Supplementary material to: "Are animals shrinking due to climate change? Temperature-mediated selection on body mass in mountain wagtails"

25 June 2018

This document presents the code used for the analysis in: "Are animals shrinking due to climate change? Temperature-mediated selection on body mass in mountain wagtails".

Trends in climate

We first explored the climate data – temperature and rainfall – for trends, using state-space models that allowed us to account for seasonal effects.

This code chunck reads in the data and prepares the temperature variable as a time series.

```
my_data <- read.csv("WagtailWeather.csv")
names(my_data)
```

```
## [1] "Year"   "Season" "Temp"   "Rain"
```

```
temp_1 <- my_data$Temp
temp_2 <- as.matrix(temp_1)
temp = ts(temp_2, start = 1976, frequency=4)
```

Then, we fit a state-space model to these data.

State Space Model for temperature

```
library(dlm)
library(Hmisc)
```

```
fit.st <- StructTS (temp, type = "BSM")

buildFun <- function(x){
  dlmTemp <- dlmModPoly(2) + dlmModSeas(4)
  diag(W(dlmTemp))[2:3] <- exp(x[1:2])
  V(dlmTemp) <- exp(x[3])
  return(dlmTemp)
}

buildFun2 <- function(x){
  dlmTemp <- dlmModPoly(2) + dlmModSeas(4)
  diag(W(dlmTemp))[2] <- exp(x[1])
  V(dlmTemp) <- exp(x[2])
  return(dlmTemp)
}

fit <- dlmMLE(temp, parm = rep(0, 3), build = buildFun)
fit2 <- dlmMLE(temp, parm = rep(0, 2), build = buildFun2)
```

```r
dlmTemp <- buildFun(fit$par)
drop(V(dlmTemp))
```

```
## [1] 0.1541838
```

```r
diag(W(dlmTemp))[2:3]
```

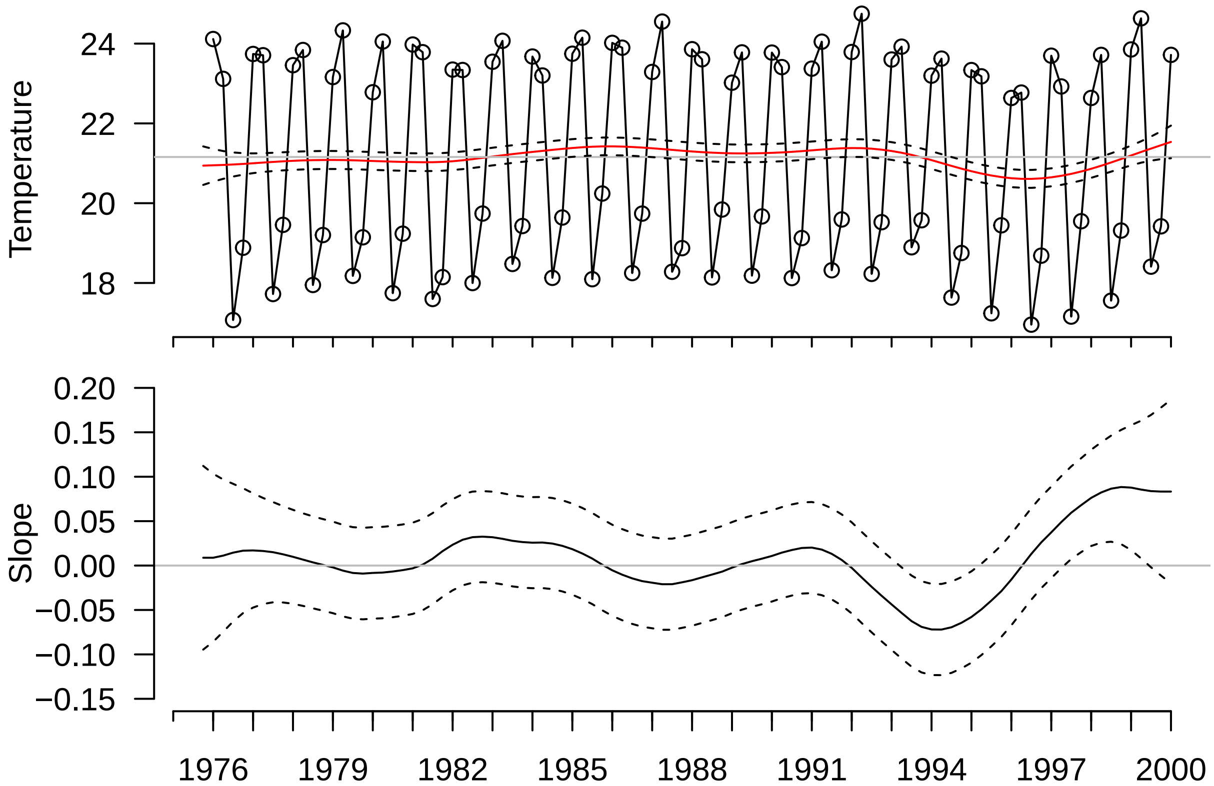
```
## [1] 4.543804e-04 2.542649e-10
```

```r
tempSmooth <- dlmSmooth(temp, mod = dlmTemp)
sm <- ts(tempSmooth$s[-1], start = 1977, frequency = 4)
```

Plot the model fit:

```r
  par(mfrow=c(2,1), mar=c(0,4,1,1), oma=c(3,0,0,0))
  plot(temp, type = "o", cex.lab = 1.5, cex.axis = 1.5, las = 1, xlim = c(1975.5, 2000),
       ylab = "Temperature", axes=F)
  minor.tick(nx = 5, ny = 0, tick.ratio = 0.5)
  axis(2, las=1)

  lines(tempSmooth$s[, 1], col = 2, type = "l", pch = 16, cex = 0.7)
  abline(h = mean(temp), col = "grey")

# ----- confidence intervals ------
  U.S <- tempSmooth$U.S
  D.S <- tempSmooth$D.S

  v <- dlmSvd2var(U.S, D.S)
  vars <- c()

  for (i in 1:(length(temp) + 1))
   { vars[i] <- v[[i]][1, 1]
   }

  pl <- tempSmooth$s[, 1] + qnorm(0.025, sd = sqrt(vars))
  pu <- tempSmooth$s[, 1] + qnorm(0.975, sd = sqrt(vars))

  lines(pl, lty = 2)
  lines(pu, lty = 2)

  plot(tempSmooth$s[, 2], ylab="Slope", xlab="Year", axes=F, ylim = c(-0.15, 0.2),
       xlim = c(1975.5, 2000), type = "l")
  abline(h=0, col="grey")
  minor.tick(nx = 5, ny = 0, tick.ratio = 0.5)
  axis(1, at=c(1976:2000))
  axis(2, las=1)

  for (i in 1:(length(temp) + 1))
  { vars[i] <- v[[i]][2, 2]
  }

  pl <- tempSmooth$s[, 2] + qnorm(0.025, sd = sqrt(vars))
  pu <- tempSmooth$s[, 2] + qnorm(0.975, sd = sqrt(vars))
```

```r
  lines(pl, lty = 2)
  lines(pu, lty = 2)
```



State Space Model for rainfall

```r
rain_1 <- my_data$Rain
rain_2 <- as.matrix(rain_1)
rain <- ts(rain_2, start=1976, frequency=4)

fit.st <- StructTS (rain, type = "BSM")

buildFun <- function(x){
  dlmTemp <- dlmModPoly(2) + dlmModSeas(4)
  diag(W(dlmTemp))[2:3] <- exp(x[1:2])
  V(dlmTemp) <- exp(x[3])
  return(dlmTemp)
}

buildFun2 <- function(x){
  dlmTemp <- dlmModPoly(2) + dlmModSeas(4)
  diag(W(dlmTemp))[2] <- exp(x[1])
  V(dlmTemp) <- exp(x[2])
  return(dlmTemp)
}
```

```r
fit <- dlmMLE(rain, parm = rep(0, 3), build = buildFun)
fit2 <- dlmMLE(rain, parm = rep(0, 2), build = buildFun2)

fit <- fit2

dlmTemp <- buildFun2(fit$par)
drop(V(dlmTemp))
```

```
## [1] 12149.89
```

```r
diag(W(dlmTemp))[2:3]
```

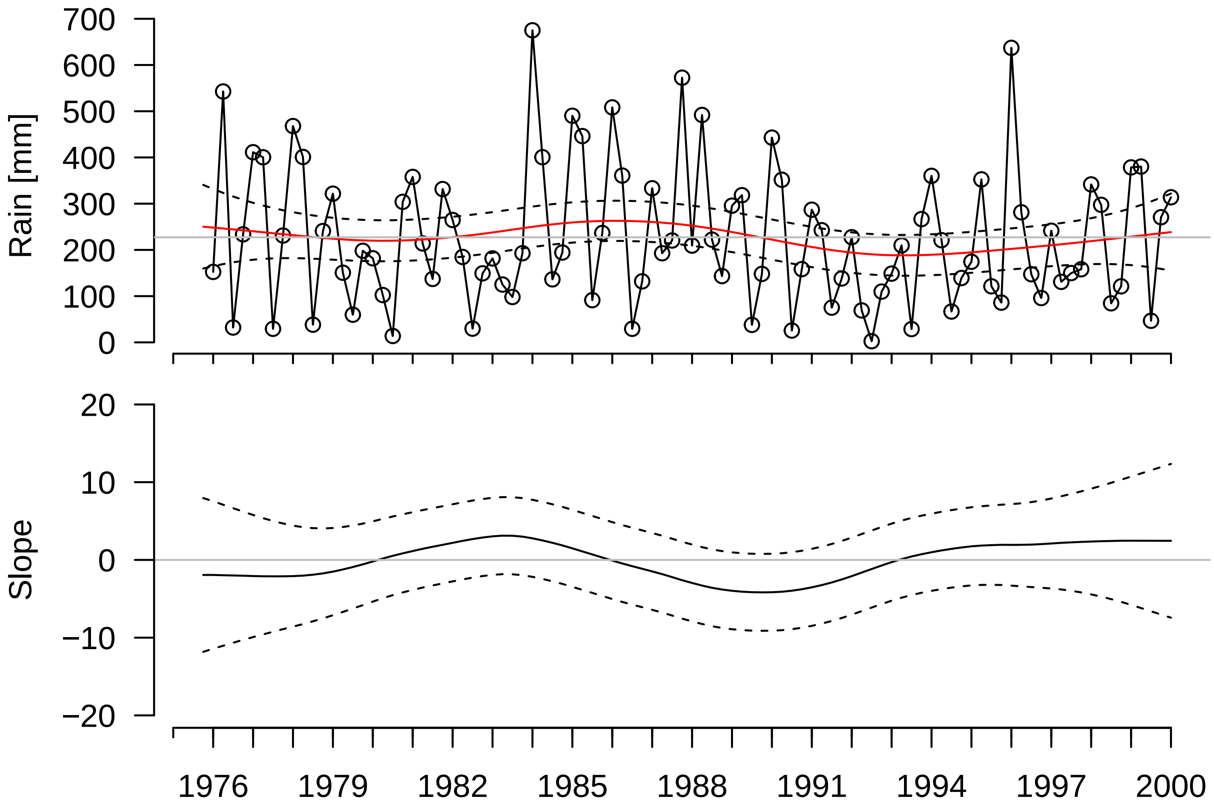
```
## [1] 2.04632 1.00000
```

```r
tempSmooth <- dlmSmooth(rain, mod = dlmTemp)
sm <- ts(tempSmooth$s[-1], start = 1977, frequency = 4)
```

Plotting the model fit:

```r
par(mfrow=c(2, 1), mar=c(0,4,1,1), oma=c(3,0,0,0))

plot(rain, type = "o", cex.lab = 1.5, cex.axis = 1.5, las = 1,
     ylab ="Rain [mm]", axes=F, xlim = c(1975.5, 2000))
 minor.tick(nx = 5, ny = 0, tick.ratio = 0.5)
 axis(2, las=1)

 lines(tempSmooth$s[, 1], col = 2, type = "l", pch = 16, cex = 0.7)
 abline(h = mean(rain), col = "grey")

# ----- confidence intervals ------
U.S <- tempSmooth$U.S
D.S <- tempSmooth$D.S

v <- dlmSvd2var(U.S, D.S)
vars <- c()

for (i in 1:(length(rain) + 1))
{ vars[i] <- v[[i]][1, 1]
}

pl <- tempSmooth$s[, 1] + qnorm(0.025, sd = sqrt(vars))
pu <- tempSmooth$s[, 1] + qnorm(0.975, sd = sqrt(vars))

lines(pl, lty = 2)
lines(pu, lty = 2)

## slope
plot(tempSmooth$s[, 2], ylab="Slope", xlab="Year", axes=F, xlim = c(1975.5, 2000),
     ylim = c(-20, 20))
abline(h=0, col="grey")
minor.tick(nx = 5, ny = 0, tick.ratio = 0.5)
axis(2, las=1)

for (i in 1:(length(rain) + 1))
{ vars[i] <- v[[i]][2, 2]
```

```
}

pl <- tempSmooth$s[, 2] + qnorm(0.025, sd = sqrt(vars))
pu <- tempSmooth$s[, 2] + qnorm(0.975, sd = sqrt(vars))

lines(pl, lty = 2)
lines(pu, lty = 2)
axis(1, at=c(1976:2000))
```



Mass trends

Then, we look at trends in the body mass of mountain wagtails. We first see whether there is any evidence for a linear change in mass over years and whether mass changes seasonally.

```
temp <- read.csv("WagtailMass.csv")

temp$time <- temp$year + (temp$season-1)/4  # time is now continuous, in quarters

temp$season <- as.factor(temp$season)

out.lm1 <- lm(mass ~ I(time-mean(time)) * season, data = temp)
summary(out.lm1)  # there is a clear year effect
```

```
##
## Call:
```

```
## lm(formula = mass ~ I(time - mean(time)) * season, data = temp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2364 -0.7344 -0.0462  0.8109  3.2804
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 19.739398   0.101340 194.784  < 2e-16 ***
## I(time - mean(time))        -0.044139   0.014923  -2.958  0.00323 **
## season2                     -0.190895   0.131751  -1.449  0.14792
## season3                      0.095626   0.152923   0.625  0.53201
## season4                      0.013339   0.169225   0.079  0.93720
## I(time - mean(time)):season2  0.027479   0.018454   1.489  0.13703
## I(time - mean(time)):season3 -0.009618   0.022241  -0.432  0.66557
## I(time - mean(time)):season4 -0.010075   0.026492  -0.380  0.70387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.219 on 566 degrees of freedom
## Multiple R-squared:  0.05705,    Adjusted R-squared:  0.04539
## F-statistic: 4.892 on 7 and 566 DF,  p-value: 2.236e-05
```

```r
anova(out.lm1)    # but no evidence for mass to depend on season
```

```
## Analysis of Variance Table
##
## Response: mass
##                              Df Sum Sq Mean Sq F value    Pr(>F)
## I(time - mean(time))          1  35.17  35.169 23.6735 1.482e-06 ***
## season                        3   7.84   2.614  1.7595    0.1538
## I(time - mean(time)):season   3   7.86   2.620  1.7639    0.1529
## Residuals                   566 840.84   1.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
out.lm2 <- lm(mass ~ I(time-mean(time)) + season, data = temp)
summary(out.lm2) # removing season-time interaction does not change the time effect
```

```
##
## Call:
## lm(formula = mass ~ I(time - mean(time)) + season, data = temp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1087 -0.7040 -0.0137  0.8387  3.1796
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          19.749029   0.100668 196.180  < 2e-16 ***
## I(time - mean(time)) -0.034705   0.007319  -4.741 2.68e-06 ***
## season2              -0.206789   0.131169  -1.577    0.115
## season3               0.096588   0.152229   0.634    0.526
## season4              -0.051253   0.159658  -0.321    0.748
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.221 on 569 degrees of freedom
## Multiple R-squared:  0.04823,    Adjusted R-squared:  0.04154
## F-statistic: 7.209 on 4 and 569 DF,  p-value: 1.148e-05
```

```
out.lm3 <- lm(mass ~ time, data = temp)
summary(out.lm3) # removing season does not change the time effect
```

```
##
## Call:
## lm(formula = mass ~ time, data = temp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2486 -0.6900 -0.0245  0.8340  3.1944
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 89.163263  14.337011   6.219 9.66e-10 ***
## time        -0.034905   0.007203  -4.846 1.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.224 on 572 degrees of freedom
## Multiple R-squared:  0.03944,    Adjusted R-squared:  0.03776
## F-statistic: 23.49 on 1 and 572 DF,  p-value: 1.623e-06
```

```
coef(out.lm3)[2] * 10/mean(temp$mass)  # percent decline in mass per decade
```

```
##        time
## -0.01773335
```

```
1+coef(out.lm3)[2] * 10/mean(temp$mass)  # percent decline in mass per decade
```

```
##      time
## 0.9822667
```

```
(1+coef(out.lm3)[2] * 10/mean(temp$mass))^0.75  # change in metabolic rate
```

```
##      time
## 0.9866703
```

Then we want to separate the among-individual and within-individual effects, following van de Pol and Wright 2009, Animal Behaviour.

```
# calculate mean time per individual
meantime <- aggregate(temp$time, by = list(temp$ringno), FUN=mean)
colnames(meantime) <- c("ringno", "meantime")

# add mean time to data frame
temp <- merge(temp, meantime, by.x = "ringno", by.y = "ringno", all.x=T)
temp$dtime <- temp$time - temp$meantime  # difference in time within individual

library(nlme)

# almost the entire effect is due to heavier individuals being replaced by lighter individuals
out.lme1 <- lme(mass ~ dtime + meantime, random =~1|ringno, data=temp)
```

```r
summary(out.lme1)
```

```
## Linear mixed-effects model fit by REML
##  Data: temp
##        AIC      BIC    logLik
##   1748.062 1769.799 -869.0311
##
## Random effects:
##  Formula: ~1 | ringno
##         (Intercept)  Residual
## StdDev:    1.038572 0.7699628
##
## Fixed effects: mass ~ dtime + meantime
##                Value Std.Error  DF   t-value p-value
## (Intercept) 73.03214 20.772130 287  3.515871  0.0005
## dtime       -0.00857  0.026532 287 -0.323024  0.7469
## meantime    -0.02686  0.010434 284 -2.574271  0.0106
##  Correlation:
##          (Intr) dtime
## dtime     0
## meantime -1      0
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.79528290 -0.46563894 -0.04904805  0.46075842  3.71350367
##
## Number of Observations: 574
## Number of Groups: 286
```

```r
# proportion of trend explained by among-individual effect
fixef(out.lme1)[3]/sum(fixef(out.lme1)[2:3])
```

```
##  meantime
## 0.7581056
```

Then look at detailed trend using GAM

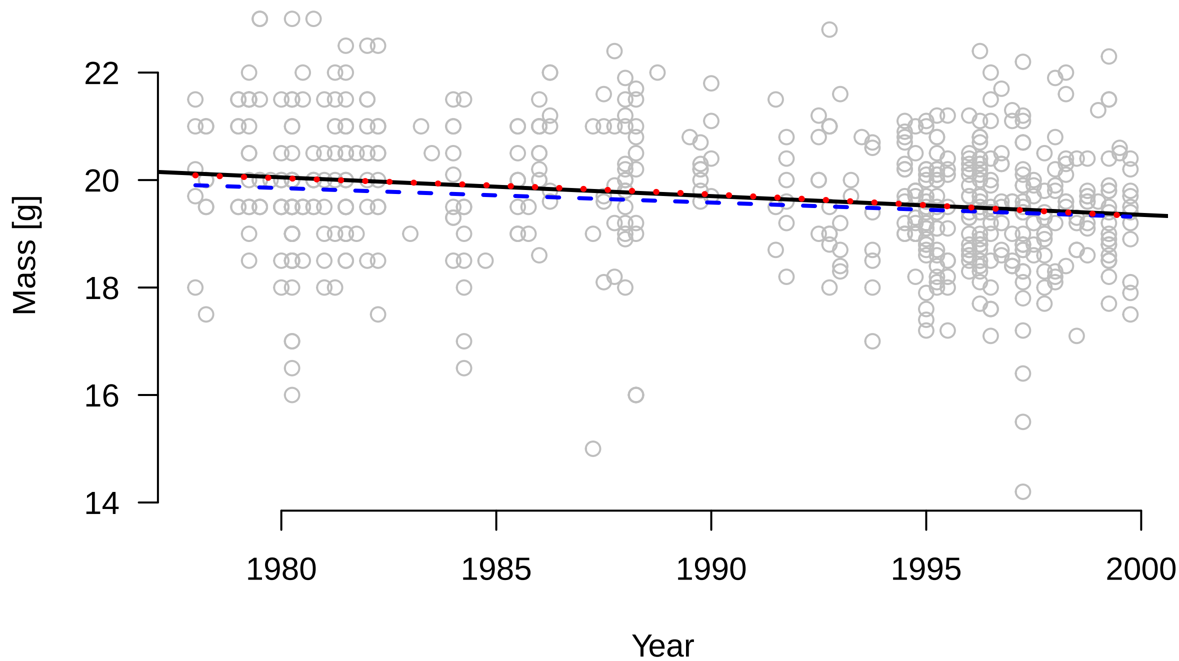
```r
library(mgcv)
```

```
## This is mgcv 1.8-23. For overview type 'help("mgcv-package")'.
```

```r
gam.mass <- gam(mass ~ s(time), data = temp)
```

Finally plot it all:

```r
plot(mass~time, data=temp, xlab="Year", ylab="Mass [g]", axes=F, col="grey")
axis(1)
axis(2,las=1)
abline(out.lm3, lwd=2)
x <- seq(min(temp$meantime), max(temp$meantime), length=50)
lines(x,fixef(out.lme1)[1] + fixef(out.lme1)[3]*x,lty=2, col="blue", lwd=2)
od<-order(temp$time)
lines(temp$time[od], predict(gam.mass)[od], lty=3, col="red", lwd=3)
```


22
20
18
16
14
Mass [g]
1980
1985
1990
1995
2000
Year