

Modeling Flight Response Summary File

Data Cleaning And Exploration

Read Libraries

```
library(lme4) # fit regressions
library(rethinking) # Bayesian data analysis and plotting
library(popbio) # logistic regression plotting
library(binom) # binomial confidence intervals
```

Read Source Files

```
source_path = paste0(dir, "/Rsrc/")

script_names = c("center_flight_data.R", # 1 function: center_data()
                  "clean_flight_data.R", # 1 function: clean_flight_data()
                  "unique_flight_data.R", # 1 function: create_delta_data()
                  "compare_models.R", # 1 function: model_comparisonsAIC()
                  "AICprobabilities.R") # 1 function: AICprobs()

for (script in script_names) {
  path = paste0(source_path, script)
  source(path)
}
```

Read the Data

```
data_path = paste0(dir, "/Dispersal/Winter_2020/stats/data/all_flight_data-Winter2020.csv")

data = read_flight_data(data_path)
data_all = data[[1]]
data_tested = data[[2]]
d = create_delta_data(data_tested, tested_more_than_once=FALSE)
```

Repeating Plot Parameters & Functions

```
# scale/magnifications
c1 = 1.3*2 # size of points
c2 = 1.2*2 # size of text
c3 = 2*2 # size of title

# compute confidence interval
get_CI = function(x,y,m) {
  x.seq = seq(min(x) - sd(x), max(x) + sd(x), length.out=100)
  prd = data.frame(x=x.seq) # newdata
```

```

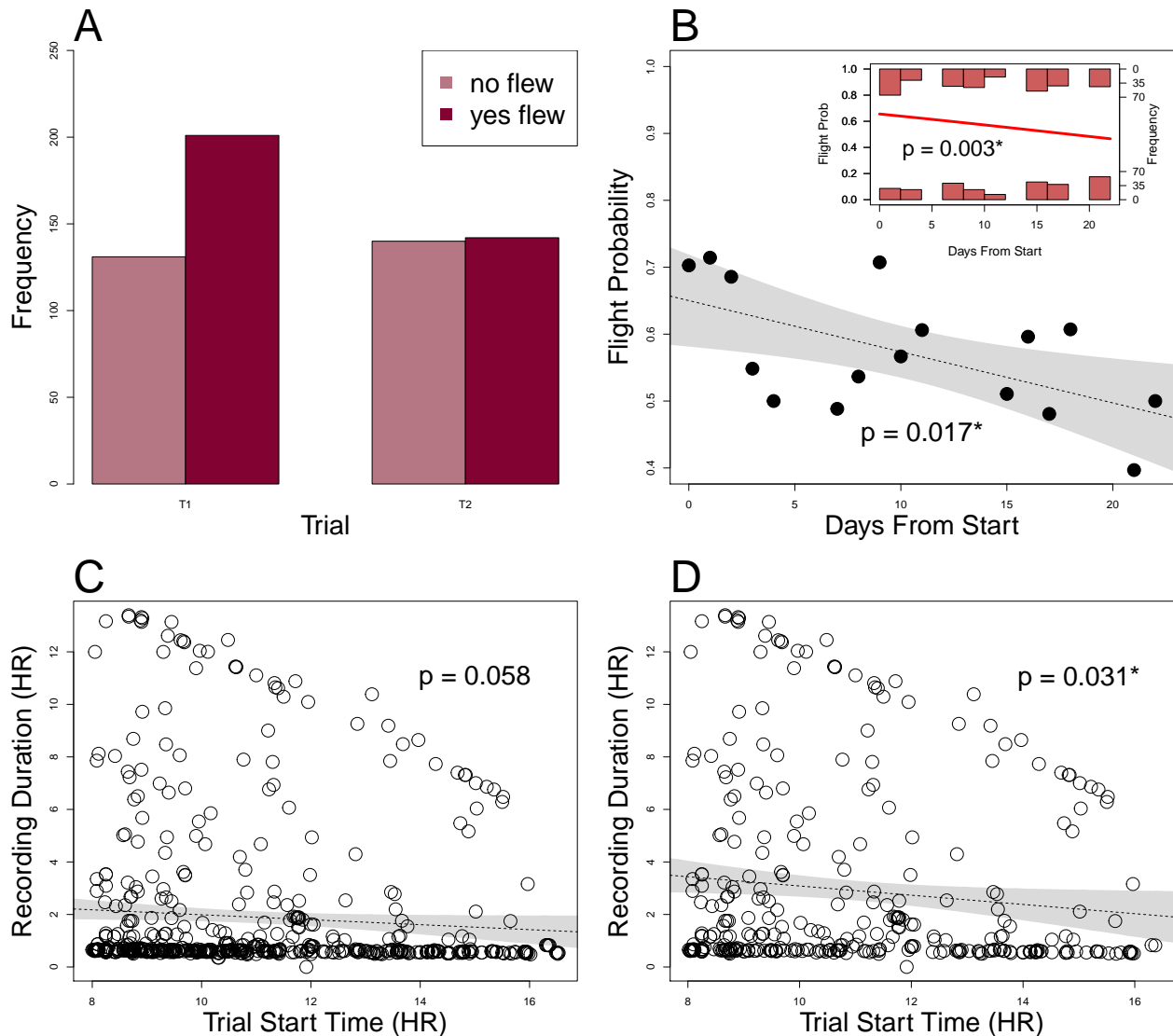
err = predict(m, newdata = prd, se.fit = TRUE)
prd$lci = err$fit - 1.96 * err$se.fit
prd$fit = err$fit
prd$uci = err$fit + 1.96 * err$se.fit
mu_ci = t(matrix(c(prd$lci, prd$uci), ncol=2))
return(list(mu_ci, prd))
}

# tailoring variables for plotting
d$mass_block=round(d$average_mass/0.005)*0.005
d$wing2body_block=round(d$wing2body, digits=2)
d$days_block=round(d$avg_days, digits=0)

```

Across-Trial Flight Response

Experimental Effects



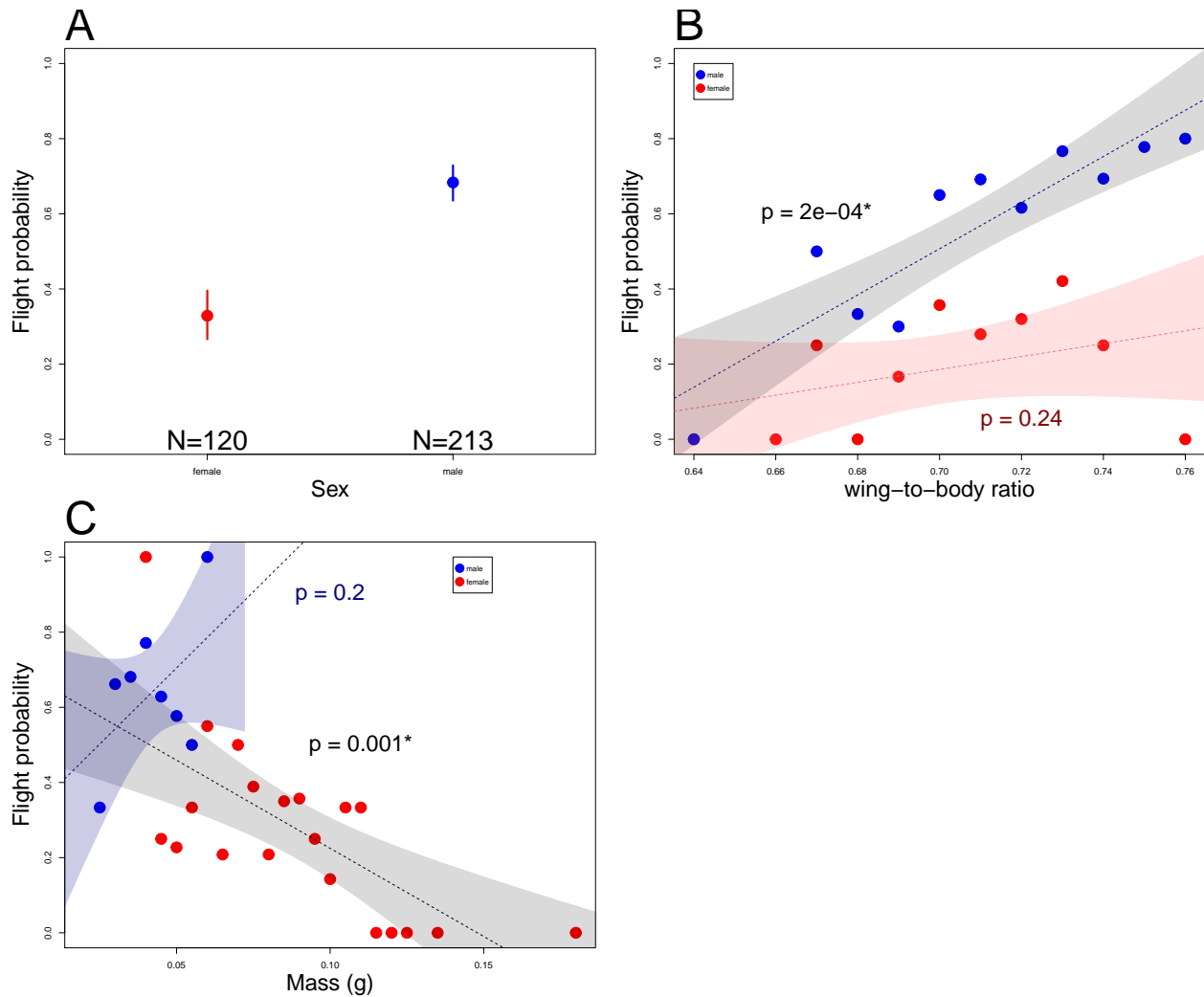
A & B. There was a negative effect of day a bug was tested but only when the full dataset is considered

(not the unique dataset). **C & D.** There was a negative effect of the trial start time but only after removing bugs that didn't fly.

Single-Variate Effects

```
# aggregate data for plotting
data_temp = aggregate(flew_prob~sex, data=d, FUN=mean)
data_temp$trials = c(sum(d$num_flew[d$sex=="F"])+d$num_notflew[d$sex=="F"]),
                    sum(d$num_flew[d$sex=="M"])+d$num_notflew[d$sex=="M"])

# calculate binomial confidence interval
data_temp$successes = c(sum(d$num_flew[d$sex=="F"]),
                       sum(d$num_flew[d$sex=="M"]))
data_temp$CI = binom.confint(data_temp$successes, data_temp$trials, methods="exact")
```



Between-Trial Flight Response (T1 vs. T2)

Read Libraries

```
library(dplyr) # data manipulation
library(zoo) # data manipulation
library(nnet) # multinomial modeling
library(kableExtra) # table formatting
library(plot.matrix) # enables heatmap plotting
```

Read Source Files

```
script_names = c( "get_warnings.R", # 1 function: withWarnings()
                  "multinom_functions.R") # 9 functions: calculate_P(),
                                          # calculate_P2(), calculate_P3(),
                                          # prediction_equations(),
                                          # prediction_equations2(),
                                          # get_significant_models(),
                                          # get_significant_modelsf(),
                                          # getting_odds()
                                          # getting_oddsf()

for (script in script_names) {
  path = paste0(source_path, script)
  source(path)
}
```

Read the Data

```
d = create_delta_data(data_tested, tested_more_than_once=TRUE)
```

Delta Flight Response Key	
Event	Encoding
flew in both trials	2
flew in T2 only	1
flew in neither trials	0
flew in T1 only	-1

We wanted to model the probability of different delta flight response cases with sex, host plant, percent changes in mass, and percent changes in egg-laying response as predictors. Since the outcomes (or response variables) were no longer binomial, we used multicategorical logit models. See the Appendix for an explanation and examples of computing multicategorical logit models.

Delta Mass Key	
Event	Sign
gained mass from T1 to T2	pos
no mass change between trails	0
lost mass from T1 to T2	neg

Host Plant Key	
Host	Encoding
Golden Rain Tree (GRT)	1
Balloon Vine (BV)	-1

Multinomial Modeling

Baseline

```
df = d %>%
  filter(!is.na(mass_diff), !is.na(flight_case))

df = df[with(df, order(mass_per)),]
n_trials = nrow(df)

df$flight_case = relevel(as.factor(df$flight_case), ref = "0")
```

Null Model

```
null = multinom(flight_case ~ 1, data = df)

## # weights: 8 (3 variable)
## initial value 385.389832
## iter 10 value 319.269929
## final value 319.269680
## converged
```

Compare Models - predictors mass %, sex, host

```
data = data.frame(R = df$flight_case,
  A = df$mass_per,
  B = df$sex_c,
  C = df$host_c)
model_script = paste0(source_path,"generic multinomial models- multinom 1RF + 3 FF.R")
model_comparisonsAIC(model_script)
```

```
##      [,1]      [,2]      [,3]      [,4]
## AICs  587.5607  591.9016  592.3168  592.4231
## models 4        7        13        12
## probs 0.7141852 0.0815063 0.06622882 0.06280119
##
## m4    multinom(formula = R ~ A + B, data = data, trace = FALSE)
## m7    multinom(formula = R ~ A + B + C, data = data, trace = FALSE)
## m13   multinom(formula = R ~ B * C + A, data = data, trace = FALSE)
## m12   multinom(formula = R ~ A * C + B, data = data, trace = FALSE)
```

```
anova(m4, m7, test="Chisq") # Adding C (host plant) does not improve fit
anova(m4, m8, test="Chisq") # Adding A*B does not improve fit
```

```
## Likelihood ratio tests of Multinomial Models
##
```

```
## Response: R
##      Model Resid. df Resid. Dev   Test    Df LR stat.   Pr(Chi)
## 1      A + B      825   569.5607
## 2 A + B + C      822   567.9016 1 vs 2     3 1.659076 0.6460701
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev   Test    Df LR stat.   Pr(Chi)
## 1 A + B      825   569.5607
## 2 A * B      822   569.4209 1 vs 2     3 0.1398496 0.9866598
```

Best Fit

```
M1 = multinom(flight_case ~ mass_per + sex_c, data = df)
model_table = calculate_P2(M1, "mass_per", "sex_c")
```

```
## # weights: 16 (9 variable)
## initial value 385.389832
## iter 10 value 286.869825
## iter 20 value 284.809036
## iter 30 value 284.797822
## final value 284.780360
## converged
##
## AIC: 587.5607
##      (Intercept) mass_per sex_c DF   SEi   SE1   SE2      zi    z1    z2
## -1      -1.015    0.043 -0.692  9 0.239 0.010 0.203  -4.248  4.390 -3.408
## 1      -6.820   -0.009 -5.626  9 0.183 0.026 0.183 -37.245 -0.348 -30.721
## 2       0.124    0.019 -0.902  9 0.167 0.008 0.159   0.742  2.334 -5.684
##      wald1 wald2 Pi > |z| P1 > |z| P2 > |z|
## -1  18.049 19.272 11.617  0.000  0.000  0.001
## 1 1387.197  0.121 943.764  0.000  0.728  0.000
## 2   0.551  5.447 32.310  0.458  0.020  0.000
```

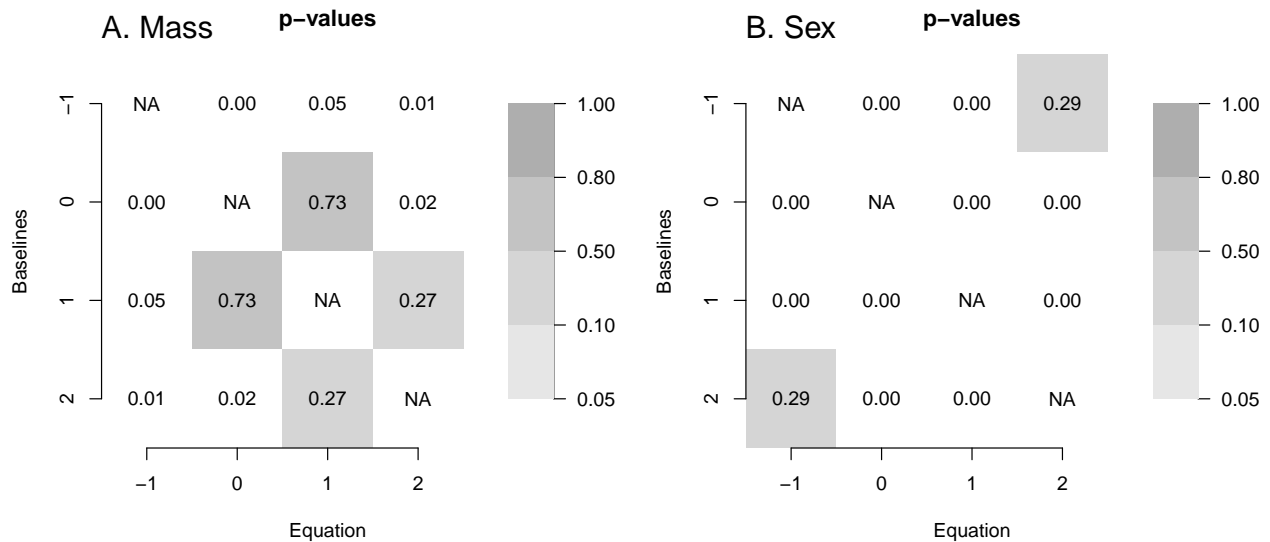
```
prediction_equations2(model_table, var1=" Mass Percent Change", var2=" Sex ")
```

```
## [1] "Where F = 1"
## [1] "log(pi_-1 / pi_1) = 5.81 + 0.05 Mass Percent Change + 4.93 Sex      Flew in T1, not T2"
## [2] "log(pi_2 / pi_-1) = 1.14 + -0.02 Mass Percent Change + -0.21 Sex    Flew in both, not T1"
## [3] "log(pi_2 / pi_1) = 6.94 + 0.03 Mass Percent Change + 4.72 Sex      Flew in both, not T2"
```

```
# define a run_multinom_model function based on the best fit model
run_multinom_model = function(d) {
  m = multinom(flight_case ~ mass_per + sex_c, trace=FALSE, data = d)
  model_table = calculate_P2(m, "mass_per", "sex_c", print_table=FALSE)
  return(model_table)
}

# determine which multinomial model equations are significant with a plot
par(mfrow=c(1,2))
MASS_ML = get_significant_models(15) # mass per
mtext("A. Mass", side=3, adj=0, line=0.5, cex=1.6, font=1)
```

```
SEX_ML = get_significant_models(16) # sex
mtext("B. Sex", side=3, adj=0, line=0.5, cex=1.6, font=1)
```



Compare Models - predictors mass %, sex, wing2body

```
df = df[with(df, order(mass_per)),]
df$wing2body_c = df$wing2body - mean(df$wing2body) # normalize

data = data.frame(R = df$flight_case,
  A = df$mass_per,
  B = df$sex_c,
  C = df$wing2body_c)
model_script = paste0(source_path, "generic multinomial models- multinom 1RF + 3 FF.R")
model_comparisonsAIC(model_script)
```

```
##      [,1]      [,2]      [,3]
## AICs  582.2678  585.1197  587.133
## models 7         12         13
## probs  0.6671688 0.1603139 0.05858546
##
## m7  multinom(formula = R ~ A + B + C, data = data, trace = FALSE)
## m12 multinom(formula = R ~ A * C + B, data = data, trace = FALSE)
## m13 multinom(formula = R ~ B * C + A, data = data, trace = FALSE)
```

```
anova(m7, m12, test="Chisq") # adding A*C does not improve fit
anova(m7, m13, test="Chisq") # Adding B*C does not improve fit
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev  Test   Df LR stat.   Pr(Chi)
## 1 A + B + C      822    558.2678
## 2 A * C + B      819    555.1197 1 vs 2    3 3.148182 0.3693379
## Likelihood ratio tests of Multinomial Models
##
```

```
## Response: R
##      Model Resid. df Resid. Dev   Test    Df LR stat.   Pr(Chi)
## 1 A + B + C      822   558.2678
## 2 B * C + A      819   557.1330 1 vs 2     3 1.134887 0.7686596
```

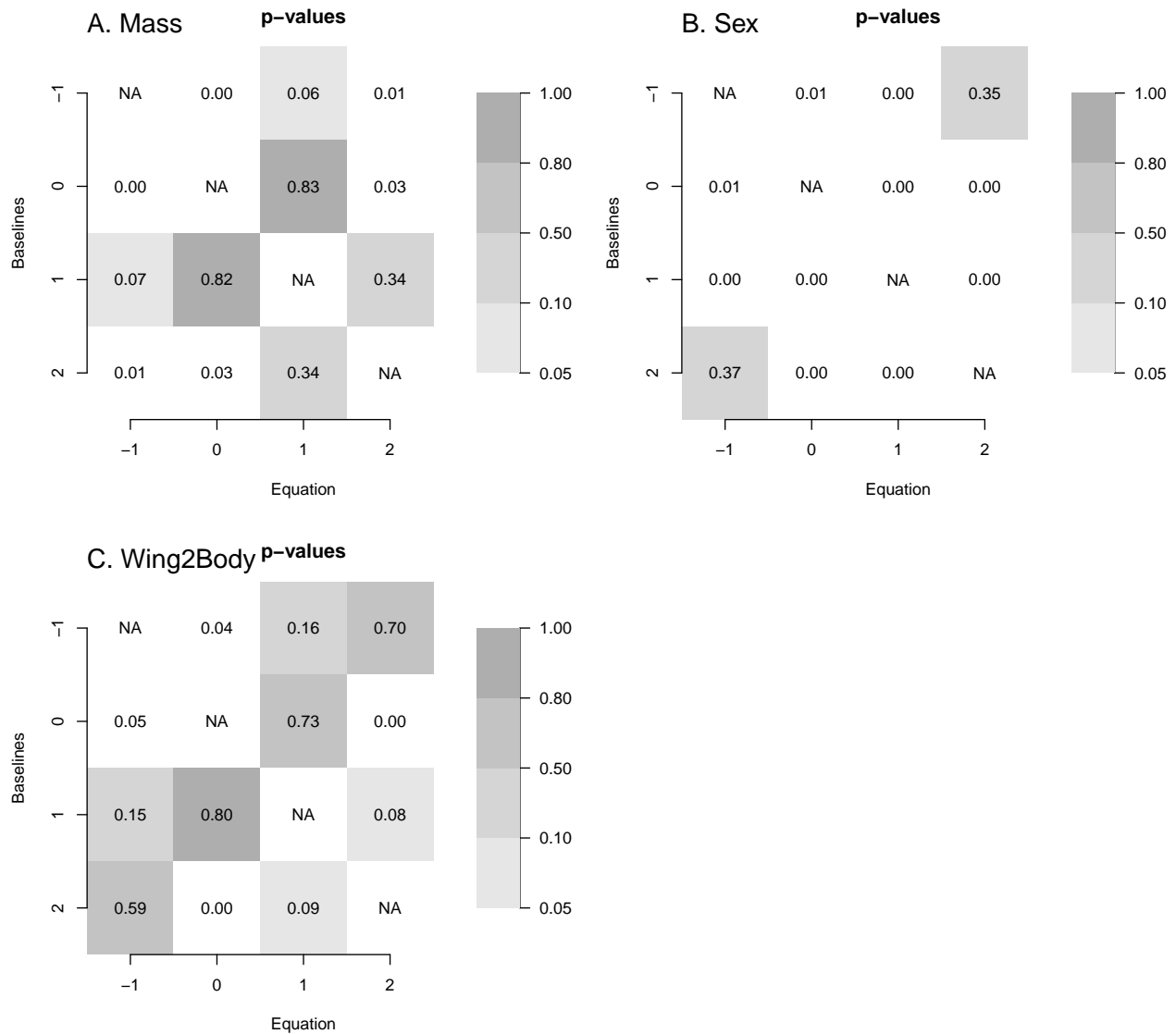
Best Fit

```
M2 = multinom(flight_case ~ mass_per + sex_c + wing2body_c, data = df)
model_table = calculate_P3(M2)
```

```
## # weights: 20 (12 variable)
## initial value 385.389832
## iter 10 value 286.740091
## iter 20 value 280.436850
## iter 30 value 279.437125
## iter 40 value 279.174660
## iter 50 value 279.134087
## final value 279.133921
## converged
##
## AIC: 582.2678
##      (Intercept) mass %      sex wing2body DF   SEi   SE1   SE2   SE3      zi
## -1      -0.935  0.041 -0.571   23.739 12 0.243 0.010 0.212 12.059 -3.854
## 1      -8.177 -0.005 -6.954    -6.595 12 0.187 0.025 0.187 18.786 -43.767
## 2       0.201  0.018 -0.760   28.094 12 0.172 0.008 0.166  9.718  1.173
##      z1      z2      z3   waldi  wald1   wald2 wald3 Pi>|z| P1>|z| P2>|z|
## -1  4.254 -2.698  1.969   14.850 18.096    7.278 3.875  0.000  0.000  0.007
## 1 -0.215 -37.102 -0.351 1915.510  0.046 1376.550 0.123  0.000  0.830  0.000
## 2  2.141 -4.590  2.891    1.375  4.585   21.071 8.357  0.241  0.032  0.000
##      P3>|z|
## -1  0.049
## 1   0.726
## 2   0.004
```

```
# define a run_multinom_model function based on the best fit model
run_multinom_model = function(d) {
  m = multinom(flight_case ~ mass_per + sex_c + wing2body_c, trace=FALSE, data = d)
  model_table = calculate_P3(m, print_table=FALSE)
  return(model_table)
}
```

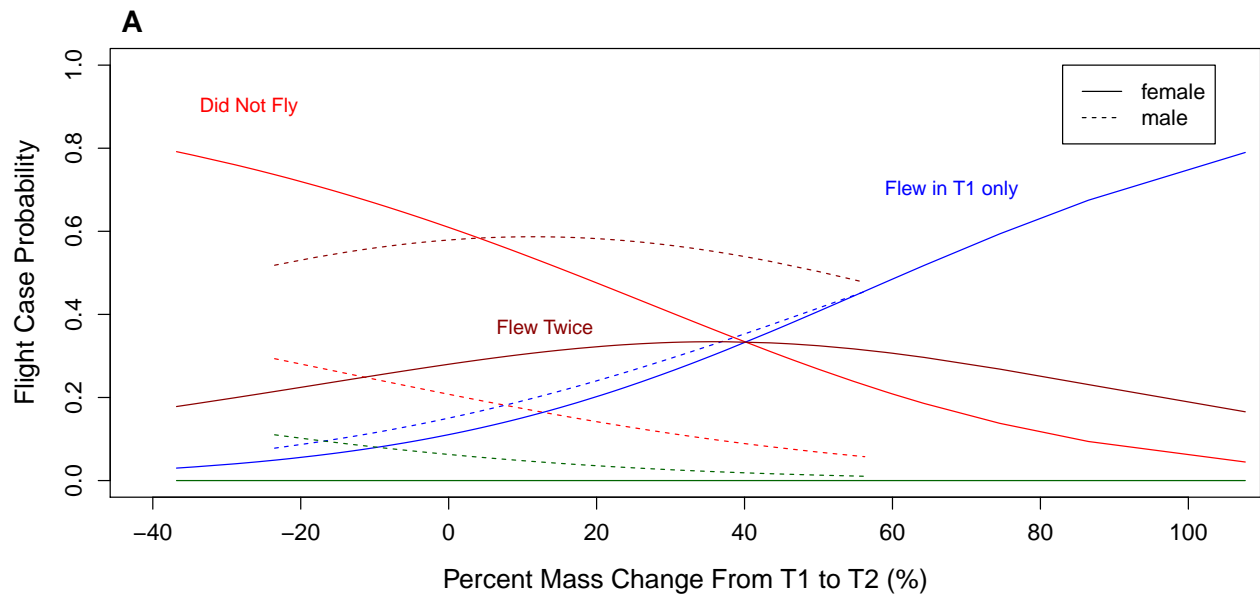
```
# determine which multinomial model equations are significant with a plot
par(mfrow=c(2,2))
MASS_PER_ML = get_significant_models(19) # mass%
mtext("A. Mass", side=3, adj=0, line=0.5, cex=1.3, font=1)
SEX_ML = get_significant_models(20) # sex
mtext("B. Sex", side=3, adj=0, line=0.5, cex=1.3, font=1)
WING2BODY_ML = get_significant_models(21) # wing2body
mtext("C. Wing2Body", side=3, adj=0, line=0.5, cex=1.3, font=1)
```

Predicted Probabilities

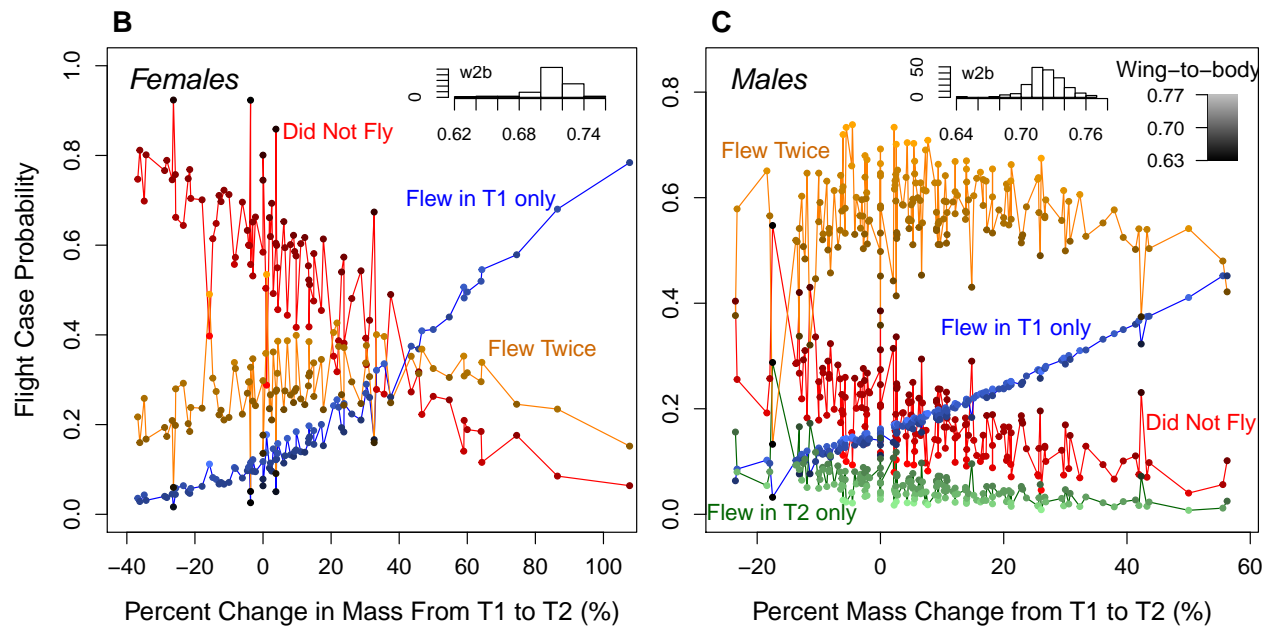
```
head(pp <- fitted(M1))
```

```
##           0           -1           1           2
## 1 0.7917303 0.03003973 4.362037e-06 0.1782256
## 2 0.7894639 0.03073036 4.325625e-06 0.1798015
## 3 0.7844677 0.03228066 4.247094e-06 0.1832474
## 4 0.7823713 0.03294258 4.214827e-06 0.1846819
## 5 0.7601763 0.04036342 3.895616e-06 0.1994564
## 6 0.7574981 0.04130977 3.859619e-06 0.2011882
```



```
head(pp <- fitted(M2)) # adding wing-to-body ratio
```

```
##          0          -1          1          2
## 1 0.7470322 0.03581826 2.459149e-07 0.2171493
## 2 0.8116776 0.02845344 2.925412e-07 0.1598686
## 3 0.6983200 0.04316854 2.166785e-07 0.2585113
## 4 0.8013507 0.03101678 2.841098e-07 0.1676322
## 5 0.7663040 0.04010556 2.581516e-07 0.1935902
## 6 0.7890842 0.03734506 2.743930e-07 0.1735705
```



Multinomial Modeling (Females Only)

Egg Case

Delta Flight Response Key	
Event	Encoding
laid eggs in both trials	2
laid eggs in T2 only	1
laid eggs in neither trials	0
laid eggs in T1 only	-1

Baseline

```
df <- d %>%
  filter(!is.na(egg_diff), !is.na(mass_diff), !is.na(flew_diff), sex_c == 1)

df <- df[with(df, order(mass_per)),]
n_tfemales = nrow(df)

df$flight_case <- relevel(as.factor(df$flight_case), ref = "0")
```

Null model

```
df$flight_case = droplevels(df$flight_case) # no female bug only flew in T2
null <- multinom(flight_case ~ 1, data = df)

## # weights:  6 (2 variable)
## initial value 102.170943
## final value 93.055466
## converged
```

Comparing Models - predictors mass diff, egg diff, host

```
data <- data.frame(R = df$flight_case,
  A = df$egg_case,
  B = df$mass_per,
  C = df$host_c)
model_script = paste0(source_path,"generic multinomial models- multinom 1RF + 3 FF.R")
model_comparisonsAIC(model_script)

##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## AICs    164.3817  165.6054  166.336   167.5638  167.9891  168.3593
## models  7         4        13        11        16        12
## probs   0.3761191 0.2039899 0.1415644 0.07661927 0.06194208 0.0514745
##
## m7  multinom(formula = R ~ A + B + C, data = data, trace = FALSE)
## m4  multinom(formula = R ~ A + B, data = data, trace = FALSE)
## m13 multinom(formula = R ~ B * C + A, data = data, trace = FALSE)
## m11 multinom(formula = R ~ A * B + C, data = data, trace = FALSE)
## m16 multinom(formula = R ~ B * C + A * B, data = data, trace = FALSE)
## m12 multinom(formula = R ~ A * C + B, data = data, trace = FALSE)
```

```
anova(m4, m7, test="Chisq") # Adding C does not improve fit
anova(m7, m13, test="Chisq") # Adding mass_diff*host does not improve fit
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev   Test      Df LR stat.   Pr(Chi)
## 1      A + B      180    153.6054
## 2 A + B + C      178    148.3817 1 vs 2      2 5.223671 0.0733997
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev   Test      Df LR stat.   Pr(Chi)
## 1 A + B + C      178    148.3817
## 2 B * C + A      176    146.3360 1 vs 2      2 2.045698 0.3595691
```

Comparing Models - predictors mass diff, egg diff, wing2body

```
##      [,1]      [,2]      [,3]      [,4]
## AICs 164.5293 164.9831 165.6054 167.7955
## models 7      13      4      12
## probs 0.3174096 0.2529723 0.1853291 0.06199495
##
## m7 multinom(formula = R ~ A + B + C, data = data, trace = FALSE)
## m13 multinom(formula = R ~ B * C + A, data = data, trace = FALSE)
## m4 multinom(formula = R ~ A + B, data = data, trace = FALSE)
## m12 multinom(formula = R ~ A * C + B, data = data, trace = FALSE)
anova(m4, m7, test="Chisq") # adding wing2body does not include fit
anova(m7, m13, test="Chisq") # Adding A*C does not improve fit
anova(m7, m12, test="Chisq") # Adding B*C does not improve fit
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev   Test      Df LR stat.   Pr(Chi)
## 1      A + B      180    153.6054
## 2 A + B + C      178    148.5293 1 vs 2      2 5.07612 0.07901956
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev   Test      Df LR stat.   Pr(Chi)
## 1 A + B + C      178    148.5293
## 2 B * C + A      176    144.9831 1 vs 2      2 3.546174 0.169808
## Likelihood ratio tests of Multinomial Models
##
## Response: R
##      Model Resid. df Resid. Dev   Test      Df LR stat.   Pr(Chi)
## 1 A + B + C      178    148.5293
## 2 A * C + B      176    147.7955 1 vs 2      2 0.7337197 0.6929067
```

Best Fit

```
M3 <- multinom(flight_case ~ mass_per + egg_case, data = df) # same best fit
```

```
## # weights: 12 (6 variable)
## initial value 102.170943
## iter 10 value 76.802714
## final value 76.802689
## converged

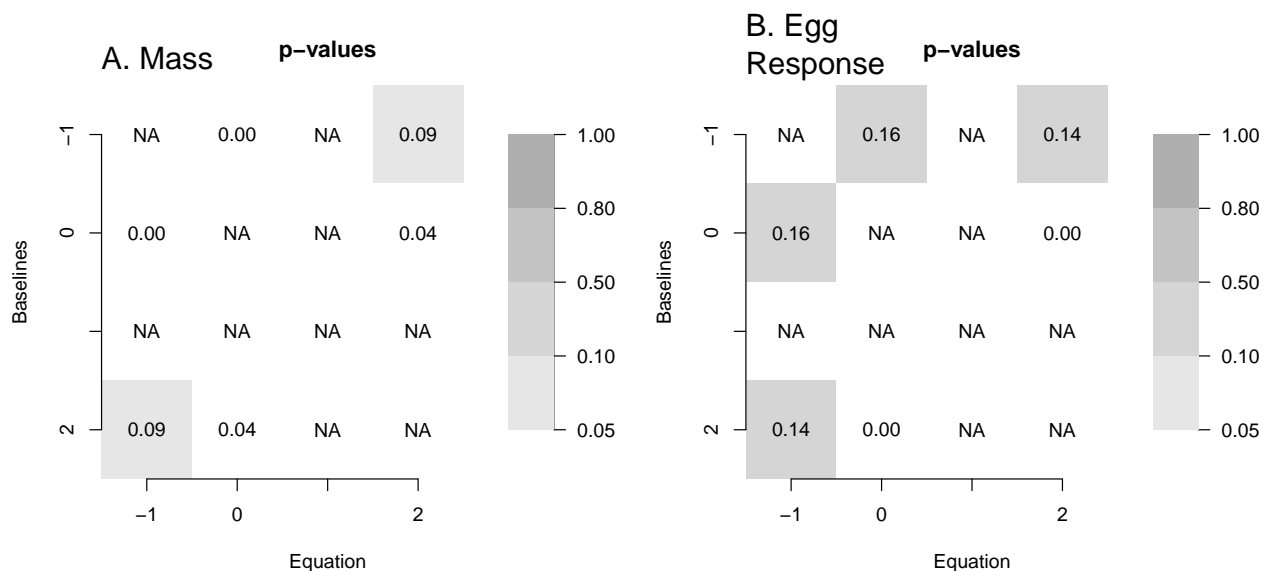
model_table = calculate_P2(M3, "mass_per", "egg_case")

## Warning in cbind(s$coefficients, c(s$edf, s$edf, s$edf), s$standard.errors[, :
## number of rows of result is not a multiple of vector length (arg 2)

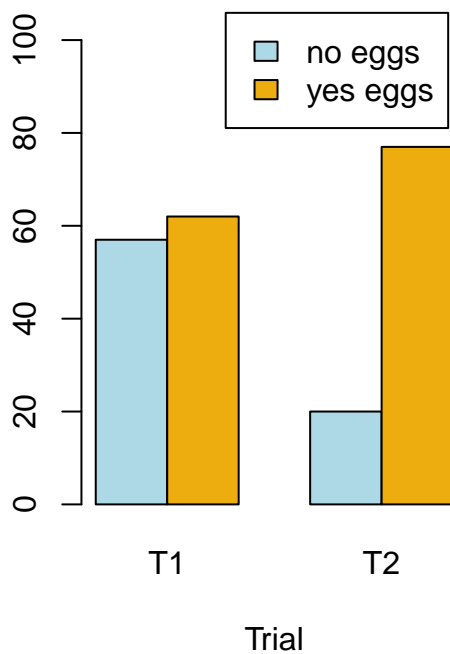
##
## AIC: 165.6054
## (Intercept) mass_per egg_case DF SE1 SE1 SE2 zi z1 z2 waldi
## -1 -0.950 0.041 -0.533 6 0.617 0.012 0.380 -1.539 3.389 -1.402 2.370
## 2 0.406 0.022 -1.098 6 0.424 0.011 0.297 0.957 2.038 -3.700 0.917
## wald1 wald2 Pi > |z| P1 > |z| P2 > |z|
## -1 11.488 1.966 0.124 0.001 0.161
## 2 4.154 13.693 0.338 0.042 0.000
```

```
# define a run_multinom_model function based on the best fit model
run_multinom_model = function(d) {
  m <- multinom(flight_case ~ mass_per + egg_case, trace=FALSE, data = d)
  model_table = calculate_P2(m, "mass_per", "egg_case", print_table=FALSE)
  return(model_table)
}

# determine which multinomial model equations are significant with a plot
par(mfrow=c(1,2))
ML_eqs = get_significant_models(15) # mass_per
mtext("A. Mass", side=3, adj=0, line=0.5, cex=1.6, font=1)
ML_eqs = get_significant_models(16) # egg_case
mtext("B. Egg \nResponse", side=3, adj=0, line=0.3, cex=1.6, font=1)
```



Barplot

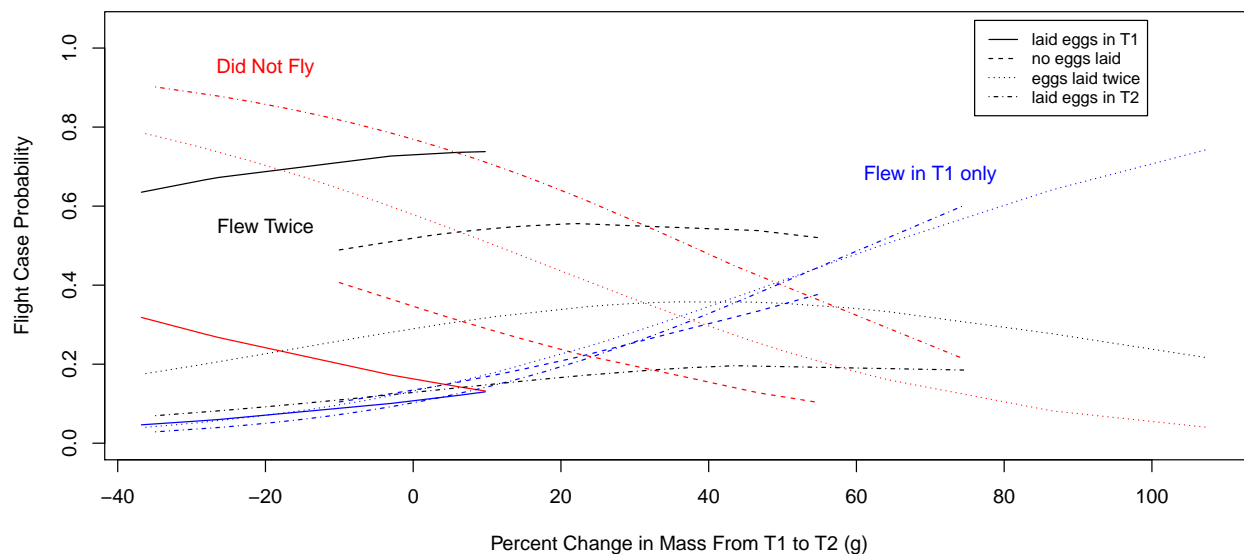


Predicted Probabilities

```
head(pp <- fitted(M3))
```

```
##           0           -1           2
## 1 0.3182776 0.04652361 0.63519881
## 2 0.7833277 0.04039434 0.17627792
## 3 0.9015654 0.02877502 0.06965959
## 4 0.7751740 0.04311454 0.18171149
## 5 0.7496332 0.05212861 0.19823820
## 6 0.2764594 0.05730902 0.66623157
```

Females Only



Flight Response Predictions

Read Libraries

```
library(cvms) # cross-validating regressions
```

Read Source Files

```
script_names = c("clean_flight_data-Fall2.R", # 1 function: clean_flight_data.Fall()
                 "unique_flight_data-Fall.R", # 1 function: create_delta_data.Fall()
                 "prediction_accuracy.R", # 1 function: calculate_accuracy()
                 "confusion_matrix.R") # 1 function: get_confusion_matrix()

for (script in script_names) {
  path = paste0(source_path, script)
  source(path)
}
```

Read the Data

```
dataFall = clean_flight_data.Fall("data/full_data-Fall2019.csv")

# extract sets with an experimental design similar to the Winter tests
ongoing_data = dataFall %>%
  filter(!is.na(mass)) %>%
  filter(set_number > 71)

# create delta data
d = create_delta_data.Fall(ongoing_data)
```

Compute predicted probabilities

```
d <- d[with(d, order(mass_per)),]

neither = c()
T1_rather_than_none = c()
T2_rather_than_none = c()
both_rather_than_none = c()

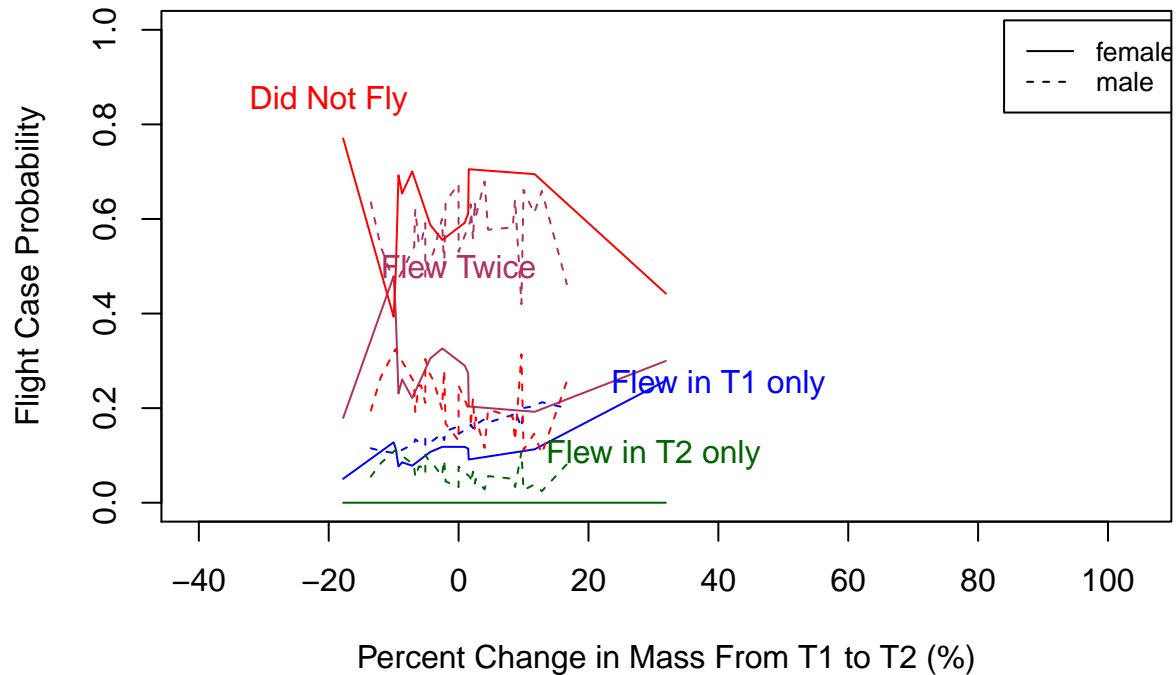
for (i in 1:nrow(d)) {
  m = d$mass_per[[i]]
  s = d$sex_c[[i]]
  #w = d$wing[[i]][1] / d$body[[i]][1]
  w = d$wing2body_c[i]
  top0 = exp(0) # just equals 1
  #top1 = exp(-17.862 + 0.041*m - 0.571*s + 23.558*w)
  #top2 = exp(-4.395 - 0.005*m - 9.580*s - 8.937*w)
  #top3 = exp(-19.931 + 0.018*m - 0.760*s + 28.019*w)
  top1 = exp(-0.935 + 0.041*m - 0.571*s + 23.739*w)
  top2 = exp(-8.177 - 0.005*m - 6.954*s - 6.595*w)
  top3 = exp(0.201 + 0.018*m - 0.760*s + 28.094*w)
  bottom = top0 + top1 + top2 + top3
```

```

neither = c(neither, top0/bottom)
T1_rather_than_none = c(T1_rather_than_none, top1/bottom)
T2_rather_than_none = c(T2_rather_than_none, top2/bottom)
both_rather_than_none = c(both_rather_than_none, top3/bottom)
}

```

Plot predicted probabilities



Overall and Grouped Accuracies

```

probs = round(cbind(neither, T1_rather_than_none, T2_rather_than_none, both_rather_than_none),2)
summary_probs = cbind(as.character(d$flight_case), as.character(d$sex), probs)
colnames(summary_probs) = c("event", "sex", "none", "T1", "T2", "both")
dataframe = as.data.frame(summary_probs)
nrow(dataframe)

```

```
## [1] 45
```

```
# overall
```

```
acc = calculate_accuracy(dataframe,3,6)
paste("Overall prediction accuracy, ", round(acc,2))

```

```
## [1] "Overall prediction accuracy, 0.6"
```

```
# by sex
```

```
femdata = dataframe %>%
```

```
  filter(sex=="F")
```

```
maledata = dataframe %>%
```

```
  filter(sex=="M")
```

```
accF = calculate_accuracy(femdata,3,6)
```

```
paste("Female prediction accuracy, ", round(accF,2))

```



```
## [1] "Female prediction accuracy, 0.38"
accM = calculate_accuracy(maledata,3,6)
paste("Male prediction accuracy, ", round(accM,2))

## [1] "Male prediction accuracy, 0.69"
```

Confusion Matrix

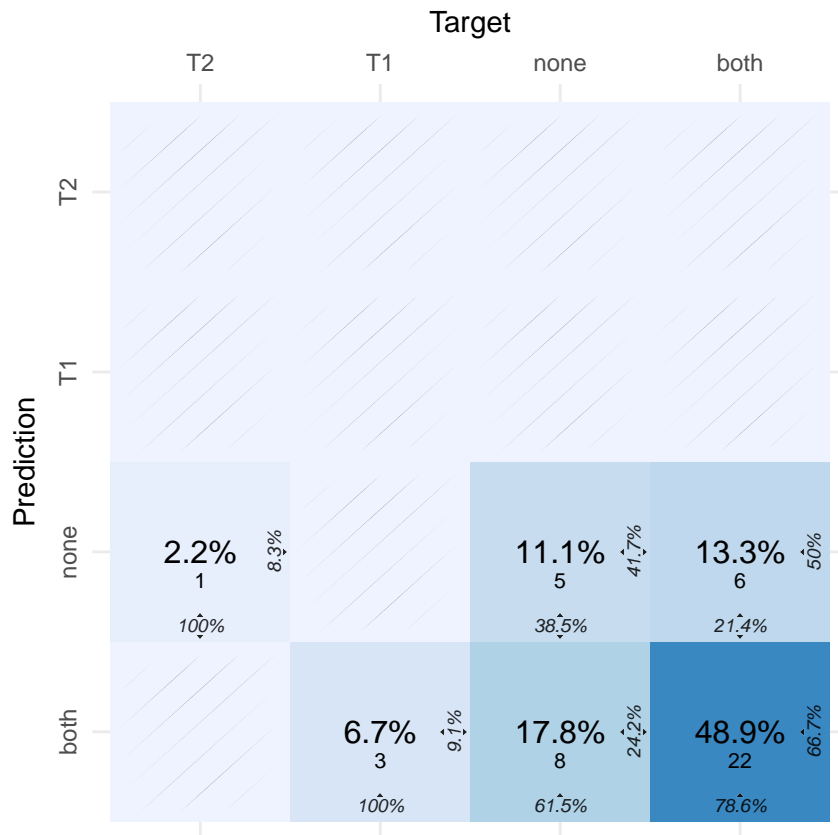
```
acc_table = get_confusion_matrix(dataframe,3,6)
acc_table

## # A tibble: 1 x 16
##   `Overall Accuracy` `Balanced Accuracy`    F1 Sensitivity Specificity
##   <dbl>              <dbl> <dbl>      <dbl>      <dbl>
## 1         0.6         0.538  NaN      0.293      0.784
## # ... with 11 more variables: Pos Pred Value <dbl>, Neg Pred Value <dbl>,
## #   Kappa <dbl>, MCC <dbl>, Detection Rate <dbl>, Detection Prevalence <dbl>,
## #   Prevalence <dbl>, Predictions <list>, Confusion Matrix <list>,
## #   Class Level Results <list>, Process <list>

confusion_matrix <- acc_table$'Confusion Matrix'[[1]]
confusion_matrix

## # A tibble: 16 x 3
##   Prediction Target    N
##   <chr>      <chr> <int>
## 1 both      both    22
## 2 none      both     6
## 3 T1        both     0
## 4 T2        both     0
## 5 both      none     8
## 6 none      none     5
## 7 T1        none     0
## 8 T2        none     0
## 9 both      T1        3
## 10 none     T1        0
## 11 T1       T1        0
## 12 T2       T1        0
## 13 both     T2        0
## 14 none     T2        1
## 15 T1       T2        0
## 16 T2       T2        0

plot_confusion_matrix(confusion_matrix, add_sums=FALSE)
```



Females Only

```
dfem = d %>%
  filter(sex=="F")
```

```
dfem <- dfem[with(dfem, order(mass_diff)),]
```

```
neither = c()
T1_rather_than_none = c()
both_rather_than_none = c()
for (i in 1:nrow(dfem)) {
  md = dfem$mass_diff[[i]]
  ed = dfem$egg_diff[[i]]
  top0 = exp(0) # just equals 1
  top1 = exp(-0.88 - 0.53*ed + 57.43*md)
  top2 = exp(-0.53 - 1.09*ed + 18.67*md)
  bottom = top0 + top1 + top2
  neither = c(neither, top0/bottom)
  T1_rather_than_none = c(T1_rather_than_none, top1/bottom)
  both_rather_than_none = c(both_rather_than_none, top2/bottom)
}
```

Compute predicted probabilities

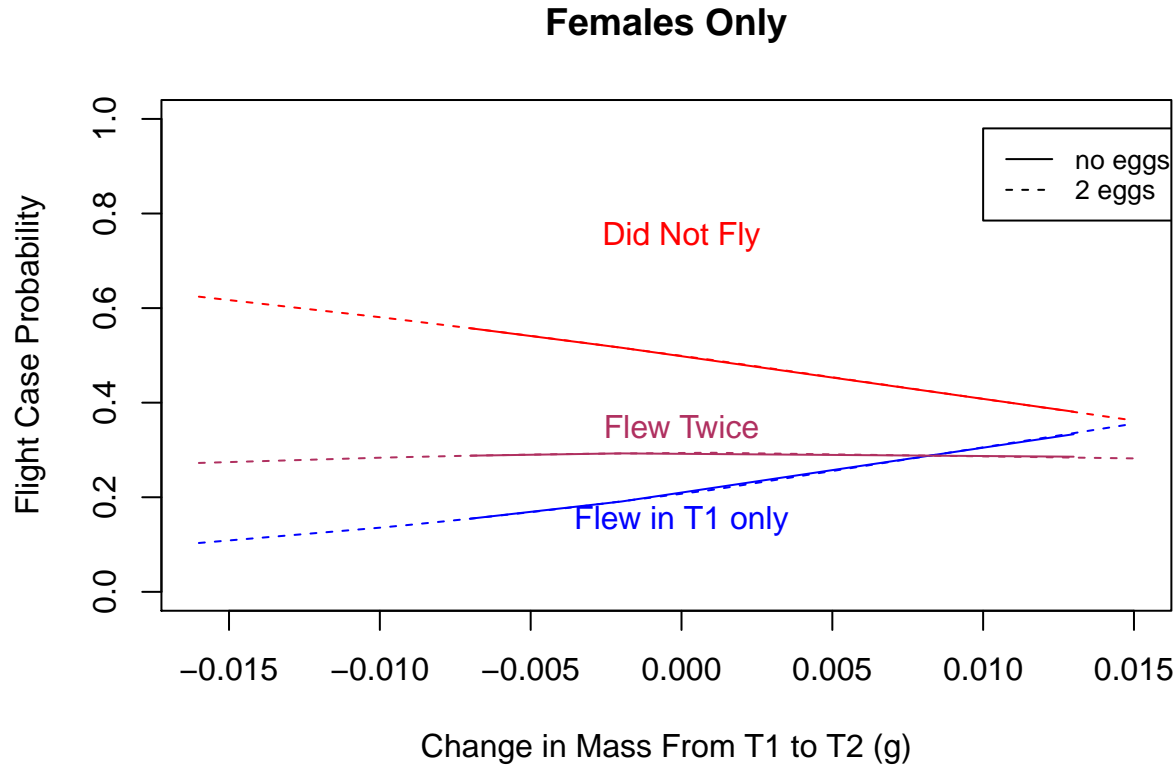
```
probs = round(cbind(neither, T1_rather_than_none, both_rather_than_none), 2)
summary_probs = cbind(as.character(dfem$flight_case), as.character(dfem$egg_diff), probs)
```

```
colnames(summary_probs) = c("event", "egg_diff", "none", "T1", "both")

egg2 = c(1,2,3,5,6,7,9,10,11,13)
noegg = c(4,8,12)

dataframe = as.data.frame(summary_probs)
dataframe$egg_cat = c(2,2,2,0,2,2,2,0,2,2,2,0,2)
```

Plot predicted probabilities



Overall and Grouped Accuracies

```
accF_eggs = calculate_accuracy(dataframe,3,5)
paste("Female prediction accuracy for mass diff and egg model, ", round(accF_eggs,2))

## [1] "Female prediction accuracy for mass diff and egg model, 0.46"
```

Confusion Matrix

```
acc_table = get_confusion_matrix(dataframe,3,5)
acc_table

## # A tibble: 1 x 16
##   `Overall Accuracy` `Balanced Accuracy` F1 Sensitivity Specificity
##   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 0.462 0.5 NaN 0.333 0.667
## # ... with 11 more variables: Pos Pred Value <dbl>, Neg Pred Value <dbl>,
## # Kappa <dbl>, MCC <dbl>, Detection Rate <dbl>, Detection Prevalence <dbl>,
```

```
## # Prevalence <dbl>, Predictions <list>, Confusion Matrix <list>,
## # Class Level Results <list>, Process <list>
```

```
confusion_matrix <- acc_table$'Confusion Matrix'[[1]]
confusion_matrix
```

```
## # A tibble: 9 x 3
##   Prediction Target      N
##   <chr>         <chr> <int>
## 1 both         both      0
## 2 none         both      6
## 3 T2           both      0
## 4 both         none      0
## 5 none         none      6
## 6 T2           none      0
## 7 both         T2        0
## 8 none         T2         1
## 9 T2           T2         0
```

```
plot_confusion_matrix(confusion_matrix, add_sums=FALSE)
```

