

```

1 # A brief illustration of mathematical coupling -----
2 # Mathematical Coupling with normal data
3 set.seed(1)
4 x<-rnorm(1000, 0, 1)
5 y<-rnorm(1000, 0, 1)
6 z<-y-x
7
8 cor(x,y)
9 summary(lm(y~x))
10 plot(x,y, ylim=c(-5,5), xlim=c(-5,5),cex.lab=1.5, cex.axis=1.5)
11 abline(a=-0.016187, b=-0.00643, lty=2, col="red")
12 # z~x
13 # (y-x)~x
14 # (-x+y)~X
15
16 cor(x,z)
17 summary(lm(z~x))
18 plot(x,z, ylim=c(-5,5), xlim=c(-5,5),
19       ylab="z (y-x)", cex.lab=1.5, cex.axis=1.5)
20 abline(a=-0.016, b=-0.99357, lty=2, col="red")
21
22 # Mathematical Coupling with Uniform Data data
23 set.seed(1)
24 x<-runif(1000, -5, 5)
25 y<-runif(1000, -5, 5)
26 z<-y-x
27
28 plot(x,y, ylim=c(-10,10), xlim=c(-10,10),cex.lab=1.5, cex.axis=1.5)
29 abline(a=-0.09713, b=-0.02923, lty=2, col="red")
30 cor(x,y)
31 summary(lm(y~x))
32 # z~x
33 # (y-x)~x
34 # (-x+y)~X
35
36 plot(x,z, ylim=c(-10,10), xlim=c(-10,10),
37       ylab="z (y-x)",cex.lab=1.5, cex.axis=1.5)
38 abline(a=-0.09713, b=-1.02923, lty=2, col="red")
39
40 cor(x,z)
41 summary(lm(z~x))
42
43
44
45
46 # Simulating over the Top of Empirical Data -----
47
48 getwd()
49 # Set the working directory to the appropriate location for the data file
50 # Then run the following code:
51 REAL<-read.csv("./data_EMPIRICAL.csv", header=TRUE)
52 head(REAL)
53 hist(REAL$II)
54 hist(REAL$CHANGE)
55
56 # Plotting all of the data
57 plot(x=REAL$II, y=REAL$CHANGE, ylab="Change (Final - Baseline FMA)",
58       xlab="Initial Impairment (66 - Baseline FMA)",
59       pch=1, #col=REAL$Group,
60       xlim=c(0, 66), ylim=c(-5,66), cex.lab=1.5, cex.axis=1.5)
61 summary(lm(CHANGE~0+II, data=REAL))
62 abline(a=0, b=0.41865, lty=2, lwd=3, col="red")
63
64 # Minimum Effects Test against 0.50:
65
66 pt(q=(0.41865-0.50)/0.01767, df=(373-2),

```

```

67     lower.tail = TRUE)*2
68
69 # Plotting data with non-fitters removed
70 FITTERS <- subset(REAL, Group == "fitters")
71
72 # Plotting all of the data
73 plot(x=FITTERS$II, y=FITTERS$CHANGE, ylab="Change (Final - Baseline FMA)",
74       xlab="Initial Impairment (66 - Baseline FMA)",
75       xlim=c(0, 66), ylim=c(-0,66), cex.lab=1.5, cex.axis=1.5)
76 summary(lm(CHANGE~0+II, data=FITTERS))
77 abline(a=0, b=0.76921, lty=2, lwd=3, col="red")
78
79
80 # Simulating Random Change Scores -----
81 # Set the Population size
82 N = 10000
83
84 # Create a population of uniform scores from
85 II<-rep(NA,N)
86 CHANGE<-rep(NA, N)
87 FINAL<-rep(NA, N)
88
89 SIM_POP<-data.frame(II, CHANGE, FINAL)
90
91 set.seed(1)
92 for (i in c(1:N)){
93   # Randomly samples from the real distribution of II
94   x<-sample(REAL$II, 1, replace=TRUE)
95
96   # Avoid going over the boundaries
97   ifelse(x==0,
98         SIM_POP$II[i]<-0, # Returns if above is TRUE
99         ifelse(x==66, # Returns if the above is FALSE
100              SIM_POP$II[i]<-66, # Returns if the above is TRUE
101              SIM_POP$II[i]<-x+round(runif(1,-1,1), digits=0)))
102   # Generates a random change score somewhere between II and the test max
103   SIM_POP$CHANGE[i] <- round(runif(1, min=0, max=(0+SIM_POP$II[i]), digits=0)
104   # Add the change score to the baseline score to get a final score
105   SIM_POP$FINAL[i] <- (66-SIM_POP$II[i])+SIM_POP$CHANGE[i]
106   print(i)
107 }
108
109
110 # Contrast our simulated density of II with the REAL density
111 hist(REAL$II, main="Distribution of Real Data",
112       xlab="Initial Impairment (66 - Baseline FMA)",
113       ylab="Count", cex.lab=1.5, cex.axis=1.5)
114 hist(SIM_POP$II, main="Distribution of Simulated Data",
115       xlab="Initial Impairment (66 - Baseline FMA)",
116       ylab="Count", cex.lab=1.5, cex.axis=1.5)
117
118 pop_mod<-lm(CHANGE~0+II, data = SIM_POP)
119 summary(pop_mod)
120 cor(SIM_POP$II, SIM_POP$CHANGE)
121
122 plot(x=SIM_POP$II, y=SIM_POP$CHANGE,
123       ylab="Change (Final - Baseline FMA)",
124       xlab="Initial Impairment (66 - Baseline FMA)",
125       xlim=c(0, 66), ylim=c(-0,66), cex.lab=1.5, cex.axis=1.5)
126 abline(a=0, b=0.496501, lty=2, lwd=3, col="red")
127
128
129 # Sampling from the Empirical Data
130 k_samples <- 10000
131 sample_size<-30
132

```

```

133 INT<-rep(NA, k_samples)
134 SLOPE<-rep(NA, k_samples)
135
136 SAMPLES <-data.frame(INT, SLOPE)
137
138 set.seed(3)
139 for (i in c(1:k_samples)){
140   samp<-SIM_POP[sample(nrow(SIM_POP), sample_size),]
141   x<-lm(CHANGE~II, data = samp)
142   SAMPLES$INT[i] <- x$coefficients[1]
143   SAMPLES$SLOPE[i] <- x$coefficients[2]
144   print(i)
145 }
146
147 set.seed(5)
148 samp1<-SIM_POP[sample(nrow(SIM_POP), sample_size),]
149 head(samp1)
150
151 mod01<-lm(CHANGE~II, data=samp1)
152 summary(mod01)
153 plot(samp1$II, samp1$CHANGE, main=paste("Random Sample of N =", sample_size),
154       ylab="Change (Final - Baseline FMA)",
155       xlab="Initial Impairment (66 - Baseline FIM)",
156       xlim=c(0, 66), ylim=c(0,66), cex.axis=1.5, cex.lab=1.5)
157 abline(a=-1.0036, b=0.6154, lty=2, lwd=2, col="red")
158 abline(a=0, b=1, lty=1, lwd=1, col="black")
159
160
161 plot(density(SAMPLES$SLOPE), col="royal blue", lwd=2, lty=2,
162       main=paste("Distribution of Sample Slopes, N =", sample_size),
163       xlab="Observed Slope",
164       xlim=c(-1,1), cex.lab=1.5, cex.axis=1.5)
165
166 # Calculate the standard error
167 SE<-sd(SAMPLES$SLOPE)
168 alt_dist<-dnorm(c(seq(from=0, to=1, by=0.01)), mean=0.5, sd=SE)
169 null_dist<-dnorm(c(seq(from=-0.5, to=0.5, by=0.01)), mean=0, sd=SE)
170 plot(c(seq(from=0, to=1, by=0.01)),
171       alt_dist,
172       type = "l", col="royal blue", lwd=2, lty=2,
173       main=paste("Distribution of Sample Slopes, N =", sample_size),
174       xlab="Observed Slope",
175       ylab="Density",
176       xlim=c(-1,1), cex.lab=1.5, cex.axis=1.5)
177 lines(c(seq(from=-0.5, to=0.5, by=0.01)),
178       null_dist, col="black", lty=2, lwd=2)
179 abline(v=0.615, col="red", lwd=1)
180
181
182 # Incorrectly doing a one sample t-test against zero:
183 pt(q=(0.615-0)/0.132, df=(sample_size-2),
184     lower.tail = FALSE)*2
185
186 # Correctly doing a one sample t-test against the mean of the
187 # sampling distribution, approximately 0.5
188 pt(q=(0.615-mean(SAMPLES$SLOPE))/0.132, df=(sample_size-2),
189     lower.tail = FALSE)*2
190
191
192
193 # Plotting a biased distribution wih non-fitters excluded -----
194 # Using Hierarchical Cluster Analysis to Identify Non-Fitters
195 head(samp1)
196 samp1$pred<-0.70*samp1$II
197 samp1$resid<-samp1$CHANGE-samp1$pred
198 head(samp1)

```

```

199
200 # Install the ecodist package to get access to mahalanobis distances.
201 library(ecodist)
202 ?ecodist::distance
203 clusters <- hclust(distance(samp1[, c("CHANGE", "pred")], method="mahalanobis"),
204                   method="ward.D2") # In Winters et al. 2015, their reference suggests
                                   Ward 1963
205 plot(clusters)
206 cutree(clusters, 2)
207 samp1$cluster<-cutree(clusters, 2)
208 samp1
209
210
211 # Note that we arrive at similar outlier exclusions even though the
212 # population data were random
213 plot(samp1$II, samp1$CHANGE, main=paste("Random Sample of N =", sample_size),
214      ylab="Change (Final - Baseline FMA)",
215      xlab="Initial Impairment (66 - Baseline FIM)",
216      pch=21, col=samp1$cluster,
217      xlim=c(0, 66), ylim=c(0,66), cex.axis=1.5, cex.lab=1.5)
218
219 summary(lm(CHANGE~II, data=samp1[samp1$cluster==2,]))
220 summary(lm(CHANGE~II, data=samp1))
221
222 # Biased sampling distribution
223 sample_size <- 30
224 k_samples <- 50
225
226 INT_TOTAL<-rep(NA, k_samples)
227 SLOPE_TOTAL<-rep(NA, k_samples)
228
229 BIASED <-data.frame(INT_TOTAL, SLOPE_TOTAL)
230
231 set.seed(1)
232 for (i in c(1:k_samples)){
233   # Obtain a sample
234   samp<-SIM_POP[sample(nrow(SIM_POP), sample_size),]
235
236   # Get the total slopes and intercepts
237   tot_mod <-lm(CHANGE~II, data=samp)
238   BIASED$INT_TOTAL[i] <- tot_mod$coefficients[1]
239   BIASED$SLOPE_TOTAL[i] <- tot_mod$coefficients[2]
240
241   # Get the residuals and fitted values for your model
242   samp$pred<-0.70*samp$II
243   samp$resid<-samp$CHANGE-samp$pred
244
245   # Get your clusters
246   clusters <- hclust(distance(samp[, c("CHANGE", "pred")], method="mahalanobis"),
247                       method="ward.D2")
248   samp$cluster<-cutree(clusters, 2)
249
250   # Write the results slopes and intercepts
251   # Cluster 1
252   BIASED$C1_INT_BIAS[i] <- lm(CHANGE~II, data = samp[samp$cluster==1,])$coefficients[1]
253   BIASED$C1_SLOPE_BIAS[i] <- lm(CHANGE~II, data = samp[samp$cluster==1,])$coefficients[2]
254   BIASED$C1_size[i]<-nrow(samp[samp$cluster==1,])
255   BIASED$C1_min_II[i]<-min(samp$II[samp$cluster==1])
256   BIASED$C1_max_II[i]<-max(samp$II[samp$cluster==1])
257   BIASED$C1_mean_CHANGE[i]<-mean(samp$CHANGE[samp$cluster==1])
258
259
260   # Cluster 2
261   BIASED$C2_INT_BIAS[i] <- lm(CHANGE~II, data = samp[samp$cluster==2,])$coefficients[1]
262   BIASED$C2_SLOPE_BIAS[i] <- lm(CHANGE~II, data = samp[samp$cluster==2,])$coefficients[2]
263   BIASED$C2_size[i]<-nrow(samp[samp$cluster==2,])

```

```

264 BIASED$C2_min_II[i]<-min(samp$II[samp$cluster==2])
265 BIASED$C2_max_II[i]<-max(samp$II[samp$cluster==2])
266 BIASED$C2_mean_CHANGE[i]<-mean(samp$CHANGE[samp$cluster==2])
267
268 # Change the # of samples and uncomment these lines to get images of classification
269 # png(paste("sample", i, ".png"))
270 # # Create a plot
271 # plot(samp$II, samp$CHANGE, main=paste("Random Sample of N =", sample_size),
272 #       ylab="Change (Final - Baseline FMA)",
273 #       xlab="Initial Impairment (66 - Baseline FIM)",
274 #       pch=21, col=samp$cluster,
275 #       xlim=c(0, 66), ylim=c(0,66), cex.axis=1.5, cex.lab=1.5)
276 # # Close the pdf file
277 # dev.off()
278 #
279 # print(i)
280
281 }
282
283 head(BIASED)
284 getwd()
285
286 # Problematically, the cluster analysis does not identify responders and
287 # non-responders:
288 summary(BIASED$C1_SLOPE_BIAS)
289 plot(density(BIASED$C1_SLOPE_BIAS, na.rm=TRUE), col="navy", lwd=2, lty=2,
290       main=paste("Distribution of Sample Slopes, N =", sample_size),
291       xlab="Observed Slope",
292       xlim=c(-2,2), cex.lab=1.5, cex.axis=1.5)
293 abline(v=0.7, col="red", lwd=1)
294
295 summary(BIASED$C2_SLOPE_BIAS)
296 plot(density(BIASED$C2_SLOPE_BIAS, na.rm=TRUE), col="navy", lwd=2, lty=2,
297       main=paste("Distribution of Sample Slopes, N =", sample_size),
298       xlab="Observed Slope",
299       xlim=c(-2,2), cex.lab=1.5, cex.axis=1.5)
300 abline(v=0.7, col="red", lwd=1)
301
302 plot(y=BIASED$C1_SLOPE_BIAS, x=BIASED$C1_mean_CHANGE,
303       col="black", lwd=2,
304       cex.lab=1.5, cex.axis=1.5)
305
306 plot(y=BIASED$C1_SLOPE_BIAS, x=BIASED$C1_size,
307       col="black", lwd=2,
308       cex.lab=1.5, cex.axis=1.5)
309
310 # Processing the data to remove poorly obtained clusters
311 # Based on Size
312 BIASED_FILTER<-subset(BIASED,
313                       C1_size >=(0.05*sample_size) & C2_size >= (0.05*sample_size))
314 # Based on Initial Impairment
315 BIASED_FILTER<-subset(BIASED_FILTER,
316                       C1_min_II < C2_max_II & C2_min_II < C1_max_II)
317
318 plot(y=BIASED_FILTER$C1_SLOPE_BIAS, x=BIASED_FILTER$C1_size,
319       col="black", lwd=2,
320       cex.lab=1.5, cex.axis=1.5)
321
322 plot(y=BIASED_FILTER$C2_mean_CHANGE, x=BIASED_FILTER$C1_mean_CHANGE,
323       col="black", lwd=2,
324       cex.lab=1.5, cex.axis=1.5)
325
326 plot(density(BIASED_FILTER$C1_SLOPE_BIAS, na.rm=TRUE), col="navy", lwd=2, lty=2,
327       main=paste("Distribution of Sample Slopes, N =", sample_size),
328       xlab="Observed Slope",
329       xlim=c(-5,5), cex.lab=1.5, cex.axis=1.5)

```

```

330
331
332
333 # Identifying the fitter group based on their Change scores
334 BIASED_FILTER$INT_FITTERS<-ifelse(BIASED_FILTER$C1_mean_CHANGE>BIASED_FILTER$C2_mean_CHAN
GE,
335     BIASED_FILTER$C1_INT_BIAS, # Returns if above is TRUE
336     BIASED_FILTER$C2_INT_BIAS) # Return if above is FALSE
337
338 BIASED_FILTER$SLOPE_FITTERS<-ifelse(BIASED_FILTER$C1_mean_CHANGE>BIASED_FILTER$C2_mean_CH
ANGE,
339     BIASED_FILTER$C1_SLOPE_BIAS, # Returns if above is TRUE
340     BIASED_FILTER$C2_SLOPE_BIAS) # Return if above is FALSE
341
342 plot(density(BIASED_FILTER$SLOPE_FITTERS, na.rm=TRUE), col="navy", lwd=2, lty=2,
343     main=paste("Distribution of Sample Slopes, N =", sample_size),
344     xlab="Observed Slope",
345     xlim=c(-2,2),
346     cex.lab=1.5, cex.axis=1.5)
347
348 head(BIASED_FILTER)
349 write.csv(BIASED_FILTER, paste("results_HCL_n",sample_size,".csv"))
350
351 # Testing the slope observed in the REAL data
352 summary(lm(CHANGE~0+II, data=FITTERS))
353 summary(BIASED_FILTER$SLOPE_FITTERS)
354 sum(BIASED_FILTER$SLOPE_FITTERS>=0.76921, na.rm=TRUE)
355
356 plot(density(BIASED_FILTER$SLOPE_FITTERS, na.rm=TRUE), col="navy", lwd=2, lty=2,
357     main=paste("Distribution of Sample Slopes, N =", sample_size),
358     xlab="Observed Slope",
359     xlim=c(0,2), cex.lab=1.5, cex.axis=1.5)
360 abline(v=0.76921, col="red", lwd=1)
361
362

```