Supplementary Material

Appendix S1. OpenBUGS code for estimating robin density in Bushy Park in 2016 using capture-mark-recapture data for six survey blocks, then using this estimate to derive density estimates for all years from 2002-2018 based on annual transect counts. The data are shown below.

```
model
{
    # priors
    p ~ dunif(0,1)    # detection probability for Lincoln-Petersen estimates
    for (i in 1:n.years) {
        alpha[i] ~ dnorm(0,0.5)        # log of the average density (robins/ha) for year i
        log(D.ave[i]) <- alpha[i]      # convert to real value
    }
    ESW ~ dnorm(0, 0.001)              # effective strip width in metres
    sd ~ dunif(0,2)                    # sd of log(density) among lines and survey blocks
    tau <- pow(sd,-2)                  # convert sd to precision

    # assign random effect for each line and survey block
    for (i in 1:n.lines) {
        re.line[i] ~ dnorm(0,tau)
    }
    for (i in 1:n.blocks) {
        re.block[i] ~ dnorm(0,tau)
    }

    # estimate robin density in 2016 (D.ave[15])
    for (i in 1:n.blocks) {        # for each of of 6 survey blocks in 2016...
        m2[i] ~ dbin(p,n1[i])          # obtain Lincoln-Petersen estimate for no. males in block (N.males[i])
        n2[i] ~ dbin(p,N.males[i])
        N.males[i] ~ dpois(mu.males[i])     # calc. expected number in block by sampling from Poisson
        log(mu.males[i]) <- log(D.ave[15]*0.5*A.block[i])+re.block[i]  # calculate local male density
    }

    # estimate robin density for other years from transect counts (year 1 = 2002)
    for (i in 1:n.years) {         # for each year...
        for (j in 1:n.lines) {          # for each transect line...
            count[i,j] ~ dpois(mu[i,j])          # sample no. robins counted based on expected number
            mu[i,j] <- D[i,j]*A.transect[i,j]  # obtain expected no. based on local D and effective area (A)
            A.transect[i,j] <- 2*ESW*Length[i,j]/10   # calculate effective A, giving estimate of ESW
            log(D[i,j]) <- log(D.ave[i])+re.line[j] # calculate local D based on average density that year
        }
    }
}
```

Data

```
list(

n.lines=11,       # number of transect lines
n.years=17,       # 2002-2018
n.blocks=6,       # number of robin survey blocks in 2016

# Length of each transect sampled (in km) each year from 2002-2017
# Line code: 1=L9, 2=L10, 3=L11, 4=L5-6S, 5=L6N, 6=L7N, 7=L8N, 8= L7S-8S, 9=L12-13, 10=L1-2,
11=L3-4
Length=structure(.Data =c(
0.451,0.525,0.525,0.835,0.663,0.600,0.555,0,0,0,0,
0.407,0.630,0.798,0.835,0.663,0.600,0.555,0,0,0,0,
0.541,0.682,0.551,0.835,0.663,0.600,0.555,0,0,0,0,
0.668,0.562,0.386,0.835,0.663,1.064,0.555,0.538,0.944,0.885,0.727,
0.668,0.750,0.684,0.835,0.663,0.600,0.555,0.538,0.944,0.885,0.727,
0.668,0.750,0.684,0.835,0.663,0.600,0.555,0.538,0.944,0.885,0.727,
0.668,0.750,0.428,0.835,0.663,0.600,0.555,0.538,0.944,0.885,0.727,
0.741,0.863,0.798,0.835,0.663,0.600,0.555,0.538,0.944,0.708,0.727,
0.167,0.750,0.684,0.835,0.530,0.258,0.555,0.179,0.708,0.664,0.545,
0.334,0.375,0.684,0.835,0.596,0.474,0.278,0.538,0.500,0.885,0.727,
0.535,0.750,0.684,0.835,0.663,0.511,0.555,0.538,0.859,0.885,0.727,
0.296,0.326,0.220,0.835,0.663,0.386,0.336,0.456,0.223,0.885,0.744,
0.668,0.715,1.228,0.796,0.383,0.458,0.560,0.538,0.534,0.885,0.727,
0.645,0.360,0.684,0.835,0.663,0.476,0.513,0.456,1.004,0.885,0.668,
0.645,0.750,0.684,0.835,0.663,0.600,0.513,0.456,1.004,0.885,0.668,
0.645,0.750,0.684,0.835,0.663,0.600,0.605,0.511,0.944,0.885,1.026,
0.507,0.623,0.468,0.835,0.663,0.600,0.513,0.456,0.589,0.885,0.668,
),.Dim = c(17, 11)),

# Number robins recorded on each transect each year from 2002-2018
count=structure(.Data =c(
1,2,0,0,4,0,3,NA,NA,NA,NA,
1,2,0,11,0,0,1,NA,NA,NA,NA,
2,2,2,4,1,1,2,NA,NA,NA,NA,
4,2,0,2,5,2,8,8,1,3,2,
2,0,0,6,2,4,3,0,1,3,4,
1,1,1,4,2,1,4,0,3,4,3,
0,2,1,7,0,4,9,1,2,2,0,
16,4,4,6,7,12,9,3,5,7,8,
8,7,8,2,8,9,10,7,8,2,6,
5,3,2,4,7,7,3,3,3,2,1,
10,11,13,17,8,11,12,21,17,15,29,
5,5,7,4,14,9,6,11,7,20,21,
11,7,12,15,5,8,14,10,2,6,10,
4,6,11,7,1,4,4,4,5,10,5,
16,15,16,24,11,17,18,12,22,13,9,
3,6,9,5,8,7,8,6,9,11,7,
14,12,13,11,7,10,7,12,19,7,7
),.Dim = c(17, 11)),

# Data for Lincoln-Petersen estimates in 6 survey blocks in October 2016
n1=c(4,4,8,10,9,7),                    # number of males banded on 1st occasion
m2=c(4,3,7,9,8,7),                     # number of banded males seen on 2nd occasion
n2=c(10,8,9,13,9,10),                  # total number of males seen on 2nd occasion
A.block=c(3.15,2.87,3.3,3.3,3.51,3.06) # area of survey block (ha)

)
```

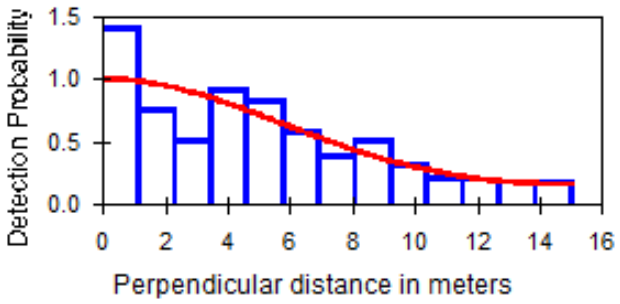
Appendix S2. OpenBUGS code for fitting logistic and exponential trends to annual density estimates (and standard errors) for bird species at Bushy Park.

```
model
{
    # priors
    alpha ~ dnorm(0,0.1)          # log(D) in first year, where D = density (individuals/ha)
    D0 <- exp(alpha)              # convert to real value
    r.ave ~ dnorm(0,0.1)          # intrinsic rate of increase after fence installed (2005)
    b.w ~ dnorm(0,0.1)            # potential negative effect of poor weather on estimate
    K ~ dunif(0,10)               # density at carrying capacity
    sd.year ~ dunif(0,1)          # sd in log(D) among years
    tau.year <- pow(sd.year,-2)   # convert sd to precision

    # model fitting
    for (i in 1:n.years) {                          # for each year ...
        logD.est[i] <- log(D.est[i])                # log-transform estimated D
        logD.SE[i] <- D.SE[i]/D.est[i]              # calculate SE for log(D) using delta method
        logD.tau[i] <- pow(logD.SE[i],-2)           # convert uncertainty in log(D) to precision
        logD.est[i] ~ dnorm(logD[i],logD.tau[i])    # sample real log(D)
        re.year[i] ~ dnorm(0,tau.year)              # sample this year's random effect on log(D)

        # fit logistic or exponential growth after fence installed in 2005
        D.ave[i] <- K/(1+((K-D0)/D0)*exp(-r.ave*max(0, year[i]-max(2005,year[1]))))
#       log(D.ave[i]) <- alpha+r.ave*max(0, year[i]-max(2005,year[1]))
        logD[i] <- log(max(0,D.ave[i]))+re.year[i]+b.w*weather[i] # add weather & random year effects
    }
}

Grey Warbler Data
n.years=13,      # inestimable in 2007, 2008, 2009, 2012
year=c(2002,2003,2004,2005,2006,2010,2011,2013, 2014, 2015,2016,2017,2018),
weather=c(0,0,0,0,0,0,1,0,0,1,0,1,0),  # poor weather in 2011, 2015, 2017
D.est=c(1.753,3.960,3.783,2.421,1.959, 0.319,0.126, 0.812,0.104,0.359,0.575,0.284,0.785),
D.SE=c(0.838,1.676,1.283,0.826,0.737,0.207,0.124,0.456,0.127,0.176,0.346,0.194,0.520)

Fantail Data
list(
n.years=17,
year=c(2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,201
8),
weather=c(0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0),  # poor weather in 2011, 2015, 2017
D.est=c(3.616,4.711,4.359,4.184,3.861,3.139,1.882,3.698,6.320,2.378,2.320,2.115,1.928,1.182,2.74
0,1.313,2.291),
D.SE=c(1.129,1.266,0.839,0.768,0.954,0.410,0.546,1.738,4.507,0.539,0.689,0.703,0.472,0.356,0.70
4,0.293,0.656)
)

Kereru Data
list(
n.years=17,
year=c(2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,201
8),
weather=c(0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0),  # poor weather in 2011, 2015, 2017
D.est=c(1.826,4.617,2.979,2.125,2.726,1.789,2.274,1.720,6.591,1.788,1.670,1.993,0.685,3.671,1.85
9,2.412,1.699),
D.SE=c(0.543,1.629,1.058,0.513,0.634,0.454,0.520,0.580,2.788,0.497,0.431,0.555,0.336,0.836,1.07
7,0.461,0.434))
```

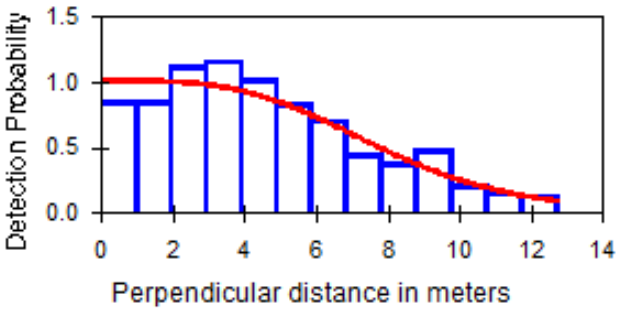
```
Silvereye Data
list(
n.years=14,      # inestimable in 2004, 2005, 2009
year=c(2002,2003, 2006,2007,2008, 2010,2011,2012,2013,2014,2015,2016,2017,2018),
weather=c(0,0,0,0,0,0,1,0,0,0,1,0,1,0),  # poor weather in 2011, 2015, 2017
D.est=c(0.866,2.082, 3.067,1.012,1.531, 1.492,2.700,1.850,0.387,1.813,0.256,0.540,0.106,0.128),
D.SE=c(0.835,0.771, 6.264,0.416,1.019, 1.145,2.268,0.829,0.281,1.150,0.170,0.446,0.101,0.132)
)

Tomtit Data
list(
n.years=13,      # inestimable in 2003, 2005, 2006, 2018
year=c(2002,2004,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017),
weather=c(0,0,0,0,0,0,1,0,0,0,1,0,1),  # poor weather in 2011, 2015, 2017
D.est=c(0.850,1.355,0.521,0.399,0.870,0.341,0.322,0.140,1.716,0.434,2.022,0.127,0.255),
D.SE=c(0.551,0.860,0.297,0.215,0.351,0.214,0.232,0.137,0.901,0.250,0.795,0.120,0.183)
)

Hihi Data
list(
n.years=6,
year=c(2013,2014,2015,2016,2017,2018),
weather=c(0,0,1,0,1,0),  # poor weather in 2015, 2017
D.est=c(2.576,2.619,0.325,4.384,0.681,1.845),
D.SE=c(1.118,0.819,0.237,1.402,0.262,0.666)
)

Saddleback Data
list(
n.years=13,
year=c(2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018),
weather=c(0,0,0,0,0,1,0,0,0,1,0,1,0),  # poor weather in 2011, 2015, 2017
D.est=c(0.265,0.513,0.798,0.506,1.741,1.190,6.117,8.404,7.837,1.830,8.466,4.451,7.776),
D.SE=c(0.126,0.471,0.287,0.179,0.432,0.473,1.191,1.448,1.288,0.446,1.670,0.618,1.176)
)

Robin Data
list(
n.years=17,
year=c(2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,201
8),
weather=c(0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0),  # poor weather in 2011, 2015, 2017
# from Distance
D.est=c(4.728,5.712,14.627,9.134,7.038,5.832,6.342,20.854,23.088,13.488,39.928,39.203,26.589,16
.911,43.764,18.107,33.726),
D.SE=c(2.109,3.783,3.015,2.911,1.949,1.343,2.365,4.507,4.728,3.172,5.957,6.778,7.070,2.998,5.51
0,2.111,5.37)

# from transects counts calibrated with capture-mark-recapture data
D.est=c(0.604,0.837,0.790,1.181,0.813,0.780,0.937,2.557,3.289,1.639,5.510,5.146,3.354,2.144,5.87
0,2.449,4.409),
D.SE=c(0.208,0.252,0.244,0.272,0.206,0.199,0.229,0.499,0.652,0.366,0.980,0.964,0.634,0.438,0.87
9,0.482,0.813)
)
```

Appendix S3. Distributions of perpendicular distances (histograms) for bird species observed in transect counts at Bushy Park, and detection functions (red lines) fitted using distance sampling. The detection function shown for each species was selected based on AICc and appropriate visual fit from six candidate models with different key functions and series expansions. These functions were used to estimate detection probability for all species except robins.

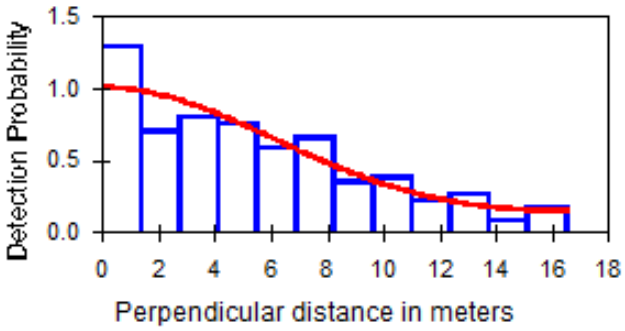
Fantail: Uniform / Cosine



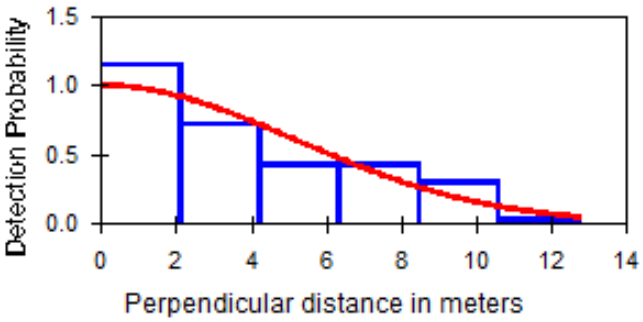
Saddleback: Half Normal Cosine



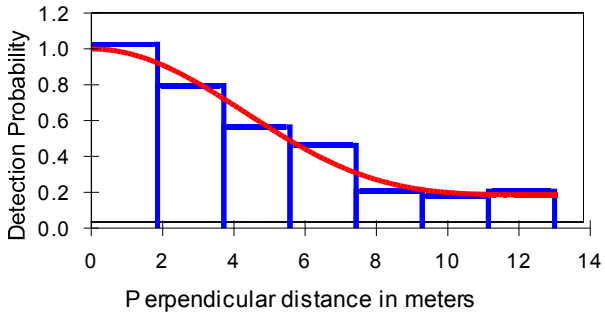
Kereru: Uniform / Cosine



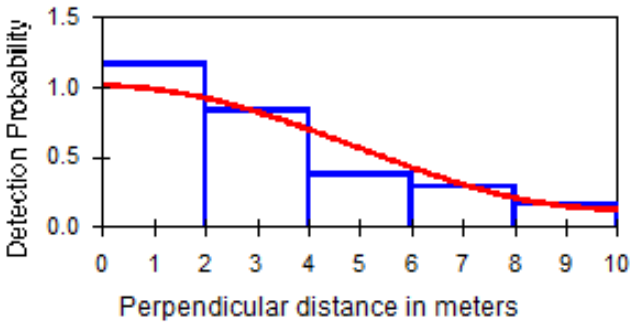
Silvereye: Half-normal / Cosine



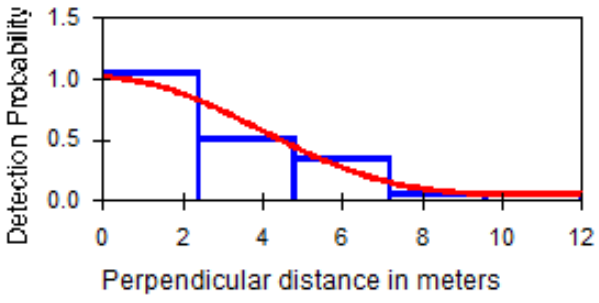
Grey warbler: Uniform Cosine



Tomtit: Uniform / Cosine



Hihi: Uniform Cosine



Robin: Half-normal / Cosine