

A R SCRIPT

```

1 # Andrew Jansen
2 # created iv.15.2018
3
4 # Miscellaneous Functions -----
5 #Import the data
6 GetData = function(file.name) {
7   test.data = data.frame(read.csv(file.name, sep = ','))
8   return(test.data)
9 }
10
11 #For ANOVA
12 SqDev = function(x) {var(x) * (length(x) - 1)}
13
14 # Analysis 1 -----
15 # correspondence b/w sample weight and diversity, by sample method and order
16 Analysis1 = function(raw.data) {
17   #Create folder for plots
18   dir.create("plots/mass-vs-count/scatter")
19   dir.create("plots/mass-vs-count/residual")
20   dir.create("plots/mass-vs-count/qqplot")
21
22   name = "mass-vs-count.txt"
23   capture.output(cat("Linear Regression by sample method and order\n\n"),
24     file = name)
25
26   #Orders that don't break the script... droplevels not removing unused orders.
27   orders = c("Coleoptera",
28     "Diptera",
29     "Hemiptera",
30     "Hymenoptera",
31     "Lepidoptera",
32     "Orthoptera")
33   for (method.type in levels(raw.data$method)) {
34     for (name.order in orders) {
35       # Regression via linear model
36       temp.mass = raw.data[with(raw.data, year == 2015
37         & method == method.type
38         & order == name.order
39         & mass > 0
40         & family_count > 0), ]$mass
41       temp.family = raw.data[with(raw.data, year == 2015
42         & method == method.type
43         & order == name.order
44         & mass > 0
45         & family_count > 0), ]$family_count
46       temp.frame = data.frame(temp.family, temp.mass)
47       lin.mod = lm(temp.frame)
48
49       # Write linear model output to console and file
50       cat(method.type, name.order)
51       data.summary = summary(lin.mod)
52       print(data.summary)
53
54       capture.output(cat(method.type, name.order), file = name, append = TRUE)
55       capture.output(data.summary, file = name, append = TRUE)
56
57       # Make scatter plots for each order
58       plot(family_count ~ mass,
59         data = raw.data[with(raw.data, year == 2015
60           & method == method.type
61           & order == name.order
62           & mass > 0
63           & family_count > 0), ],
64         xlab = "sample mass (g)",
65         ylab = "number of families",
66         main = paste(name.order, " (", method.type, ")", sep = "")
67       )
68       abline(lin.mod, col = "blue")
69       dev.copy(pdf,
70         paste("plots/mass-vs-count/scatter/2015",
71           method.type,
72           name.order,
73           "mass-vs-count.pdf",
74           sep = "_"),
75         width = 3,
76         height = 3
77       )
78       dev.off()
79
80       # Plot the residuals vs. the fitted values
81       plot(residuals(lin.mod) ~ fitted(lin.mod),
82         data = temp.frame,
83         xlab = "fitted family count",
84         ylab = "residual family count",
85         main = NULL
86       )
87       dev.copy(pdf,
88         paste("plots/mass-vs-count/residual/2015",
89           method.type,
90           name.order,
91           "residual.pdf",
92           sep = "_"),
93         width = 3,
94         height = 3
95       )
96       dev.off()
97
98       # Make a normal probability plot of the residuals
99       qqnorm(lin.mod$residuals,
100         ylab = "Residual Quantiles",

```

```

101     main = NULL
102   )
103   qqline(lin.mod$residuals,
104     probs = c(0.25, 0.75),
105     col = 'blue'
106   )
107
108   dev.copy(pdf,
109     paste("plots/mass-vs-count/qqplot/2015",
110       method.type,
111       name.order,
112       "qqplot.pdf",
113       sep = "_"),
114     width = 3,
115     height = 3
116   )
117   dev.off()
118 }
119 }
120 }
121
122 # Analysis 2 -----
123 # linear mixed model: sample weight and diversity per order and sample method
124 Analysis2 = function(raw.data) {
125   dir.create("plots/glmm")
126   name = "general_linear_mixed_model.txt"
127   capture.output(cat("General Linear Mixed Model, negative binomial\n\n"),
128     file = name)
129
130   # Mixed linear model with mass as fixed effect and order and method as random
131   mixed.lmer.poisson = glmer(family_count ~ mass + (1 | order) + (1 | method),
132     data = raw.data[with(raw.data, mass > 0
133       & family_count > 0), ],
134     family = poisson)
135
136   print(summary(mixed.lmer.poisson))
137
138   mixed.lmer.nb = glmer.nb(family_count ~ mass + (1 | order) + (1 | method),
139     data = raw.data[with(raw.data, mass > 0
140       & family_count > 0), ],
141     family = negative.binomial)
142
143   print(summary(mixed.lmer.nb))
144
145   print(lrttest(mixed.lmer.poisson, mixed.lmer.nb))
146
147   mixed.lmer.nomass = glmer.nb(family_count ~ (1 | order) + (1 | method),
148     data = raw.data[with(raw.data, mass > 0
149       & family_count > 0), ],
150     family = negative.binomial)
151
152   mixed.lmer.noorder = glmer.nb(family_count ~ mass + (1 | method),
153     data = raw.data[with(raw.data, mass > 0
154       & family_count > 0), ],
155     family = negative.binomial)
156
157   mixed.lmer.nomethod = glmer.nb(family_count ~ mass + (1 | order),
158     data = raw.data[with(raw.data, mass > 0
159       & family_count > 0), ],
160     family = negative.binomial)
161
162   print(lrttest(mixed.lmer.nb,
163     mixed.lmer.nomass,
164     mixed.lmer.noorder,
165     mixed.lmer.nomethod))
166
167   capture.output(summary(mixed.lmer.poisson), file = name, append = TRUE)
168   capture.output(summary(mixed.lmer.nb), file = name, append = TRUE)
169   capture.output(lrttest(mixed.lmer.poisson, mixed.lmer.nb),
170     file = name, append = TRUE)
171   capture.output(lrttest(mixed.lmer.nb,
172     mixed.lmer.nomass,
173     mixed.lmer.noorder,
174     mixed.lmer.nomethod),
175     file = name, append = TRUE)
176
177   plot(fitted(mixed.lmer.nb),
178     residuals(mixed.lmer.nb),
179     xlab = "fitted family count",
180     ylab = "residual family count"
181   )
182   dev.copy(pdf,
183     paste("plots/glmm/2015_glmm_residual.pdf",
184       sep = "_"),
185     width = 3,
186     height = 3
187   )
188   dev.off()
189
190   qqnorm(residuals(mixed.lmer.nb),
191     ylab = "Residual Quantiles",
192     main = NULL
193   )
194   qqline(residuals(mixed.lmer.nb),
195     probs = c(0.25, 0.75),
196     col = 'blue'
197   )
198   dev.copy(pdf,
199     paste("plots/glmm/2015_glmm_qqplot.pdf",
200       sep = "_"),
201     width = 3,
202     height = 3

```

```

203 )
204 dev.off()
205 }
206
207 # Analysis 3 -----
208 # comparisons between blocks for diversity and weight of individual orders
209 Analysis3 = function(raw.data) {
210   dir.create("plots/blocks")
211   dir.create("plots/blocks/tukeyHSD")
212   dir.create("plots/blocks/tukeyHSD/mass")
213   dir.create("plots/blocks/tukeyHSD/family")
214   dir.create("plots/blocks/interaction")
215   dir.create("plots/blocks/interaction/mass")
216   dir.create("plots/blocks/interaction/family")
217
218   name = "blocks.txt"
219   capture.output(cat("Analyses of Variance (order, method) \n\n"), file = name)
220
221   #Orders that don't break the script... droplevels not removing unused orders.
222   orders = c("Coleoptera",
223             "Diptera",
224             "Hemiptera",
225             "Hymenoptera",
226             "Lepidoptera",
227             "Orthoptera",
228             "Neuroptera")
229
230   raw.data.2015 = raw.data[raw.data$year == 2015, ]
231
232   for (method.type in levels(raw.data.2015$method)) {
233     for (name.order in orders) {
234       capture.output(cat(name.order, method.type, "\n\n"),
235                     file = name, append = TRUE)
236       temp.data = raw.data.2015[with(raw.data.2015,
237                                   method == method.type
238                                   & order == name.order), ]
239       temp.data$event = factor(temp.data$event)
240       mass.model = lm(mass ~ block * event, temp.data)
241       family.model = lm(family_count ~ block * event, temp.data)
242       mass.aov = aov(mass.model)
243       family.aov = aov(family.model)
244
245       capture.output(cat("sample mass", "\n\n"), file = name, append = TRUE)
246       capture.output(summary(mass.aov), file = name, append = TRUE)
247       capture.output(leveneTest(mass.model), file = name, append = TRUE)
248       capture.output(shapiro.test(residuals(mass.model)),
249                     file = name, append = TRUE)
250       capture.output(TukeyHSD(mass.aov), file = name, append = TRUE)
251
252       par(mar = c(4, 4, 2, 0.6) + 0.1)
253       plot(TukeyHSD(mass.aov))
254       dev.copy(pdf,
255               paste("plots/blocks/tukeyHSD/mass/mass",
256                     method.type,
257                     name.order,
258                     "anova.pdf",
259                     sep = "-"),
260             width = 7.5,
261             height = 200
262             )
263       dev.off()
264       par(mar = c(4, 4, 1.1, 0.6) + 0.1)
265
266       mass.anova = anova(mass.model)
267       MS.res = mass.anova$Mean[4]
268       nu = mass.anova$Df[4]
269       means = tapply(temp.data$mass,
270                     list(temp.data$block, temp.data$event),
271                     mean)
272
273       alpha = 0.05
274       n = tapply(temp.data$mass,
275                 list(temp.data$block, temp.data$event),
276                 length)
277       se = sqrt(MS.res / n)
278       t.critical = qt(1 - alpha / 2, nu)
279       inter = t.critical * se
280
281       plotCI(x = 1:5,
282             y = means[1,],
283             uiw = inter[1,],
284             type = "b",
285             xaxt = "n",
286             col = "red",
287             xlab = "Sample Week",
288             ylab = "Mean Order Mass",
289             ylim = c(-1, 8),
290             xlim = c(1, 6),
291             main = paste(name.order, " (" , method.type, ")", sep = ""))
292
293       plotCI(x = 1:5 + 0.05,
294             y = means[2,],
295             uiw = inter[2,],
296             type = "b",
297             xaxt = "n",
298             col = "blue",
299             add = TRUE)
300
301       plotCI(x = 1:5 + 0.1,
302             y = means[3,],
303             uiw = inter[3,],
304             type = "b",
305             xaxt = "n",

```

```

305     col = "green",
306     add = TRUE
307 )
308 plotCI(x = 1:5 + 0.15,
309        y = means[4,],
310        uiw = inter[4,],
311        type = "b",
312        xaxt = "n",
313        col = "black",
314        add = TRUE
315 )
316 axis(1, at = 1:5, labels = c(1, 2, 3, 4, 5))
317 legend(x = 5.25,
318        y = 0,
319        yjust = 0,
320        lty = 1,
321        col = c("red", "blue", "green", "black"),
322        legend = c("1---", "2---", "3---", "UTC"),
323        bty = "o"
324 )
325 dev.copy(pdf,
326          paste("plots/blocks/interaction/mass/mass",
327                method.type,
328                name.order,
329                "interplot.pdf",
330                sep = "-"),
331          width = 6,
332          height = 3
333 )
334 dev.off()
335
336 capture.output(cat("sample families", "\n\n"), file = name, append = TRUE)
337 capture.output(summary(family.aov), file = name, append = TRUE)
338 capture.output(leveneTest(family.model), file = name, append = TRUE)
339 capture.output(shapiro.test(residuals(family.model)),
340               file = name, append = TRUE)
341 capture.output(TukeyHSD(family.aov), file = name, append = TRUE)
342
343 par(mar = c(4, 4, 2, 0.6) + 0.1)
344 plot(TukeyHSD(family.aov))
345 dev.copy(pdf,
346          paste("plots/blocks/tukeyHSD/family/family",
347                method.type,
348                name.order,
349                "anova.pdf",
350                sep = "-"),
351          width = 7.5,
352          height = 200
353 )
354 dev.off()
355 par(mar = c(4, 4, 1.1, 0.6) + 0.1)
356
357 family.anova = anova(family.model)
358 MS.res = family.anova$Mean[4]
359 nu = family.anova$Df[4]
360 means = tapply(temp.data$family_count,
361               list(temp.data$block, temp.data$event),
362               mean)
363
364 alpha = 0.05
365 n = tapply(temp.data$family_count,
366           list(temp.data$block, temp.data$event),
367           length)
368 se = sqrt(MS.res / n)
369 t.critical = qt(1 - alpha / 2, nu)
370 inter = t.critical * se
371
372 plotCI(x = 1:5,
373        y = means[1,],
374        uiw = inter[1,],
375        type = "b",
376        xaxt = "n",
377        col = "red",
378        xlab = "Sample Week",
379        ylab = "Mean Family Count",
380        ylim = c(-1, 25),
381        xlim = c(1, 6),
382        main = paste(name.order, " (" , method.type, ")", sep = "")
383 )
384 plotCI(x = 1:5 + 0.05,
385        y = means[2,],
386        uiw = inter[2,],
387        type = "b",
388        xaxt = "n",
389        col = "blue",
390        add = TRUE
391 )
392 plotCI(x = 1:5 + 0.1,
393        y = means[3,],
394        uiw = inter[3,],
395        type = "b",
396        xaxt = "n",
397        col = "green",
398        add = TRUE
399 )
400 plotCI(x = 1:5 + 0.15,
401        y = means[4,],
402        uiw = inter[4,],
403        type = "b",
404        xaxt = "n",
405        col = "black",
406        add = TRUE
407 )

```

```

407     axis(1, at = 1:5, labels = c(1, 2, 3, 4, 5))
408     legend(x = 5.25,
409           y = 0,
410           yjust = 0,
411           lty = 1,
412           col = c("red", "blue", "green", "black"),
413           legend = c("1---", "2---", "3---", "UTC"),
414           bty = "o"
415     )
416     dev.copy(pdf,
417             paste("plots/blocks/interaction/family/family",
418                   method.type,
419                   name.order,
420                   "interplot.pdf",
421                   sep = "_"),
422             width = 6,
423             height = 3
424     )
425     dev.off()
426   }
427 }
428 }
429
430 # Analysis 4 -----
431 # comparisons between blocks for bulk diversity and sample weight
432 # all comparisons (here and below) by sample method and in bulk
433 Analysis4 = function(raw.data) {
434   dir.create("plots/bulk")
435
436   name = "bulk.txt"
437   capture.output(cat("Analyses of Variance (bulk) \n\n"), file = name)
438
439   raw.data.2015 = raw.data[raw.data$year == 2015, ]
440   raw.data.2015$event = factor(raw.data.2015$event)
441
442   bulk.data = aggregate(formula = (cbind(mass, family_count)
443                               ~ site + event + block + method),
444                         data = raw.data.2015,
445                         FUN = sum)
446
447   bulk.lm = lm(family_count ~ mass,
448               data = bulk.data[with(bulk.data, mass > 0
449                                   & family_count > 0), ])
451
452   plot(family_count ~ mass,
453        data = bulk.data[with(bulk.data, mass > 0 & family_count > 0), ],
454        xlab = "sample mass (g)",
455        ylab = "number of families",
456        main = "Bulk (by site and event)"
457   )
458   abline(bulk.lm, col = "blue")
459   dev.copy(pdf,
460           paste("plots/bulk/2015_bulk_scatter.pdf",
461                 sep = "_"),
462           width = 3,
463           height = 3
464   )
465   dev.off()
466
467   # TODO Break down by method, then go to glmer's and MANOVA
468
469   mixed.lmer.poisson = glmer(family_count ~ mass + (1 | method),
470                             data = bulk.data[with(bulk.data, mass > 0
471                                                     & family_count > 0), ],
472                             family = poisson)
473
474   mixed.lmer.nb = glmer.nb(family_count ~ mass + (1 | method),
475                            data = bulk.data[with(bulk.data, mass > 0
476                                                    & family_count > 0), ],
477                            family = negative.binomial)
478
479   print(summary(bulk.lm))
480   print(summary(mixed.lmer.poisson))
481   print(summary(mixed.lmer.nb))
482   print(lrttest(mixed.lmer.poisson, mixed.lmer.nb))
483
484   capture.output(summary(bulk.lm), file = name, append = TRUE)
485   capture.output(summary(mixed.lmer.poisson), file = name, append = TRUE)
486   capture.output(summary(mixed.lmer.nb), file = name, append = TRUE)
487   capture.output(lrttest(mixed.lmer.poisson, mixed.lmer.nb),
488                 file = name, append = TRUE)
489
490   plot(fitted(mixed.lmer.nb),
491        residuals(mixed.lmer.nb),
492        xlab = "fitted family count",
493        ylab = "residual family count"
494   )
495   dev.copy(pdf,
496           paste("plots/bulk/2015_bulk_residual.pdf",
497                 sep = "_"),
498           width = 3,
499           height = 3
500   )
501   dev.off()
502
503   qqnorm(residuals(mixed.lmer.nb),
504          ylab = "Residual Quantiles",
505          main = NULL
506   )
507   qqline(residuals(mixed.lmer.nb),
508          probs = c(0.25, 0.75),

```

```

509     col = 'blue'
510 )
511 dev.copy(pdf,
512     paste("plots/bulk/2015_bulk_qqplot.pdf",
513         sep = "_"),
514     width = 3,
515     height = 3
516 )
517 dev.off()
518 }
519
520 # Analysis 5 -----
521 #   comparisons between blocks for bulk diversity of pollinator families
522 Analysis5 = function(raw.data) {
523 }
524 }
525
526 # Analysis 6 -----
527 #   comparisons between blocks for diversity of pollinator families, per order
528 Analysis6 = function(raw.data) {
529 }
530 }
531
532 # Main Function -----
533 main = function() {
534     #Clear workspace
535     rm(list = ls())
536
537     #Load gplots for plotCI
538     library(gplots)
539     library(car)
540     library(lme4)
541     library(lmtest)
542
543     #Create folder for plots
544     dir.create("plots")
545
546     #Minimal margins on graphs
547     par(mar = c(4, 4, 1.1, 0.6) + 0.1)
548
549     #Run analysis functions
550     readline(prompt = "reading raw_data.txt: press enter to continue.")
551     raw.data = GetData("raw_data.txt")
552
553     # readline(prompt = "first analysis: press enter to continue.")
554     # Analysis1(raw.data)
555     #
556     # readline(prompt = "second analysis: press enter to continue.")
557     # Analysis2(raw.data)
558     # #
559     # readline(prompt = "third analysis: press enter to continue.")
560     # Analysis3(raw.data)
561     # #
562     readline(prompt = "fourth analysis: press enter to continue.")
563     Analysis4(raw.data)
564     #
565     # readline(prompt = "fifth analysis: press enter to continue.")
566     # Analysis5(raw.data)
567     #
568     # readline(prompt = "sixth analysis: press enter to continue.")
569     # Analysis6(raw.data)
570 }
571
572 # Execute Code -----
573 main()

```

analysis.R