EEB313 Project: Complete cleaned-up code

2022-12-08

##Setup: required libraries

```
## Warning: package 'rgdal' was built under R version 4.2.2

## Warning: package 'mapproj' was built under R version 4.2.2

## Warning: package 'usethis' was built under R version 4.2.2

## Warning: package 'ggmap' was built under R version 4.2.2

## Warning: package 'mapview' was built under R version 4.2.2

## Warning: package 'arules' was built under R version 4.2.2

## Warning: package 'report' was built under R version 4.2.2

## Warning: package 'geosphere' was built under R version 4.2.2

## Warning: package 'osmdata' was built under R version 4.2.2

## Warning: package 'arsenal' was built under R version 4.2.2
```

##Input data

```
morph <- read.csv("Morphology_metadata2.csv")
```

##PCA

```
morph_means <- morph %>%
  group_by(Species) %>%
  filter(!is.na(Wing) & !is.na(Culmen) & !is.na(Mass)) %>%
  summarize(mean(Wing), mean(Culmen), mean(Mass))
```

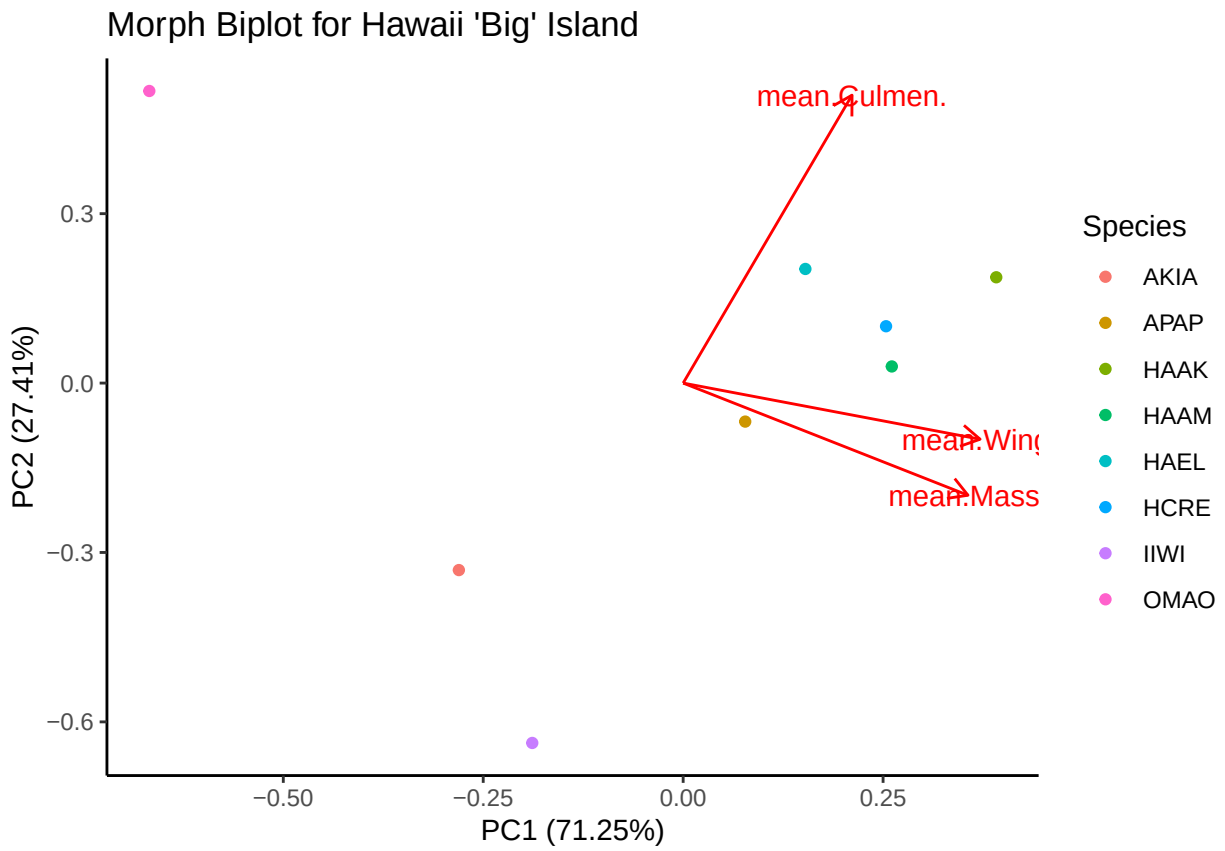
```
df <- data.frame(morph_means[,-1])

morph_pc <- prcomp(df, scale = TRUE, center = TRUE, retx = T)
summary(morph_pc)
```

```
## Importance of components:
##                           PC1    PC2     PC3
## Standard deviation     1.4620 0.9069 0.20055
## Proportion of Variance 0.7125 0.2741 0.01341
## Cumulative Proportion  0.7125 0.9866 1.00000
```

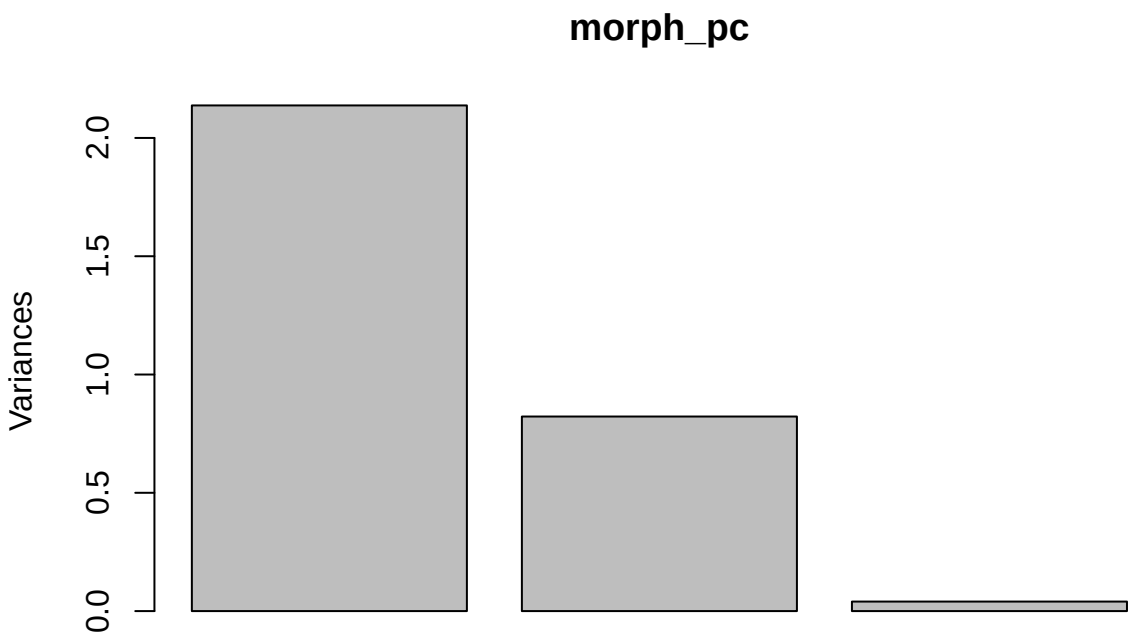
```
morph_pc$rotation <- -1*morph_pc$rotation
morph_pc$rotation
```

```
##                     PC1        PC2        PC3
## mean.Wing.    0.6676279 -0.1788466  0.7226942
## mean.Culmen. 0.3797288  0.9167602 -0.1239225
## mean.Mass.    0.6403741 -0.3571619 -0.6799679
```

```
autoplot(morph_pc, data = morph_means, colour = 'Species', loadings = TRUE,
         loadings.label = TRUE) +
labs(title = "Morph Biplot for Hawaii 'Big' Island") +
theme_classic()
```




```
plot(morph_pc)
```



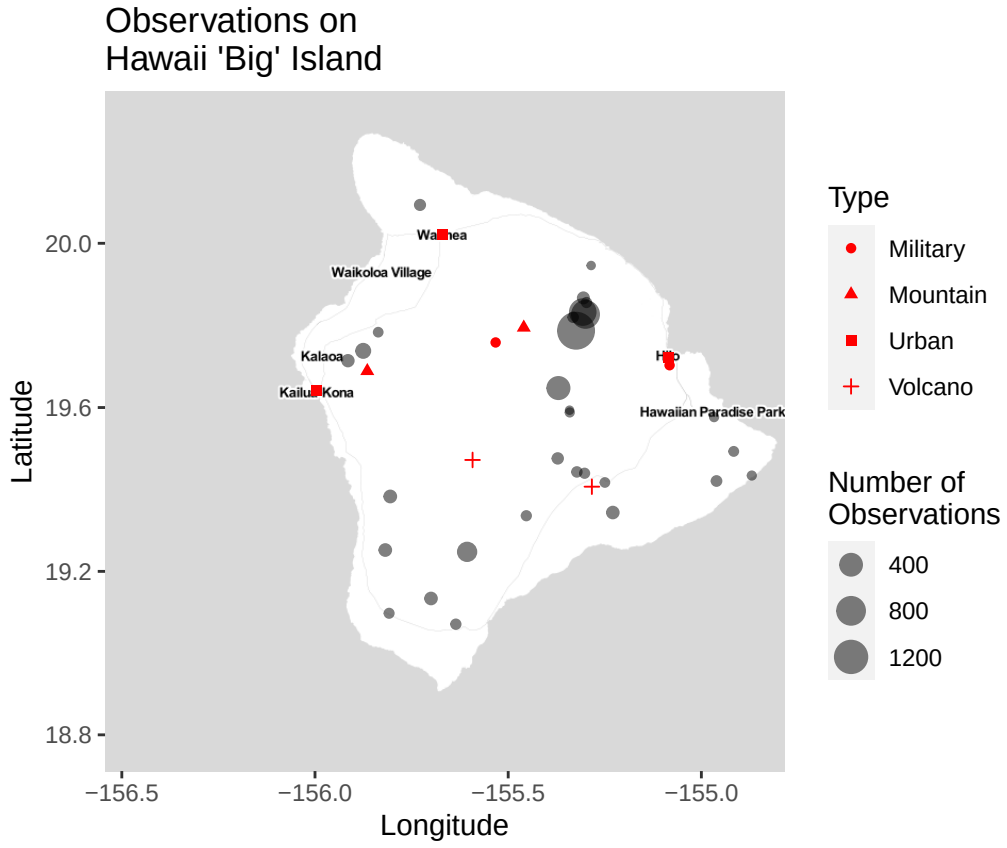
##Conversion to lat-long and adding landmarks

```
# Identifies ESPG as 6334.
morph_sf <- st_as_sf(morph, coords = c("UTM_X", "UTM_Y"), crs = 6334)

# Transforms coordinates to standard lat-lon format.
morph_wgs84 <- st_transform(morph_sf, crs = 4326)

# Converts geometry class into separate lat-lon columns.
morph <- morph_wgs84 %>%
  extract(geometry, c('lon', 'lat'), '\\((.*), (.*)\\)', convert = TRUE) %>%
  as.data.frame()

# Selects for relavent columns.
morph_spa <- morph %>%
  select(Species, Date, Sex, Wing, Culmen, Mass, lat, lon)

# Previews the data.
#head(morph_spa)

# Uses ggmap to extract map of Hawaii.
register_google(key = "AIzaSyCAOX4U9l7oFpljVfFnF1U-IW-0ve92QTk")
map_hi <- get_map(location = 'Island of Hawaii', zoom = 9, source = "stamen",
                  maptype = "toner-lite")

# List of natural and anthropogenic effects that can be analyzed.
```

```
effects <- list(Mauna_Loa = c(geocode("Mauna Loa"), Type = "Volcano"),
                Kīlauea = c(geocode("Kīlauea"), Type = "Volcano"),
                Hilo = c(geocode("Hilo"), Type = "Urban"),
                Waimea = c(geocode("Waimea"), Type = "Urban"),
                Kona = c(geocode("Kona"), Type = "Urban"),
                Hualalai = c(geocode("Hualalai"), Type = "Mountain"),
                Mauna_Kea = c(geocode("Mauna Kea Access Rd, Hilo, HI 96720, United States"),
                              Type = "Mountain"),
                Pohakuloa = c(geocode("Pohakuloa Training Area"), Type = "Military"),
                Army_Reserve = c(geocode("470 W Lanikaula St, Hilo, HI 96720, United States"),
                                 Type = "Military"))
```

```
## Warning: "Kona" not uniquely geocoded, using "kailua-kona, hi, usa"
```

```
# Formats list as a dataframe.
effects <- as.data.frame(bind_rows(effects, .id = "Name"))

# Plots the data.

ggmap(map_hi) +
  geom_count(data = morph_spa, aes(x = lon, y = lat), alpha = 0.5) +
  labs(title = "Observations on \nHawaii 'Big' Island", x = "Longitude", y = "Latitude",
       size = "Number of \nObservations") +
  geom_point(data = effects, aes(x = lon, y = lat, shape = Type), color = "Red")
```

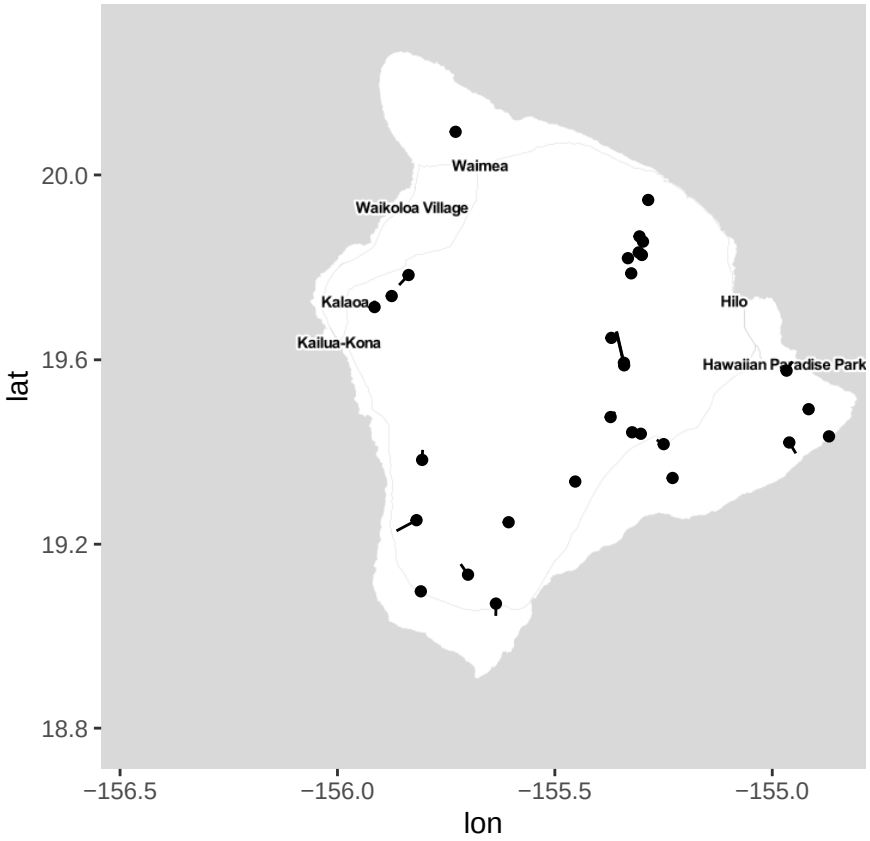


##Distance to landmarks calculation

```
#extract points lat-long data into a matrix
points <- as.matrix(morph_spa[,7:8])

#names of the distance columns in a list
name_loc <- c("dist_mau_loa","dist_kil","dist_hil","dist_wai","dist_kon",
              "dist_hua","dist_mau_kea","dist_poh","dist_arm_res")

#Distance calculation for loop
for(i in 1:9){
  loc <- as.numeric(c(effects[i,3], effects[i,2]))
  dist_x <-  as.data.frame(spDistsN1(points,loc,longlat=T))
  colnames(dist_x) <- name_loc[i]
  morph_spa <- cbind(morph_spa,dist_x)
}
```

##Distance to shoreline calculation

```
# Uses the previously created sf_object to identify unique sample sites.
morph_wgs84_distinct <- morph_wgs84 %>%
  distinct(geometry)



# Creates a bounding box of Hawaii coastline data.
osm_box <- getbb(place_name = "Hawaii") %>%
  opq() %>%
  add_osm_feature("natural", "coastline") %>%
  osmdata_sf()


# Uses the dist2line function in geosphere.
dist_coast <- geosphere::dist2Line(p = st_coordinates(morph_wgs84_distinct),
                        line = st_coordinates(osm_box$osm_lines)[,1:2])

# Creates a lat lon version of unique sampling sites.
morph_spa_distinct <- morph_spa %>%
  distinct(lat, lon)

# Combine initial data with distance to coastline.
morph_spa_coast.distance <- cbind(morph_spa_distinct %>%
                                    rename(y = lat, x = lon), dist_coast) %>%
                                    mutate(kilometers = distance/1000)

# Plot distances
ggmap(map_hi) +
  geom_point(data = morph_spa_distinct,
             aes(x = lon, y = lat)) +
  geom_segment(data = morph_spa_coast.distance,
               aes(x = x,
                   y = y,
                   xend = lon,
                   yend = lat))
```




```
morph_spa_coast.binned <- morph_spa_coast.distance %>%
  mutate(shr_bin = discretize(kilometers, method="interval", breaks=3,
                              labels=c("near","middle","far")))
```

```
morph_spa <- left_join(morph_spa, morph_spa_coast.binned %>%
                         select(y, shr_bin),
                       by = c("lat" = "y"))
morph_spa$shr_bin <- as.character(morph_spa$shr_bin)
```

##Filtering out N/A values

```
#Filter out NAs
morph_spa_noNA <- morph_spa %>%
  filter(!is.na(Wing) & !is.na(Culmen) & !is.na(Mass))
```

##Binning distance to random effects

```
#Sort Natural Landmarks (random effect) into categories
morph_spa_bins <- morph_spa_noNA %>%
  mutate(mau_loa_bins = discretize(dist_mau_loa, method="interval", breaks=3,
                                   labels=c("near","middle","far"))) %>%
  mutate(kil_bins = discretize(dist_kil, method="interval", breaks=3,
                               labels=c("near","middle","far"))) %>%
  mutate(hua_bins = discretize(dist_hua, method="interval", breaks=3,
```

```
                                  labels=c("near","middle","far"))) %>%
  mutate(mau_kea_bins = discretize(dist_mau_kea, method="interval", breaks=3,
                                   labels=c("near","middle","far")))
```

##Numericise Year

```
#Extract last 2 digits -- Years as numeric variable
morph_spa_bins <- morph_spa_bins %>%
  mutate(Year=as.numeric(str_sub(morph_spa_bins$Date,-2,-1)))
```

##Base Linerar Mixed Models

Here are the base linear mixed models, one for each of the morphological traits. Note that all fixed effects, random effects and covariate (year) are included.

```
#Mixed model for Wing
wing_lmer <- lmer(Wing~Year+dist_hil+dist_wai+dist_kon+dist_poh+dist_arm_res+
                    (1|mau_loa_bins)+(1|kil_bins)+(1|hua_bins)+(1|mau_kea_bins)+
                    (1|shr_bin),data=morph_spa_bins,REML=F)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(wing_lmer)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula:
## Wing ~ Year + dist_hil + dist_wai + dist_kon + dist_poh + dist_arm_res +
##     (1 | mau_loa_bins) + (1 | kil_bins) + (1 | hua_bins) + (1 |
##     mau_kea_bins) + (1 | shr_bin)
##    Data: morph_spa_bins
##
##      AIC      BIC   logLik deviance df.resid
##  19231.8  19308.0  -9602.9  19205.8     2600
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2325 -0.7888 -0.1196  0.5708  3.6110
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  mau_loa_bins (Intercept)  0.00    0.000
##  kil_bins     (Intercept)  0.00    0.000
##  hua_bins     (Intercept)  0.00    0.000
##  mau_kea_bins (Intercept)  0.00    0.000
##  shr_bin      (Intercept)  0.00    0.000
##  Residual                 91.12    9.546
## Number of obs: 2613, groups:
## mau_loa_bins, 3; kil_bins, 3; hua_bins, 3; mau_kea_bins, 3; shr_bin, 3
##
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
```

```
## (Intercept)    72.66206    6.78137 2613.00000  10.715  < 2e-16 ***
## Year           -0.21921    0.11851 2613.00000  -1.850 0.064459 .
## dist_hil       -0.59539    0.86858 2613.00000  -0.685 0.493102
## dist_wai        0.13930    0.06592 2613.00000   2.113 0.034688 *
## dist_kon        0.03917    0.07079 2613.00000   0.553 0.580087
## dist_poh       -0.34452    0.10245 2613.00000  -3.363 0.000783 ***
## dist_arm_res    0.66755    0.88408 2613.00000   0.755 0.450270
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Year   dst_hl dist_w dst_kn dst_ph
## Year        -0.348
## dist_hil     0.456 -0.423
## dist_wai    -0.418  0.190 -0.823
## dist_kon    -0.831 -0.019  0.061 -0.045
## dist_poh     0.790  0.005  0.326 -0.531 -0.724
## dist_arm_rs -0.530  0.409 -0.995  0.828  0.029 -0.405
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
#Mixed model for Culmen
cul_lmer <- lmer(Culmen~Year+dist_hil+dist_wai+dist_kon+dist_poh+dist_arm_res+
                   (1|mau_loa_bins)+(1|kil_bins)+(1|hua_bins)+(1|mau_kea_bins)+
                   (1|shr_bin),data=morph_spa_bins,REML=F)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(cul_lmer)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula:
## Culmen ~ Year + dist_hil + dist_wai + dist_kon + dist_poh + dist_arm_res +
##     (1 | mau_loa_bins) + (1 | kil_bins) + (1 | hua_bins) + (1 |
##     mau_kea_bins) + (1 | shr_bin)
##    Data: morph_spa_bins
##
##      AIC      BIC   logLik deviance df.resid
##  16321.8  16398.1  -8147.9  16295.8     2600
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9587 -0.7670 -0.3508  1.0416  2.5514
##
## Random effects:
##  Groups       Name        Variance  Std.Dev.
##  mau_loa_bins (Intercept) 3.160e-08 0.0001778
##  kil_bins     (Intercept) 3.559e+00 1.8865553
##  hua_bins     (Intercept) 0.000e+00 0.0000000
##  mau_kea_bins (Intercept) 0.000e+00 0.0000000
##  shr_bin      (Intercept) 0.000e+00 0.0000000
##  Residual                 2.979e+01 5.4579038
```

```
## Number of obs: 2613, groups:
## mau_loa_bins, 3; kil_bins, 3; hua_bins, 3; mau_kea_bins, 3; shr_bin, 3
##
## Fixed effects:
##               Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    21.32011    4.33784  226.43137   4.915  1.7e-06 ***
## Year            0.07541    0.07310 2255.96598   1.032   0.3024
## dist_hil        0.94917    0.71718 1113.29140   1.323   0.1860
## dist_wai        0.02247    0.04951  197.63578   0.454   0.6505
## dist_kon       -0.01714    0.04271 1217.45699  -0.401   0.6882
## dist_poh       -0.14580    0.06900  253.55074  -2.113   0.0356 *
## dist_arm_res   -0.97412    0.72710 1112.86812  -1.340   0.1806
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Year   dst_hl dist_w dst_kn dst_ph
## Year        -0.256
## dist_hil     0.437 -0.020
## dist_wai    -0.525  0.069 -0.717
## dist_kon    -0.813  0.003 -0.009  0.155
## dist_poh     0.764 -0.062  0.207 -0.624 -0.749
## dist_arm_rs -0.487  0.014 -0.998  0.725  0.071 -0.260
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
#Mixed model for Mass
mas_lmer <- lmer(Mass~Year+dist_hil+dist_wai+dist_kon+dist_poh+dist_arm_res+
                   (1|mau_loa_bins)+(1|kil_bins)+(1|hua_bins)+(1|mau_kea_bins)+
                   (1|shr_bin),data=morph_spa_bins,REML=F)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(mas_lmer)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula:
## Mass ~ Year + dist_hil + dist_wai + dist_kon + dist_poh + dist_arm_res +
##     (1 | mau_loa_bins) + (1 | kil_bins) + (1 | hua_bins) + (1 |
##     mau_kea_bins) + (1 | shr_bin)
##    Data: morph_spa_bins
##
##      AIC      BIC   logLik deviance df.resid
##  18441.0  18517.2  -9207.5  18415.0     2600
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0267 -0.4783 -0.2540  0.0876  4.8573
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  mau_loa_bins (Intercept)  0.00    0.000
```

```
##  kil_bins     (Intercept)  0.00    0.000
##  hua_bins     (Intercept)  0.00    0.000
##  mau_kea_bins (Intercept)  0.00    0.000
##  shr_bin      (Intercept)  0.00    0.000
##  Residual                 67.33    8.205
## Number of obs: 2613, groups:
## mau_loa_bins, 3; kil_bins, 3; hua_bins, 3; mau_kea_bins, 3; shr_bin, 3
##
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)    15.07076    5.82909 2613.00000   2.585  0.00978 **
## Year           -0.18504    0.10186 2613.00000  -1.817  0.06940 .
## dist_hil        0.14581    0.74661 2613.00000   0.195  0.84517
## dist_wai        0.05237    0.05666 2613.00000   0.924  0.35546
## dist_kon        0.07763    0.06085 2613.00000   1.276  0.20219
## dist_poh       -0.21687    0.08806 2613.00000  -2.463  0.01386 *
## dist_arm_res   -0.06444    0.75994 2613.00000  -0.085  0.93243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Year   dst_hl dist_w dst_kn dst_ph
## Year        -0.348
## dist_hil     0.456 -0.423
## dist_wai    -0.418  0.190 -0.823
## dist_kon    -0.831 -0.019  0.061 -0.045
## dist_poh     0.790  0.005  0.326 -0.531 -0.724
## dist_arm_rs -0.530  0.409 -0.995  0.828  0.029 -0.405
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

##Model Dredging

Model Dredging using MuMIn function dredge()

```
wing_lmer_full <- lmer(Wing~Year+dist_hil+dist_wai+dist_kon+dist_poh+dist_arm_res+
                         (1|mau_loa_bins)+(1|kil_bins)+(1|hua_bins)+(1|mau_kea_bins)+
                         (1|shr_bin),data=morph_spa_bins,REML=F, na.action="na.fail")
```

```
## boundary (singular) fit: see help('isSingular')
```

```
wing_dredge <- dredge(wing_lmer_full, rank=AICc)
```

```
## Fixed term is "(Intercept)"
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
```

```
wing_dredge
```

```
## Global model call: lmer(formula = Wing ~ Year + dist_hil + dist_wai + dist_kon +
##     dist_poh + dist_arm_res + (1 | mau_loa_bins) + (1 | kil_bins) +
##     (1 | hua_bins) + (1 | mau_kea_bins) + (1 | shr_bin), data = morph_spa_bins,
##     REML = F, na.action = "na.fail")
## ---
## Model selection table
##    (Int) dst_arm_res    dst_hil   dst_kon   dst_poh    dst_wai     Yer df
## 57 78.06                                  -0.246400  0.0898200 -0.2200 10
## 58 78.26    0.016640                      -0.276000  0.1019000 -0.2541 11
## 59 78.32              0.0157700           -0.274300  0.1005000 -0.2539 11
## 61 78.95                        -0.009222 -0.255000  0.0973300 -0.2415 11
## 25 74.62                                  -0.273100  0.0988500          9
## 60 75.78    0.653300 -0.6249000           -0.303400  0.1409000 -0.2180 12
## 62 74.78    0.064270              0.042150 -0.321600  0.1021000 -0.2536 12
## 63 75.38              0.0574800  0.037610 -0.313200  0.0980800 -0.2558 12
## 28 71.24    1.319000 -1.3000000           -0.305000  0.1639000          11
## 26 74.37    0.008321                      -0.290200  0.1065000          10
## 27 74.40              0.0078020           -0.289200  0.1057000          10
## 29 74.72                        -0.002038 -0.275700  0.1010000          10
## 8  89.48   -1.937000  1.7430000 -0.182700                               10
## 64 72.66    0.667600 -0.5954000  0.039170 -0.344500  0.1393000 -0.2192 13
## 40 90.96   -1.823000  1.6460000 -0.167400                     -0.1818 11
## 41 79.96                                  -0.171000           -0.2196  9
## 54 88.92   -0.183000            -0.184500            0.0708700 -0.1927 11
## 44 79.98   -1.563000  1.5570000           -0.191800            -0.2028 11
## 55 88.71             -0.1850000 -0.190200            0.0796900 -0.1823 11
## 9  76.13                                  -0.179000                      8
## 30 71.45    0.048000              0.036050 -0.327600  0.1061000          11
## 12 77.07   -1.693000  1.6810000           -0.204000                      10
## 31 71.83              0.0436000  0.032600 -0.323200  0.1042000          11
## 22 86.58   -0.195000            -0.191000            0.0689300          10
## 23 85.96             -0.1929000 -0.187300            0.0750500          10
## 16 87.67   -1.939000  1.7740000 -0.155000 -0.035000                      11
## 42 80.34   -0.018600                      -0.153400           -0.2157 10
## 24 89.55   -2.078000  1.8860000 -0.178600           -0.0087950          11
## 45 79.10                         0.016230 -0.170100           -0.2217 10
## 38 88.31   -0.147000            -0.134800                     -0.1880 10
## 43 80.32             -0.0176100           -0.154000            -0.2165 10
## 48 88.68   -1.823000  1.6840000 -0.131800 -0.046570            -0.1837 12
## 6  86.31   -0.159700            -0.147200                               9
## 56 90.96   -1.829000  1.6520000 -0.167300           -0.0003261 -0.1817 12
## 39 87.76             -0.1418000 -0.128100                     -0.1922 10
## 10 77.09   -0.025640                      -0.161800                      9
## 11 77.04             -0.0243600           -0.162300                      9
## 7  85.66             -0.1542000 -0.140500                               9
## 13 75.40                         0.019800 -0.185800                      9
## 36 77.35   -1.647000  1.5270000                                 -0.2086 10
## 32 85.81   -1.632000  1.4890000 -0.137700 -0.068050  0.0189400          12
## 46 86.06   -0.112100            -0.099140 -0.040720            -0.1929 11
## 47 79.14             -0.0005375  0.015750 -0.169600            -0.2215 11
```

```
## 14 86.46   -0.161900            -0.149500  0.002526                    10
## 52 80.50   -2.196000  2.0900000                      -0.0511300 -0.2424 11
## 15 82.19             -0.1030000 -0.087050 -0.061340                    10
## 4  74.23   -1.199000  1.0750000                                          9
## 20 78.71   -3.091000  2.9700000                      -0.0869100          10
## 2  73.44   -0.097490                                                     8
## 34 76.34   -0.092420                                           -0.1783   9
## 35 76.32             -0.0896600                                -0.1817   9
## 3  73.40             -0.0953900                                          8
## 18 71.17   -0.109700                                 0.0369400           9
## 50 76.87   -0.087980                                -0.0083180 -0.1851 10
## 19 70.98             -0.1057000                      0.0376400           9
## 51 77.01             -0.0845400                     -0.0107700 -0.1913 10
## 33 71.78                                                       -0.2303   8
## 49 66.47                                            0.0559100 -0.1635   9
## 1  67.51                                                                7
## 17 64.17                                            0.0477400           8
## 37 72.14                       -0.005008                       -0.2325   9
## 53 68.80                       -0.035170            0.0644500 -0.1966 10
## 5  68.11                       -0.010950                                8
## 21 64.60                       -0.007891            0.0491900           9
##       logLik    AICc delta weight
## 57 -9603.494 19227.1  0.00  0.207
## 58 -9603.291 19228.7  1.61  0.092
## 59 -9603.305 19228.7  1.64  0.091
## 61 -9603.406 19228.9  1.84  0.082
## 25 -9605.790 19229.6  2.58  0.057
## 60 -9603.031 19230.2  3.11  0.044
## 62 -9603.113 19230.3  3.27  0.040
## 63 -9603.163 19230.4  3.37  0.038
## 28 -9604.722 19231.5  4.47  0.022
## 26 -9605.752 19231.6  4.52  0.022
## 27 -9605.756 19231.6  4.53  0.022
## 29 -9605.787 19231.7  4.59  0.021
## 8  -9605.905 19231.9  4.82  0.019
## 64 -9602.878 19231.9  4.82  0.019
## 40 -9604.966 19232.0  4.96  0.017
## 41 -9607.123 19232.3  5.24  0.015
## 54 -9605.330 19232.8  5.69  0.012
## 44 -9605.468 19233.0  5.96  0.010
## 55 -9605.505 19233.1  6.04  0.010
## 9  -9608.536 19233.1  6.06  0.010
## 30 -9605.624 19233.3  6.28  0.009
## 12 -9606.641 19233.4  6.29  0.009
## 31 -9605.651 19233.4  6.33  0.009
## 22 -9606.710 19233.5  6.43  0.008
## 23 -9606.860 19233.8  6.73  0.007
## 16 -9605.876 19233.9  6.78  0.007
## 42 -9606.899 19233.9  6.81  0.007
## 24 -9605.898 19233.9  6.82  0.007
## 45 -9606.914 19233.9  6.84  0.007
## 38 -9606.916 19233.9  6.85  0.007
## 43 -9606.929 19233.9  6.87  0.007
## 48 -9604.916 19234.0  6.88  0.007
```

```
## 6  -9607.947 19234.0  6.89  0.007
## 56 -9604.966 19234.1  6.98  0.006
## 39 -9607.219 19234.5  7.45  0.005
## 10 -9608.240 19234.5  7.48  0.005
## 11 -9608.284 19234.6  7.57  0.005
## 7  -9608.295 19234.7  7.59  0.005
## 13 -9608.317 19234.7  7.63  0.005
## 36 -9607.732 19235.5  8.48  0.003
## 32 -9605.866 19235.9  8.78  0.003
## 46 -9606.893 19235.9  8.82  0.003
## 47 -9606.914 19235.9  8.86  0.002
## 14 -9607.947 19236.0  8.91  0.002
## 52 -9607.059 19236.2  9.15  0.002
## 15 -9608.241 19236.6  9.49  0.002
## 4  -9609.374 19236.8  9.75  0.002
## 20 -9608.648 19237.4 10.31  0.001
## 2  -9611.269 19238.6 11.52  0.001
## 34 -9610.267 19238.6 11.53  0.001
## 35 -9610.453 19239.0 11.90  0.001
## 3  -9611.510 19239.1 12.00  0.001
## 18 -9611.053 19240.2 13.10  0.000
## 50 -9610.260 19240.6 13.53  0.000
## 19 -9611.342 19240.8 13.68  0.000
## 51 -9610.433 19241.0 13.88  0.000
## 33 -9613.300 19242.7 15.58  0.000
## 49 -9612.822 19243.7 16.64  0.000
## 1  -9614.910 19243.9 16.79  0.000
## 17 -9614.130 19244.3 17.24  0.000
## 37 -9613.290 19244.6 17.58  0.000
## 53 -9612.466 19245.0 17.94  0.000
## 5  -9615.050 19246.2 19.08  0.000
## 21 -9614.110 19246.3 19.22  0.000
## Models ranked by AICc(x)
## Random terms (all models):
##   1 | mau_loa_bins, 1 | kil_bins, 1 | hua_bins, 1 | mau_kea_bins, 1 | shr_bin
```

```
cul_lmer_full <- lmer(Culmen~Year+dist_hil+dist_wai+dist_kon+dist_poh+dist_arm_res+
                        (1|mau_loa_bins)+(1|kil_bins)+(1|hua_bins)+(1|mau_kea_bins)+
                        (1|shr_bin),data=morph_spa_bins,REML=F, na.action="na.fail")
```

```
## boundary (singular) fit: see help('isSingular')
```

```
cul_dredge <- dredge(cul_lmer_full, rank=AICc)
```

```
## Fixed term is "(Intercept)"
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
cul_dredge
```

```
## Global model call: lmer(formula = Culmen ~ Year + dist_hil + dist_wai + dist_kon +
##     dist_poh + dist_arm_res + (1 | mau_loa_bins) + (1 | kil_bins) +
##     (1 | hua_bins) + (1 | mau_kea_bins) + (1 | shr_bin), data = morph_spa_bins,
##     REML = F, na.action = "na.fail")
## ---
## Model selection table
##    (Int) dst_arm_res   dst_hil   dst_kon  dst_poh   dst_wai     Yer df
## 25 18.79                                 -0.16360  0.062100          9
## 12 21.42   -1.162000  1.163000           -0.14680                    10
## 57 17.53                                 -0.17660  0.068390 0.07845 10
## 27 18.06              0.022770           -0.18710  0.068630          10
## 44 20.61   -1.219000  1.214000           -0.14900           0.07128 11
## 29 19.56                      -0.016240 -0.16190  0.064370          10
## 26 18.02    0.022330                     -0.18650  0.069300          10
## 9  19.73                                 -0.09892                     8
## 28 20.55   -0.898000  0.903800           -0.16430  0.023470          11
## 16 23.28   -1.154000  1.128000 -0.020300 -0.12330                    11
## 59 17.09              0.017850           -0.19080  0.071620 0.06800 11
## 61 18.24                      -0.012740 -0.17160  0.068650 0.06964 11
## 58 17.07    0.017100                     -0.19000  0.072040 0.06854 11
## 41 18.86                                 -0.10340           0.05739  9
## 60 19.67   -0.937600  0.935900           -0.16810  0.025880 0.07370 12
## 48 22.35   -1.211000  1.180000 -0.020190 -0.12610           0.07298 12
## 13 20.18                      -0.008341 -0.09827                     9
## 31 18.51              0.016420 -0.005279 -0.18030  0.067830          11
## 11 19.54              0.006989           -0.10490                     9
## 30 18.82    0.011350           -0.008863 -0.17460  0.067320          11
## 10 19.60    0.004912                     -0.10320                     9
## 32 21.92   -0.928600  0.915700 -0.014120 -0.14560  0.020210          12
## 63 17.62              0.010320 -0.006160 -0.18280  0.070650 0.06821 12
## 62 17.98    0.004220           -0.010150 -0.17620  0.069670 0.06904 12
## 45 19.23                      -0.006369 -0.10170           0.05279 10
## 43 18.77              0.003796           -0.10590           0.05500 10
## 14 22.26   -0.030680          -0.032460 -0.06982                    10
## 42 18.82    0.001370                     -0.10420           0.05653 10
## 6  26.41   -0.095500          -0.085220                              9
## 64 21.32   -0.974100  0.949200 -0.017140 -0.14580  0.022470 0.07541 13
## 15 21.58             -0.021030 -0.024610 -0.07835                    10
## 8  28.71   -0.732500  0.613200 -0.102000                             10
## 7  26.00             -0.091250 -0.082170                              9
## 24 28.98   -1.303000  1.199000 -0.082870          -0.041630          11
## 46 21.50   -0.034790          -0.033340 -0.07002           0.05817 11
## 38 25.67   -0.099740          -0.086250                    0.05761 10
## 40 28.20   -0.840200  0.712900 -0.108000                    0.06849 11
## 47 20.82             -0.024750 -0.025280 -0.07863           0.05619 11
## 22 26.83   -0.103900          -0.092450           0.008941          10
## 39 25.27             -0.095140 -0.083080                    0.05537 10
## 23 26.54             -0.102200 -0.091780           0.012030          10
## 56 28.33   -1.352000  1.242000 -0.086390          -0.040820 0.06490 12
```

```
## 54 26.11   -0.109100           -0.094180            0.009861 0.05901 11
## 55 25.83             -0.107200 -0.093450            0.013060 0.05733 11
## 20 24.41   -2.220000  2.180000                     -0.126600         10
## 52 23.70   -2.268000  2.223000                     -0.127800 0.05769 11
## 5  17.23                       -0.028480                              8
## 4  15.02   -1.490000  1.489000                                        9
## 1  15.51                                                             7
## 36 13.74   -1.622000  1.616000                               0.08354 10
## 18 20.27   -0.029240                                -0.045660         9
## 19 20.20             -0.028520                      -0.045000         9
## 17 18.32                                            -0.045750         8
## 21 18.26                       -0.025390            -0.019800         9
## 37 16.59                       -0.027620                     0.03212  9
## 33 14.62                                                     0.04750  8
## 2  16.11   -0.009352                                                  8
## 3  16.04             -0.008256                                       8
## 50 19.70   -0.031510                                -0.046030 0.03978 10
## 51 19.64             -0.030680                      -0.045300 0.03856 10
## 49 17.45                                            -0.040320 0.02961  9
## 53 17.59                       -0.025020            -0.017630 0.02802 10
## 34 15.28   -0.012270                                          0.05461  9
## 35 15.22             -0.011040                               0.05353  9
##       logLik    AICc delta weight
## 25 -8149.430 16316.9  0.00  0.103
## 12 -8148.552 16317.2  0.26  0.091
## 57 -8148.831 16317.7  0.82  0.068
## 27 -8149.078 16318.2  1.31  0.053
## 44 -8148.093 16318.3  1.36  0.052
## 29 -8149.106 16318.3  1.37  0.052
## 26 -8149.109 16318.3  1.37  0.052
## 9  -8151.158 16318.4  1.44  0.050
## 28 -8148.442 16319.0  2.06  0.037
## 16 -8148.481 16319.1  2.13  0.035
## 59 -8148.661 16319.4  2.49  0.030
## 61 -8148.665 16319.4  2.50  0.029
## 58 -8148.687 16319.5  2.55  0.029
## 41 -8150.838 16319.7  2.81  0.025
## 60 -8147.957 16320.0  3.10  0.022
## 48 -8147.994 16320.1  3.18  0.021
## 13 -8151.027 16320.1  3.19  0.021
## 31 -8149.071 16320.2  3.31  0.020
## 11 -8151.102 16320.3  3.34  0.019
## 30 -8149.091 16320.3  3.35  0.019
## 10 -8151.129 16320.3  3.40  0.019
## 32 -8148.403 16320.9  4.00  0.014
## 63 -8148.652 16321.4  4.49  0.011
## 62 -8148.663 16321.4  4.52  0.011
## 45 -8150.768 16321.6  4.69  0.010
## 43 -8150.824 16321.7  4.80  0.009
## 14 -8150.828 16321.7  4.81  0.009
## 42 -8150.836 16321.8  4.83  0.009
## 6  -8151.866 16321.8  4.87  0.009
## 64 -8147.892 16321.9  4.99  0.008
## 15 -8150.939 16322.0  5.03  0.008
```

```
## 8  -8151.034 16322.2  5.22  0.008
## 7  -8152.149 16322.4  5.44  0.007
## 24 -8150.476 16323.1  6.12  0.005
## 46 -8150.517 16323.1  6.21  0.005
## 38 -8151.561 16323.2  6.28  0.004
## 40 -8150.622 16323.3  6.42  0.004
## 47 -8150.649 16323.4  6.47  0.004
## 22 -8151.795 16323.7  6.74  0.004
## 39 -8151.867 16323.8  6.89  0.003
## 23 -8152.030 16324.1  7.21  0.003
## 56 -8150.099 16324.3  7.39  0.003
## 54 -8151.476 16325.1  8.12  0.002
## 55 -8151.729 16325.6  8.63  0.001
## 20 -8153.533 16327.2 10.22  0.001
## 52 -8153.228 16328.6 11.63  0.000
## 5  -8157.945 16331.9 15.02  0.000
## 4  -8157.009 16332.1 15.16  0.000
## 1  -8159.130 16332.3 15.37  0.000
## 36 -8156.379 16332.8 15.91  0.000
## 18 -8157.393 16332.9 15.92  0.000
## 19 -8157.484 16333.0 16.11  0.000
## 17 -8158.590 16333.2 16.31  0.000
## 21 -8157.827 16333.7 16.79  0.000
## 37 -8157.849 16333.8 16.84  0.000
## 33 -8158.918 16333.9 16.96  0.000
## 2  -8159.054 16334.2 17.23  0.000
## 3  -8159.077 16334.2 17.28  0.000
## 50 -8157.248 16334.6 17.65  0.000
## 51 -8157.347 16334.8 17.85  0.000
## 49 -8158.515 16335.1 18.17  0.000
## 53 -8157.756 16335.6 18.67  0.000
## 34 -8158.783 16335.6 18.70  0.000
## 35 -8158.816 16335.7 18.77  0.000
## Models ranked by AICc(x)
## Random terms (all models):
##   1 | mau_loa_bins, 1 | kil_bins, 1 | hua_bins, 1 | mau_kea_bins, 1 | shr_bin
```

```
mass_lmer_full <- lmer(Mass~Year+dist_hil+dist_wai+dist_kon+dist_poh+dist_arm_res+
                         (1|mau_loa_bins)+(1|kil_bins)+(1|hua_bins)+(1|mau_kea_bins)+
                         (1|shr_bin),data=morph_spa_bins,REML=F, na.action="na.fail")
```

```
## boundary (singular) fit: see help('isSingular')
```

```
mass_dredge <- dredge(mass_lmer_full, rank=AICc)
```

```
## Fixed term is "(Intercept)"
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```
mass_dredge
```

```
## Global model call: lmer(formula = Mass ~ Year + dist_hil + dist_wai + dist_kon +
##     dist_poh + dist_arm_res + (1 | mau_loa_bins) + (1 | kil_bins) +
##     (1 | hua_bins) + (1 | mau_kea_bins) + (1 | shr_bin), data = morph_spa_bins,
##     REML = F, na.action = "na.fail")
## ---
## Model selection table
##    (Int) dst_arm_res   dst_hil   dst_kon  dst_poh   dst_wai     Yer df
## 57 20.95                                  -0.14580  0.063680 -0.1850 10
## 61 19.95                         0.010320 -0.13610  0.055260 -0.1609 11
## 58 20.90   -0.003613                      -0.13930  0.061050 -0.1776 11
## 59 20.89             -0.003466            -0.13960  0.061330 -0.1775 11
## 63 14.81              0.082790  0.077780 -0.21990  0.056350 -0.1815 12
## 62 14.55    0.083300             0.076900 -0.22250  0.061470 -0.1766 12
## 13 17.13                         0.035130 -0.09373                    9
## 45 19.34                         0.026240 -0.08635           -0.1149 10
## 29 16.94                         0.026390 -0.13100  0.039040         10
## 44 23.86   -0.712200  0.689200            -0.08672           -0.2016 11
## 48 17.32   -0.646000  0.713500  0.080140 -0.17370           -0.2029 12
## 41 22.34                                  -0.09327           -0.1716  9
## 25 18.75                                  -0.16060  0.058990          9
## 47 14.42              0.079110  0.091000 -0.16550           -0.1337 11
## 10 20.04   -0.036610                      -0.06022                    9
## 42 21.63   -0.025150                      -0.06344           -0.1238 10
## 27 18.98             -0.024930            -0.11210  0.043460         10
## 26 19.06   -0.025210                      -0.11150  0.042000         10
## 11 19.98             -0.035650            -0.06008                    9
## 15 13.20              0.058880  0.084410 -0.15350                   10
## 43 21.60             -0.023740            -0.06440           -0.1240 10
## 14 13.73    0.050070             0.076710 -0.14270                   10
## 60 21.25   -0.092690  0.087430            -0.13550  0.055590 -0.1826 12
## 46 15.39    0.060270             0.075710 -0.14480           -0.1225 11
## 30 12.64    0.062880             0.077840 -0.19580  0.042530         11
## 31 13.17              0.056450  0.073800 -0.18760  0.038360         11
## 64 15.07   -0.064440  0.145800  0.077630 -0.21690  0.052370 -0.1850 13
## 12 20.53   -0.318800  0.280200            -0.06886                   10
## 2  19.26   -0.062060                                                8
## 3  19.17             -0.060740                                      8
## 28 17.45    0.465200 -0.478100            -0.13680  0.074830         11
## 16 14.08   -0.251000  0.301300  0.078850 -0.15430                   11
## 9  18.79                                  -0.08879                    8
## 34 20.66   -0.052980                                          -0.1113  9
## 32 11.38    0.500000 -0.428500  0.075560 -0.21600  0.071940         12
## 35 20.52             -0.052010                                -0.1070  9
## 38 23.94   -0.074740           -0.033120                     -0.1189 10
## 7  21.85             -0.079210 -0.028090                              9
## 39 23.44             -0.071230 -0.029920                     -0.1126 10
## 18 19.69   -0.059350                                -0.011070          9
## 6  21.81   -0.076500           -0.036560                              9
```

```
## 19 19.51            -0.058540                    -0.008677          9
## 40 26.27   -0.721800  0.639800 -0.056730                    -0.1818 11
## 50 20.89   -0.051520                             -0.007600 -0.1064 10
## 51 20.71            -0.050820                    -0.005726 -0.1039 10
## 52 23.96   -1.185000  1.138000                    -0.049540 -0.1897 11
## 36 20.46   -0.541900  0.491000                              -0.1462 10
## 20 20.97   -0.480900  0.417500                    -0.029220         10
## 54 25.19   -0.089570           -0.071600           0.029570 -0.1385 11
## 4  18.43   -0.217000  0.160300                                       9
## 8  22.41   -0.356500  0.273900 -0.043450                            10
## 22 22.46   -0.089030           -0.055640           0.018290         10
## 55 24.94            -0.089410 -0.072640           0.033380 -0.1329 11
## 23 22.39            -0.089900 -0.058230           0.023150         10
## 56 26.23   -0.878400  0.802700 -0.046850          -0.013800 -0.1881 12
## 33 17.48                                                    -0.1443  8
## 24 22.48   -0.294000  0.208700 -0.047960           0.006040         11
## 1  14.86                                                             7
## 37 17.11                       0.008172                     -0.1472  9
## 49 17.04                                            0.005688 -0.1408  9
## 5  13.93                       0.016580                              8
## 17 14.60                                            0.004041         8
## 53 19.44                       0.049540          -0.054290 -0.2059 10
## 21 14.38                       0.021510          -0.011010          9
##       logLik    AICc delta weight
## 57 -9208.314 18436.7  0.00  0.132
## 61 -9208.165 18438.4  1.72  0.056
## 58 -9208.301 18438.7  1.99  0.049
## 59 -9208.302 18438.7  1.99  0.049
## 63 -9207.485 18439.1  2.38  0.040
## 62 -9207.500 18439.1  2.41  0.040
## 13 -9210.540 18439.1  2.44  0.039
## 45 -9209.677 18439.4  2.73  0.034
## 29 -9209.717 18439.5  2.81  0.032
## 44 -9208.776 18439.7  2.94  0.030
## 48 -9207.908 18439.9  3.22  0.026
## 41 -9210.937 18439.9  3.23  0.026
## 25 -9210.988 18440.0  3.33  0.025
## 47 -9209.056 18440.2  3.50  0.023
## 10 -9211.087 18440.2  3.53  0.023
## 42 -9210.095 18440.3  3.56  0.022
## 27 -9210.125 18440.3  3.62  0.022
## 26 -9210.150 18440.4  3.67  0.021
## 11 -9211.164 18440.4  3.68  0.021
## 15 -9210.184 18440.5  3.74  0.020
## 43 -9210.190 18440.5  3.75  0.020
## 14 -9210.291 18440.7  3.95  0.018
## 60 -9208.295 18440.7  4.00  0.018
## 46 -9209.320 18440.7  4.03  0.018
## 30 -9209.330 18440.8  4.05  0.017
## 31 -9209.389 18440.9  4.17  0.016
## 64 -9207.481 18441.1  4.39  0.015
## 12 -9210.804 18441.7  4.98  0.011
## 2  -9212.877 18441.8  5.10  0.010
## 3  -9212.903 18441.9  5.15  0.010
```

```
## 28 -9209.900 18441.9  5.19  0.010
## 16 -9209.965 18442.0  5.32  0.009
## 9  -9213.039 18442.1  5.42  0.009
## 34 -9212.074 18442.2  5.50  0.008
## 32 -9209.130 18442.4  5.67  0.008
## 35 -9212.172 18442.4  5.70  0.008
## 38 -9211.303 18442.7  5.98  0.007
## 7  -9212.329 18442.7  6.01  0.007
## 39 -9211.523 18443.1  6.42  0.005
## 18 -9212.723 18443.5  6.80  0.004
## 6  -9212.783 18443.6  6.92  0.004
## 19 -9212.810 18443.7  6.98  0.004
## 40 -9210.880 18443.9  7.15  0.004
## 50 -9212.002 18444.1  7.38  0.003
## 51 -9212.132 18444.3  7.64  0.003
## 52 -9211.335 18444.8  8.06  0.002
## 36 -9212.416 18444.9  8.20  0.002
## 20 -9212.431 18444.9  8.23  0.002
## 54 -9211.448 18445.0  8.28  0.002
## 4  -9213.526 18445.1  8.41  0.002
## 8  -9212.547 18445.2  8.47  0.002
## 22 -9212.591 18445.3  8.55  0.002
## 55 -9211.592 18445.3  8.57  0.002
## 23 -9212.645 18445.4  8.66  0.002
## 56 -9210.841 18445.8  9.09  0.001
## 33 -9215.299 18446.7  9.94  0.001
## 24 -9212.540 18447.2 10.47  0.001
## 1  -9216.599 18447.2 10.53  0.001
## 37 -9215.258 18448.6 11.87  0.000
## 49 -9215.285 18448.6 11.93  0.000
## 5  -9216.386 18448.8 12.11  0.000
## 17 -9216.591 18449.2 12.52  0.000
## 53 -9214.658 18449.4 12.69  0.000
## 21 -9216.349 18450.8 14.05  0.000
## Models ranked by AICc(x)
## Random terms (all models):
##   1 | mau_loa_bins, 1 | kil_bins, 1 | hua_bins, 1 | mau_kea_bins, 1 | shr_bin
```

##Model Averaging (after dredge)

```
top_wing_avg <- model.avg(wing_dredge, subset = delta <=2)
summary(top_wing_avg)
```

```
##
## Call:
## model.avg(object = wing_dredge, subset = delta <= 2)
##
## Component model call:
## lmer(formula = Wing ~ <4 unique rhs>, data = morph_spa_bins, REML = F,
##      na.action = na.fail)
##
## Component models:
##      df   logLik     AICc delta weight
```

```
## 456  10 -9603.49 19227.07  0.00   0.44
## 1456 11 -9603.29 19228.68  1.61   0.20
## 2456 11 -9603.30 19228.71  1.64   0.19
## 3456 11 -9603.41 19228.91  1.84   0.17
##
## Term codes:
## dist_arm_res     dist_hil     dist_kon     dist_poh     dist_wai         Year
##            1            2            3            4            5            6
##
## Model-averaged coefficients:
## (full average)
##               Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  78.302721   1.792316    1.793124  43.668  < 2e-16 ***
## dist_poh     -0.259061   0.049408    0.049429   5.241    2e-07 ***
## dist_wai      0.095545   0.035454    0.035470   2.694  0.00707 **
## Year         -0.236927   0.102463    0.102510   2.311  0.02082 *
## dist_arm_res  0.003251   0.013302    0.013307   0.244  0.80698
## dist_hil      0.003040   0.012861    0.012865   0.236  0.81319
## dist_kon     -0.001607   0.009817    0.009821   0.164  0.87002
##
## (conditional average)
##               Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  78.302721   1.792316    1.793124  43.668  < 2e-16 ***
## dist_poh     -0.259061   0.049408    0.049429   5.241    2e-07 ***
## dist_wai      0.095545   0.035454    0.035470   2.694  0.00707 **
## Year         -0.236927   0.102463    0.102510   2.311  0.02082 *
## dist_arm_res  0.016636   0.026130    0.026142   0.636  0.52453
## dist_hil      0.015767   0.025634    0.025646   0.615  0.53869
## dist_kon     -0.009222   0.021973    0.021984   0.419  0.67486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
top_cul_avg <- model.avg(cul_dredge, subset = delta <=2)
summary(top_cul_avg)
```

```
##
## Call:
## model.avg(object = cul_dredge, subset = delta <= 2)
##
## Component model call:
## lmer(formula = Culmen ~ <8 unique rhs>, data = morph_spa_bins, REML =
##      F, na.action = na.fail)
##
## Component models:
##      df   logLik     AICc delta weight
## 45    9 -8149.43 16316.93  0.00   0.20
## 124  10 -8148.55 16317.19  0.26   0.17
## 456  10 -8148.83 16317.75  0.82   0.13
## 245  10 -8149.08 16318.24  1.31   0.10
## 1246 11 -8148.09 16318.29  1.36   0.10
## 345  10 -8149.11 16318.30  1.37   0.10
## 145  10 -8149.11 16318.30  1.37   0.10
## 4     8 -8151.16 16318.37  1.44   0.10
##
```

```
## Term codes:
## dist_arm_res     dist_hil     dist_kon     dist_poh     dist_wai         Year
##            1            2            3            4            5            6
##
## Model-averaged coefficients:
## (full average)
##               Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  19.280376   1.896051    1.896501  10.166  < 2e-16 ***
## dist_poh     -0.159236   0.042965    0.042979   3.705 0.000211 ***
## dist_wai      0.041576   0.041073    0.041081   1.012 0.311511
## dist_arm_res -0.321657   0.588920    0.588973   0.546 0.584975
## dist_hil      0.325823   0.584794    0.584847   0.557 0.577453
## Year          0.017436   0.046880    0.046891   0.372 0.710010
## dist_kon     -0.001618   0.006917    0.006919   0.234 0.815095
##
## (conditional average)
##              Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  19.28038    1.89605     1.89650  10.166  < 2e-16 ***
## dist_poh     -0.15924    0.04297     0.04298   3.705 0.000211 ***
## dist_wai      0.06597    0.03267     0.03269   2.018 0.043590 *
## dist_arm_res -0.86216    0.68090     0.68102   1.266 0.205518
## dist_hil      0.86592    0.66422     0.66434   1.303 0.192430
## Year          0.07534    0.07165     0.07168   1.051 0.293212
## dist_kon     -0.01624    0.01558     0.01559   1.041 0.297648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
top_mass_avg <- model.avg(mass_dredge, subset = delta <=2)
summary(top_mass_avg)
```

```
##
## Call:
## model.avg(object = mass_dredge, subset = delta <= 2)
##
## Component model call:
## lmer(formula = Mass ~ <4 unique rhs>, data = morph_spa_bins, REML = F,
##      na.action = na.fail)
##
## Component models:
##      df   logLik     AICc delta weight
## 456  10 -9208.31 18436.71  0.00   0.46
## 3456 11 -9208.17 18438.43  1.72   0.20
## 1456 11 -9208.30 18438.70  1.99   0.17
## 2456 11 -9208.30 18438.71  1.99   0.17
##
## Term codes:
## dist_arm_res     dist_hil     dist_kon     dist_poh     dist_wai         Year
##            1            2            3            4            5            6
##
## Model-averaged coefficients:
## (full average)
##                Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  20.7345217  1.5866660   1.5873604  13.062  < 2e-16 ***
## dist_poh     -0.1417178  0.0402988   0.0403173   3.515  0.00044 ***
```

```
## dist_wai      0.0611804  0.0301997   0.0302136   2.025  0.04287 *
## Year         -0.1777220  0.0871752   0.0872153   2.038  0.04158 *
## dist_kon      0.0020213  0.0093087   0.0093122   0.217  0.82817
## dist_arm_res -0.0006174  0.0093854   0.0093897   0.066  0.94758
## dist_hil     -0.0005920  0.0092006   0.0092048   0.064  0.94872
##
## (conditional average)
##               Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)  20.734522   1.586666    1.587360  13.062  < 2e-16 ***
## dist_poh     -0.141718   0.040299    0.040317   3.515  0.00044 ***
## dist_wai      0.061180   0.030200    0.030214   2.025  0.04287 *
## Year         -0.177722   0.087175    0.087215   2.038  0.04158 *
## dist_kon      0.010320   0.018889    0.018898   0.546  0.58498
## dist_arm_res -0.003613   0.022464    0.022474   0.161  0.87229
## dist_hil     -0.003466   0.022038    0.022048   0.157  0.87508
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefTable(top_wing_avg,full=T)
```

```
##                Estimate Std. Error
## (Intercept)  78.3027212     1.7931
## dist_poh     -0.2590612     0.0494
## dist_wai      0.0955451     0.0355
## Year         -0.2369269     0.1025
## dist_arm_res  0.0032513     0.0133
## dist_hil      0.0030401     0.0129
## dist_kon     -0.0016071     0.0098
```

```
coefTable(top_cul_avg,full=T)
```

```
##               Estimate Std. Error
## (Intercept)  19.280376     1.8965
## dist_poh     -0.159236     0.0430
## dist_wai      0.041576     0.0411
## dist_arm_res -0.321657     0.5890
## dist_hil      0.325823     0.5848
## Year          0.017436     0.0469
## dist_kon     -0.001618     0.0069
```

```
coefTable(top_mass_avg,full=T)
```

```
##                 Estimate Std. Error
## (Intercept)  20.73452166     1.5874
## dist_poh     -0.14171783     0.0403
## dist_wai      0.06118039     0.0302
## Year         -0.17772200     0.0872
## dist_kon      0.00202126     0.0093
## dist_arm_res -0.00061737     0.0094
## dist_hil     -0.00059201     0.0092
```

#Appendix A; exporting cleaned-up Dataset

```
write.csv(morph_spa_bins, "Cleaned_Dataset_final.csv", row.names = FALSE)
```

R package Citations

```
packs <- c("tidyverse","ggplot2","maps","sp","sf","s2","rgdal","mapproj","ggfortify","usethis",
           "ggmap","mapview","arules","stringr","lme4","lmerTest","nlme","MuMIn","report",
           "geosphere","osmdata","arsenal")
for(i in 1:length(packs)){
  citation(packs[i])
}
```