,
        main = 'I - NO EXTREME YEAR')
```

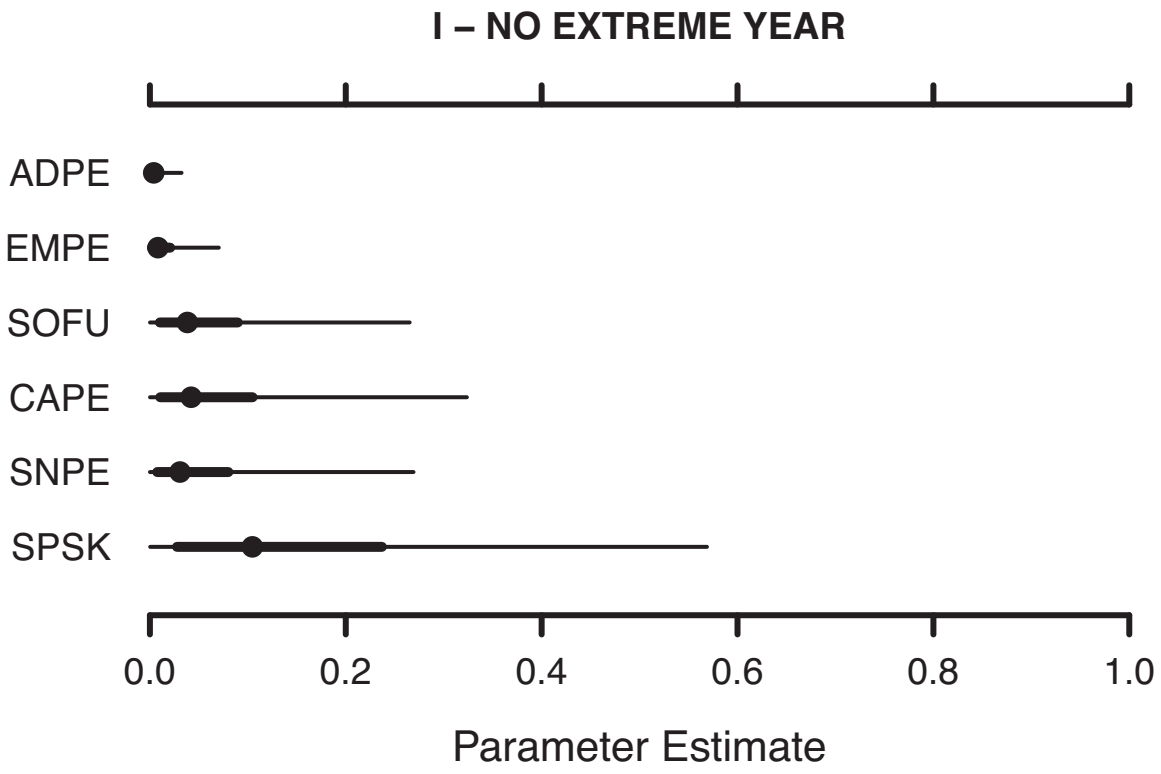

Figure A8-18: Parameter estimates for I parameters - synchrony index for all years except 2013 (1994, 1996, 1998-2012, 2014-2016)

Effect of environmental factors on community synchrony

$$F_{ij} \sim Binom(E_{ij}, p_{cov_{ij}}) \tag{A8-3}$$

$$logit(p_{cov_{ij}}) = \mu_{cov_j} + \sum_{k=1}^{4} (\beta_{k_j} * c_{k_i}) + \delta_{cov_i} + \epsilon_{cov_{ij}}$$

$$\beta_{k_j} \sim N(0, \sigma^2_{\beta_k})$$
$$\delta_{cov_i} \sim N(0, \sigma^2_{\delta_{cov}})$$
$$\epsilon_{cov_{ij}} \sim N(0, \sigma^2_{\epsilon_{cov_j}})$$

L-H model with SIC

```
#load SIC data

setwd('Data/SIC')


SIC <- read.csv('SIC_150.csv')
```

```
#center data
SIC_SMN_sc <- scale(SIC$SMN, scale = FALSE)
SIC_WMN_sc <- scale(SIC$WMN, scale = FALSE)

SIC_data <- data.frame(YEAR = SIC$YEAR,
                       SIC_S = SIC_SMN_sc,
                       SIC_S_L1 = c(NA, SIC_SMN_sc[-NROW(SIC_SMN_sc)]),
                       SIC_W = SIC_WMN_sc,
                       SIC_W_L1 = c(NA, SIC_WMN_sc[-NROW(SIC_WMN_sc)]))

SIC_mrg <- left_join(ADPE_f, SIC_data, by = 'YEAR')


#  BS as a function of time - DATA ------------------------------------------------

DATA_cov <- list(
  E1 = ADPE_f$ABUN * 2, #two eggs
  E2 = EMPE_f$ABUN, #one egg
  E3 = SOFU_f$ABUN, #one egg
  E4 = CAPE_f$ABUN, #one egg
  E5 = SNPE_f$ABUN, #one egg
  E6 = SPSK_f$ABUN * 2, #two eggs
  F1 = ADPE_f$CHICKS,
  F2 = EMPE_f$CHICKS,
  F3 = SOFU_f$CHICKS,
  F4 = CAPE_f$CHICKS,
  F5 = SNPE_f$CHICKS,
  F6 = SPSK_f$CHICKS,
  N = NROW(ADPE_f),
  #Summer SIC
  SIC_S = SIC_mrg$SIC_S,
  #Summer SIC - Lag 1
  SIC_S_L1 = SIC_mrg$SIC_S_L1,
  #Winter SIC
  SIC_W = SIC_mrg$SIC_W,
  #Winter SIC - Lag 1
  SIC_W_L1 = SIC_mrg$SIC_W_L1
)

DATA <- DATA_cov
```

```
{
sink("LM_model_cov.jags")

cat("
```

```
model {

# Likelihood -------------------------------------------------------------

for (i in 1:N)
{
F1[i] ~ dbin(rho[1,i], E1[i])
F2[i] ~ dbin(rho[2,i], E2[i])
F3[i] ~ dbin(rho[3,i], E3[i])
F4[i] ~ dbin(rho[4,i], E4[i])
F5[i] ~ dbin(rho[5,i], E5[i])
F6[i] ~ dbin(rho[6,i], E6[i])
# where Es[i] = # eggs laid for that species * number of breeding pairs in year i
}

# Priors and regressions -------------------------------------------------
for (i in 1:N)
{
# Logistic regressions on productivity (with random terms)
#bs are offsets for each species

logit(rho[1,i]) <- b0[1] + b1[1] * SIC_S[i] +
b2[1] * SIC_S_L1[i] +
b3[1] * SIC_W[i] +
b4[1] * SIC_W_L1[i] +
d[i] + e[1,i]

logit(rho[2,i]) <- b0[2] + b1[2] * SIC_S[i] +
b2[2] * SIC_S_L1[i] +
b3[2] * SIC_W[i] +
b4[2] * SIC_W_L1[i] +
d[i] + e[2,i]

logit(rho[3,i]) <- b0[3] + b1[3] * SIC_S[i] +
b2[3] * SIC_S_L1[i] +
b3[3] * SIC_W[i] +
b4[4] * SIC_W_L1[i] +
d[i] + e[3,i]

logit(rho[4,i]) <- b0[4] + b1[4] * SIC_S[i] +
b2[4] * SIC_S_L1[i] +
b3[4] * SIC_W[i] +
b4[4] * SIC_W_L1[i] +
d[i] + e[4,i]

logit(rho[5,i]) <- b0[5] + b1[5] * SIC_S[i] +
b2[5] * SIC_S_L1[i] +
b3[5] * SIC_W[i] +
b4[5] * SIC_W_L1[i] +
d[i] + e[5,i]

logit(rho[6,i]) <- b0[6] + b1[6] * SIC_S[i] +
b2[6] * SIC_S_L1[i] +
```

```
    b3[6] * SIC_W[i] +
    b4[6] * SIC_W_L1[i] +
    d[i] + e[6,i]


    #priors
    d[i] ~ dnorm(0, tau.d)

    for (j in 1:6)
    {

    e[j, i] ~ dnorm(0, tau.e[j])
    }
    }

    tau.d  <- 1/var.d
    var.d <- sd.d * sd.d
    sd.d ~ dunif(0, 3)

    for (j in 1:6)
    {
    b0[j] ~ dunif(-5,5)
    b1[j] ~ dnorm(0, tau.b1)
    b2[j] ~ dnorm(0, tau.b2)
    b3[j] ~ dnorm(0, tau.b3)
    b4[j] ~ dnorm(0, tau.b4)

    tau.e[j] <- 1/var.e[j]
    var.e[j] <- sd.e[j] * sd.e[j]
    sd.e[j] ~ dunif(0,3)

    # Synchrony indices
    I[j] <- var.d/(var.e[j] + var.d)
    }

    tau.b1 <- pow(sd.b1, -2)
    tau.b2 <- pow(sd.b2, -2)
    tau.b3 <- pow(sd.b3, -2)
    tau.b4 <- pow(sd.b4, -2)

    sd.b1 ~ dunif(0, 3)
    sd.b2 ~ dunif(0, 3)
    sd.b3 ~ dunif(0, 3)
    sd.b4 ~ dunif(0, 3)


    }",fill = TRUE)

sink()
}


# Starting values -------------------------------------------------------
```

```
Inits_1 <- list(b0 = rep(0, 6),
                b1 = rep(0, 6),
                b2 = rep(0, 6),
                b3 = rep(0, 6),
                b4 = rep(0, 6),
                sd.b1 = runif(1),
                sd.b2 = runif(1),
                sd.b3 = runif(1),
                sd.b4 = runif(1),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Mersenne-Twister",
                .RNG.seed = 1)

Inits_2 <- list(b0 = rep(0, 6),
                b1 = rep(0, 6),
                b2 = rep(0, 6),
                b3 = rep(0, 6),
                b4 = rep(0, 6),
                sd.b1 = runif(1),
                sd.b2 = runif(1),
                sd.b3 = runif(1),
                sd.b4 = runif(1),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Wichmann-Hill",
                .RNG.seed = 2)

Inits_3 <- list(b0 = rep(0, 6),
                b1 = rep(0, 6),
                b2 = rep(0, 6),
                b3 = rep(0, 6),
                b4 = rep(0, 6),
                sd.b1 = runif(1),
                sd.b2 = runif(1),
                sd.b3 = runif(1),
                sd.b4 = runif(1),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Marsaglia-Multicarry",
                .RNG.seed = 3)

F_Inits <- list(Inits_1, Inits_2, Inits_3)



# Parameters to track -------------------------------------------------------

Pars <- c('b0',
```

```
          'b1',
          'b2',
          'b3',
          'b4',
          'd',
          'var.d',
          'var.e',
          'I')


# Inputs for MCMC -----------------------------------------------------------

JAGS_FILE <- 'LM_model_cov.jags'
n_adapt <- 8000  # number for initial adapt
n_burn <- 30000000 # number burnin
n_draw <- 100000000  # number of final draws to make
n_thin <- 500    # thinning rate
n_chain <- 3  # number of chains


# Run model (parallel) ------------------------------------------------------------------

#number of chains
cl <- parallel::makeCluster(n_chain)

pid <- NA
for(i in 1:n_chain)
{
  pidNum <- capture.output(cl[[i]])
  start <- regexpr("pid", pidNum)[[1]]
  end <- nchar(pidNum)
  pid[i] <- substr(pidNum, (start + 4), end)
}

parallel::clusterExport(cl,
                        c('DATA',
                          'n_adapt',
                          'n_burn',
                          'n_draw',
                          'n_thin',
                          'Pars',
                          'pid',
                          'F_Inits',
                          'JAGS_FILE'
                        ))


ptm <- proc.time()
out.1 <- parallel::clusterEvalQ(cl,
                                {
                                  require(rjags)
                                  processNum <- which(pid==Sys.getpid())
                                  m.inits <- F_Inits[[processNum]]
```

```
                                  jm = jags.model(data = DATA,
                                                  file = paste0(JAGS_FILE),
                                                  inits = m.inits,
                                                  n.chains = 1,
                                                  n.adapt = n_adapt)

                                  update(jm,
                                         n.iter = n_burn)

                                  samples = coda.samples(jm,
                                                         n.iter = n_draw,
                                                         variable.names = Pars,
                                                         thin = n_thin)
                                  return(samples)
                                })


out_cov <- coda::mcmc.list(out.1[[1]][[1]],
                           out.1[[2]][[1]],
                           out.1[[3]][[1]])

invisible(file.remove(JAGS_FILE))
```

Posterior summary

```
MCMCsummary(out_cov, round = 3)
```

```
##            mean    sd   2.5%    50%  97.5% Rhat
## I[1]      0.025 0.022  0.001  0.019  0.082 1.00
## I[2]      0.054 0.045  0.001  0.044  0.169 1.00
## I[3]      0.165 0.120  0.004  0.141  0.463 1.00
## I[4]      0.235 0.157  0.006  0.211  0.598 1.00
## I[5]      0.194 0.139  0.004  0.168  0.527 1.00
## I[6]      0.334 0.201  0.010  0.316  0.773 1.00
## b0[1]    -1.236 0.581 -2.429 -1.208 -0.125 1.02
## b0[2]     0.195 0.360 -0.515  0.194  0.919 1.00
## b0[3]     0.766 0.224  0.332  0.762  1.224 1.00
## b0[4]     0.771 0.170  0.436  0.770  1.111 1.00
## b0[5]    -0.112 0.191 -0.490 -0.112  0.265 1.00
## b0[6]    -0.774 0.148 -1.071 -0.773 -0.482 1.00
## b1[1]    -0.057 0.038 -0.125 -0.061  0.002 1.01
## b1[2]    -0.014 0.016 -0.047 -0.013  0.013 1.00
## b1[3]     0.002 0.010 -0.018  0.002  0.023 1.00
## b1[4]    -0.004 0.009 -0.022 -0.004  0.013 1.00
## b1[5]    -0.006 0.009 -0.026 -0.006  0.011 1.00
## b1[6]     0.002 0.007 -0.013  0.001  0.017 1.00
## b2[1]    -0.010 0.019 -0.062 -0.004  0.014 1.00
## b2[2]    -0.007 0.013 -0.039 -0.003  0.013 1.00
## b2[3]     0.002 0.009 -0.016  0.001  0.023 1.00
## b2[4]    -0.008 0.009 -0.029 -0.007  0.005 1.00
## b2[5]    -0.002 0.008 -0.020 -0.001  0.014 1.00
## b2[6]    -0.006 0.008 -0.023 -0.005  0.007 1.00
## b3[1]    -0.022 0.069 -0.206 -0.006  0.087 1.00
```

```
## b3[2]    -0.016 0.055 -0.154 -0.005  0.082 1.00
## b3[3]    -0.028 0.049 -0.149 -0.015  0.046 1.00
## b3[4]    -0.002 0.037 -0.084 -0.001  0.077 1.00
## b3[5]     0.014 0.041 -0.061  0.007  0.112 1.00
## b3[6]     0.017 0.036 -0.046  0.010  0.101 1.00
## b4[1]     0.026 0.081 -0.104  0.008  0.240 1.00
## b4[2]    -0.016 0.066 -0.180 -0.005  0.105 1.00
## b4[3]     0.000 0.101 -0.200  0.000  0.199 1.00
## b4[4]    -0.014 0.040 -0.107 -0.008  0.059 1.00
## b4[5]    -0.042 0.058 -0.182 -0.026  0.039 1.00
## b4[6]     0.001 0.039 -0.084  0.001  0.083 1.00
## d[1]     -0.099 0.262 -0.661 -0.078  0.400 1.00
## d[2]     -0.024 0.234 -0.503 -0.018  0.443 1.00
## d[3]      0.120 0.269 -0.384  0.096  0.701 1.00
## d[4]     -0.370 0.279 -0.958 -0.353  0.094 1.00
## d[5]     -0.064 0.229 -0.536 -0.053  0.386 1.00
## d[6]      0.066 0.241 -0.405  0.053  0.567 1.00
## d[7]     -0.113 0.250 -0.645 -0.095  0.360 1.00
## d[8]      0.086 0.239 -0.371  0.070  0.593 1.00
## d[9]      0.163 0.245 -0.282  0.141  0.692 1.00
## d[10]     0.118 0.240 -0.334  0.100  0.628 1.00
## d[11]    -0.026 0.235 -0.507 -0.020  0.447 1.00
## d[12]     0.249 0.259 -0.200  0.227  0.802 1.00
## d[13]     0.221 0.247 -0.217  0.202  0.745 1.00
## d[14]    -0.126 0.235 -0.617 -0.110  0.320 1.00
## d[15]     0.007 0.230 -0.457  0.005  0.476 1.00
## d[16]    -0.024 0.251 -0.539 -0.017  0.481 1.00
## d[17]     0.392 0.297 -0.092  0.371  1.027 1.00
## d[18]    -0.532 0.362 -1.311 -0.505  0.038 1.00
## d[19]    -0.169 0.301 -0.838 -0.134  0.368 1.00
## d[20]     0.041 0.245 -0.444  0.032  0.551 1.00
## d[21]     0.032 0.269 -0.507  0.022  0.590 1.00
## var.d     0.125 0.097  0.003  0.105  0.366 1.00
## var.e[1]  5.551 1.788  2.477  5.454  8.763 1.00
## var.e[2]  2.410 0.929  1.177  2.222  4.753 1.00
## var.e[3]  0.711 0.417  0.205  0.618  1.769 1.00
## var.e[4]  0.422 0.198  0.158  0.383  0.912 1.00
## var.e[5]  0.564 0.262  0.219  0.511  1.217 1.00
## var.e[6]  0.254 0.150  0.055  0.224  0.628 1.00
```

β parameters

```
#beta params
SP <- c('ADPE', 'EMPE', 'SOFU', 'CAPE', 'SNPE', 'SPSK')
LGS <- c('S_L0', 'S_L1', 'W_L0', 'W_L1')
L_SIC <- c()
for (k in 1:4)
{
  for (j in 1:6)
  {
    L_SIC <- c(L_SIC, paste0('SIC_', LGS[k], '_', SP[j]))
  }
}
```

```
LABS <- L_SIC

MCMCplot(out_cov, params = c('b1', 'b2', 'b3', 'b4'),
         ISB = FALSE,
         ylim = c(-0.2, 0.2),
         labels = LABS,
         thin_sz = 1.5,
         thick_sz = 3, med_sz = 1.2,
         horiz = FALSE,
         ref_ovl = FALSE,
         main = expression(paste(beta, ' - Effect of covariates on Breeding Success')))
```

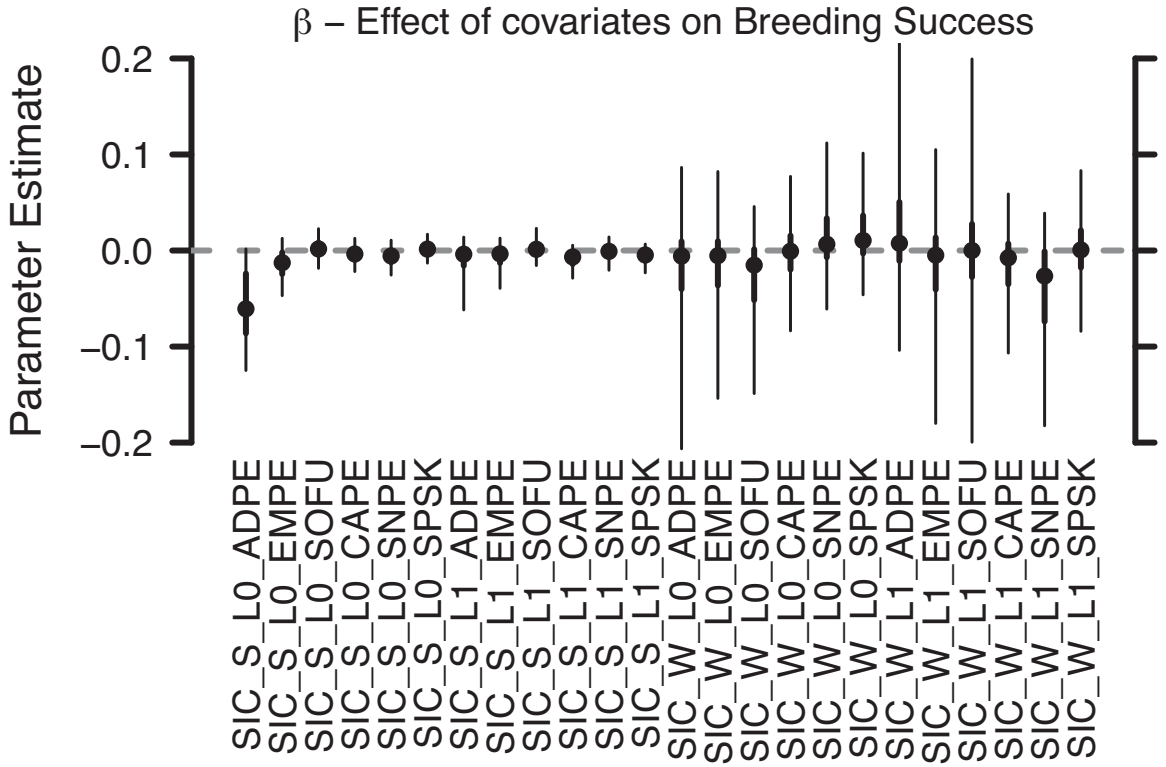

Figure A8-19: Posterior estimates for β parameters - the effect of unlagged (L0) and lagged one year (L1) summer (S) and winter (W) sea ice concentration (SIC) on the breeding productivity of Adélie penguin (ADPE), emperor penguin (EMPE), southern fulmar (SOFU), cape petrel (CAPE), snow petrel (SNPE), south polar skua (SPSK). Data from all years were used (1994, 1996, 1998-2016).

Proportion of synchronous variation explained by the covariates

$$C_\delta = 1 - \frac{\widehat{\sigma^2_{\delta_{cov}}}}{\widehat{\sigma^2_\delta}} \tag{A8-4}$$

```
NUM <- MCMCsummary(out_cov, params = 'var.d')[,4][1]
DEN <- MCMCsummary(out_LM, params = 'var.d')[,4][1]

(C_d <- 1 - (NUM/DEN)) #degree of synchonous variation explained by covariates
```

```
## [1] 0.2758756
```

L-H model with SIC without identified extreme year (2013)

A model identical to the covaraite model above was fit, except that it excluded the identified extreme year (2013).

```
SIC_mrg_ne <- left_join(ADPE_f_ne, SIC_data, by = 'YEAR')


#  BS as a function of time - DATA ------------------------------------------------

DATA_cov <- list(
  E1 = ADPE_f_ne$ABUN * 2, #two eggs
  E2 = EMPE_f_ne$ABUN, #one egg
  E3 = SOFU_f_ne$ABUN, #one egg
  E4 = CAPE_f_ne$ABUN, #one egg
  E5 = SNPE_f_ne$ABUN, #one egg
  E6 = SPSK_f_ne$ABUN * 2, #two eggs
  F1 = ADPE_f_ne$CHICKS,
  F2 = EMPE_f_ne$CHICKS,
  F3 = SOFU_f_ne$CHICKS,
  F4 = CAPE_f_ne$CHICKS,
  F5 = SNPE_f_ne$CHICKS,
  F6 = SPSK_f_ne$CHICKS,
  N = NROW(ADPE_f_ne),
  #Summer SIC
  SIC_S = SIC_mrg_ne$SIC_S,
  #Summer SIC - Lag 1
  SIC_S_L1 = SIC_mrg_ne$SIC_S_L1,
  #Winter SIC
  SIC_W = SIC_mrg_ne$SIC_W,
  #Winter SIC - Lag 1
  SIC_W_L1 = SIC_mrg_ne$SIC_W_L1
)

DATA <- DATA_cov
```

```
# Starting values ----------------------------------------------------------


Inits_1 <- list(b0 = rep(0, 6),
                b1 = rep(0, 6),
                b2 = rep(0, 6),
                b3 = rep(0, 6),
                b4 = rep(0, 6),
                sd.b1 = runif(1),
                sd.b2 = runif(1),
                sd.b3 = runif(1),
                sd.b4 = runif(1),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Mersenne-Twister",
                .RNG.seed = 1)
```

```
Inits_2 <- list(b0 = rep(0, 6),
                b1 = rep(0, 6),
                b2 = rep(0, 6),
                b3 = rep(0, 6),
                b4 = rep(0, 6),
                sd.b1 = runif(1),
                sd.b2 = runif(1),
                sd.b3 = runif(1),
                sd.b4 = runif(1),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Wichmann-Hill",
                .RNG.seed = 2)

Inits_3 <- list(b0 = rep(0, 6),
                b1 = rep(0, 6),
                b2 = rep(0, 6),
                b3 = rep(0, 6),
                b4 = rep(0, 6),
                sd.b1 = runif(1),
                sd.b2 = runif(1),
                sd.b3 = runif(1),
                sd.b4 = runif(1),
                d = rnorm(DATA$N),
                e = matrix(rnorm(DATA$N*6), nrow = 6),
                sd.e = runif(6, 0, 2),
                .RNG.name = "base::Marsaglia-Multicarry",
                .RNG.seed = 3)

F_Inits <- list(Inits_1, Inits_2, Inits_3)



# Parameters to track -----------------------------------------------------

Pars <- c('b0',
          'b1',
          'b2',
          'b3',
          'b4',
          'd',
          'var.d',
          'var.e',
          'I')


# Inputs for MCMC ---------------------------------------------------------

JAGS_FILE <- 'LM_model_cov.jags'
n_adapt <- 8000  # number for initial adapt
n_burn <- 30000000 # number burnin
n_draw <- 100000000  # number of final draws to make
```

```
n_thin <- 500    # thinning rate
n_chain <- 3  # number of chains


# Run model (parallel) ------------------------------------------------------------

#number of chains
cl <- parallel::makeCluster(n_chain)

pid <- NA
for(i in 1:n_chain)
{
  pidNum <- capture.output(cl[[i]])
  start <- regexpr("pid", pidNum)[[1]]
  end <- nchar(pidNum)
  pid[i] <- substr(pidNum, (start + 4), end)
}

parallel::clusterExport(cl,
                        c('DATA',
                          'n_adapt',
                          'n_burn',
                          'n_draw',
                          'n_thin',
                          'Pars',
                          'pid',
                          'F_Inits',
                          'JAGS_FILE'
                        ))


ptm <- proc.time()
out.1 <- parallel::clusterEvalQ(cl,
                                {
                                  require(rjags)
                                  processNum <- which(pid==Sys.getpid())
                                  m.inits <- F_Inits[[processNum]]

                                  jm = jags.model(data = DATA,
                                                  file = paste0(JAGS_FILE),
                                                  inits = m.inits,
                                                  n.chains = 1,
                                                  n.adapt = n_adapt)

                                  update(jm,
                                         n.iter = n_burn)

                                  samples = coda.samples(jm,
                                                         n.iter = n_draw,
                                                         variable.names = Pars,
                                                         thin = n_thin)
                                  return(samples)
                                })
```

```
out_cov <- coda::mcmc.list(out.1[[1]][[1]],
                           out.1[[2]][[1]],
                           out.1[[3]][[1]])

invisible(file.remove(JAGS_FILE))
```

Posterior summary

```
MCMCsummary(out_cov_ne, round = 3)
```

```
##            mean    sd   2.5%    50%  97.5% Rhat
## I[1]      0.012 0.016  0.000  0.006  0.054 1.00
## I[2]      0.018 0.022  0.000  0.010  0.077 1.00
## I[3]      0.073 0.082  0.000  0.047  0.296 1.00
## I[4]      0.083 0.094  0.000  0.051  0.339 1.00
## I[5]      0.065 0.079  0.000  0.037  0.286 1.00
## I[6]      0.185 0.186  0.000  0.129  0.663 1.00
## b0[1]    -0.948 0.440 -1.810 -0.935 -0.064 1.02
## b0[2]     0.239 0.368 -0.483  0.240  0.968 1.00
## b0[3]     0.831 0.202  0.445  0.825  1.246 1.00
## b0[4]     0.799 0.169  0.466  0.798  1.137 1.00
## b0[5]    -0.101 0.198 -0.494 -0.100  0.289 1.00
## b0[6]    -0.717 0.120 -0.958 -0.717 -0.481 1.00
## b1[1]    -0.029 0.027 -0.086 -0.023  0.004 1.00
## b1[2]    -0.008 0.013 -0.038 -0.005  0.015 1.00
## b1[3]     0.006 0.009 -0.010  0.004  0.025 1.00
## b1[4]    -0.002 0.008 -0.018 -0.001  0.013 1.00
## b1[5]    -0.004 0.008 -0.023 -0.003  0.012 1.00
## b1[6]     0.004 0.006 -0.007  0.004  0.017 1.00
## b2[1]    -0.002 0.012 -0.030  0.000  0.021 1.00
## b2[2]    -0.004 0.011 -0.032 -0.002  0.015 1.00
## b2[3]     0.007 0.010 -0.007  0.004  0.032 1.00
## b2[4]    -0.006 0.008 -0.026 -0.004  0.007 1.00
## b2[5]    -0.002 0.008 -0.020 -0.001  0.014 1.00
## b2[6]    -0.001 0.006 -0.015 -0.001  0.011 1.00
## b3[1]    -0.021 0.061 -0.180 -0.007  0.079 1.00
## b3[2]    -0.015 0.053 -0.145 -0.005  0.080 1.00
## b3[3]    -0.024 0.044 -0.131 -0.014  0.048 1.00
## b3[4]    -0.001 0.036 -0.079  0.000  0.077 1.00
## b3[5]     0.011 0.040 -0.064  0.005  0.106 1.00
## b3[6]     0.026 0.034 -0.029  0.019  0.102 1.00
## b4[1]     0.032 0.079 -0.085  0.010  0.244 1.00
## b4[2]    -0.015 0.065 -0.177 -0.004  0.104 1.00
## b4[3]     0.000 0.097 -0.194  0.000  0.195 1.00
## b4[4]    -0.015 0.037 -0.100 -0.008  0.054 1.00
## b4[5]    -0.040 0.057 -0.180 -0.023  0.041 1.00
## b4[6]    -0.002 0.034 -0.075  0.000  0.068 1.00
## d[1]     -0.043 0.170 -0.444 -0.018  0.275 1.00
## d[2]     -0.033 0.158 -0.393 -0.015  0.280 1.00
## d[3]      0.034 0.172 -0.301  0.013  0.432 1.00
## d[4]     -0.167 0.206 -0.666 -0.115  0.112 1.00
## d[5]     -0.022 0.154 -0.367 -0.009  0.290 1.00
```

```
## d[6]      0.034 0.163 -0.284  0.014  0.408 1.00
## d[7]     -0.023 0.164 -0.394 -0.009  0.310 1.00
## d[8]      0.034 0.160 -0.280  0.014  0.400 1.00
## d[9]      0.076 0.168 -0.211  0.041  0.477 1.00
## d[10]     0.063 0.164 -0.230  0.033  0.451 1.00
## d[11]     0.006 0.156 -0.324  0.002  0.343 1.00
## d[12]     0.133 0.189 -0.145  0.085  0.594 1.00
## d[13]     0.110 0.178 -0.169  0.068  0.541 1.00
## d[14]    -0.077 0.168 -0.478 -0.042  0.213 1.00
## d[15]    -0.026 0.158 -0.381 -0.011  0.291 1.00
## d[16]    -0.051 0.168 -0.446 -0.023  0.263 1.00
## d[17]     0.120 0.189 -0.169  0.073  0.581 1.00
## d[18]    -0.117 0.201 -0.621 -0.065  0.188 1.00
## d[19]    -0.012 0.159 -0.358 -0.004  0.322 1.00
## d[20]    -0.062 0.178 -0.487 -0.029  0.258 1.00
## var.d     0.041 0.049  0.000  0.026  0.171 1.00
## var.e[1]  4.197 1.722  1.646  3.908  8.160 1.00
## var.e[2]  2.605 1.000  1.284  2.398  5.157 1.00
## var.e[3]  0.579 0.334  0.174  0.504  1.425 1.00
## var.e[4]  0.495 0.215  0.215  0.452  1.031 1.00
## var.e[5]  0.685 0.302  0.291  0.624  1.437 1.00
## var.e[6]  0.188 0.109  0.045  0.166  0.462 1.00
```

β parameters when excluding extreme year

```
#beta params
SP <- c('ADPE', 'EMPE', 'SOFU', 'CAPE', 'SNPE', 'SPSK')
LGS <- c('S_L0', 'S_L1', 'W_L0', 'W_L1')
L_SIC <- c()
for (k in 1:4)
{
  for (j in 1:6)
  {
    L_SIC <- c(L_SIC, paste0('SIC_', LGS[k], '_', SP[j]))
  }
}

LABS <- L_SIC

MCMCplot(out_cov_ne, params = c('b1', 'b2', 'b3', 'b4'),
         ISB = FALSE,
         ylim = c(-0.2, 0.2),
         labels = LABS,
         thin_sz = 1.5,
         thick_sz = 3, med_sz = 1.2,
         horiz = FALSE,
         ref_ovl = FALSE,
         main = expression(paste(beta, ' - Effect of covariates on Breeding Success - NO EX')))
```

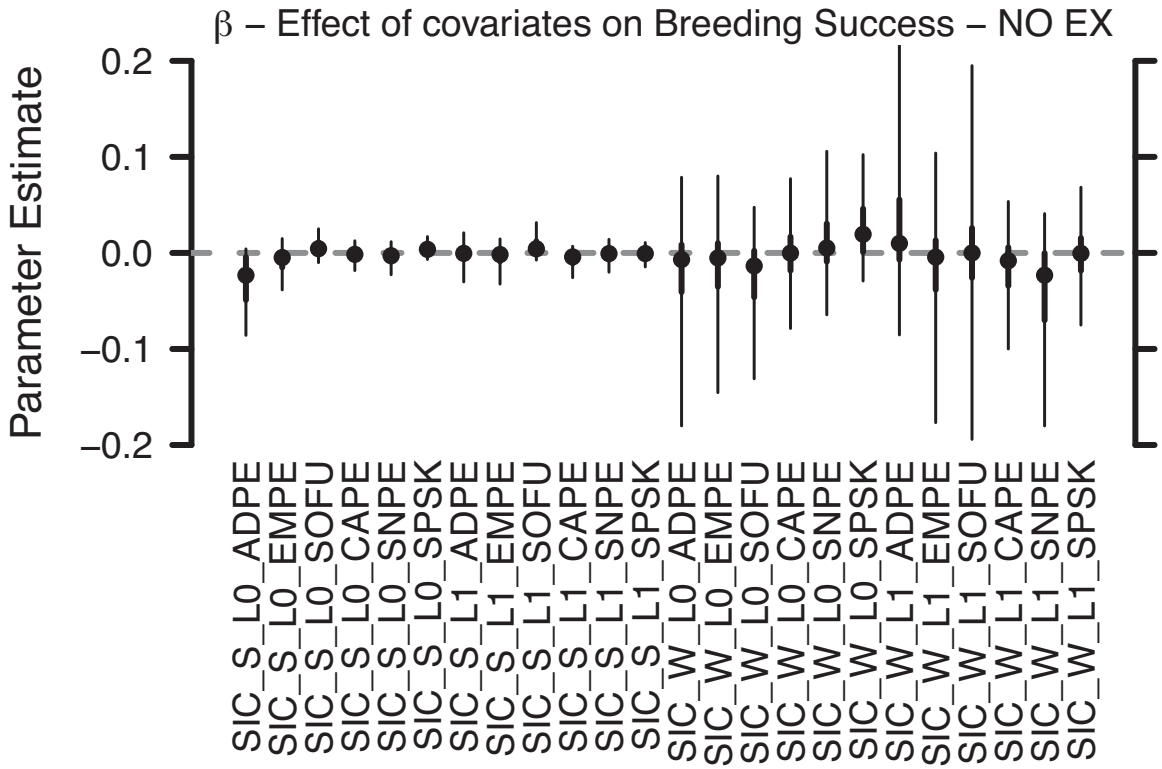

Figure A8-20: Posterior estimates for β parameters - the effect of unlagged (L0) and lagged one year (L1) summer (S) and winter (W) sea ice concentration (SIC) on the breeding productivity of Adélie penguin (ADPE), emperor penguin (EMPE), southern fulmar (SOFU), cape petrel (CAPE), snow petrel (SNPE), south polar skua (SPSK). Data from all years except 2013 were used (1994, 1996, 1998-2012, 2014-2016).

Proportion of synchronous variation explained by the covariates without extreme year

```
NUM_ne <- MCMCsummary(out_cov_ne, params = 'var.d')[,4][1]
DEN_ne <- MCMCsummary(out_LM_ne, params = 'var.d')[,4][1]

(C_d_ne <- 1 - (NUM_ne/DEN_ne)) #degree of synchonous variation explained by covariates
```

```
## [1] -0.2050254
```

Negative value indicates that the variance in the year effect in the model with covariates is larger than the year effect in the model without covariates (though slight - see posterior summary above). In other words, the covariates explained no additional variation in the model when the extreme year is included.

Appendix 9

This appendix contains supplemental analyses for Chapter 4.

Initial set up

```
#install packages if they don't exist - then load them
if('pacman' %in% rownames(installed.packages()) == FALSE)
{
  install.packages('pacman')
}

pacman::p_load(dplyr,
               knitr)
```

Correlation beween Breeding Success and Abundance - significance testing

Detrend data

```
setwd('Data')

data <- read.csv('pub_data.csv', header=TRUE)
```

```
ADPE <- filter(data, SPECIES == 'ADPE')
EMPE <- filter(data, SPECIES == 'EMPE')
SOFU <- filter(data, SPECIES == 'SOFU')
CAPE <- filter(data, SPECIES == 'CAPE')
SNPE <- filter(data, SPECIES == 'SNPE')
SPSK <- filter(data, SPECIES == 'SPSK')


#function that detrends - can use residuals() if no NA - this accomodates NAs
detrend_fun <- function(INPUT)
{
  tfit <- summary(lm(INPUT ~ c(1:length(INPUT))))
  alpha <- tfit$coefficients[1,1]
  beta <- tfit$coefficients[2,1]

  TOUT <- c()
  for (i in 1:length(INPUT))
  {
    temp <- alpha + beta*i
    temp2 <- INPUT[i] - temp
    TOUT <- c(TOUT, temp2)
  }

  return(TOUT)
}


#NO DETREND
ADPE_ndt <- data.frame(YEAR = ADPE$YEAR, BS = ADPE$BS, L_ABUN = log(ADPE$ABUN))
EMPE_ndt <- data.frame(YEAR = EMPE$YEAR, BS = EMPE$BS, L_ABUN = log(EMPE$ABUN))
```

```
SOFU_ndt <- data.frame(YEAR = SOFU$YEAR, BS = SOFU$BS, L_ABUN = log(SOFU$ABUN))
CAPE_ndt <- data.frame(YEAR = CAPE$YEAR, BS = CAPE$BS, L_ABUN = log(CAPE$ABUN))
SNPE_ndt <- data.frame(YEAR = SNPE$YEAR, BS = SNPE$BS, L_ABUN = log(SNPE$ABUN))
SPSK_ndt <- data.frame(YEAR = SPSK$YEAR, BS = SPSK$BS, L_ABUN = log(SPSK$ABUN))

#DETREND
ADPE_dt <- data.frame(YEAR = ADPE$YEAR, apply(ADPE_ndt[,-1], 2, detrend_fun))
EMPE_dt <- data.frame(YEAR = EMPE$YEAR, apply(EMPE_ndt[,-1], 2, detrend_fun))
SOFU_dt <- data.frame(YEAR = SOFU$YEAR, apply(SOFU_ndt[,-1], 2, detrend_fun))
CAPE_dt <- data.frame(YEAR = CAPE$YEAR, apply(CAPE_ndt[,-1], 2, detrend_fun))
SNPE_dt <- data.frame(YEAR = SNPE$YEAR, apply(SNPE_ndt[,-1], 2, detrend_fun))
SPSK_dt <- data.frame(YEAR = SPSK$YEAR, apply(SPSK_ndt[,-1], 2, detrend_fun))
```

Determine if any of the correlations are significant at the $\alpha = 0.05$ level after controlling for multiple comparisons using a Benjamini-Hochberg correction

```
p_fun <- function(IN_ABUN, IN_BS, i)
{
  tt <- data.frame(ABUN = IN_ABUN[-c(1:i)], BS = IN_BS[-c((LEN - (i-1)):LEN)])
  pv <- cor.test(tt$ABUN, tt$BS, use = 'pairwise.complete.obs')$p.value
  temp2 <- data.frame(Lag = i, p_val = pv)
  return(temp2)
}

ADPE_t <- data.frame(Lag = 0,
                     p_val = cor.test(ADPE_dt$L_ABUN, ADPE_dt$BS,
                                      use = 'pairwise.complete.obs')$p.value)
EMPE_t <- data.frame(Lag = 0,
                     p_val = cor.test(EMPE_dt$L_ABUN, EMPE_dt$BS,
                                      use = 'pairwise.complete.obs')$p.value)
SOFU_t <- data.frame(Lag = 0,
                     p_val = cor.test(SOFU_dt$L_ABUN, SOFU_dt$BS,
                                      use = 'pairwise.complete.obs')$p.value)
CAPE_t <- data.frame(Lag = 0,
                     p_val = cor.test(CAPE_dt$L_ABUN, CAPE_dt$BS,
                                      use = 'pairwise.complete.obs')$p.value)
SNPE_t <- data.frame(Lag = 0,
                     p_val = cor.test(SNPE_dt$L_ABUN, SNPE_dt$BS,
                                      use = 'pairwise.complete.obs')$p.value)
SPSK_t <- data.frame(Lag = 0,
                     p_val = cor.test(SPSK_dt$L_ABUN, SPSK_dt$BS,
                                      use = 'pairwise.complete.obs')$p.value)

LEN <- length(1980:2016)
for (i in 1:8)
{
  ADPE_out <- p_fun(ADPE_dt$L_ABUN, ADPE_dt$BS, i)
  ADPE_t <- rbind(ADPE_t, ADPE_out)
  EMPE_out <- p_fun(EMPE_dt$L_ABUN, EMPE_dt$BS, i)
  EMPE_t <- rbind(EMPE_t, EMPE_out)
  SOFU_out <- p_fun(SOFU_dt$L_ABUN, SOFU_dt$BS, i)
  SOFU_t <- rbind(SOFU_t, SOFU_out)
  CAPE_out <- p_fun(CAPE_dt$L_ABUN, CAPE_dt$BS, i)
```

```
  CAPE_t <- rbind(CAPE_t, CAPE_out)
  SNPE_out <- p_fun(SNPE_dt$L_ABUN, SNPE_dt$BS, i)
  SNPE_t <- rbind(SNPE_t, SNPE_out)
  SPSK_out <- p_fun(SPSK_dt$L_ABUN, SPSK_dt$BS, i)
  SPSK_t <- rbind(SPSK_t, SPSK_out)
}

#p values
mrg_pval <- rbind(ADPE_t, EMPE_t, SOFU_t, CAPE_t, SNPE_t, SPSK_t)

pd <- data.frame(LAG = 0:8,
                 ADPE = ADPE_t$p_val, EMPE = EMPE_t$p_val, SOFU = SOFU_t$p_val,
                 CAPE = CAPE_t$p_val, SNPE = SNPE_t$p_val, SPSK = SPSK_t$p_val)

pdr <- data.frame(LAG = pd$LAG, apply(pd[,-1], 2, function(x) round(x, 3)))
```

Table A9-1: Unadjusted p-values. ADPE – Adélie penguin; EMPE – emperor penguin; SOFU – southern fulmar; CAPE – cape petrel; SNPE – snow petrel; SPSK – south polar skua

LAG	ADPE	EMPE	SOFU	CAPE	SNPE	SPSK
0	0.129	0.146	0.014	0.531	0.765	0.643
1	0.015	0.923	0.946	0.430	0.122	0.939
2	0.766	0.365	0.387	0.555	0.461	0.226
3	0.790	0.394	0.486	0.336	0.212	0.267
4	0.590	0.285	0.972	0.875	0.012	0.793
5	0.414	0.876	0.671	0.178	0.027	0.643
6	0.290	0.026	0.487	0.239	0.032	0.040
7	0.441	0.072	0.124	0.611	0.182	0.392
8	0.969	0.123	0.237	0.215	0.049	0.839

No values fall under the adjusted significance threshold.

```
sum(p.adjust(reshape2::melt(pdr[,-1])[,2], method = 'BH') < 0.05)
```

```
## [1] 0
```

Estimation of the effect of Aundance on Breeding Success (density dependent effects on breeding success)

```
dd_fun <- function(IN, MAIN)
{
  #remove NAs
  to.rm <- which(is.na(IN$BS) | is.na(IN$L_ABUN))
  if(length(to.rm)>0)
  {
    IN_m <- IN[-to.rm,]
  }else{
    IN_m <- IN
  }
  #fit linear model
  fit <- lm(IN_m$BS ~ IN_m$L_ABUN)
  s_fit <- summary(fit)
  p.val <- s_fit$coefficients[2,4] #extract pval
```

```
  r2 <- s_fit$adj.r.squared #extract r^2
  r_p.val <- round(p.val, digits = 3) #round
  r_r2 <- round(r2, digits = 2) #round

  r2_txt <- paste0('r^2 = ', r_r2) #put in txt
  pval_txt <- paste0('p value = ', r_p.val)

  lgd <- rbind(r2_txt, pval_txt) #craete object for legend

  #plot data and fit
  plot(IN_m$L_ABUN, IN_m$BS, pch = 19, main = MAIN, xlab = 'log(Abundance)', ylab = 'Breeding Success')
  abline(fit, col = 'red')
  legend('topright', legend = lgd, bty = 'n', pch = NA, text.col = 'red') #create legend
}

dd_fun(ADPE_ndt, MAIN = 'Adélie penguin')
```

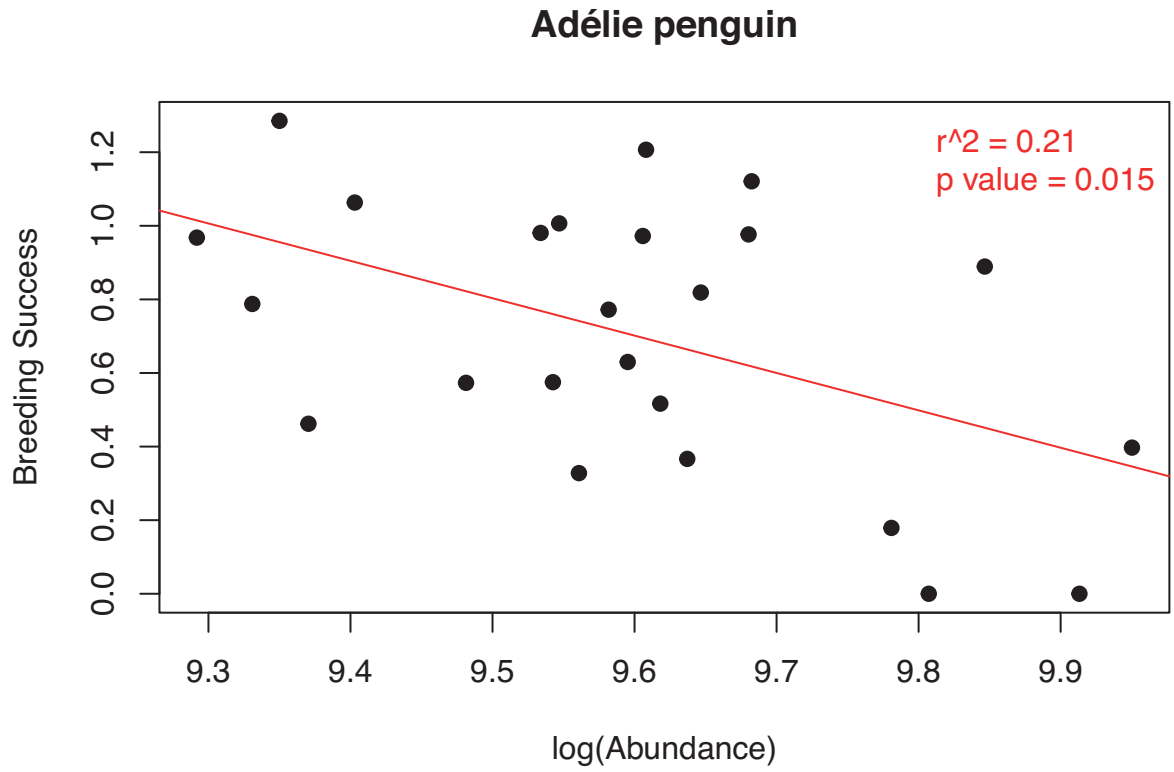

Figure A9-1: Breeding Success as a function of log(Abundance) for Adélie penguin

```
dd_fun(EMPE_ndt, MAIN = 'Empreror penguin')
```

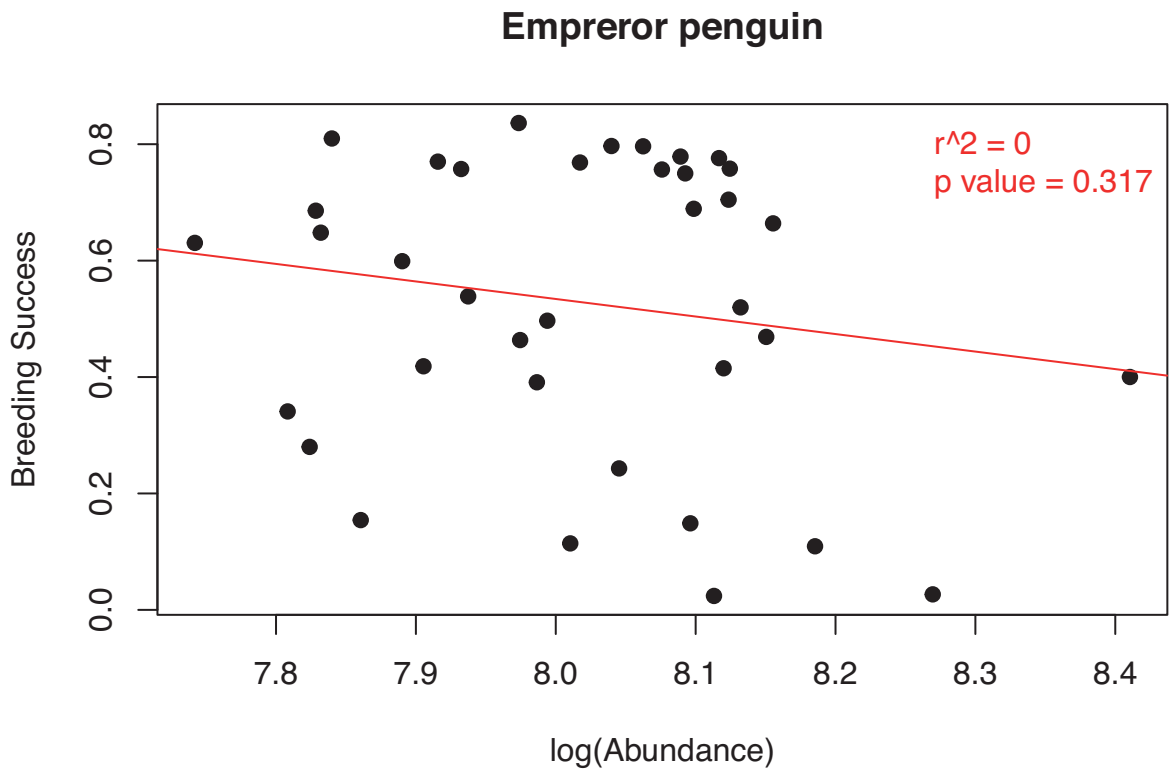

Figure A9-2: Breeding Success as a function of log(Abundance) for emperor penguin

```
dd_fun(SOFU_ndt, MAIN = 'Southern fulmar')
```

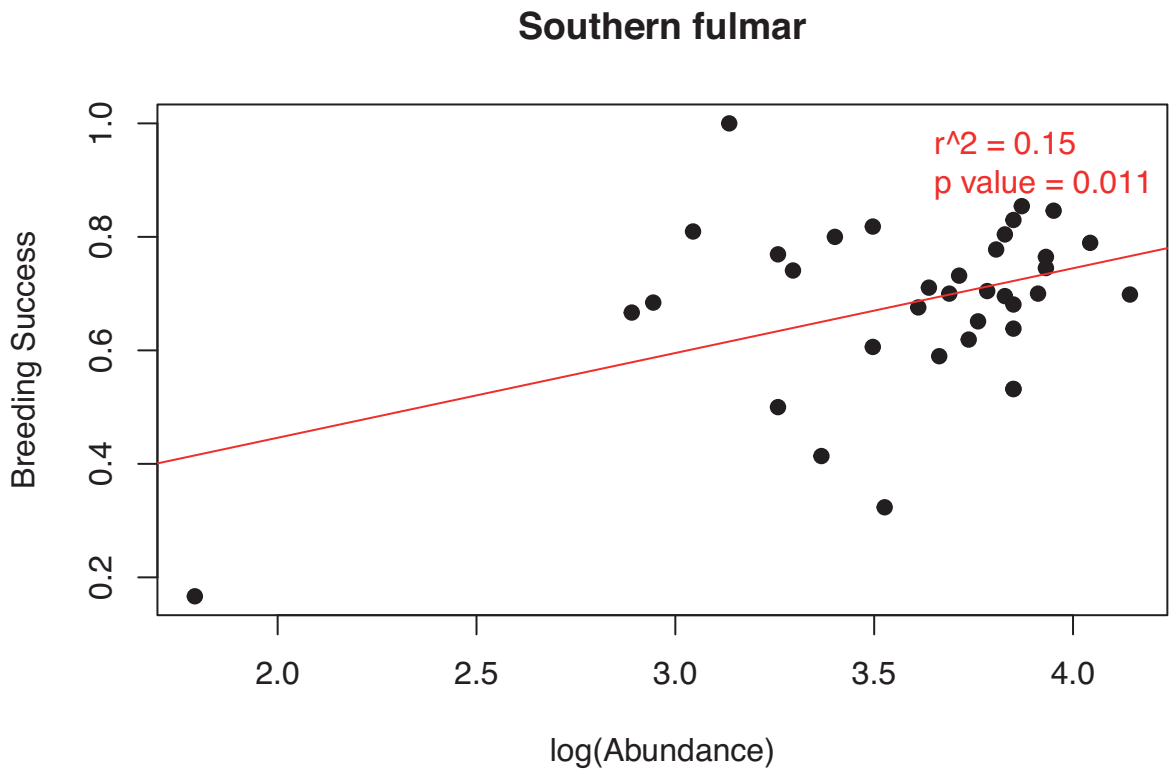

Figure A9-3: Breeding Success as a function of log(Abundance) for southern fulmar

```
dd_fun(CAPE_ndt, MAIN = 'Cape petel')
```

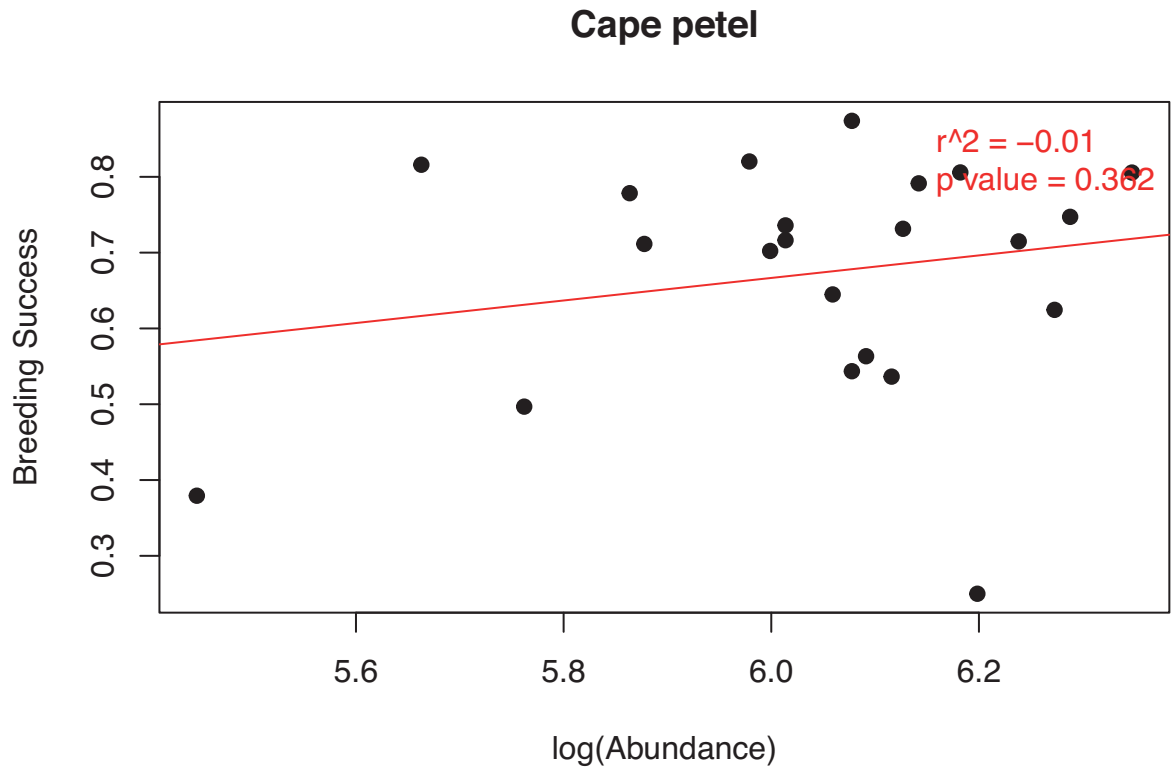

Figure A9-4: Breeding Success as a function of log(Abundance) for cape petrel

```
dd_fun(SNPE_ndt, MAIN = 'Snow petrel')
```

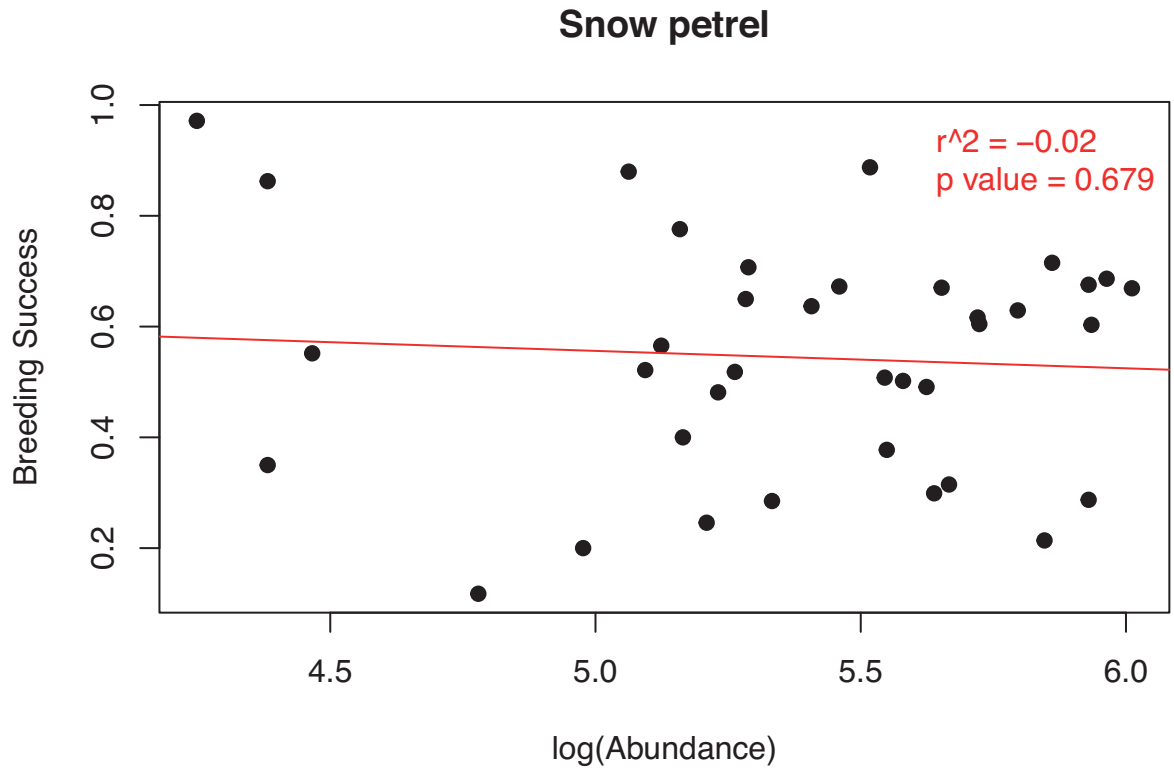

Figure A9-5: Breeding Success as a function of log(Abundance) for snow petrel

```
dd_fun(SPSK_ndt, MAIN = 'South polar skua')
```

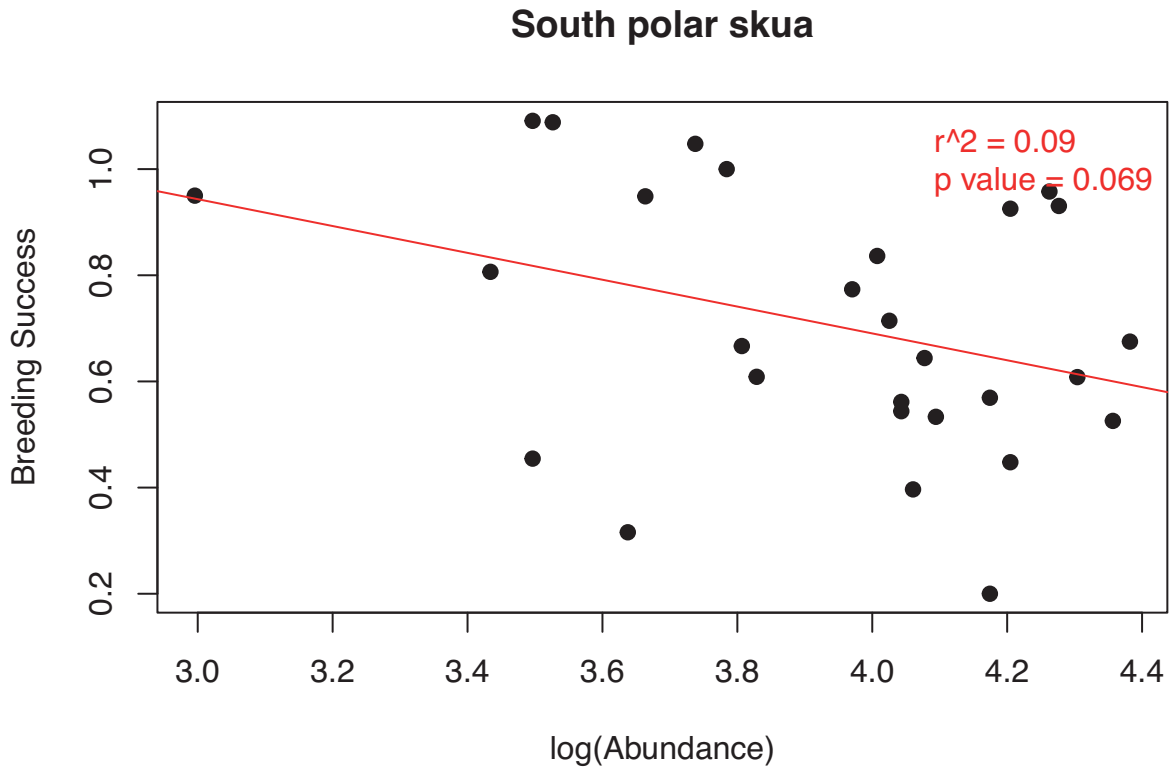

Figure A9-6: Breeding Success as a function of log(Abundance) for south polar skua

March-April SIC at TA from 1979-2016

500km is used because they are leaving the colony at that time. March-April is the end of the breeding season. Birds are starting their migration.

```
setwd('Data/SIC')
SIC_500_ma <- read.csv('SIC_500_ma.csv')
```

```
#plot 500km March April SIC
plot(SIC_500_ma$YEAR, SIC_500_ma$SMN, pch = 19, type = 'l', ylim = c(5, 35),
     main = 'March-April SIC',
     xlab = 'YEAR',
     ylab = 'SIC',
     lwd = 2)
fit <- lm(SMN ~ YEAR, data = SIC_500_ma)
abline(fit, col = 'red', lwd = 2)
s_fit <- summary(fit)
#plot r^s and pval on plot
p.val <- s_fit$coefficients[2,4] #extract pval
r2 <- s_fit$adj.r.squared #extract r^2
r_p.val <- round(p.val, digits = 3) #round
r_r2 <- round(r2, digits = 2) #round
r2_txt <- paste0('r^2 = ', r_r2) #put in txt
pval_txt <- paste0('p value = ', r_p.val)
lgd <- rbind(r2_txt, pval_txt) #craete object for legend
legend('topleft', legend = lgd, bty = 'n', pch = NA, text.col = 'red') #create legend
```

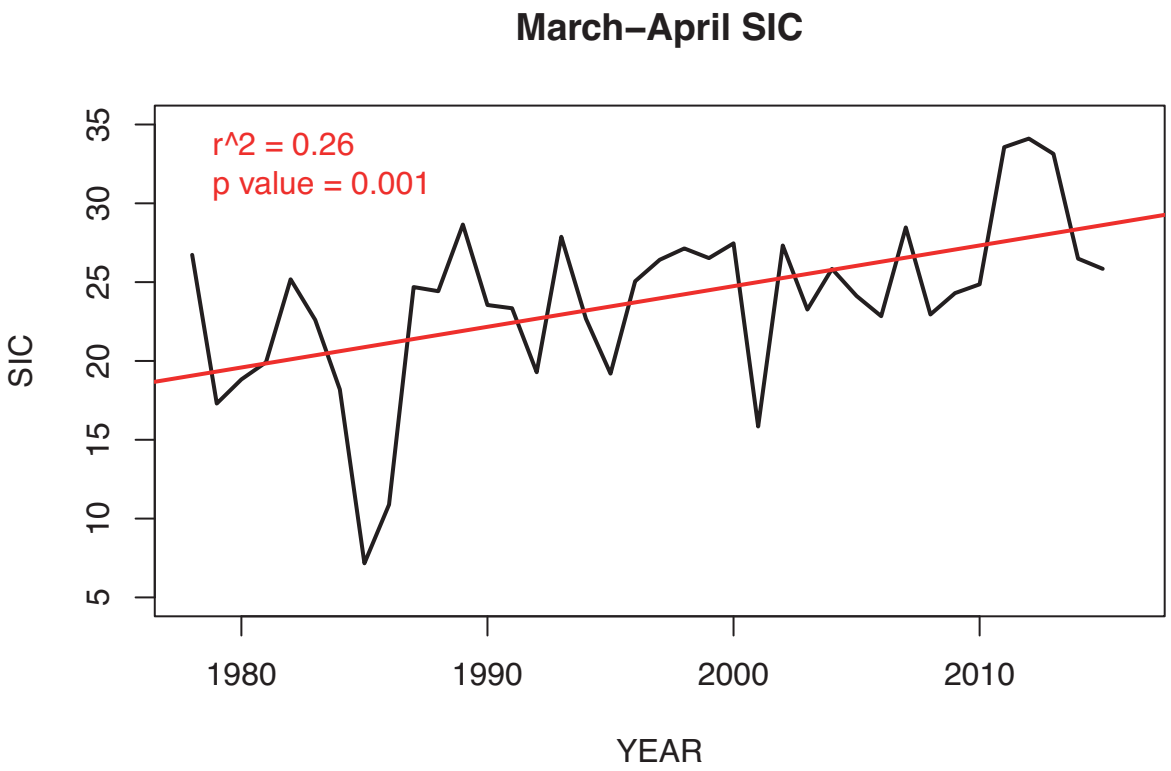

Figure A9-7: March-Aril sea ice concentration at the breeding site

Linear regression shown in red. Sea ice during this period is increasing over time. ADPE at least need sea ice to overwinter.

This might be resulting in higher juvenile survival, a notion that is consistent with the diverging trends of Abundance and Breeding Success. Future work should explore this notion further.

Bad years for one species aren't bad years for all species

Exceptionally poor years, defined here as any year that is in the fifth quantile or lower for breding success, were calculated for each of the six species. Only years where data on all species was available was used.

```
BS_mrg <- data.frame(YEAR = ADPE$YEAR,
                     ADPE_BS = ADPE$BS,
                     EMPE_BS = EMPE$BS,
                     SOFU_BS = SOFU$BS,
                     CAPE_BS = CAPE$BS,
                     SNPE_BS = SNPE$BS,
                     SPSK_BS = SPSK$BS)

#which rows have no NAs (data for every species)
BS_ind <- which(table(which(!is.na(BS_mrg), arr.ind=TRUE)[,1]) == 7)

BS_comp <- BS_mrg[BS_ind,]


#find exceptionally poor years

species <- c('Adélie penguin', 'Emperor penguin', 'Southern fulmar',
             'Cape petrel', 'Snow petrel', 'South polar skua')


low_fun <- function(IN, TH = 0.05) #TH is threshold for quantile cutoff
{
  #IN <- BS_comp_dt
  l_th <- apply(IN[,-1], 2, function(x) quantile(x, probs = TH))

  OUT1 <- data.frame()
  for (i in 1:6) #6 is number of species
  {
    #i <- 1
    l_yrs <- IN[which(IN[,i+1] <= l_th[i]), 1]
    ind <- rep(species[i], length(l_yrs))
    temp <- data.frame(SPECIES = ind, POOR_YEAR = l_yrs)
    OUT1 <- rbind(OUT1, temp)
  }
  return(OUT1)
}


#Antarctic seabirds
#find years in the bottom 5% for breeding success
(lyr_BS <- low_fun(BS_comp, 0.05))
```

```
##             SPECIES POOR_YEAR
## 1    Adélie penguin      2013
## 2    Adélie penguin      2016
## 3   Emperor penguin      1994
```

```
## 4   Emperor penguin      2014
## 5   Southern fulmar      2001
## 6   Southern fulmar      2013
## 7       Cape petrel      2013
## 8       Cape petrel      2014
## 9       Snow petrel      2011
## 10      Snow petrel      2014
## 11 South polar skua      1998
## 12 South polar skua      2013
```

While some exceptionally poor years are shared among species, species exhibit 'extreme years' independent of the rest of the community as well.

The number of species that exhibit an exceptionally poor year in the following years is shown below.

```
table(lyr_BS[,2])
```

```
##
## 1994 1998 2001 2011 2013 2014 2016
##    1    1    1    1    4    3    1
```

Appendix 10

This appendix contains supplemental information and supplemental analyses for work presented in Chapter 5.

Guano sampling

A total of 108 *Pygoscelis* spp. penguin (Adélie penguin *Pygoscelis adeliae*, gentoo penguin *Pygoscelis papua*, and chinstrap penguin *Pygoscelis antarcticum*) guano samples were collected from 16 unique breeding colonies colonies in the Antarctic Peninsula region (Figure A10-1) during December of 2014 and December of 2015. While the Adélie penguin is the principle focus of this study due to its circumpolar distribution and well understood population dynamics, samples were collected from gentoo and chinstrap penguins as well to increase the dietary breadth and overall number of samples collected. For instance, previous localized dietary studies at penguin breeding colonies in the Antarctic Peninsula region have suggested that Adélie penguin diet in this region is far less flexible than gentoo penguin diet (Ratcliffe and Trathan 2012). Increasing sampling breadth allowed us to characterize the spectra of a wider range of diets, ultimately resulting in more robust dietary predictions. Similar physiology of these closely related species suggests that spectra/diet relationships would hold true across all three *Pygoscelis* species.

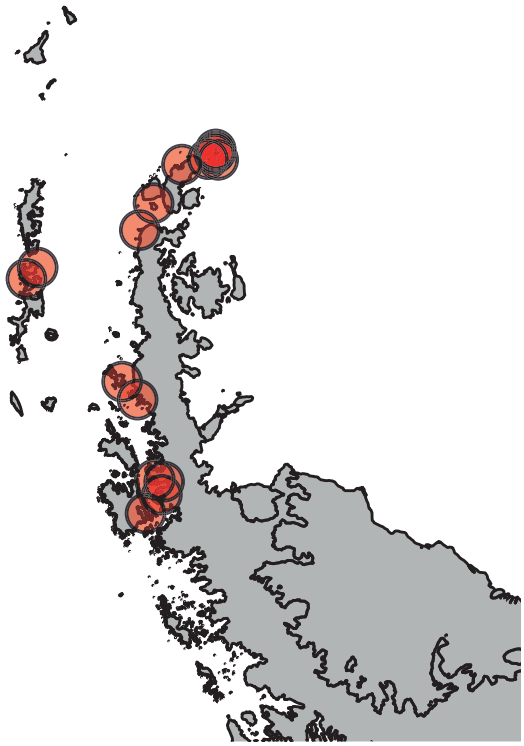

Figure A10-1: Guano sampling sites (16) across the Antarctic Peninsula region.

Guano reflectance

Reflectance measurements and $\delta^{15}N$ values (a proxy for the trophic level at which an animal is feeding) were obtained for each guano sample. Guano spectra (350 – 2500 nm, derived from an ASD FieldSpec 4 instrument) was convolved to simulate relevant satellite sensor bands (Landsat 4 TM, Landsat 5 TM, Landsat 7 ETM+) using the 'hsdar' package (Lehnert et al. 2018) in the R statistical environment (R Core Team 2016) (Figure A10-2).

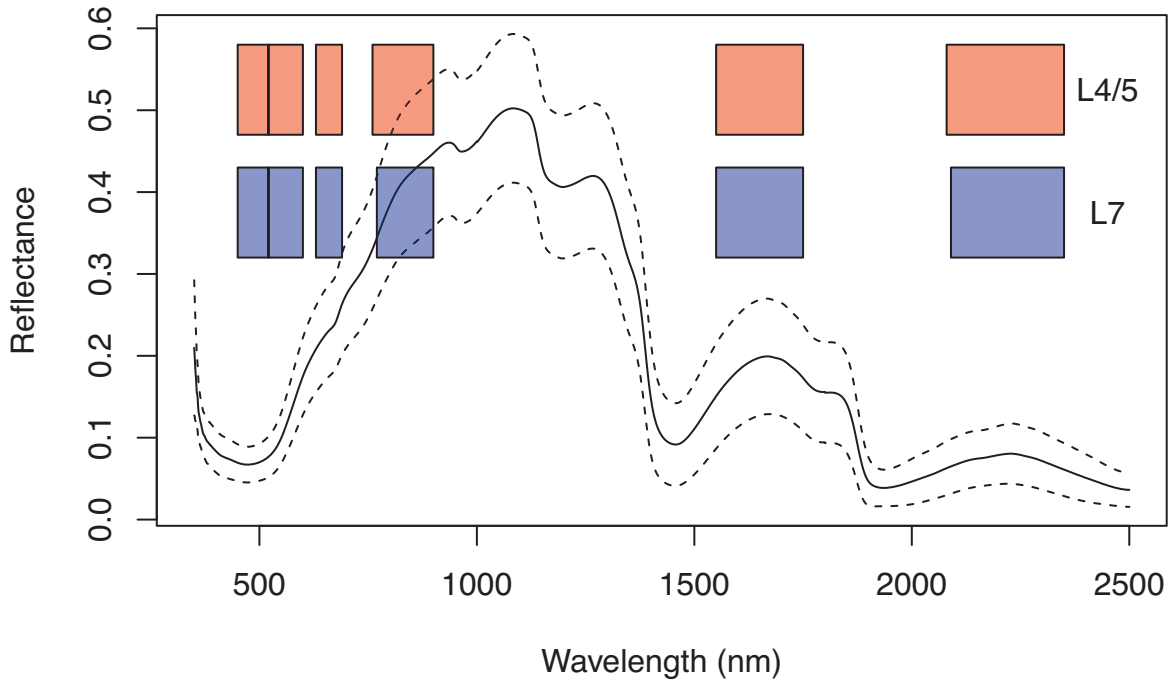

Figure A10-2: Spectral reflectance of penguin guano samples used in analyses. The solid black line represents the mean spectra of all samples, while dotted lines represent 95th quantiles. Colored boxes represent the spectral range of the Landsat 4 and Landsat TM (red), and Landsat 7 ETM+ (blue) sensors.

Guano spectra and $\delta^{15}N$

$\delta^{15}N$ obtained from stable isotope analysis (SIA) was modeled as a function of convolved guano spectra using a partial least squares regression to develop a quantitative relationship between guano spectra and $\delta^{15}N$ (dietary proxy for trophic level at which an organism is feeding at). As partial least squares regression is sensitive to outliers (Hubert and Branden 2003), Cook's distance was used to remove any samples with anomalous spectra properties. Following (Bollen and Jackman 1990), any sample with a computer Cook's D of greater than 4 x mean (Cook's D) was removed from the spectral calibration (Figure A10-3).

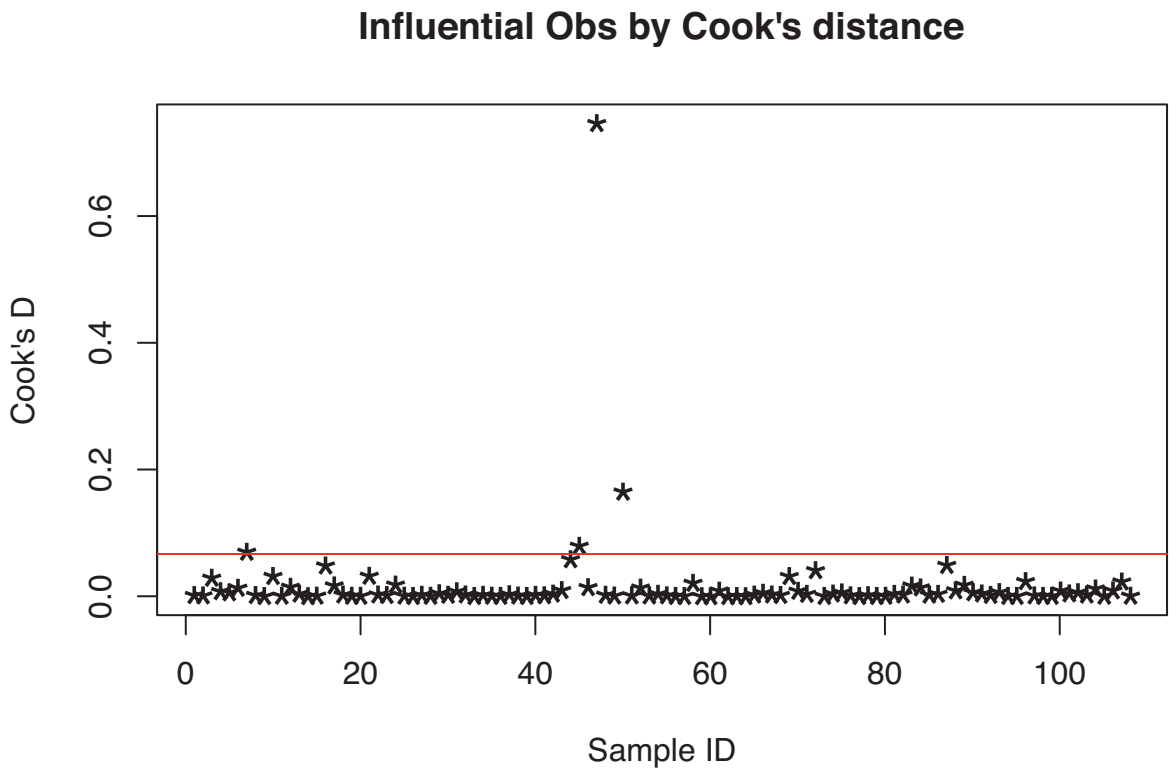

Figure A10-3: Cooks distance for each guano sample. Higher values represent anomalous spectra qualities for that sample. The red line represents the mean value for Cook's D.

In total, 4 outliers were identified, leaving 104 samples for the spectral calibration. Laboratory analysis revealed a range of $\delta^{15}N$ values from 1.43 - 14.22 $^{0}/_{00}$ with a mean of 6.13 $^{0}/_{00}$ (sd = 2.44 $^{0}/_{00}$; Figure A10-4).

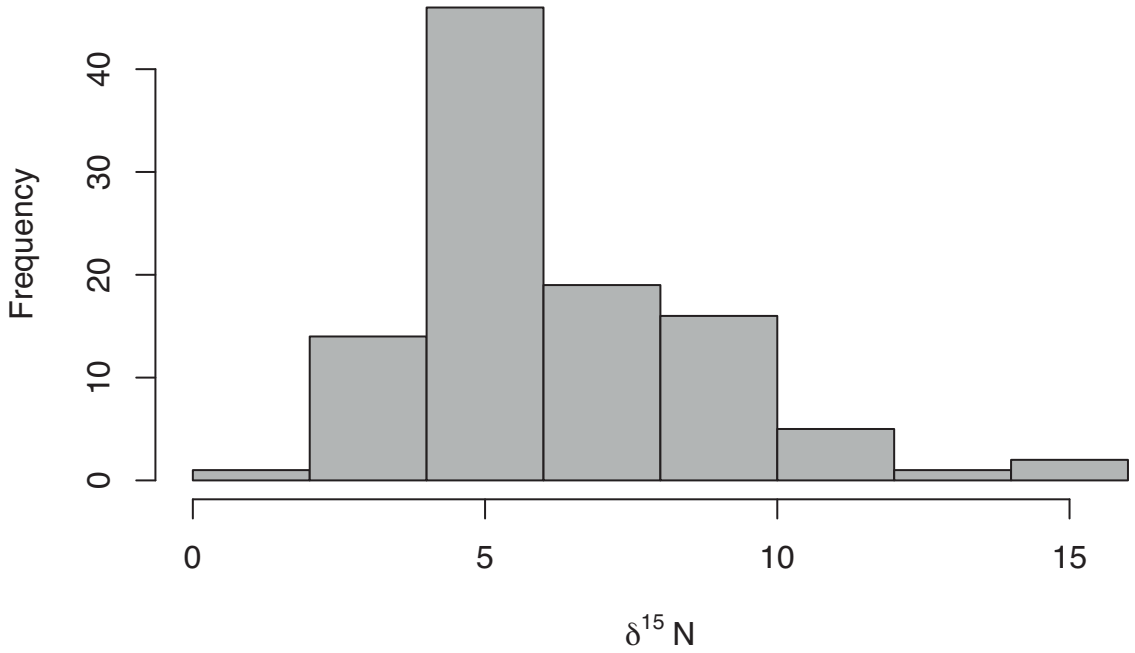

Figure A10-4: Histogram of $\delta^{15}N$ values for the 104 guano samples used in the PLSR analysis.

Samples were divided into separate calibration and validation sets, whereby the calibration set would be used to calibrate the model, and some samples will be held out for further validation. Eighty percent of the dataset were randomly selected for calibration, 20% for out of sample validation

Number of samples in Caliration set: 83

Number of samples in Validation set: 21

Partial least squares (PLS) regression was implemented using the 'pls' package (Mevik and Wehrens 2007). Separate models were fit for each sensor, as the bands cover slightly different regions of the electromagnetic spectrum (Figure A10-1). PLS decomposes covariates (in this case, reflectance values obtained from satellite sensors) into 'principal components', which are fit to predict the response variable (i.e., $\delta^{15}N$ values obtained from SIA). To determine the optimal number of principal components, the PRESS statistic (predicted sum-of-squares) should be minimized. However, this may result in overfitting. When not obvious using visual assessment, a t-test was used to determine whether adding an additional component significantly reduced the PRESS statistic. If a component significantly reduced the statistic, the larger number of components was used.

Landsat 4

Cross-validation (CV) was used to evaluate PRESS statistic. The PRESS statistic was calculated using a randomly selected 80% of the calibration set at each iteration.

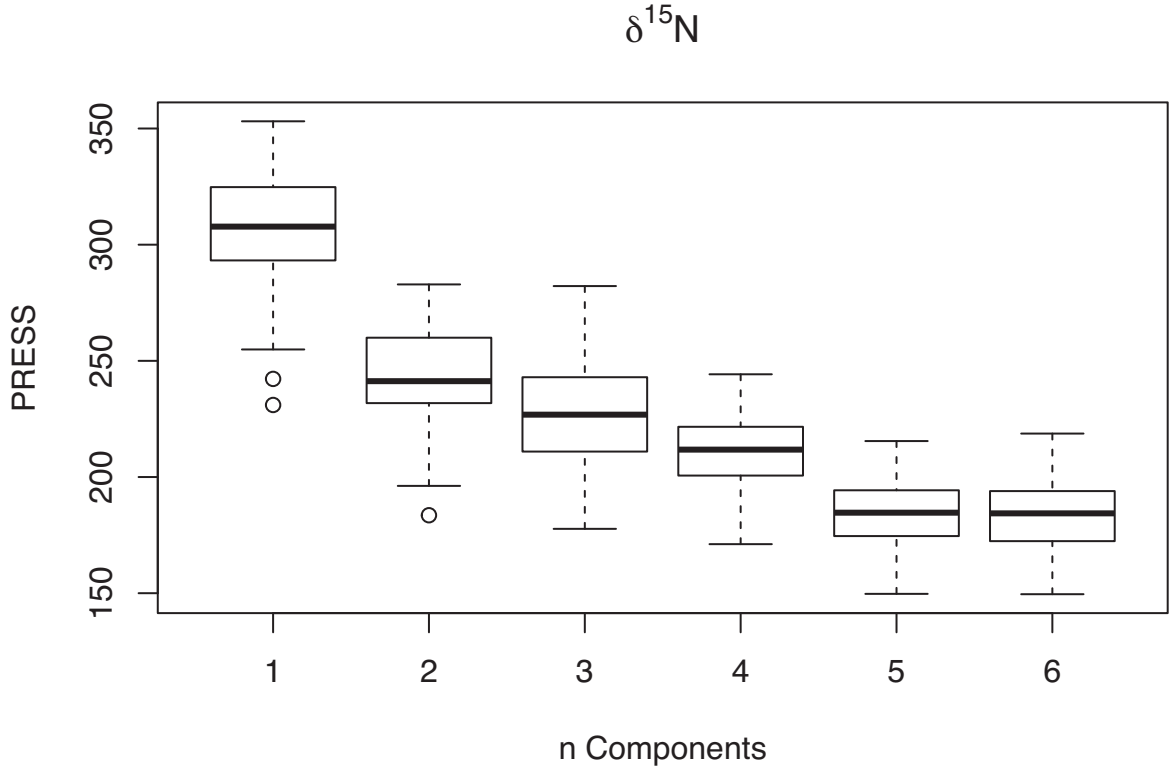

Figure A10-5: Landsat 4 - PRESS statistic as determined by the PLS analysis (lower being a better fit). The thick lines represent median values derived from the k-fold procedure, while box edges represent quartiles. The top and bottom of the whiskers are 1.5 × interquartile range from the upper and lower boxes, respectively. Data beyond this range are plotted as points.

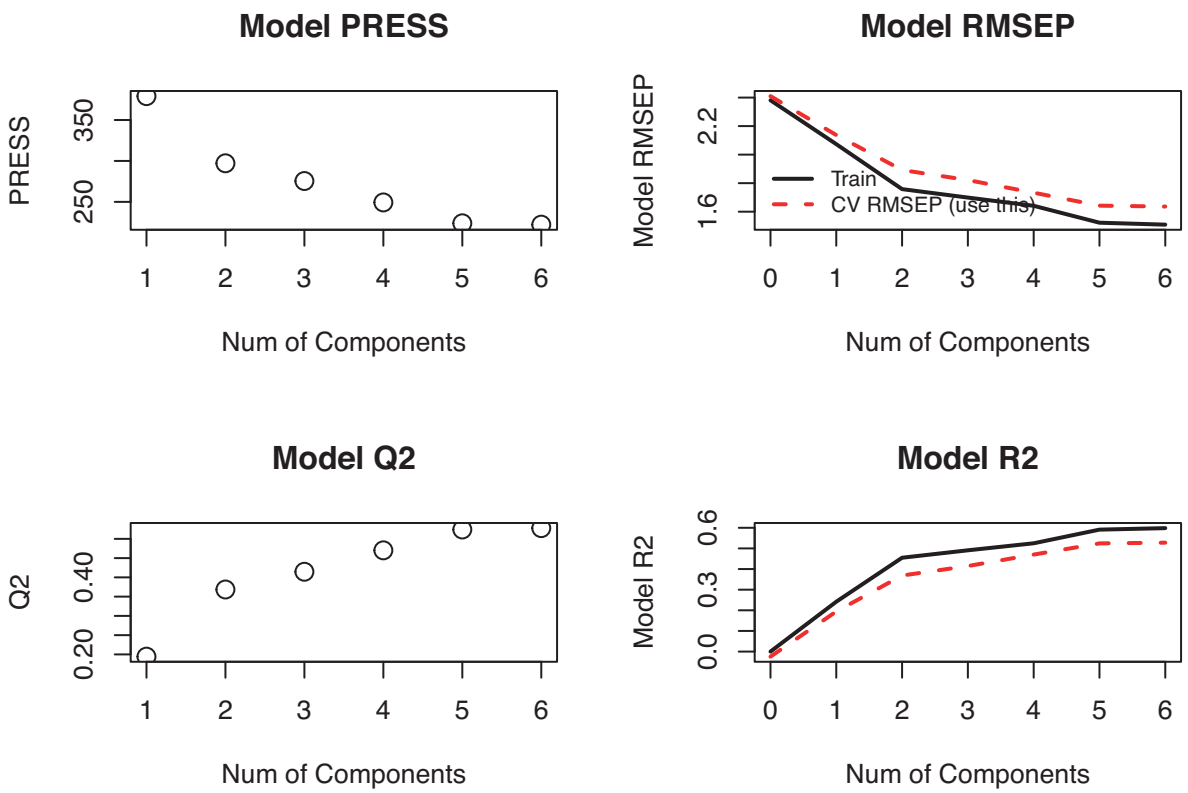

Figure A10-6: Landsat 4 - Model PRESS, RMSEP (Root Mean Square Error of Prediction), Q2, and model R^2 at various numbers of components. PRESS, RMSEP should be minimized, while Q2 and R^2 should be maximized.

A t-test was used to decided whether 6 components significantly improves model fit over 5 components.

t-test p-val: 0.723

Landsat 4 number of components: 5

Some structure in the residuals is apparent, with a tendency to underestimate high values for $\delta^{15}N$, but overall diagnostics show a reasonable model fit (Figure A10-7).

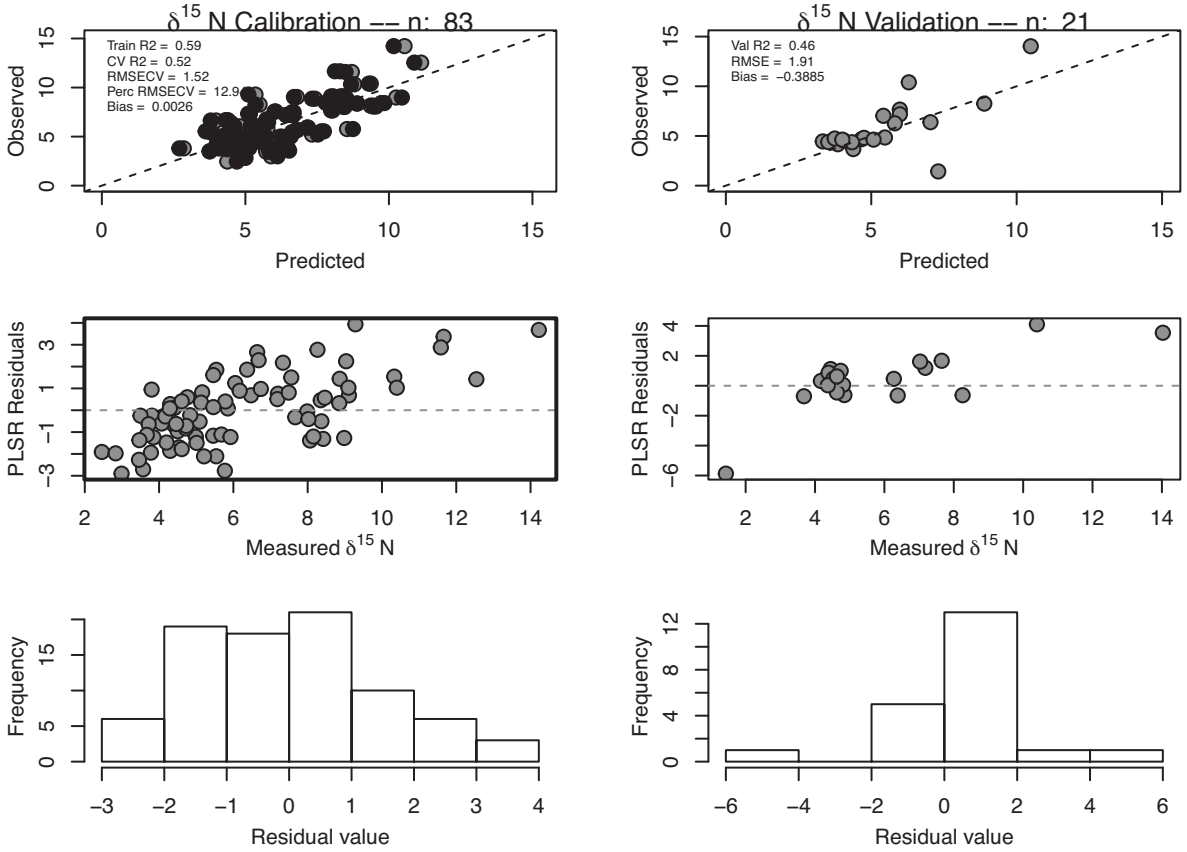

Figure A10-7: Landsat 4 - Observed vs. predicted values as determined by the PLS analysis, residuals as a function of $\delta^{15}N$, and histograms of the residuals for the calibration (left column) and out of sample validation sets (right column).

Coefficient values and Variable Importance Factors (VIP) were calculated for each sensor band (Figure A10-8). Coefficients are how each band contributes to the predicted response. VIP is used to determine which predictor variables contribute most to the prediction of the response variable. Higher VIP values indicate greater importance of that predictor in the PLSR. The NIR band (Band 4 for Landsat 4/5 [760nm to 900nm], Band 4 for Landsat 7 [770nm to 900nm], and Band 5 for Landsat 8 [851nm to 879nm]) shows increased importance in the prediction of $\delta^{15}N$ values from guano spectra.

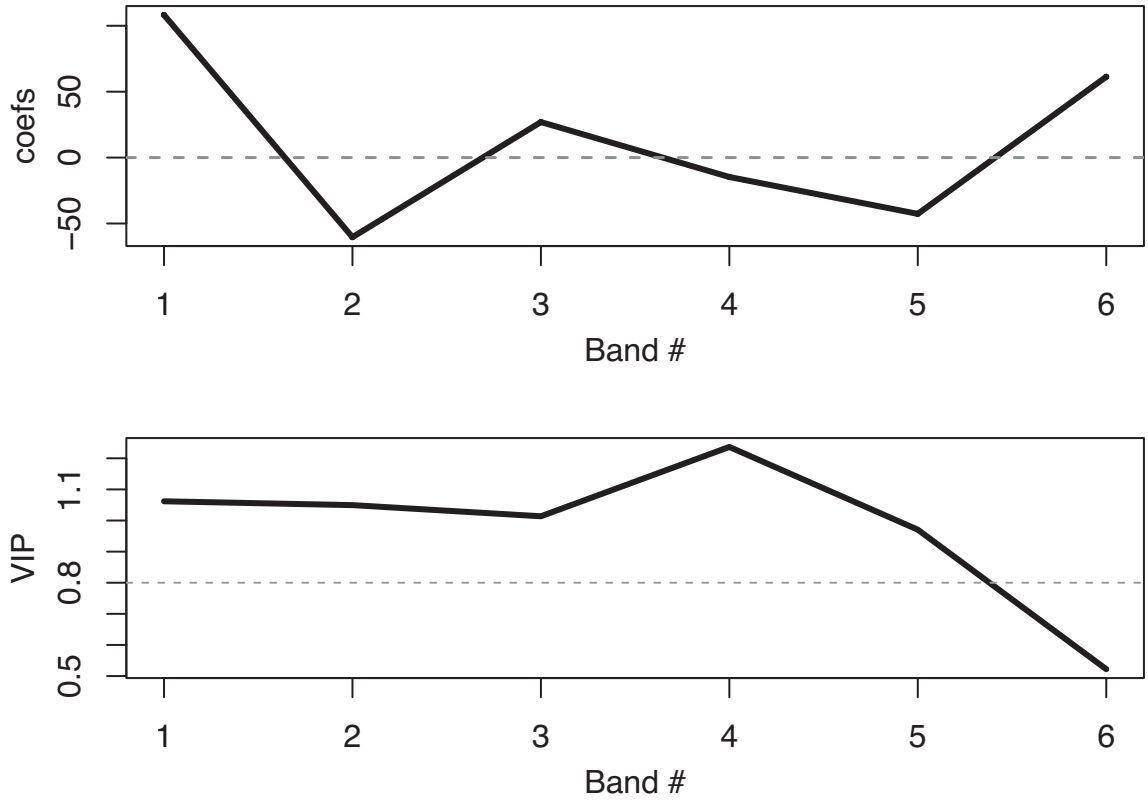

Figure A10-8: Landsat 4 - Coefficients (top panel) and Variable Importance Factors (VIP) (bottom panel) for each sensor band.

CV was used to evaluate the model R^2, RMSE, and model bias. For each of 1000 iterations, the calibration set was subset into two groups, one for CV calibration (80% of the data), one for CV validation (20% of the data) and the aforementioned statistics were calculated using the CV validation set. Overall fit was good with no strong systematic biases (Figure A10-9).

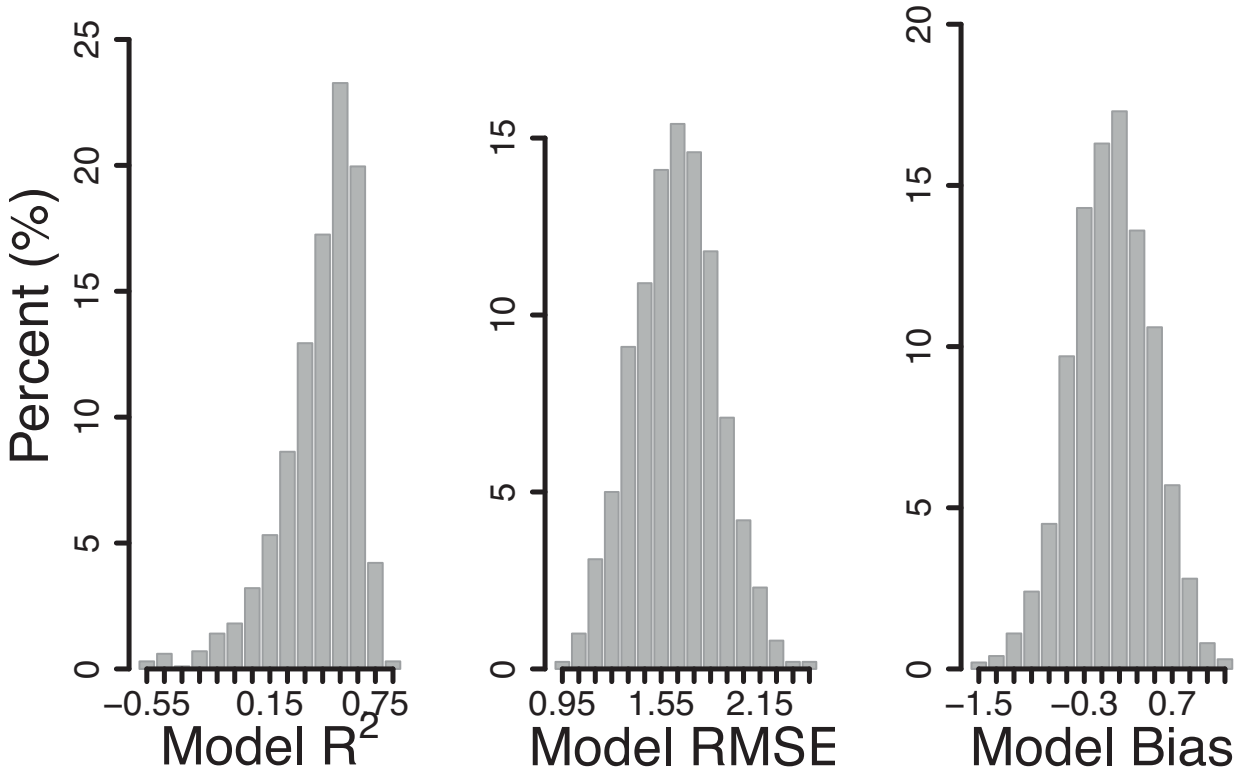

Figure A10-9: Landsat 4 - Model R^2, RMSE, and model bias at each iteration of the CV validation

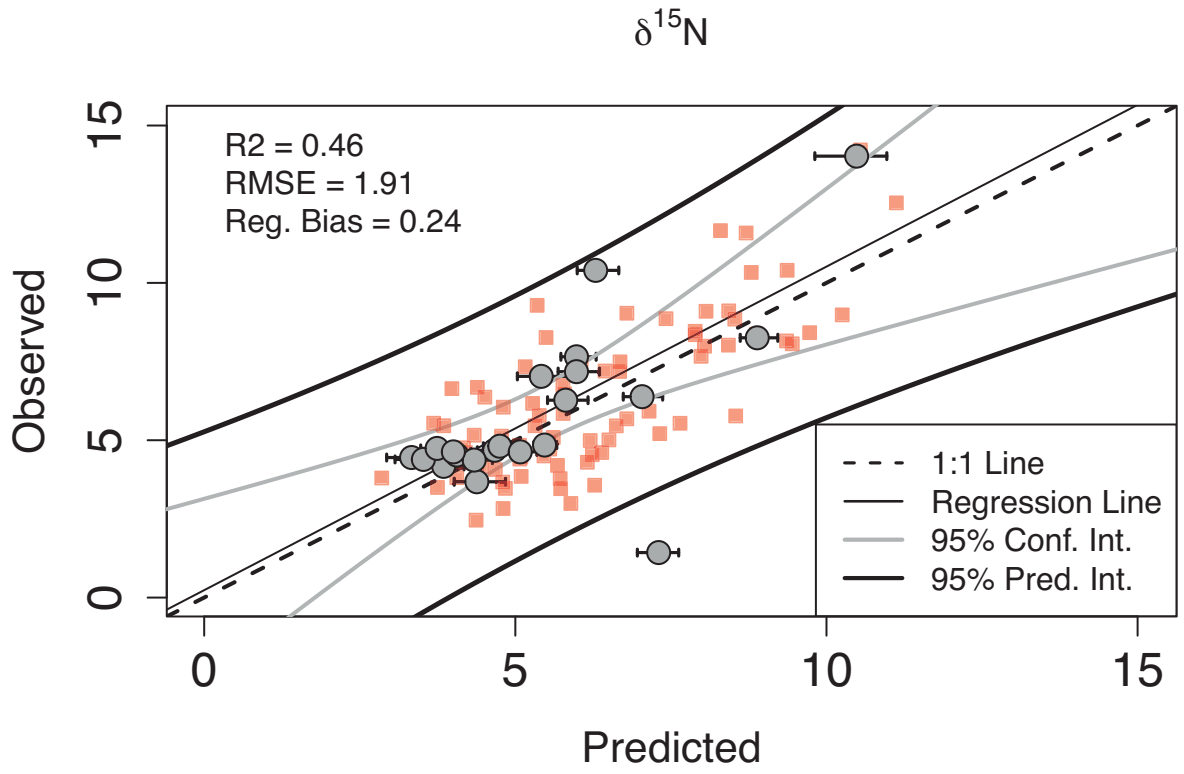

Figure A10-10: Landsat 4 - Validation results for the PLS model. Grey dots are predicted values from the PLS model using the validation set, error bars denote the 95% confidence intervals for each predicted value, red squares represent predicted values from the calibration set, grey lines represent 95% confidence intervals, thick black lines represent 95% prediction intervals, the dotted black line represents the 1:1 line, and the thin solid black line represents the regression. Values for R^2, RMSE, and model bias are shown in the top left.

Landsat 5

The same analyses are conducted above, but for Landsat 5. All methods and assessments that applied previously apply below as well.

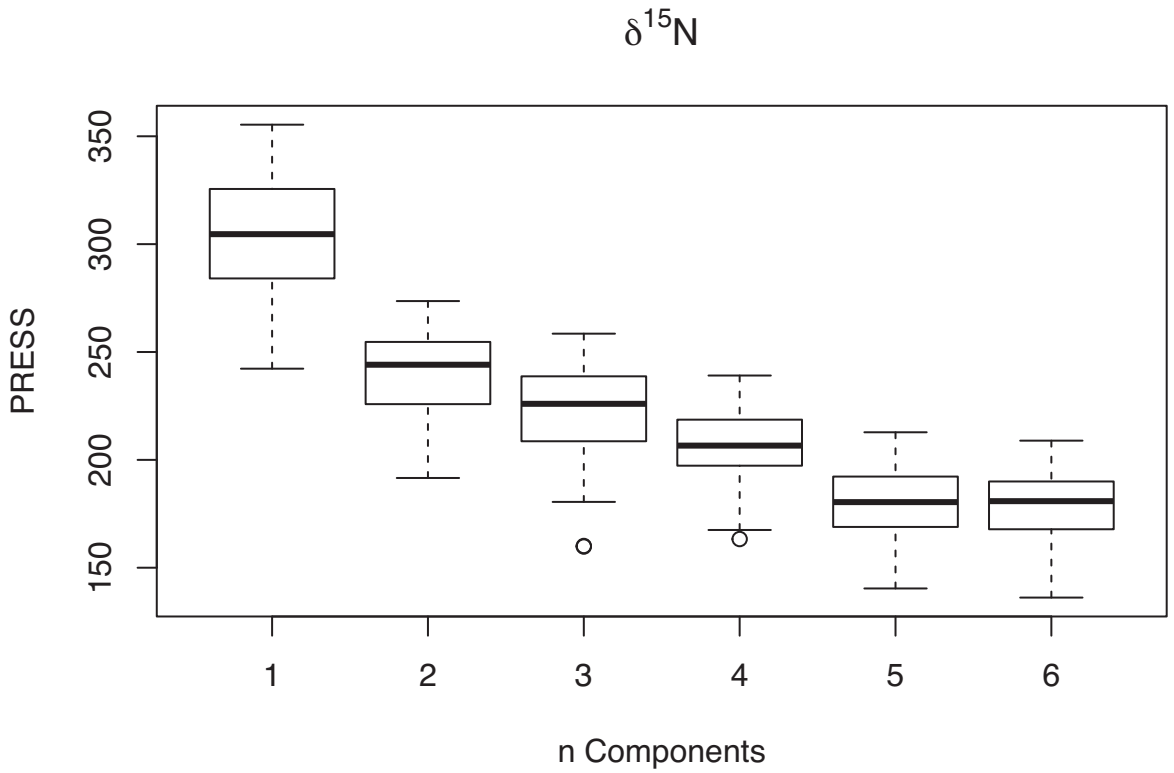

Figure A10-11: Landsat 5 - PRESS statistic as determined by the PLS analysis (lower being a better fit). The thick lines represent median values derived from the k-fold procedure, while box edges represent quartiles. The top and bottom of the whiskers are 1.5 × interquartile range from the upper and lower boxes, respectively. Data beyond this range are plotted as points.

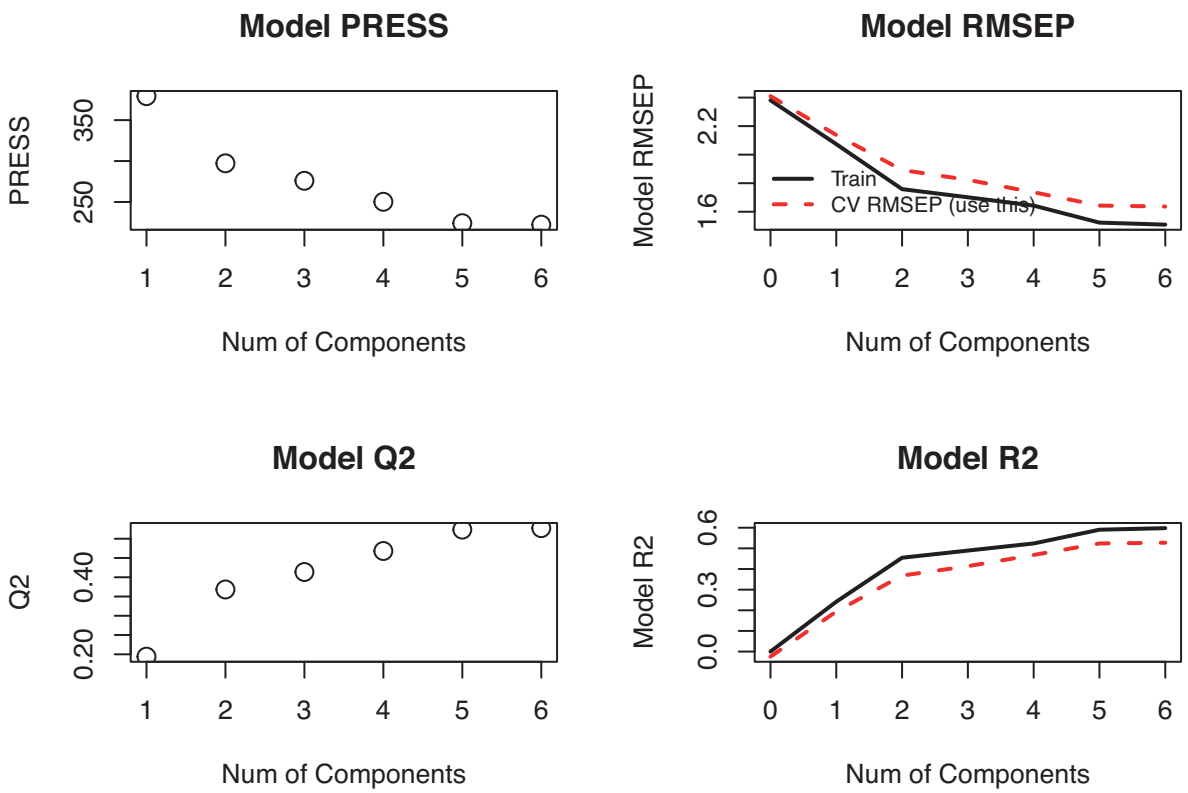

Figure A10-12: Landsat 5 - Model PRESS, RMSEP (Root Mean Square Error of Prediction), Q2, and model R^2 at various numbers of components. PRESS, RMSEP should be minimized, while Q2 and R^2 should be maximized.

t-test p-val: 0.652

Landsat 5 number of components: 5

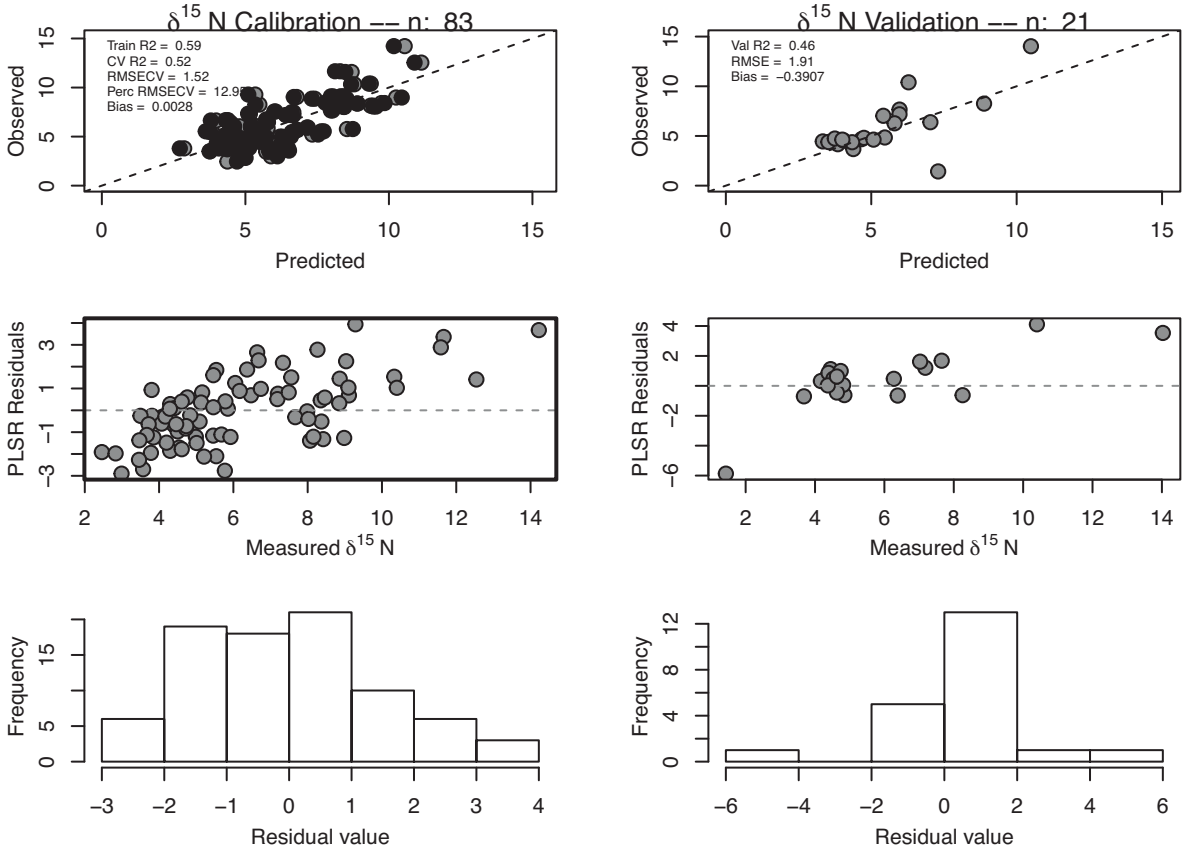

Figure A10-13: Landsat 5 - Observed vs. predicted values as determined by the PLS analysis, residuals as a function of $\delta^{15}N$, and histograms of the residuals for the calibration (left column) and out of sample validation sets (right column).

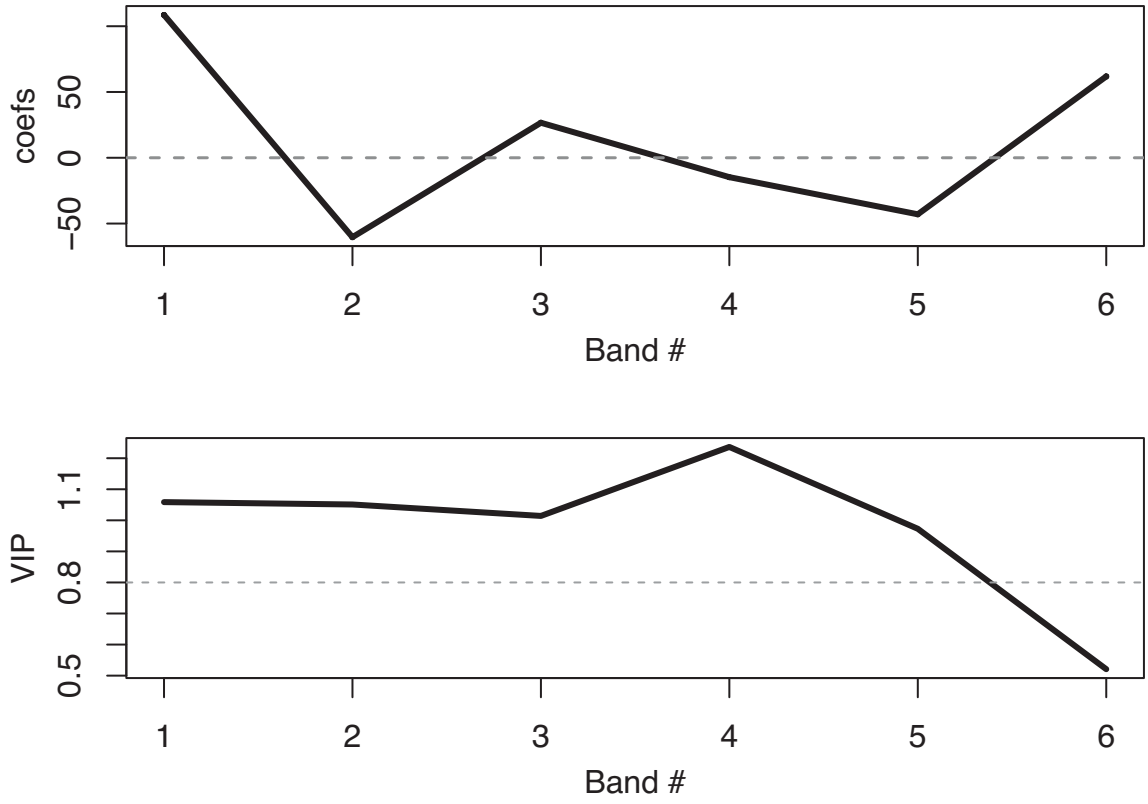

Figure A10-14: Landsat 5 - Coefficients (top panel) and Variable Importance Factors (VIP) (bottom panel) for each sensor band.

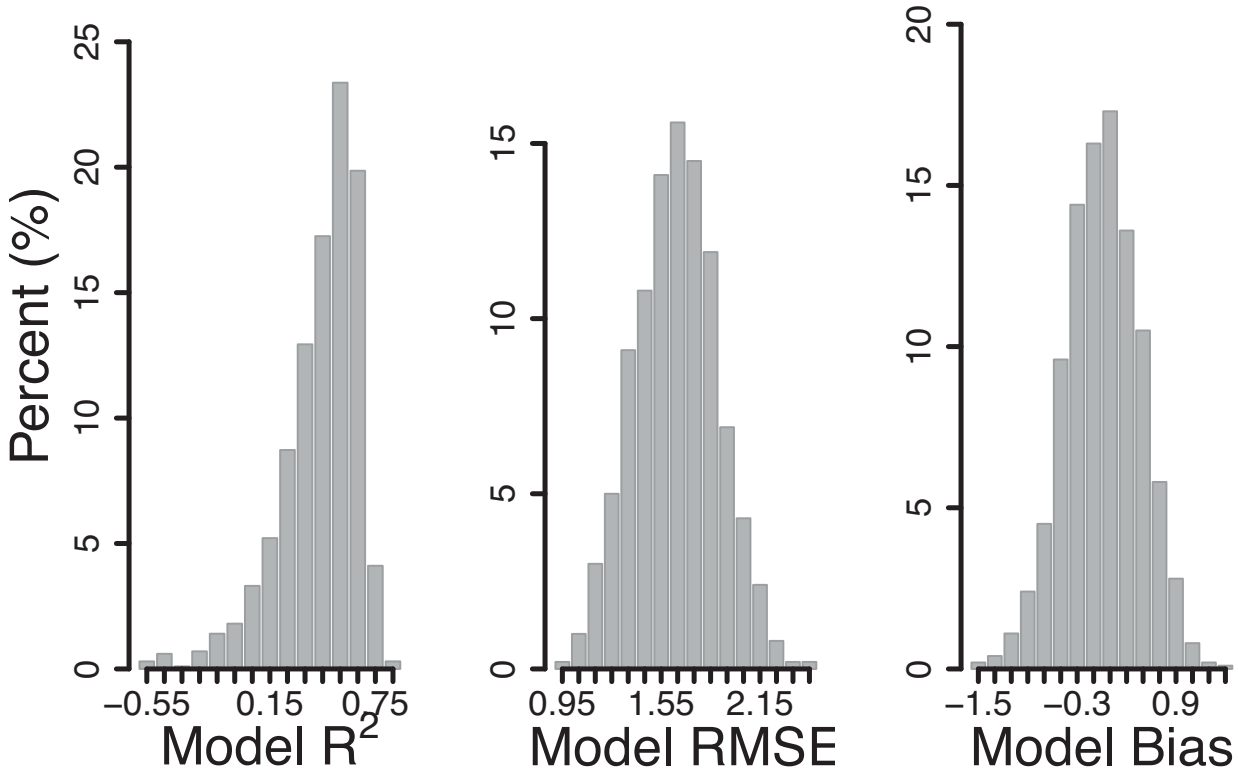

Figure A10-15: Landsat 5 - Model R^2, RMSE, and model bias at each iteration of the CV validation

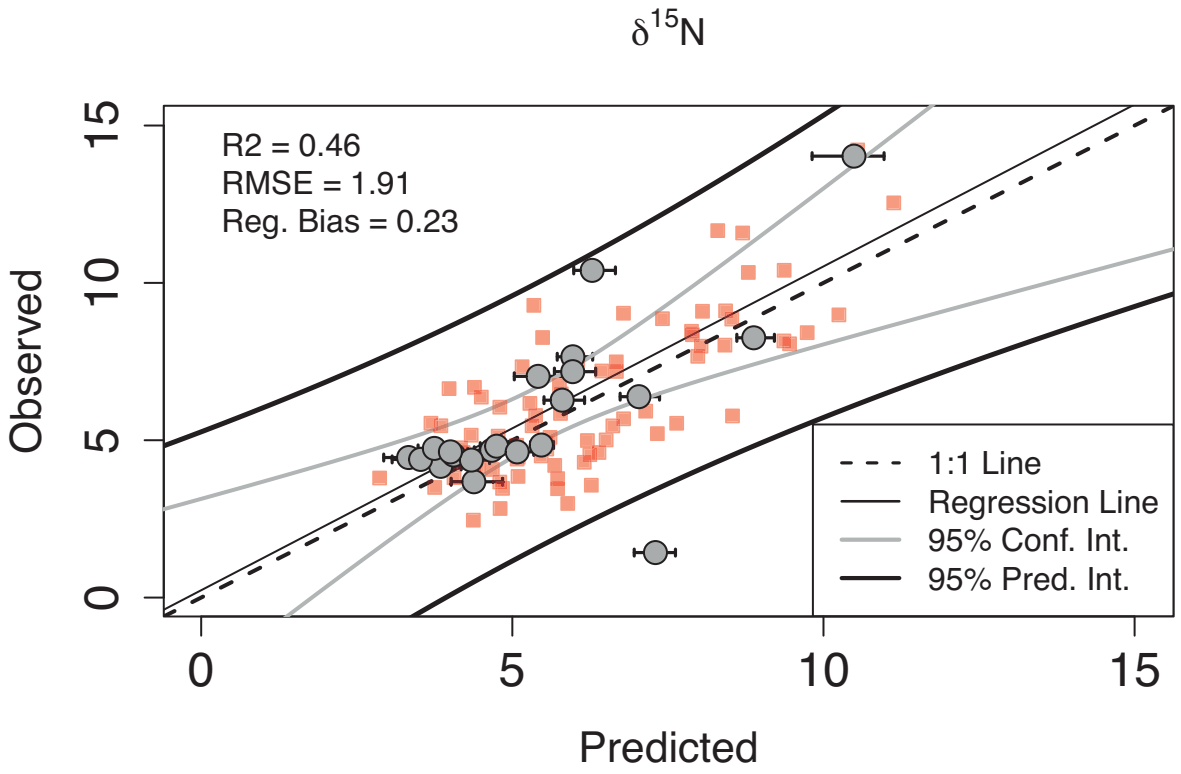

Figure A10-16: Landsat 5 - Validation results for the PLS model. Grey dots are predicted values from the PLS model using the validation set, error bars denote the 95% confidence intervals for each predicted value, red squares represent predicted values from the calibration set, grey lines represent 95% confidence intervals, thick black lines represent 95% prediction intervals, the dotted black line represents the 1:1 line, and the thin solid black line represents the regression. Values for R^2, RMSE, and model bias are shown in the top left.

Landsat 7

The same analyses are conducted above, but for Landsat 7. All methods and assessments that applied previously apply below as well.

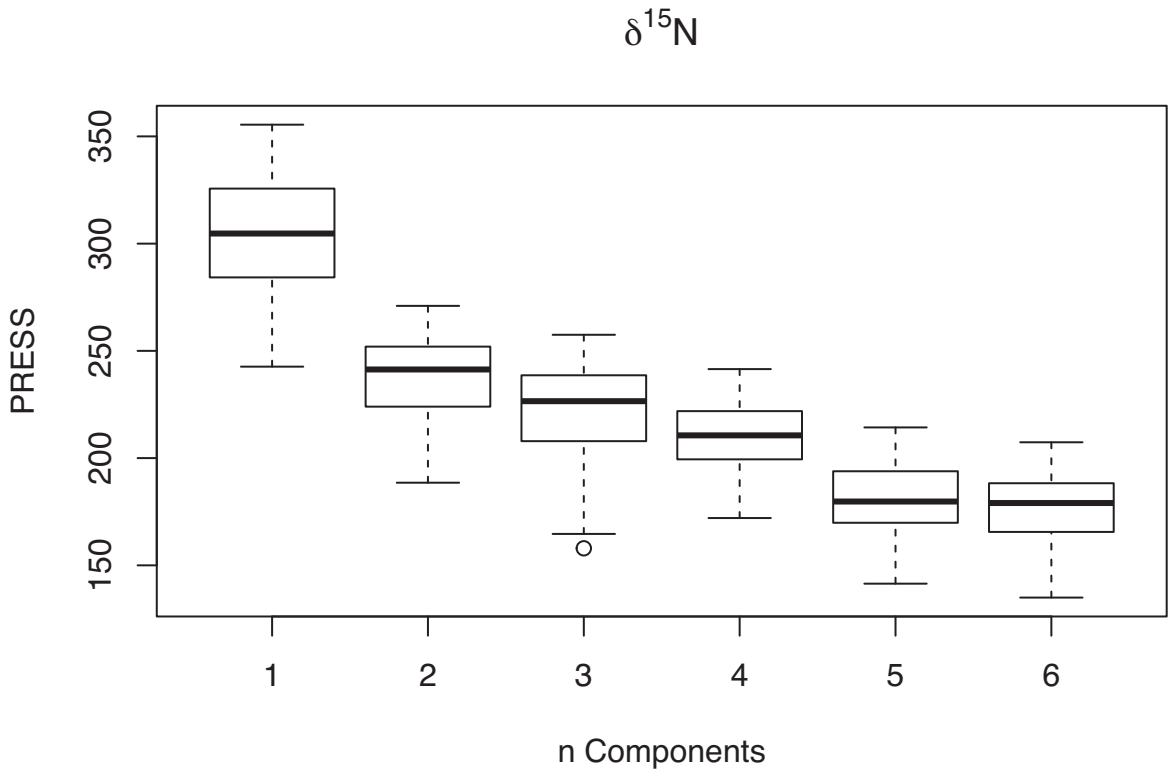

Figure A10-17: Landsat 7 - PRESS statistic as determined by the PLS analysis (lower being a better fit). The thick lines represent median values derived from the k-fold procedure, while box edges represent quartiles. The top and bottom of the whiskers are 1.5 × interquartile range from the upper and lower boxes, respectively. Data beyond this range are plotted as points.

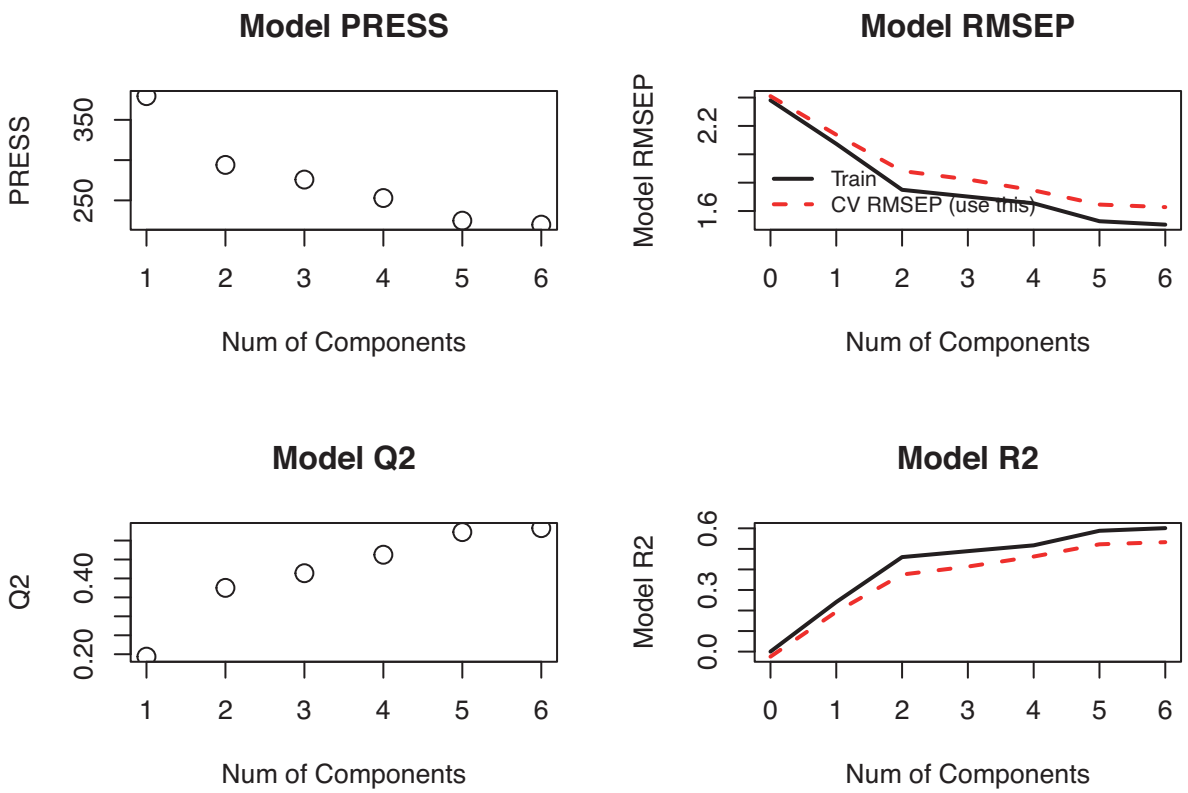

Figure A10-18: Landsat 7 - Model PRESS, RMSEP (Root Mean Square Error of Prediction), Q2, and model R^2 at various numbers of components. PRESS, RMSEP should be minimized, while Q2 and R^2 should be maximized.

t-test p-val: 0.103

Landsat 7 number of components: 5

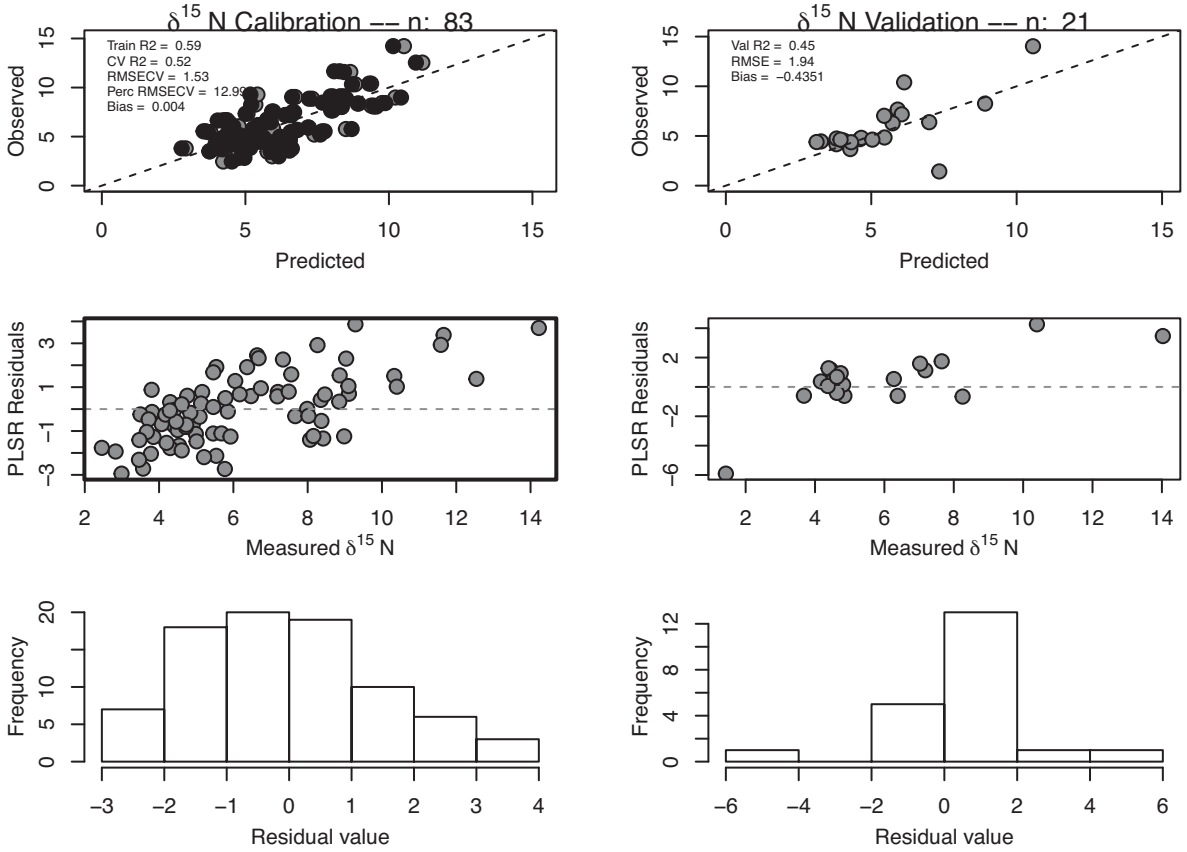

Figure A10-19: Landsat 7 - Observed vs. predicted values as determined by the PLS analysis, residuals as a function of $\delta^{15}N$, and histograms of the residuals for the calibration (left column) and out of sample validation sets (right column).

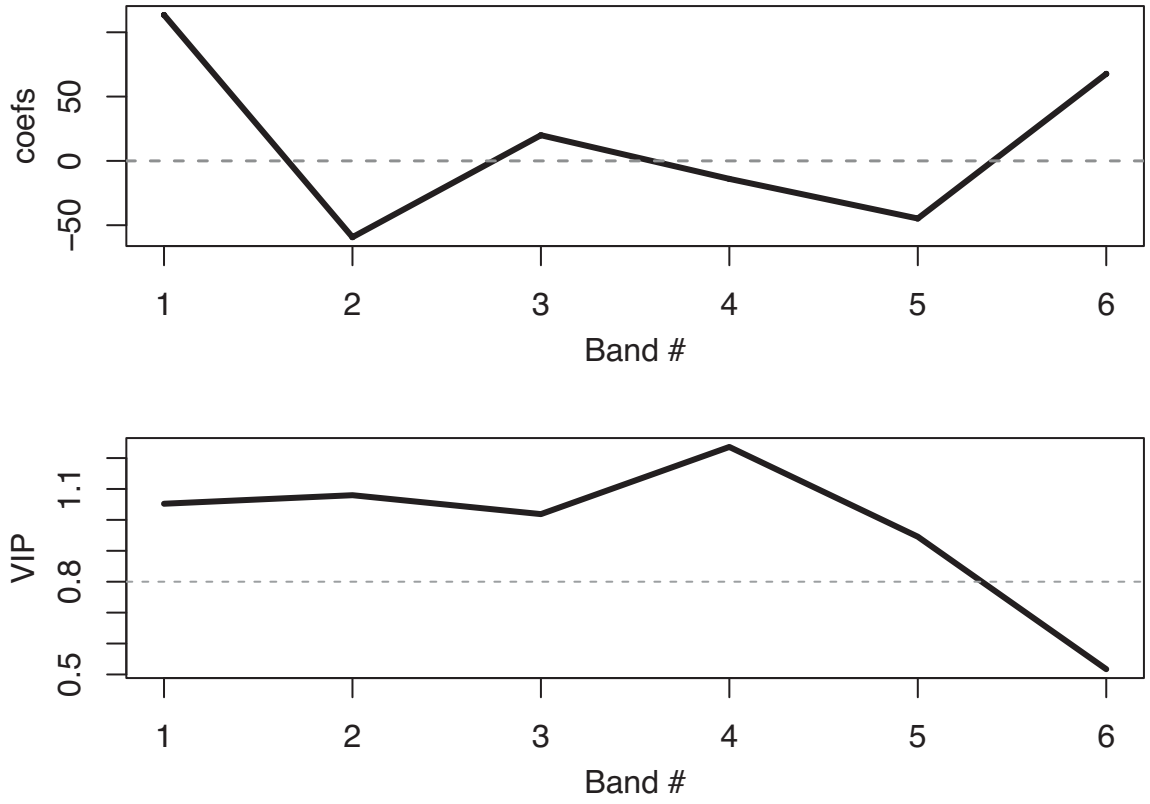

Figure A10-20: Landsat 7 - Coefficients (top panel) and Variable Importance Factors (VIP) (bottom panel) for each sensor band.

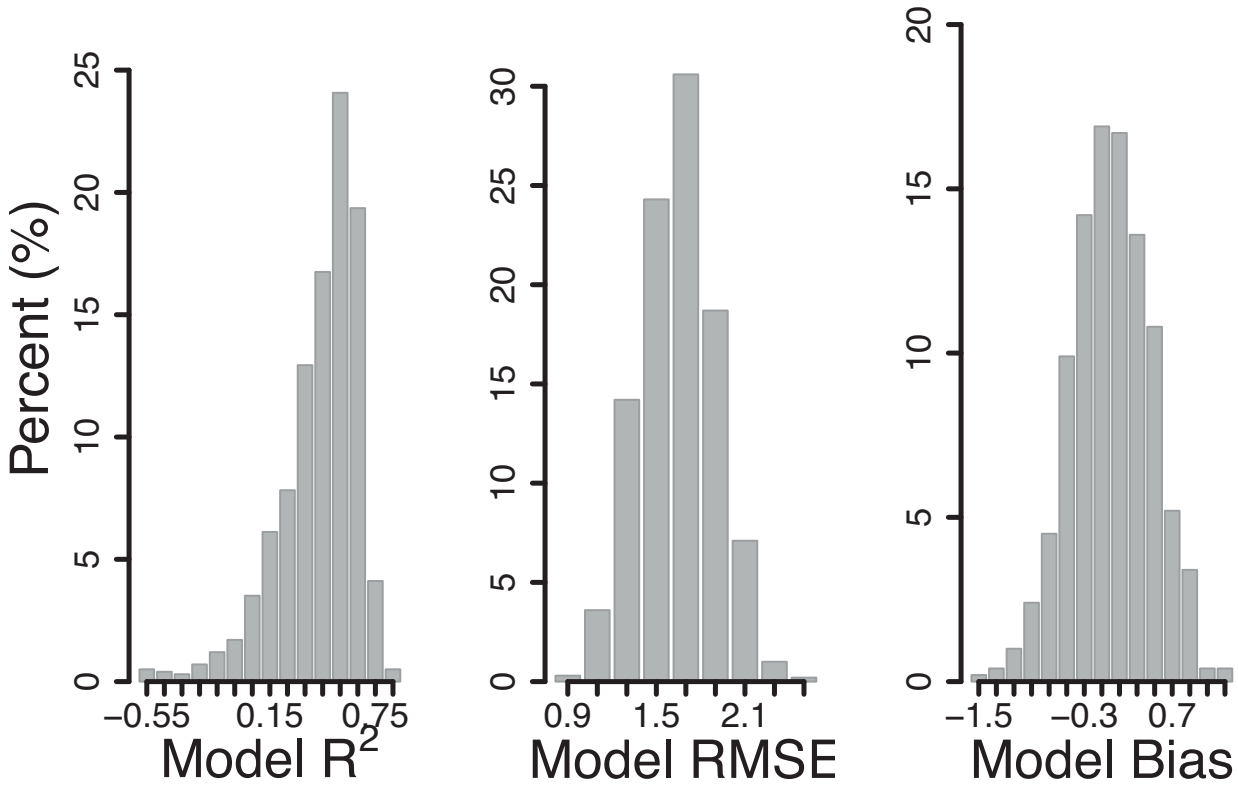

Figure A10-21: Landsat 7 - Model R^2, RMSE, and model bias at each iteration of the CV validation

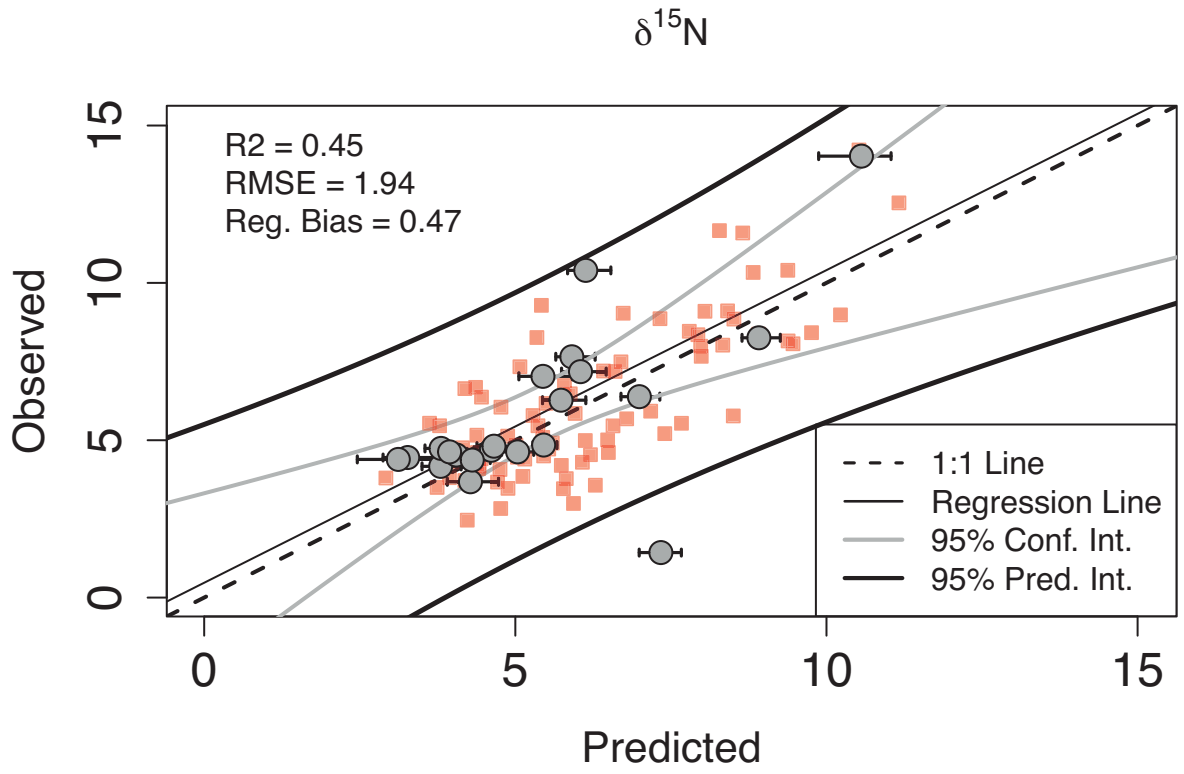

Figure A10-22: Landsat 7 - Validation results for the PLS model. Grey dots are predicted values from the PLS model using the validation set, error bars denote the 95% confidence intervals for each predicted value, red squares represent predicted values from the calibration set, grey lines represent 95% confidence intervals, thick black lines represent 95% prediction intervals, the dotted black line represents the 1:1 line, and the thin solid black line represents the regression. Values for R^2, RMSE, and model bias are shown in the top left.

Table A10-1: Results of the PLSR models for each satellite sensor

Sensor	No. components	R^2	RMSE
Landsat 4 TM	5	0.47	1.73
Landsat 5 TM	5	0.47	1.74
Landsat 7 ETM+	5	0.46	1.75

Predict $\delta^{15}N$

The derived relationship between guano spectra and $\delta^{15}N$ values was used to predict $\delta^{15}N$ values for Adélie penguin colony spectra. Adélie penguin colonies were identified from Landsat imagery using the algorithm outlined by Schwaller et al. (2013). All cloud free scenes that intersected known Adélie penguin colonies during December - January from 1984-2012 were used. $\delta^{15}N$ values were predicted for each pixel identified as guano in each scene.

Landsat scene availability

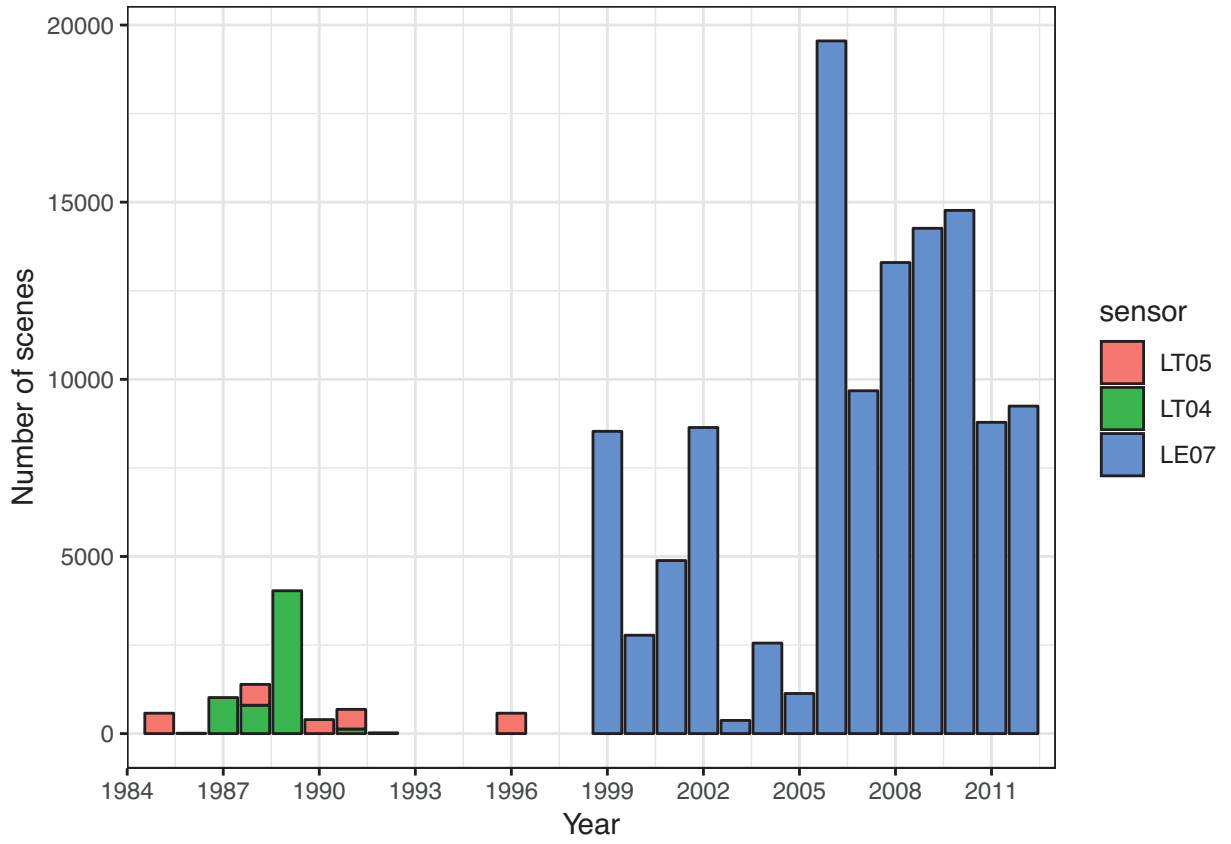

Figure A10-23: For each year, the number of cloud-free Landsat scenes that intersected known Adélie penguin colonies. Results sorted by satellite sensor. A total of 998 site/scenes were available over this period.

Calculate mean variance in PLS residuals

The residuals from the PLS analysis were used to decide the degree of observation error in $d^{15}N$ values in subsequent models (Equation A10-1).

Mean residual variation: 2.347

$\delta^{15}N$ **~ time**

$\delta^{15}N$ was modeled in a hierarchical Bayesian framework. Models were fit using the R package 'rjags' (Plummer 2016), to interface with JAGS (Plummer 2003) in the R statistical environment (R Core Team 2016). Inferences were obtained from 50,000 samples drawn from six chains with a thinning rate of 20, following a 'burn-in' period of 30,000 draws and an adaptation phase of 8000 draws. All computation was performed on high performance computing resources at the Institute for Advanced Computational Science - each compute node on this cluster consisted of two Intel Xeon E5-2683v3 CPUs operating at 2.0 Ghz, with 28 total cores and 128 GB of available RAM.

The following was used to assess changes in $\delta^{15}N$ through time:

$$y_{ijks} \sim N(z_{scene_{ijks}}, 2.35) \tag{A10-1}$$

$$z_{scene_{ijks}} \sim N(z_{pix_{ijk}}, \sigma^2_{pix})$$

$$z_{pix_{ijk}} \sim N(z_{year_{ij}}, \sigma^2_{year})$$

$$z_{year_{ij}} \sim N(\mu_{z_{ij}}, \sigma^2_z)$$

$$\mu_{z_{ij}} = \alpha_j + \beta_j * X_i$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix}, \Sigma\right)$$

$$\Sigma = \begin{pmatrix} \sigma^2_\alpha & \rho * \sigma_\alpha * \sigma_\beta \\ \rho * \sigma_\alpha * \sigma_\beta & \sigma^2_\beta \end{pmatrix}$$

$$\sigma_{scene} \sim U(0, 5)$$

$$\sigma_{year} \sim U(0, 5)$$

$$\sigma_z \sim U(0, 5)$$

$$\bar{\alpha} \sim N(0, 1000)$$

$$\bar{\beta} \sim N(0, 100)$$

$$\sigma_a \sim U(0, 20)$$

$$\sigma_b \sim U(0, 5)$$

$$\rho \sim U(-1, 1)$$

where i is year, j is site, k is a unique pixel for that site, s is scene number, y is the $\delta^{15}N$ value predicted by the PLSR model, z_{scene} is the $\delta^{15}N$ value for a given scene/pixel/site/year, z_{pix} is the mean $\delta^{15}N$ value for a given pixel/site/year, z_{year} is the mean $\delta^{15}N$ value for a given site/year, α is the intercept parameter, β is the slope parameter, X is year, ρ represents the correlation between α and β, and all σ parameters represent standard deviations.

Table A10-2: For each site used in the analysis, the site ID, median number of pixels over all Landsat scenes (indicative of colony size), and number of seasons of data. Sufficient Landsat imagery was available for 136 sites across the Antarctic continent.

site_id	med_pix	seasons_data
ADAR	587	10
AMBU	27	2
ANDR	2	3
ANDS	5	9
ARDL	48	2
ARTH	15	7
AUST	2	6
AVIA	48	7
BATT	77	6
BEAG	369	3
BEAU	24	13
BERK	8	7
BERT	8	3
BIEN	44	10
BOLI	4	12
BONG	2	2
BRAS	206	3
BRDN	45	13
BRDS	2	6
BROW	69	7
BSON	6	10
CHAL	171	12
CHAP	18	8
CHI1	1	2
CHI2	3	9
CHUN	17	11
CMID	5	2
COCK	3	6
CONE	6	4
COTT	46	10
CROZ	362	13
CRUZ	1	2
CSOU	28	9
CURZ	18	9
DARW	13	3
DAVI	6	9
DEMS	10	7
DENI	4	8
DEVI	10	5
DOU2	1	6
DOWN	7	11
DUKE	8	10
DURO	20	7
DURV	56	8
DUTH	16	2
EARL	29	4
EDEN	102	6
EDWI	112	12

Table A10-2 (cont.): For each site used in the analysis, the site ID, median number of pixels over all Landsat scenes (indicative of colony size), and number of seasons of data. Sufficient Landsat imagery was available for 136 sites across the Antarctic continent.

	site_id	med_pix	seasons_data
49	EMPE	8	4
50	FORB	1	9
51	FRAW	46	13
52	FRNC	3	6
53	GING	6	6
54	GNEY	45	12
55	GOUR	124	5
56	HASW	1	3
57	HBAY	66	9
58	HERO	322	3
59	HOLL	37	10
60	HOPE	33	4
61	IFOI	2	2
62	INEX	52	14
63	IVAN	2	9
64	JULE	86	11
65	KIRB	3	8
66	KIRT	2	5
67	KNOB	10	7
68	LAGO	9	5
69	LAUF	2	7
70	LEWI	4	7
71	LION	7	3
72	LLAN	35	3
73	LONG	2	13
74	LOVI	34	11
75	LOWT	6	11
76	LSAY	120	12
77	MACK	170	11
78	MADD	49	7
79	MADI	6	4
80	MAHE	1	8
81	MALL	4	8
82	MAWS	9	10
83	MBIS	388	5
84	MEDL	7	5
85	MIDG	18	7
86	MRII	1	3
87	NEIS	4	4
88	NMED	4	4
89	NORF	22	13
90	NUTT	1	2
91	ODBE	22	10
92	OLDH	3	10
93	PATE	87	2
94	PAUL	194	6
95	PEIN	11	2
96	PGEO	24	10

Table A10-2 (cont.): For each site used in the analysis, the site ID, median number of pixels over all Landsat scenes (indicative of colony size), and number of seasons of data. Sufficient Landsat imagery was available for 136 sites across the Antarctic continent.

	site_id	med_pix	seasons_data
97	PHIL	4	10
98	PIGE	94	11
99	PISL	23	11
100	PLAT	72	3
101	PMAR	70	11
102	POSS	364	10
103	PROC	8	6
104	PTHO	23	2
105	RBSN	1	5
106	REDI	6	6
107	RNVG	18	13
108	ROOK	53	13
109	ROYD	4	14
110	RRRI	3	5
111	RUMP	1	4
112	SAXU	4	3
113	SCUL	20	12
114	SHEE	14	6
115	SHEI	58	10
116	SHLY	30	8
117	SIMS	19	11
118	SOTH	3	2
119	STAN	1	6
120	STEI	1	5
121	STRA	32	2
122	SVEN	50	10
123	SVIS	75	12
124	TAYH	5	3
125	TRYN	25	13
126	UFSI	1	7
127	UNN7	2	7
128	VESN	95	13
129	VESS	325	11
130	VORT	4	5
131	WAIT	2	5
132	WATT	7	10
133	WAYA	5	11
134	WHTY	12	8
135	WIDE	6	5
136	WOPT	38	10

$\delta^{15}N$ ~ Sea Ice Concentration + Shelf Area

Sea ice concentration (SIC) data were derived from the satellite-based Nimbus 7, SMMR, and SSM/I-SSMIS passive microwave sensors, processed by the NASA Team algorithm (Cavalieri et al. 1995) via the National Snow and Ice Data

Center (Cavalieri et al. 1996). Mean sea ice concentration (fraction of water covered by sea ice) from December - January within a 150 km radius of breeding sites coincides with the maximum foraging range for this species during breeding season (Ainley 2002, Emmerson et al. 2015).

Continental shelf area (SA) was calculated within 150 km radius of breeding sites using bathymetry data from the International Bathymetric Chart of the Southern Ocean (https://www.scar.org/science/ibcso/ibcso/). Continental shelf area was defined as the total area of ocean floor less than 1000m depth, following Clarke et al. (1998).

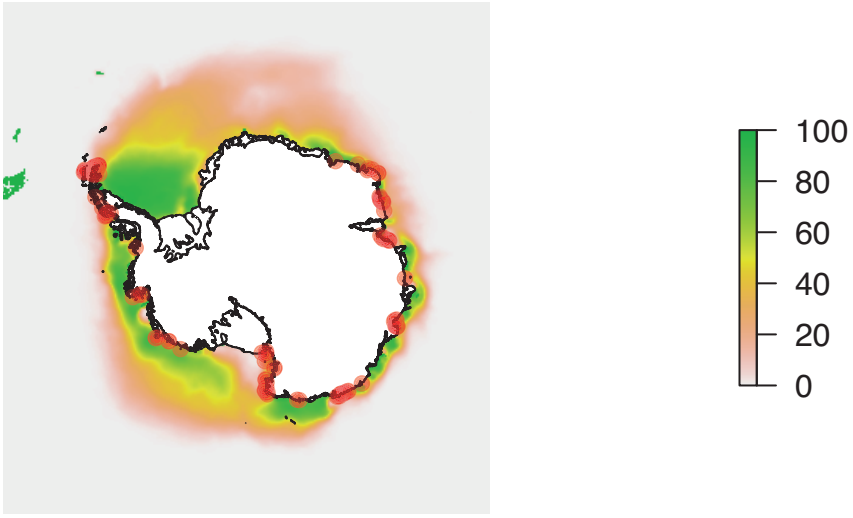

Figure A10-24: Average sea ice concentration (percent sea surface covered with ice) from Dec - Jan around the Antarctic continent. Red dots indicate Adélie penguin colonies considered in this study.

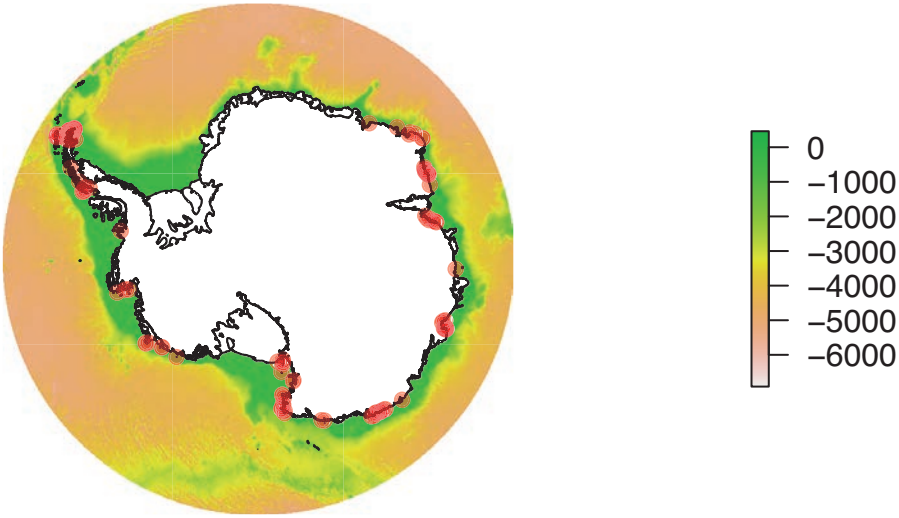

Figure A10-25: Sea floor depth (m) around the Antarctic continent. Red dots indicate Adélie penguin colonies considered in this study.

Inferences were obtained from 10,000 samples drawn from three chains, following a 'burn-in' period of 10,000 draws and an adaptation phase of 8000 draws.

The following was used to assess the effect of sea ice concentration and shelf area on $\delta^{15}N$ values: