Supplementary Material for *'Tackling Early Medieval Dietary Transitions Using a Hierarchical and Multi-isotope Approach'*

Further Statistical Information

They are a mixture of various statistical and graphical methods for determining the optimal number of clusters, 30 of which are automatically computed as part of the "NbClust" package, and the other graphical indices were generated using a combination of "NbClust" and "factoextra" packages in R (Charrad et al., 2014; Kassambara, 2017; Kassambara and Mundt, 2017). All indices were used for every UML iteration, and the majority rule adhered to (optimal number of clusters determined by agreement of the highest number of indices) to avoid user determination. The pre-sets for all indices were kept (generally this also means a 95% confidence interval if they were used by the index, see code and "NbClust" documentation for mathematical details), and the algorithm run for the clustering method used (Ward2 hierarchical).

Hierarchical Clustering "NbClust" outputs

Bone

```
NbClust(data = EMEUboneclean, diss = NULL, distance = "euclidean", min.nc = 2, max.nc =
15, method = "ward.D2")
*** : The Hubert index is a graphical method of determining the number of clusters.
          In the plot of Hubert index, we seek a significant knee that corresponds to a
          significant increase of the value of the measure i.e the significant peak in Hubert
          index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
          In the plot of D index, we seek a significant knee (the significant peak in Dindex
          second differences plot) that corresponds to a significant increase of the value of
          the measure.

*******************************************************************
* Among all indices:
* 3 proposed 2 as the best number of clusters
* 8 proposed 3 as the best number of clusters
* 2 proposed 4 as the best number of clusters
* 1 proposed 5 as the best number of clusters
* 6 proposed 6 as the best number of clusters
* 1 proposed 7 as the best number of clusters
* 1 proposed 12 as the best number of clusters
* 1 proposed 15 as the best number of clusters

            ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3


*******************************************************************
$All.index
```

```
      KL      CH  Hartigan     CCC     Scott  Marriot     TrCovW   TraceW Friedman  Rubin Cindex
DB Silhouette   Duda
2  2.9470 1923.879 1552.1379 -22.9149  3595.199 28062699 10559187.3 5648.984   1.3085
1.4650 0.1413 1.4502     0.3921 0.5151
3  5.2031 2098.406 1291.8626 -42.5995  5804.395 37025991  3765891.2 4107.801   2.1228
2.0147 0.1388 1.0617     0.4075 0.6043
4  1.1856 2265.962  966.0241 -38.5583  8121.882 37602469  1738443.2 3130.121   3.4543
2.6440 0.1565 1.0465     0.3121 0.5344
5  1.5564 2337.445  868.9175 -36.8777  9827.143 38914172  1151233.5 2537.343   4.6931
3.2617 0.1441 1.0428     0.3162 0.5629
6  0.1082 2436.193  337.8742 -34.5102 11286.202 39389006   999386.8 2096.652   5.9249
3.9472 0.1183 1.1218     0.3068 0.6733
7  1.4673 2251.894  367.1101 -39.2452 12000.497 45114845   707889.5 1938.203   6.6749
4.2699 0.1106 1.1897     0.2896 0.5654
8  1.3385 2153.608  357.8757 -41.9811 12646.659 50408444   677554.7 1780.053   7.3405
4.6493 0.1221 1.1786     0.2866 0.6735
9  1.1280 2091.884  324.2138 -43.8069 13311.730 54328058   677271.5 1638.139   8.1065
5.0521 0.1205 1.1299     0.2902 0.5711
10 2.0277 2040.953  348.1011 -45.3730 13966.314 57260534   512580.7 1518.902   8.9695
5.4487 0.1156 1.1308     0.2811 0.5685
11 2.1568 2026.036  345.6806 -45.9072 14668.331 58476296   378339.8 1400.805   9.9641
5.9080 0.1138 1.0910     0.2821 0.4325
12 2.0891 2027.022  329.8978 -45.9650 15291.359 59866562   370837.8 1292.565  10.8585
6.4028 0.1462 1.0017     0.2826 0.6694
13 1.2932 2033.633  328.1841 -45.8458 15957.638 59813276   287846.6 1196.890  11.9718
6.9146 0.1411 1.0205     0.2557 0.6409
14 0.2107 2051.260  244.2454 -45.3911 16548.071 60147063   285571.4 1108.703  12.9476
7.4646 0.1323 1.0586     0.2523 0.5438
15 1.7699 2034.476  246.7738 -45.9523 17020.621 61596685   281030.2 1046.725  13.8212
7.9066 0.1260 1.0494     0.2570 0.5299
   Pseudot2  Beale Ratkowsky     Ball Ptbiserial    Frey McClain   Dunn Hubert SDindex Dindex
SDbw
2  1146.7191 0.9407    0.3673 2824.4922     0.4729 -0.4888  0.3803 0.0067  2e-04  6.3540
0.9403 1.4088
3  1910.0016 0.6546    0.4087 1369.2671     0.5423  1.7266  0.3885 0.0072  2e-04  4.5541
0.8232 0.8481
4   816.4711 0.8704    0.3940  782.5302     0.4589  0.0132  0.9319 0.0040  2e-04  4.4837
0.7153 0.9543
5  1518.3939 0.7763    0.3723  507.4687     0.4816  0.6693  0.9502 0.0040  3e-04  4.0522
0.6508 0.7618
6   465.7589 0.4847    0.3528  349.4420     0.4437  0.4731  1.3848 0.0040  3e-04  4.2693
0.5742 0.6447
7   214.4304 0.7658    0.3308  276.8862     0.4345  0.0040  1.5027 0.0040  3e-04  4.9696
0.5426 0.5867
8   142.5110 0.4831    0.3132  222.5067     0.4371 -0.0846  1.4936 0.0044  3e-04  4.8725
0.5256 0.5206
```

```
9   491.1249 0.7498    0.2985  182.0154     0.4397  0.4829  1.4809 0.0044  3e-04  4.6677
0.5135 0.4678
10  202.6979 0.7563    0.2857  151.8902     0.4331  0.0175  1.5512 0.0044  3e-04  4.6145
0.4920 0.4626
11  356.9464 1.3075    0.2748  127.3459     0.4349 -0.1083  1.5427 0.0044  3e-04  4.3367
0.4798 0.4260
12  641.5641 0.4935    0.2652  107.7137     0.4367  2.3670  1.5319 0.0058  3e-04  5.5693
0.4732 0.4464
13  359.1052 0.5594    0.2565   92.0684     0.3687  0.2772  2.2394 0.0058  4e-04  7.0510
0.4486 0.4399
14  575.4163 0.8376    0.2487   79.1931     0.3624  0.3663  2.3322 0.0058  4e-04  7.1806
0.4299 0.4010
15  187.1673 0.8829    0.2413   69.7817     0.3544  0.0838  2.4430 0.0058  4e-04  7.1968
0.4162 0.3780

$All.CriticalValues
   CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
2         0.6110          775.3291       0.3905
3         0.6360         1669.3758       0.5197
4         0.6012          621.6597       0.4189
5         0.6259         1168.4713       0.4602
6         0.6021          634.3506       0.6160
7         0.5345          243.0072       0.4654
8         0.5383          252.2106       0.6171
9         0.5853          463.2995       0.4727
10        0.5312          235.6094       0.4699
11        0.5326          238.6958       0.2713
12        0.6133          819.1438       0.6105
13        0.5844          455.9063       0.5717
14        0.5876          481.4464       0.4330
15        0.5126          200.6259       0.4144

$Best.nc
                   KL       CH Hartigan      CCC    Scott Marriot  TrCovW   TraceW Friedman   Rubin
Cindex     DB Silhouette
Number_clusters 3.0000    6.000   6.0000   2.0000    4.000       6       3   3.0000   4.0000
6.0000 7.0000 12.0000     3.0000
Value_Index     5.2031 2436.193 531.0433 -22.9149 2317.486 5251004 6793296 563.5026
1.3315 -0.3629 0.1106  1.0017     0.4075
                 Duda PseudoT2  Beale Ratkowsky     Ball PtBiserial Frey McClain   Dunn Hubert
SDindex Dindex   SDbw
Number_clusters 6.0000   6.0000 2.0000    3.0000    3.000     3.0000    1  2.0000 3.0000      0
5.0000      0 15.000
Value_Index     0.6733 465.7589 0.9407    0.4087 1455.225     0.5423   NA  0.3803 0.0072
0  4.0522      0  0.378

$Best.partition TOO MANY TO PRINT
```

Dentine

```
NbClust(data = EMEUdentineclean, diss = NULL, distance = "euclidean", min.nc = 2, max.nc =
15, method = "ward.D2")
*** : The Hubert index is a graphical method of determining the number of clusters.
            In the plot of Hubert index, we seek a significant knee that corresponds to a
            significant increase of the value of the measure i.e the significant peak in Hubert
            index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
            In the plot of D index, we seek a significant knee (the significant peak in Dindex
            second differences plot) that corresponds to a significant increase of the value of
            the measure.

*******************************************************************
* Among all indices:
* 1 proposed 2 as the best number of clusters
* 11 proposed 3 as the best number of clusters
* 2 proposed 4 as the best number of clusters
* 5 proposed 5 as the best number of clusters
* 2 proposed 11 as the best number of clusters
* 1 proposed 13 as the best number of clusters
* 1 proposed 15 as the best number of clusters

              ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3


*******************************************************************
$All.index
      KL      CH Hartigan     CCC    Scott  Marriot     TrCovW    TraceW Friedman   Rubin Cindex
DB Silhouette   Duda
2  1.1605 689.8837 638.3127 -3.0678 1164.951  943438.6 512552.084 1067.3651   1.9708
1.7426 0.1490 1.1517     0.4481 0.4261
3  2.6473 900.1366 253.6577 -2.3540 1970.244  893803.0  92376.167  632.6639   3.9210
2.9399 0.1860 0.7636     0.4883 0.5277
4  0.7343 847.7568 307.3965 -4.1147 2462.697  936265.0  91811.213  496.8547   5.7963
3.7435 0.2032 0.8730     0.4005 0.6008
5  1.0312 922.5099 119.0435 -1.4182 3000.736  820792.8  36394.359  373.1251   8.5692
4.9849 0.1680 0.9714     0.3386 0.6702
6  1.0541 855.7673  96.4745 -3.7506 3272.885  882355.4  22899.105  330.6215  10.5059
5.6258 0.1540 1.1333     0.2984 0.6277
7  1.3558 802.7278  91.9402 -5.7153 3483.187  958153.5  14718.625  299.3955  12.1124
6.2125 0.1442 1.1617     0.2933 0.4261
8  1.9342 768.8166  95.4013 -7.0374 3623.368 1076532.1  14011.121  272.3009  12.9927
6.8307 0.1390 1.0359     0.2830 0.6240
```

```
9  0.8749 753.3544  80.4051 -7.6688 3840.429 1079138.0   8810.811  246.7924  15.0569
7.5367 0.1321 1.1063     0.2787 0.4945
10 2.5513 736.1811  83.6286 -8.3866 3959.482 1172345.9   7775.685  226.9967  15.8773
8.1940 0.1264 1.0674     0.2896 0.6703
11 6.6935 730.2941  82.3841 -8.6536 4093.848 1227897.2   7708.636  208.1007  17.0057
8.9380 0.1171 1.0821     0.2998 0.5143
12 0.1012 730.0500  89.0134 -8.6907 4231.375 1260623.2   6410.307  190.9973  18.2310
9.7384 0.1633 1.0967     0.2933 0.4115
13 0.5441 740.6428  93.8351 -8.2839 4379.846 1261393.5   6026.755  174.1311  19.6721
10.6816 0.2065 1.0337     0.2956 0.5413
14 1.2512 759.9423  84.2959 -7.5343 4561.080 1204142.0   5887.249  157.9826  21.9124
11.7734 0.1913 1.0063     0.2988 0.5232
15 1.6569 775.7033  72.4795 -6.9416 4712.309 1175055.7   5004.047  144.6826  23.7635
12.8557 0.1802 0.9768     0.3089 0.3920
   Pseudot2  Beale Ratkowsky     Ball Ptbiserial    Frey McClain   Dunn Hubert SDindex Dindex
SDbw
2  507.7796 1.3433    0.4536 533.6826     0.4960 -0.5753  0.4751 0.0062 0.0007  3.9937
0.8394 1.2743
3  307.0376 0.8926    0.4689 210.8880     0.6042  1.1814  0.4414 0.0090 0.0011  2.9269
0.7184 0.6612
4  365.4913 0.6633    0.4280 124.2137     0.5918  1.4329  0.5445 0.0106 0.0012  2.9958
0.6352 0.5551
5  180.1015 0.4907    0.3997  74.6250     0.5044  1.6433  1.0051 0.0105 0.0013  3.3165
0.5475 0.5778
6  132.8464 0.5904    0.3700  55.1036     0.4357  0.6402  1.4856 0.0105 0.0014  4.5300
0.5036 0.4619
7  290.8936 1.3405    0.3461  42.7708     0.4183  1.5417  1.6717 0.0105 0.0014  4.5532
0.4793 0.5807
8   70.5056 0.5975    0.3266  34.0376     0.3855  0.2080  2.0231 0.0105 0.0014  4.5331
0.4476 0.6078
9  151.2832 1.0153    0.3104  27.4214     0.3838  0.4343  2.0624 0.0105 0.0014  4.3006
0.4303 0.4894
10  89.5206 0.4892    0.2963  22.6997     0.3753  0.3896  2.1756 0.0105 0.0014  4.8752
0.4102 0.4409
11  30.2145 0.9156    0.2841  18.9182     0.3621  0.0127  2.3532 0.0105 0.0015  4.7917
0.3952 0.4127
12  70.0885 1.4018    0.2734  15.9164     0.3629  0.0078  2.3400 0.0148 0.0015  4.8309
0.3876 0.3185
13 130.5143 0.8420    0.2640  13.3947     0.3641  0.3873  2.3210 0.0190 0.0015  4.5056
0.3772 0.2832
14 122.1056 0.9045    0.2557  11.2845     0.3503  0.1081  2.4950 0.0190 0.0015  4.6258
0.3560 0.2872
15 105.4624 1.5284    0.2480   9.6455     0.3501  0.2101  2.4658 0.0190 0.0015  4.7841
0.3452 0.2635

$All.CriticalValues
   CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
```

```
2        0.5549         302.3658      0.2616
3        0.5488         281.9660      0.4101
4        0.5767         403.7767      0.5153
5        0.5530         295.7861      0.6124
6        0.5175         208.8197      0.5545
7        0.5146         203.7829      0.2628
8        0.4555         139.8775      0.5510
9        0.4802         160.2064      0.3635
10        0.4998         182.1721      0.6135
11        0.2585          91.8049      0.4055
12        0.3361          96.7755      0.2511
13        0.4841         164.1092      0.4318
14        0.4701         151.0575      0.4060
15        0.3867         107.8309      0.2206

$Best.nc
                KL       CH Hartigan     CCC    Scott  Marriot   TrCovW    TraceW Friedman   Rubin
Cindex     DB
Number_clusters 11.0000   5.0000    3.000  5.0000   3.0000      5.0      3.0   3.0000   5.0000
5.0000 11.0000 3.0000
Value_Index      6.6935 922.5099  384.655 -1.4182 805.2928 177034.9 420175.9 298.8919
2.7728 -0.6005  0.1171 0.7636
            Silhouette   Duda PseudoT2  Beale Ratkowsky     Ball PtBiserial Frey McClain   Dunn
Hubert SDindex Dindex
Number_clusters     3.0000 4.0000   4.0000 2.0000    3.0000   3.0000     3.0000    1  3.0000
13.000      0  3.0000      0
Value_Index        0.4883 0.6008 365.4913 1.3433    0.4689 322.7946     0.6042   NA  0.4414
0.019      0  2.9269      0
                SDbw
Number_clusters 15.0000
Value_Index      0.2635

$Best.partition
 CND_01  CND_02  CND_03  CND_04  CND_05  CND_06  CND_07  CND_08  CND_09  CND_10
CND_11  CND_12  CND_13  CND_14  CND_15  CND_16
     1       1       1       1       1       1       1       2       1       1       2       2       2       2       2       2
 CND_17  CND_18  CND_19  CND_20  CND_21  CND_22  CND_23  CND_24  CND_25  CND_26
CND_27  CND_28  CND_29  CND_30  CND_31  CND_32
     2       2       2       2       2       2       2       1       2       2       2       2       2       2       2       2
 CND_33  CND_34  CND_35  CND_36  CND_37  CND_38  CND_39  CND_40  CND_41  CND_42
CND_43  CND_44  CND_45  CND_46  CND_47  CND_48
     2       2       2       2       2       2       2       2       1       2       2       2       2       2       1       1
 CND_49  CND_50  CND_51  CND_52  CND_53  CND_54  CND_55  CND_56  CND_57  CND_58
CND_59  CND_60  CND_61  CND_62  CND_63  CND_64
     1       2       2       2       1       2       2       2       2       2       2       2       2       2       2       2
 CND_65  CND_66  CND_67  CND_68  CND_69  CND_70  CND_71  CND_72  CND_73  CND_74
CND_75  CND_76  CND_77  CND_78  CND_79  CND_80
```

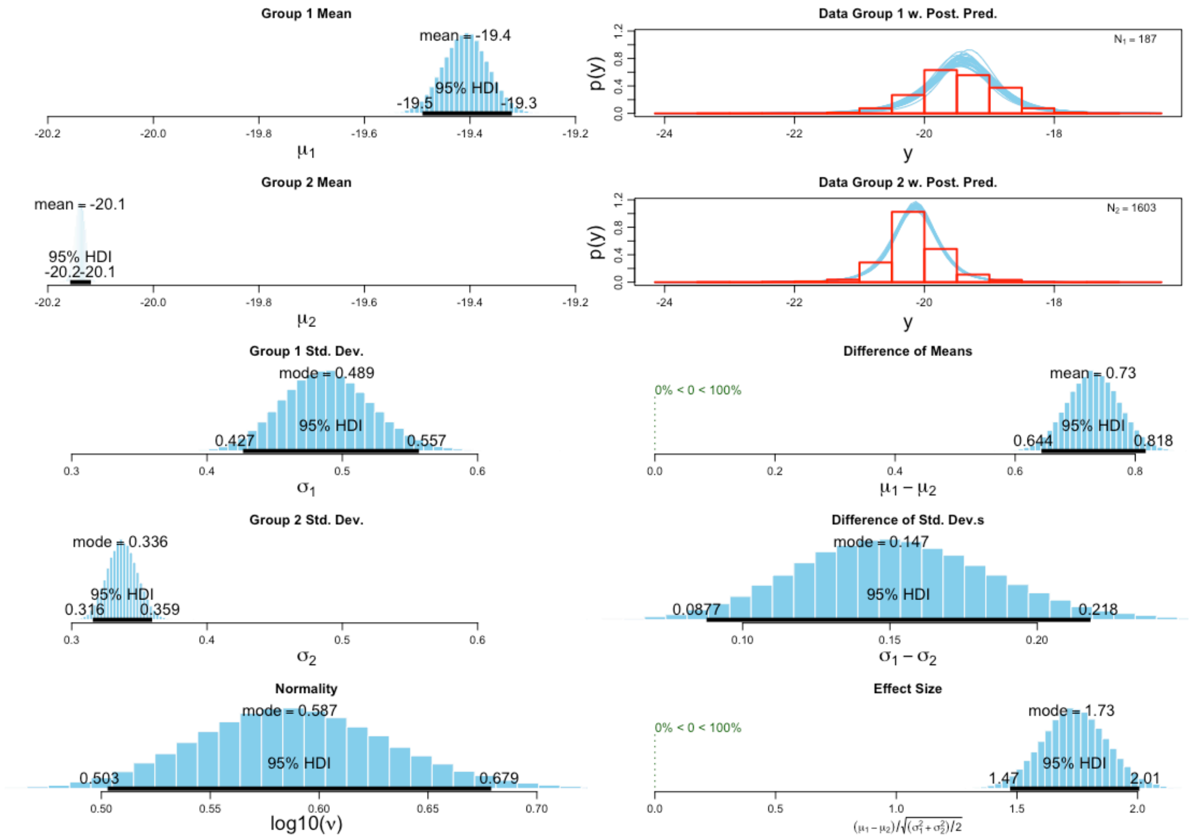
```
      2       2       2       1       1       1       2       2       2       1       2       2       1       2       1       1 
 CND_81  CND_82  CND_83  CND_84  CND_85  CND_86  CND_87  CND_88  CND_89  CND_90 
CND_91  CND_92  CND_93  CND_94  CND_95  CND_96 
      1       2       2       1       1       1       1       1       1       1       1       1       1       2       2       2 
 CND_97  CND_98  CND_99 CND_100 CND_101 CND_102 CND_103 CND_104 CND_105 
CND_106 CND_107 CND_108 CND_109 CND_110 CND_111 CND_112 
      2       2       2       1       2       2       1       1       1       2       1       1       1       1       1       1 
CND_113 CND_114 CND_115 CND_116 CND_117 CND_118 CND_119 CND_120 CND_121 
CND_122 CND_123 CND_124 CND_125 CND_126 CND_127 CND_128 
      1       1       1       1       1       1       1       1       1       2       2       2       2       1       1       2 
CND_129 CND_130 CND_131 CND_132 CND_133 CND_134 CND_135 CND_136 CND_137 
CND_138 CND_139 CND_140 CND_141 CND_142 CND_143 CND_144 
      2       1       1       1       2       2       2       1       1       1       2       2       2       1       2       1 
CND_145 CND_146 CND_147 CND_148 CND_149 CND_150 CND_151 CND_152 CND_153 
CND_154 CND_155 CND_156 CND_157 CND_158 CND_159 CND_160 
      1       1       1       1       2       2       2       1       1       1       2       2       2       1       1       1 
CND_161 CND_162 CND_163 CND_164 CND_165 CND_166 CND_167 CND_168 CND_169 
CND_170 CND_171 CND_172 CND_173 CND_174 CND_175 CND_176 
      1       1       1       2       1       1       1       1       1       1       1       1       1       1       1       2 
CND_177 CND_178 CND_179 CND_180 CND_181 CND_182 CND_183 CND_184 CND_185 
CND_186 CND_187 CND_188 CND_189 CND_190 CND_191 CND_192 
      2       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 
CND_193 CND_194 CND_195 CND_196 CND_197 CND_198 CND_199 CND_200 CND_201 
CND_202 CND_203 CND_204 CND_205 CND_206 CND_207 CND_208 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       2       1 
CND_209 CND_210 CND_211 CND_212 CND_213 CND_214 CND_215 CND_216 CND_217 
CND_218 CND_219 CND_220 CND_221 CND_222 CND_223 CND_224 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 
CND_225 CND_226 CND_227 CND_228 CND_229 CND_230 CND_231 CND_232 CND_233 
CND_234 CND_235 CND_236 CND_237 CND_238 CND_239 CND_240 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 
CND_241 CND_242 CND_243 CND_244 CND_245 CND_246 CND_247 CND_248 CND_249 
CND_250 CND_251 CND_252 CND_253 CND_254 CND_255 CND_256 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 
CND_257 CND_258 CND_259 CND_260 CND_261 CND_262 CND_263 CND_264 CND_265 
CND_266 CND_267 CND_268 CND_269 CND_270 CND_271 CND_272 
      1       1       1       1       1       1       1       1       1       1       1       1       1       2       1       1 
CND_273 CND_274 CND_275 CND_276 CND_277 CND_278 CND_279 CND_280 CND_281 
CND_282 CND_283 CND_284 CND_285 CND_286 CND_287 CND_288 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 
CND_289 CND_290 CND_291 CND_292 CND_293 CND_294 CND_295 CND_296 CND_297 
CND_298 CND_299 CND_300 CND_301 CND_302 CND_303 CND_304 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 
CND_305 CND_306 CND_307 CND_308 CND_309 CND_310 CND_311 CND_312 CND_313 
CND_314 CND_315 CND_316 CND_317 CND_318 CND_319 CND_320 
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       2       1 
```

```
CND_321 CND_322 CND_323 CND_324 CND_325 CND_326 CND_327 CND_328 CND_329
CND_330 CND_331 CND_332 CND_333 CND_334 CND_335 CND_336
    1     1     2     1     1     2     1     2     1     1     2     1     1     1     2     2
CND_337 CND_338 CND_339 CND_340 CND_341 CND_342 CND_343 CND_344 CND_345
CND_346 CND_347 CND_348 CND_349 CND_350 CND_351 CND_352
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
CND_353 CND_354 CND_355 CND_356 CND_357 CND_358 CND_359 CND_360 CND_361
CND_362 CND_363 CND_364 CND_365 CND_366 CND_367 CND_368
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
CND_369 CND_370 CND_371 CND_372 CND_373 CND_374 CND_375 CND_376 CND_377
CND_378 CND_379 CND_380 CND_381 CND_382 CND_383 CND_384
    1     1     2     1     1     1     1     1     1     1     1     1     1     1     1     1
CND_385 CND_386 CND_387 CND_388 CND_389 CND_390 CND_391 CND_392 CND_393
CND_394 CND_395 CND_396 CND_397 CND_398 CND_399 CND_400
    2     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
CND_401 CND_402 CND_403 CND_404 CND_405 CND_406 CND_407 CND_408 CND_409
CND_410 CND_411 CND_412 CND_413 CND_414 CND_415 CND_416
    1     1     1     1     1     1     1     1     1     1     1     2     1     1     1     1
CND_417 CND_418 CND_419 CND_420 CND_421 CND_422 CND_423 CND_424 CND_425
CND_426 CND_427 CND_428 CND_429 CND_430 CND_431 CND_432
    1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
CND_433 CND_434 CND_435 CND_436 CND_437 CND_438 CND_439 CND_440 CND_441
CND_442 CND_443 CND_444 CND_445 CND_446 CND_447 CND_448
    1     2     1     1     1     1     1     1     1     1     1     1     1     1     1     1
CND_449 CND_450 CND_451 CND_452 CND_453 CND_454 CND_455 CND_456 CND_457
CND_458 CND_459 CND_460 CND_461 CND_462 CND_463 CND_464
    1     1     1     1     1     1     1     1     1     2     1     1     1     1     3     1
CND_465 CND_466 CND_467 CND_468 CND_469 CND_470 CND_471 CND_472 CND_473
CND_474 CND_475 CND_476 CND_477 CND_478 CND_479 CND_480
    3     3     3     3     3     3     3     3     3     3     3     3     3     3     3     3
CND_481 CND_482 CND_483 CND_484 CND_485 CND_486 CND_487 CND_488 CND_489
CND_490 CND_491 CND_492 CND_493 CND_494 CND_495 CND_496
    3     3     3     3     3     3     3     3     3     3     3     3     3     2     2     2
CND_497 CND_498 CND_499 CND_500 CND_501 CND_502 CND_503 CND_504 CND_505
CND_506 CND_507 CND_508 CND_509 CND_510 CND_511 CND_512
    2     1     2     1     1     1     2     2     2     2     2     2     2     2     2     2
CND_513 CND_514 CND_515 CND_516 CND_517 CND_518 CND_519 CND_520 CND_521
CND_522 CND_523 CND_524 CND_525 CND_526 CND_527 CND_528
    2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2
CND_529 CND_530 CND_531 CND_532 CND_533 CND_534 CND_535 CND_536 CND_537
CND_538 CND_539 CND_540 CND_541 CND_542 CND_543 CND_544
    1     1     1     2     2     2     2     2     2     2     1     2     2     2     2     2
CND_545 CND_546 CND_547 CND_548 CND_549 CND_550 CND_551 CND_552 CND_553
CND_554 CND_555 CND_556 CND_557 CND_558 CND_559 CND_560
    2     2     2     2     2     2     2     2     2     1     1     1     1     1     1     1
CND_561 CND_562 CND_563 CND_564 CND_565 CND_566 CND_567 CND_568 CND_569
CND_570 CND_571 CND_572 CND_573 CND_574 CND_575 CND_576
```

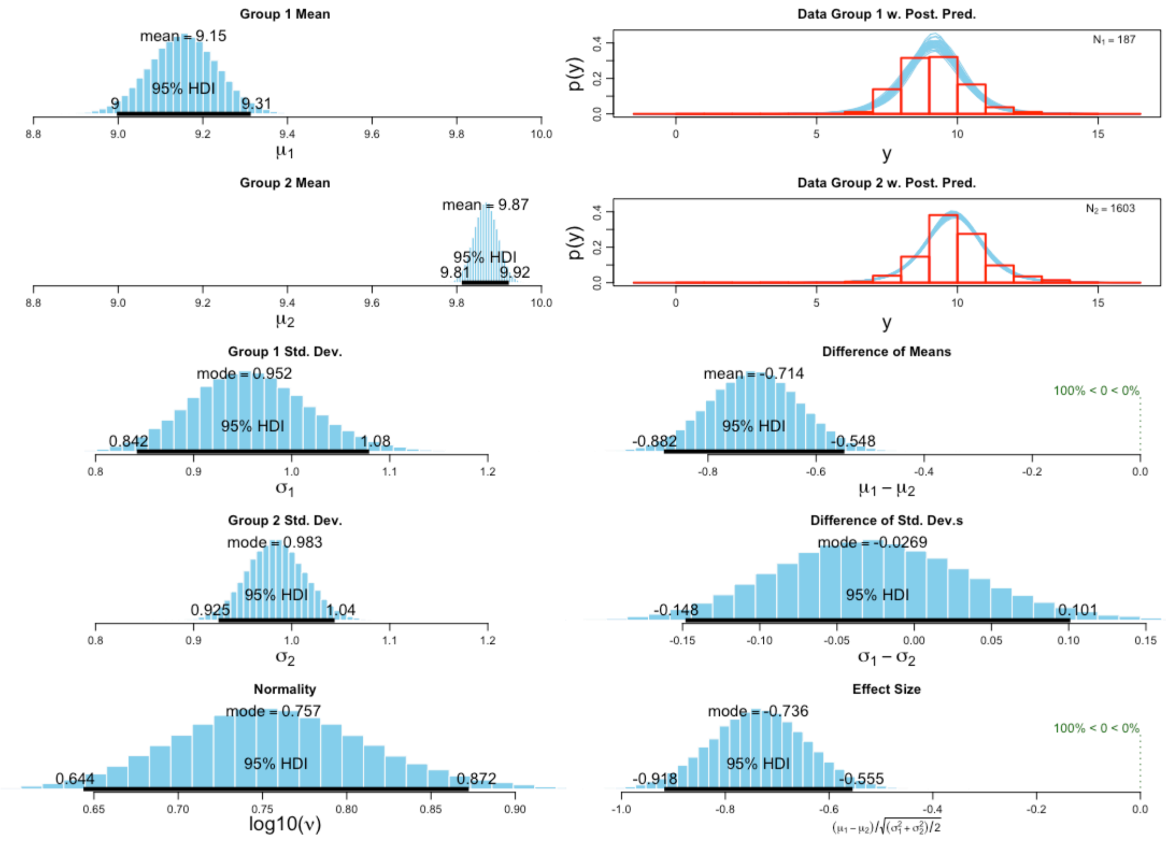
```
      2       1       1       1       1       1       1       1       2       1       1       1       2       2       2       2 
CND_577 CND_578 CND_579 CND_580 CND_581 CND_582 CND_583 CND_584 CND_585 
CND_586 CND_587 CND_588 CND_589 CND_590 CND_591 CND_592 
      2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2 
CND_593 CND_594 CND_595 CND_596 CND_597 CND_598 CND_599 CND_600 CND_601 
CND_602 CND_603 CND_604 CND_605 CND_606 CND_607 CND_608 
      2       2       2       2       2       2       2       1       1       1       1       1       1       1       1       1 
CND_609 CND_610 CND_611 CND_612 CND_613 CND_614 CND_615 CND_616 CND_617 
CND_618 CND_619 CND_620 CND_621 CND_622 CND_623 CND_624 
      1       1       1       1       1       1       1       2       2       2       2       2       2       2       2       2 
CND_625 CND_626 CND_627 CND_628 CND_629 CND_630 CND_631 CND_632 CND_633 
CND_634 CND_635 CND_636 CND_637 CND_638 CND_639 CND_640 
      2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2 
CND_641 CND_642 CND_643 CND_644 CND_645 CND_646 CND_647 CND_648 CND_649 
CND_650 CND_651 CND_652 CND_653 CND_654 CND_655 CND_656 
      2       2       2       2       2       2       2       1       1       1       2       2       2       2       2       2 
CND_657 CND_658 CND_659 CND_660 CND_661 CND_662 CND_663 CND_664 CND_665 
CND_666 CND_667 CND_668 CND_669 CND_670 CND_671 CND_672 
      2       1       2       2       2       2       2       2       2       2       2       1       1       1       1       1 
CND_673 CND_674 CND_675 CND_676 CND_677 CND_678 CND_679 CND_680 CND_681 
CND_682 CND_683 CND_684 CND_685 CND_686 CND_687 CND_688 
      1       2       2       1       1       1       1       1       1       2       2       2       2       2       2       2 
CND_689 CND_690 CND_691 CND_692 CND_693 CND_694 CND_695 CND_696 CND_697 
CND_698 CND_699 CND_700 CND_701 CND_702 CND_703 CND_704 
      2       2       2       2       2       2       1       1       1       1       1       1       1       1       2       1 
CND_705 CND_706 CND_707 CND_708 CND_709 CND_710 CND_711 CND_712 CND_713 
CND_714 CND_715 CND_716 CND_717 CND_718 CND_719 CND_720 
      2       1       2       2       1       1       1       1       1       2       1       1       1       1       2       2 
CND_721 CND_722 CND_723 CND_724 CND_725 CND_726 CND_727 CND_728 CND_729 
CND_730 CND_731 CND_732 CND_733 CND_734 CND_735 CND_736 
      2       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2 
CND_737 CND_738 CND_739 CND_740 CND_741 CND_742 CND_743 CND_744 CND_745 
CND_746 CND_747 CND_748 CND_749 CND_750 CND_751 CND_752 
      2       2       2       1       1       1       1       1       1       1       1       1       2       2       1       1 
CND_753 CND_754 CND_755 CND_756 CND_757 CND_758 CND_759 CND_760 CND_761 
CND_762 CND_763 CND_764 CND_765 CND_766 CND_767 CND_768 
      1       1       1       1       2       2       2       2       2       2       2       2       2       2       2       1 
CND_769 CND_770 CND_771 CND_772 CND_773 CND_774 CND_775 CND_776 CND_777 
CND_778 CND_779 CND_780 CND_781 CND_782 CND_783 CND_784 
      1       2       2       2       2       2       2       2       2       2       2       2       2       2       2       2 
CND_785 CND_786 CND_787 CND_788 CND_789 CND_790 CND_791 CND_792 CND_793 
CND_794 CND_795 CND_796 CND_797 CND_798 CND_799 CND_800 
      2       2       2       2       2       2       1       1       1       1       1       1       1       2       2       2 
CND_801 CND_802 CND_803 CND_804 CND_805 CND_806 CND_807 CND_808 CND_809 
CND_810 CND_811 CND_812 CND_813 CND_814 CND_815 CND_816 
      1       1       1       1       1       2       1       2       2       2       2       2       1       1       1       2 
```

```
CND_817 CND_818 CND_819 CND_820 CND_821 CND_822 CND_823 CND_824 CND_825
CND_826 CND_827 CND_828 CND_829 CND_830 CND_831 CND_832
      1       2       2       2       2       2       2       2       2       2       1       2       1       1       1       1
CND_833 CND_834 CND_835 CND_836 CND_837 CND_838 CND_839 CND_840 CND_841
CND_842 CND_843 CND_844 CND_845 CND_846 CND_847 CND_848
      1       1       2       2       2       1       1       2       2       2       2       1       1       1       1       2
CND_849 CND_850 CND_851 CND_852 CND_853 CND_854 CND_855 CND_856 CND_857
CND_858 CND_859 CND_860 CND_861 CND_862 CND_863 CND_864
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1
CND_865 CND_866 CND_867 CND_868 CND_869 CND_870 CND_871 CND_872 CND_873
CND_874 CND_875 CND_876 CND_877 CND_878 CND_879 CND_880
      1       1       1       1       1       1       1       2       2       2       2       1       3       3       1       1
CND_881 CND_882 CND_883 CND_884 CND_885 CND_886 CND_887 CND_888 CND_889
CND_890 CND_891 CND_892 CND_893 CND_894 CND_895 CND_896
      1       1       1       1       1       1       1       1       1       1       1       1       1       1       3       1
CND_897 CND_898 CND_899 CND_900 CND_901 CND_902 CND_903 CND_904 CND_905
CND_906 CND_907 CND_908 CND_909 CND_910 CND_911 CND_912
      1       1       1       1       1       3       1       1       1       1       1       1       1       1       1       1
CND_913 CND_914 CND_915 CND_916 CND_917 CND_918 CND_919 CND_920 CND_921
CND_922 CND_923 CND_924 CND_925 CND_926 CND_927 CND_928
      1       1       2       2       1       1       1       1       1       1       1       1       1       2       1       1
CND_929 CND_930 CND_931
      1       2       1
```

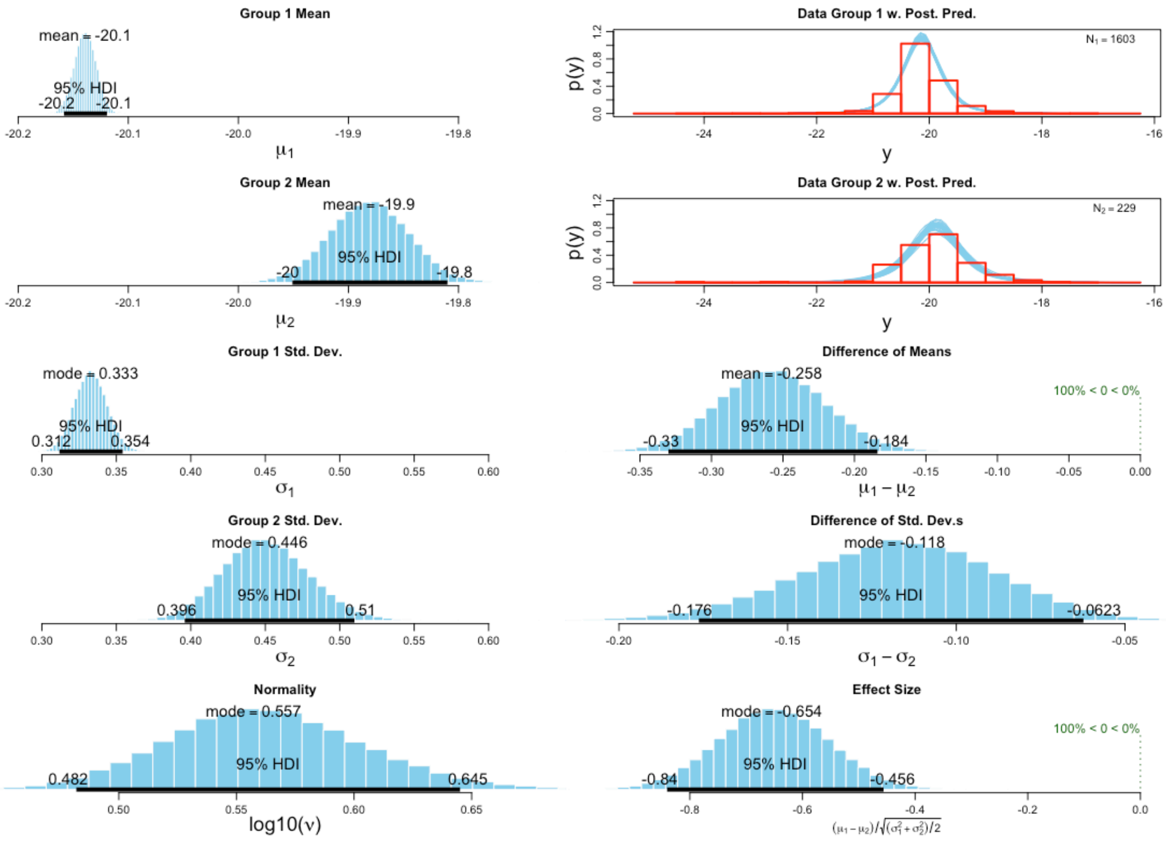
BEST test outputs



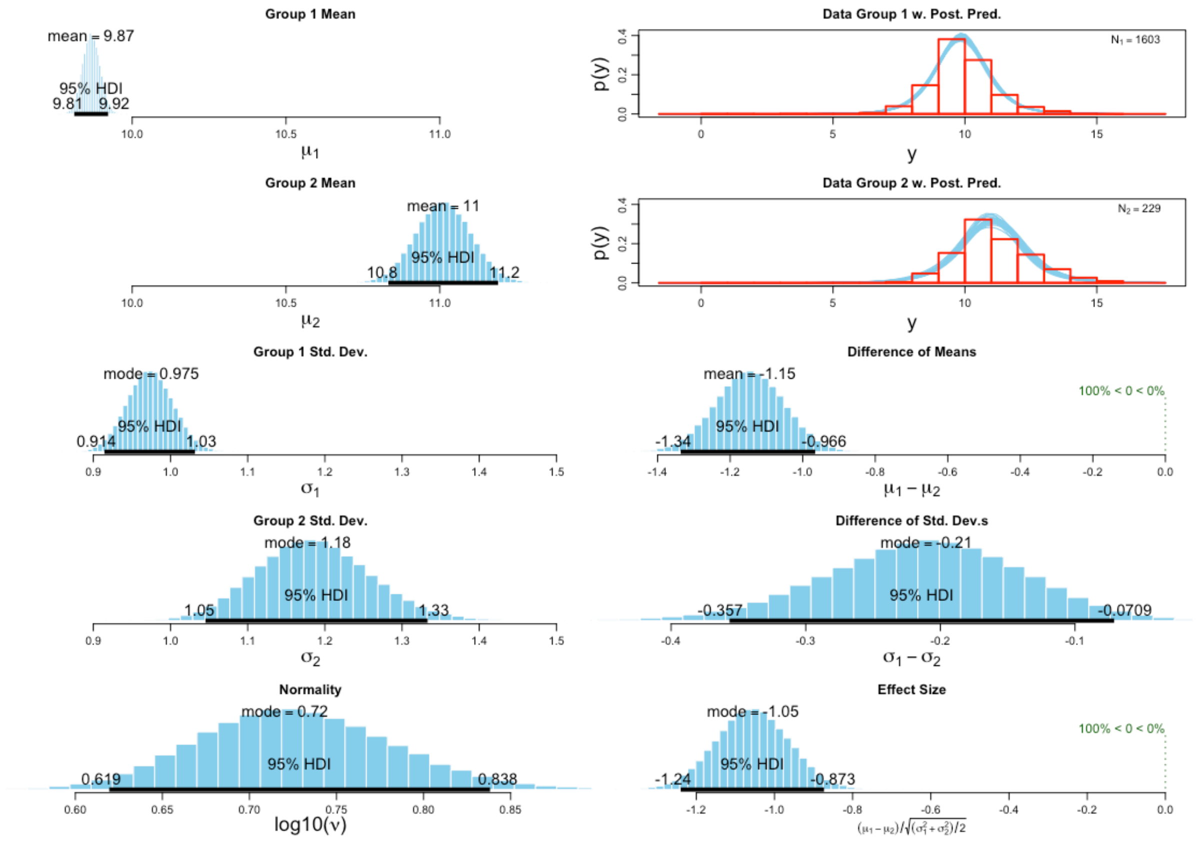
Supplementary Figure 1: BEST test output for c. 200BC – 450 AD (group 1) vs c. 400-790 AD (group 2) bone $\delta^{13}C_{coll}$ values for England (all sexes).



Supplementary Figure 2: BEST test output for c. 200BC – 450 AD (group 1) vs c. 400-790 AD (group 2) bone δ^{15}N$_{coll}$ values for England (all sexes).



Supplementary Figure 3: BEST test output for c. 350-790 AD (group 1) vs c.790-1200 AD (group 2) bone δ^{13}C$_{coll}$ values for England (all sexes).



Supplementary Figure 4: BEST test output for c. 350-790 AD (group 1) vs c.790-1200 AD (group 2) bone $\delta^{15}N_{coll}$ values for England (all sexes).

R code (.R file upload not supported on Editorial assistant, code pasted here as plain text)

```
# 15 March 2021
# REDACTED FOR ANONYMITY
# Script for 'Tackling Early Medieval Dietary Transitions Using a Hierarchical and Multi-
isotope Approach'

#data soon to be published as a data paper in "Ecology" reference redacted for anonymity

#loading packages - some just in case
library(readr)
library(readxl)
library(ggplot2)
library(tidyverse)
library(ggsci)
library(ggExtra)
library(ggpmisc)
library(ggpubr)
library(forcats)
library(viridis)
library(ggridges)
library(dplyr)
library(magrittr)
library(ggdendro)
library(ape)
source("http://addictedtor.free.fr/packages/A2R/lastVersion/R/code.R") # load code of
A2R function
library(factoextra)
library(cluster)
library(NbClust)
library(tidyr)
library(scales)
library(ggthemes)
library(ztable)
library(BEST) #make sure you have the latest JAGS and rjags installed, if you're having
problems, quit R, reinstall JAGS, open R and reinstall both rjags and BEST
library(cowplot)
library(gridExtra)
library(grid)
library(rlang)

#colour blind friendly palettes
# The palette with grey:
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
"#D55E00", "#CC79A7")

# The palette with black:
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
"#D55E00", "#CC79A7")
```

```
#read in data
C_N_Database_bone <- read.csv("~/medieval_palaeoeco_human_bone_CN.csv") #read in
all europe bone data
View(C_N_Database_bone)
C_N_dentine<-read.csv("~/medieval_palaeoeco_human_dentine_CN.csv")
View(C_N_dentine)

Oxygen_Sr_Database <- read.csv("~/medieval_palaeoeco_human_apatite.csv")
View(Oxygen_Sr_Database)
summary(Oxygen_Sr_Database)
#remove bone data
OSr_teeth<-subset(Oxygen_Sr_Database, `Bone/Tooth`!="Bone")
head(OSr_teeth)
summary(OSr_teeth)
#Ireland still seems to be skewing the data - to do with Ryan et al. 2018 - REMOVE this
data
OSr_teeth<-subset(OSr_teeth, Reference!='Ryan SE, Reynard LM, Crowley QG, Snoeck C,
Tuross N (2018). "Early Medieval reliance on the land and the local: An Integrated multi-
isotope study (87Sr/86Sr, δ18O, δ13C, δ15N) of diet and mirgration in Co. Meath, Ireland."
Journal of Archaeological Science 98: 59-71.')
summary(OSr_teeth)



#subset for only English material
England_CN_bone<- subset(C_N_Database_bone, Country=="England")
View(England_CN_bone)
summary(England_CN_bone)
England_CN_dentine<-subset(C_N_dentine, Country=="England")
Oxy_England<- subset(OSr_teeth, Country=="England")
summary(Oxy_England)

#defining european regions
#bone
C_N_Database_bone$EuRegion<-C_N_Database_bone$County
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Split-Dalmatia', 'Croatia'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Zadar', 'Croatia'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Vis', 'Croatia'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Vukovar-Srijem', 'Croatia'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Šibenik-Knin', 'Croatia'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Roskilde', 'Skagerrak-Kattegat-Jutland Basin'))
```

```
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Zealand', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Funen', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Ålborg', 'Skagerrak-Kattegat-Jutland Basin'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Saare', 'Baltic'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Turku and Pori', Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Nord-Trøndelag', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Nordland', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Troms', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Rogaland', 'Atlantic & Arctic Norway'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Sogn og Fjordane', 'Atlantic & Arctic Norway'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Vest-Agder', 'Skagerrak-Kattegat-Jutland Basin'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Vestfold', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Hordaland', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Hedmark', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Trøndelag', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Telemark', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Akershus', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Oppland', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Sør Trøndelag', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Uppland', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Uppsala', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Kalmar', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Stockholm', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Gotland', 'Baltic'))
```

```
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Öland', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Gävleborg', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Adelsö', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Leningrad Oblast', 'Baltic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Nord Trøndelag', 'Atlantic & Arctic Norway'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Aarhus', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Hjørring', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Skive', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Oslo', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Isle of Man', 'Irish Sea'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Pembrokeshire', 'Irish Sea'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Vale of Glamorgan', 'Irish Sea'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Meath', 'Irish Sea'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Dublin', 'Irish Sea'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Ireland', 'Irish Sea'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Outer Hebrides', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Scotland', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Rousay', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Highlands', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Orkney', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Mainland', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Highlands, Scotland', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Sanday', 'Scotland and Scottish Isles'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Dumfries and Galloway', 'Scotland and Scottish Isles'))
```

```
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Bedfordshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Cambridgeshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Dorset', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Derbyshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'East Sussex', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Sussex', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Kent', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Hampshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Wiltshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Warwickshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Suffolk', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Oxfordshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Lincolnshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Tyne & Wear', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Northumberland', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Yorkshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Rutland', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Nottinghamshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Hertfordshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Buckinghamshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'North Lincolnshire', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Somerset', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Norfolk', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Gloucestershire', 'England'))
```

```
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Surrey', 'England'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Northamptonshire', 'England'))
#C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Kujalleq', 'North Atlantic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Mosfell', 'North Atlantic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Lake Myvatn', 'North Atlantic'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Piemont', 'Po Valley'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Friuli-Venezia Giulia', 'Po Valley'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Piedmont', 'Po Valley'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Ibiza', 'Balearic & Tyrrhenian Seas'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Rome', 'Balearic & Tyrrhenian Seas'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Barcelona', 'Balearic & Tyrrhenian Seas'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Valencia', 'Balearic & Tyrrhenian Seas'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Languedoc', 'Balearic & Tyrrhenian Seas'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Normandy', 'Normandy/Neustria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Flanders', 'Normandy/Neustria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Wallonia', 'Normandy/Neustria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Aragon', 'Inland & Western Iberia'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Pontevedra', 'Inland & Western Iberia'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Madrid', 'Inland & Western Iberia'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Setubal', 'Inland & Western Iberia'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Alentejo', 'Inland & Western Iberia'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Lower Saxony', 'Frisia & Saxony'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Friesland', 'Frisia & Saxony'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Saxony-Anhalt', 'Frisia & Saxony'))
```

```
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Rhineland', 'Austrasia & Burgundy'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Grand Est', 'Austrasia & Burgundy'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Rhein-Kreis Neuss, Nordrhine-Westphalia', 'Austrasia & Burgundy'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Gem. Bedburg-Königshoven, Rhein-Erft-Kreis, Northrhine-Westphalia',
'Austrasia & Burgundy'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Baden-Württemberg', 'Austrasia & Burgundy'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Bavaria', 'Austro-Hungary & Bavaria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Lower Austria', 'Austro-Hungary & Bavaria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Northern Hungary', 'Austro-Hungary & Bavaria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Tyrol', 'Austro-Hungary & Bavaria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Vienna', 'Austro-Hungary & Bavaria'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Crete', 'Greece'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Peloponnese', 'Greece'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(EuRegion = replace(EuRegion,
EuRegion == 'Western Macedonia', 'Greece'))

C_N_Database_bone$EuRegion = factor(C_N_Database_bone$EuRegion,
                              levels=c("North Atlantic", "Atlantic & Arctic Norway","Skagerrak-
Kattegat-Jutland Basin","Baltic", "Scotland and Scottish Isles", "Irish Sea", "England",
"Frisia & Saxony", "Normandy/Neustria", "Austrasia & Burgundy","Austro-Hungary &
Bavaria", "Po Valley", "Croatia", "Balearic & Tyrrhenian Seas","Inland & Western Iberia",
"Greece"),ordered=TRUE)

#dentine
C_N_dentine$EuRegion<-C_N_dentine$County
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Bedfordshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Lincolnshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Northumberland', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Nottinghamshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Kent', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Wiltshire', 'England'))
```

```
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Cambridgeshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Rutland', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Hertfordshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Yorkshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Northamptonshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Dorset', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Warwickshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Suffolk', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Lincolnshire', 'England'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Vukovar-Srijem', 'Croatia'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Funen', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Troms', 'Atlantic & Arctic Norway'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Nordland', 'Atlantic & Arctic Norway'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Oslo', 'Skagerrak-Kattegat-Jutland Basin'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Öland', 'Baltic'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Piedmont', 'Po Valley'))
C_N_dentine <- C_N_dentine %>% mutate(EuRegion = replace(EuRegion, EuRegion ==
'Highlands, Scotland', 'Scotland and Scottish Isles'))

C_N_dentine$EuRegion = factor(C_N_dentine$EuRegion,
                    levels=c("Atlantic & Arctic Norway","Skagerrak-Kattegat-Jutland Basin",
"Baltic","Scotland and Scottish Isles", "England", "Po Valley", "Croatia"),ordered=TRUE)




#hierarchical clustering and dendrograms
#dendrograms as an alternative to k means clustering which doesn't require you to select
the number of cluster

#prepping the data for hierarchical clustering
EMEUboneclean <- C_N_Database_bone
```

```
class(as.data.frame(EMEUboneclean))
head(EMEUboneclean)
EMEUboneclean <- data.frame(EMEUboneclean)
rownames(EMEUboneclean) <- EMEUboneclean[,1]
EMEUboneclean <- EMEUboneclean[,-1]
head(EMEUboneclean)
show(EMEUboneclean)
EMEUboneclean<-EMEUboneclean[c(32,33)]
head(EMEUboneclean)
EMEUboneclean <- scale(EMEUboneclean)
EMEUboneclean <- na.omit(EMEUboneclean)
head(EMEUboneclean)
#hierarchical clustering
ddEMEU_CN_bone<-dist(EMEUboneclean, method = "euclidean")
hcEMEU_CN_bone<-hclust(ddEMEU_CN_bone, method = "ward.D2")
ggdendrogram(hcEMEU_CN_bone)
ggdendrogram(hcEMEU_CN_bone, rotate = TRUE,)
plot(as.phylo(hcEMEU_CN_bone), type = "fan")
set.seed(123)
fviz_nbclust(EMEUboneclean, hcut, method = "silhouette")
set.seed(123)
gap_stat <- clusGap(EMEUboneclean, FUN = hcut, nstart = 25, K.max = 10, B = 50)
print(gap_stat, method = "firstmax")
print(gap_stat, method = "globalmax")
fviz_gap_stat(gap_stat)
set.seed(123)
fviz_nbclust(EMEUboneclean, hcut, method = "wss")
NbClust(data = EMEUboneclean, diss = NULL, distance = "euclidean", min.nc = 2, max.nc =
15, method = "ward.D2")
A2Rplot(hcEMEU_CN_bone, k = 2, boxes = FALSE, col.up = "gray50", col.down = cbbPalette)
A2Rplot(hcEMEU_CN_bone, k = 3, boxes = FALSE, col.up = "gray50", col.down = cbbPalette)
A2Rplot(hcEMEU_CN_bone, k = 6, boxes = FALSE, col.up = "gray50", col.down = cbbPalette)

#import! manually entered scatterplot groups from full size print out of dendrogram
above
CNB_hclusters <- read_excel("~/CNB_hclusters.xlsx")
#View(CNB_hclusters)
C_N_Database_bone$`CNB_hcluster`<-CNB_hclusters$CNB_EMEU_hclust_group

#with 3 clusters from Nbclust, need to combine sub-clusters
C_N_Database_bone$`CNB_hcluster3`<-C_N_Database_bone$`CNB_hcluster`
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '1.1.1', '1'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '1.1.2', '1'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '1.2.1', '1'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '1.2.2', '1'))
```

```
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '2.1.1', '2.1'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '2.1.2', '2.1'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '2.2.1', '2.2'))
C_N_Database_bone <- C_N_Database_bone %>% mutate(CNB_hcluster3 =
replace(CNB_hcluster3, CNB_hcluster3 == '2.2.2', '2.2'))


#scatterplot for EMEU C&N dendrogram
tiff("Dropbox/Publications/EJA/Leggett_Fig3.tiff", units="in", width=18.04, height=10.76,
res=300)
ggplot(C_N_Database_bone,aes(d13C, d15N, color=`CNB_hcluster3`))+ #3groups
  theme_bw()+
  geom_point(size=3,shape=16)+
  scale_colour_manual(values=cbbPalette, name="Cluster Number")+
  ylab(expression(paste(delta^{15},N["bone (AIR)"], "
(\u2030)")))+xlab(expression(paste(delta^{13},C["bone (PDB)"], " (\u2030)")))+
  scale_x_continuous(limits=c(-24.5,-12.5),breaks=seq(-24.5,-
12.5,1))+scale_y_continuous(limits=c(0,20),breaks=seq(0,20,1))+
  theme(axis.text=element_text(size=18),axis.title=element_text(size=22),legend.title
=element_text(size=22), legend.text = element_text(size=18))
dev.off()

#stacked bar plots
#stacked barchart with country and cluster
#counts for country categories

CNBcleanplus<- C_N_Database_bone
class(as.data.frame(CNBcleanplus))
head(CNBcleanplus)
CNBcleanplus <- data.frame(CNBcleanplus)
rownames(CNBcleanplus) <-CNBcleanplus[,1]
CNBcleanplus <- CNBcleanplus[,-1]
head(CNBcleanplus)
View(CNBcleanplus)
CNBcleanplus<-CNBcleanplus[c(2,3,37,40)] #country, country code, region and cluster
group
head(CNBcleanplus)
CNBcleanplus <- na.omit(CNBcleanplus)
head(CNBcleanplus)

CNBdfr <- CNBcleanplus %>%
  mutate(Country = as.factor(Country)     # categorical values to factor
        , `CNB_hcluster2` = as.ordered(`CNB_hcluster3`))# character to ordered factor (like a
grade)
```

```
#fct_reorder(carbcleanplus$Country, carbcleanplus$`Country Code`)

#dfr <- na.omit(dfr)
CNBdfr_prop <- CNBdfr %>%
  count(Country, `CNB_hcluster3`) %>%          # group_by() & summarise(n = n()) are
implicit
  mutate(prop = prop.table(n))    # prop = n/sum(n) works too
as.data.frame(CNBdfr_prop)

CNBdfr_prop2 <- CNBdfr %>%
  count(`CNB_hcluster3`, Country) %>%          # group_by() & summarise(n = n()) are
implicit
  mutate(prop = prop.table(n))    # prop = n/sum(n) works too
as.data.frame(CNBdfr_prop2)

ggplot(CNBdfr_prop, aes(CNBdfr_prop$`CNB_hcluster3`,CNBdfr_prop$prop,)) +
  geom_bar(colour = "black", aes(fill = Country, weight=`CNB_hcluster3`, outline.colour =
"black"), position = "fill", stat="identity") +
  scale_fill_viridis(discrete = TRUE, name = "Country")+
  xlab(expression(paste("Cluster")))+ylab(expression(paste("Proportion")))+
  theme_bw()

ggplot(CNBdfr_prop, aes(Country,CNBdfr_prop$prop,)) +
  geom_bar(colour = "black", aes(fill = CNBdfr_prop$`CNB_hcluster3`, weight=Country,
outline.colour = "black"), position = "fill", stat="identity") +
  scale_fill_manual(values=cbbPalette, name="Cluster")+
  xlab(expression(paste("Country")))+ylab(expression(paste("Proportion")))+
  theme_bw()


CNBdfr2 <- CNBcleanplus %>%
  mutate(EuRegion = as.factor(EuRegion), `CNB_hcluster3` = as.ordered(`CNB_hcluster3`))

CNBdfr2_prop <- CNBdfr2 %>%
  count(EuRegion, `CNB_hcluster3`) %>%          # group_by() & summarise(n = n()) are
implicit
  mutate(prop = prop.table(n))    # prop = n/sum(n) works too
as.data.frame(CNBdfr2_prop)

ggplot(CNBdfr2_prop, aes(CNBdfr2_prop$`CNB_hcluster3`,CNBdfr2_prop$prop,)) +
  theme_bw()+
  geom_bar(colour = "black", aes(fill = EuRegion, weight=`CNB_hcluster3`, outline.colour =
"black"), position = "fill", stat="identity") +
  scale_fill_viridis(discrete = TRUE, name = "Region")+
  xlab(expression(paste("Cluster")))+ylab(expression(paste("Proportion")))+
  theme(axis.text=element_text(size=16),axis.title=element_text(size=22), legend.text
=element_text(size=18),legend.title = element_text(size = 22))

ggplot(CNBdfr2_prop, aes(EuRegion,CNBdfr2_prop$prop,)) +
  theme_bw()+
  geom_bar(colour = "black", aes(fill = CNBdfr2_prop$`CNB_hcluster3`, weight=EuRegion,
outline.colour = "black"), position = "fill", stat="identity") +
  scale_fill_manual(values=cbbPalette, name="Cluster")+
  xlab(expression(paste("Region")))+ylab(expression(paste("Proportion")))+
  theme(axis.text=element_text(size=16),axis.text.x = element_text(angle = 45, hjust =
1),axis.title=element_text(size=22), legend.text =element_text(size=18),legend.title =
element_text(size = 22))


#dentine hierarchical clustering
#prepping the data for hierarchical clustering
EMEUdentineclean <- C_N_dentine
class(as.data.frame(EMEUdentineclean))
head(EMEUdentineclean)
EMEUdentineclean <- data.frame(EMEUdentineclean)
rownames(EMEUdentineclean) <- EMEUdentineclean[,1]
EMEUdentineclean <- EMEUdentineclean[,-1]
head(EMEUdentineclean)
View(EMEUdentineclean)
EMEUdentineclean<-EMEUdentineclean[c(33,34)] #choosing just d13C and d15N
head(EMEUdentineclean)
EMEUdentineclean <- scale(EMEUdentineclean)
EMEUdentineclean <- na.omit(EMEUdentineclean)
head(EMEUdentineclean)
#hierarchical clustering
ddEMEU_CN_dentine<-dist(EMEUdentineclean, method = "euclidean")
hcEMEU_CN_dentine<-hclust(ddEMEU_CN_dentine, method = "ward.D2")
ggdendrogram(hcEMEU_CN_dentine)
ggdendrogram(hcEMEU_CN_dentine, rotate = TRUE,)
plot(as.phylo(hcEMEU_CN_dentine), type = "fan")
set.seed(123)
fviz_nbclust(EMEUdentineclean, hcut, method = "silhouette")
set.seed(123)
gap_stat <- clusGap(EMEUdentineclean, FUN = hcut, nstart = 25, K.max = 10, B = 50)
print(gap_stat, method = "firstmax")
print(gap_stat, method = "globalmax")
fviz_gap_stat(gap_stat)
set.seed(123)
fviz_nbclust(EMEUdentineclean, hcut, method = "wss")
NbClust(data = EMEUdentineclean, diss = NULL, distance = "euclidean", min.nc = 2, max.nc
= 15, method = "ward.D2")
A2Rplot(hcEMEU_CN_dentine, k = 7, boxes = FALSE, col.up = "gray50", col.down =
c("black", "#E69F00","#0072B2","#009E73","#CC79A7","#56B4E9","#D55E00"))
A2Rplot(hcEMEU_CN_dentine, k = 8, boxes = FALSE, col.up = "gray50", col.down =
cbbPalette)
A2Rplot(hcEMEU_CN_dentine, k = 6, boxes = FALSE, col.up = "gray50", col.down =
cbbPalette)
```

```
A2Rplot(hcEMEU_CN_dentine, k = 5, boxes = FALSE, col.up = "gray50", col.down =
cbbPalette)
A2Rplot(hcEMEU_CN_dentine, k = 3, boxes = FALSE, col.up = "gray50", col.down =
cbbPalette)

#import manually entered scatterplot groups from above as per with bone
CND_hclusters <- read_excel("~/CND_hclusters.xlsx")
#View(CND_hclusters)
C_N_dentine$`CND_hcluster`<-CND_hclusters$CND_hcluster_group5
C_N_dentine <- C_N_dentine %>% mutate(CND_hcluster = replace(CND_hcluster,
CND_hcluster == '1.1000000000000001', '1.1'))
C_N_dentine <- C_N_dentine %>% mutate(CND_hcluster = replace(CND_hcluster,
CND_hcluster == '1.1', '1'))
C_N_dentine <- C_N_dentine %>% mutate(CND_hcluster = replace(CND_hcluster,
CND_hcluster == '1.2', '1'))
C_N_dentine <- C_N_dentine %>% mutate(CND_hcluster = replace(CND_hcluster,
CND_hcluster == '2.2.1', '2.2'))
C_N_dentine <- C_N_dentine %>% mutate(CND_hcluster = replace(CND_hcluster,
CND_hcluster == '2.2.2', '2.2'))

#scatterplot for EMEU C&N dendrogram dentine
tiff("Dropbox/Publications/EJA/Leggett_Fig7.tiff", units="in", width=18.04, height=10.76,
res=300)
ggplot(C_N_dentine,aes(d13C, d15N, color=`CND_hcluster`))+ #3 clusters
  theme_bw()+
  geom_point(size=3,shape=16)+
  scale_colour_manual(values=cbbPalette, name="Cluster Number")+
  ylab(expression(paste(delta^{15},N["dentine (AIR)"], "
(\u2030)")))+xlab(expression(paste(delta^{13},C["dentine (PDB)"], " (\u2030)")))+
  scale_x_continuous(limits=c(-24.5,-12.5),breaks=seq(-24.5,-
12.5,1))+scale_y_continuous(limits=c(0,20),breaks=seq(0,20,1))+
  theme(axis.text=element_text(size=18),axis.title=element_text(size=22),legend.title
=element_text(size=22), legend.text = element_text(size=18))
dev.off()

ggplot(C_N_dentine,aes(d13C, d15N, color=`CND_hcluster`))+ #3 clusters
  theme_bw()+
  geom_point(size=3,shape=16)+
  scale_colour_manual(values=cbbPalette, name="Cluster Number")+
  ylab(expression(paste(delta^{15},N["dentine (AIR)"], "
(\u2030)")))+xlab(expression(paste(delta^{13},C["dentine (PDB)"], " (\u2030)")))+
  scale_x_continuous(limits=c(-24.5,-12.5),breaks=seq(-24.5,-
12.5,1))+scale_y_continuous(limits=c(0,20),breaks=seq(0,20,1))+
  theme(axis.text=element_text(size=18),axis.title=element_text(size=22),legend.title
=element_text(size=22), legend.text = element_text(size=18))
ggsave("Dropbox/Publications/EJA/Leggett_Fig7_v2.tiff", width=18.04, height=10.76, dpi
= 320) #dpi still weirdly 72??? whyyyyy???

bitmap("Dropbox/Publications/EJA/Leggett_Fig7_v3.tiff", height = 10.76, width = 18.04,
    units = 'cm', type="tiff", res=300)
ggplot(C_N_dentine,aes(d13C, d15N, color=`CND_hcluster`))+ #3 clusters
  theme_bw()+
  geom_point(size=3,shape=16)+
  scale_colour_manual(values=cbbPalette, name="Cluster Number")+
  ylab(expression(paste(delta^{15},N["dentine (AIR)"], "
(\u2030)")))+xlab(expression(paste(delta^{13},C["dentine (PDB)"], " (\u2030)")))+
  scale_x_continuous(limits=c(-24.5,-12.5),breaks=seq(-24.5,-
12.5,1))+scale_y_continuous(limits=c(0,20),breaks=seq(0,20,1))+
  theme(axis.text=element_text(size=18),axis.title=element_text(size=22),legend.title
=element_text(size=22), legend.text = element_text(size=18))
dev.off()

postscript("Dropbox/Publications/EJA/Leggett_Fig7.eps", height = 10.76, width = 18.04,
        horizontal = FALSE, onefile = FALSE, paper = "special",
        colormodel = "cmyk")
ggplot(C_N_dentine,aes(d13C, d15N, color=`CND_hcluster`))+ #3 clusters
  theme_bw()+
  geom_point(size=3,shape=16)+
  scale_colour_manual(values=cbbPalette, name="Cluster Number")+
  ylab(expression(paste(delta^{15},N["dentine (AIR)"], "
(\u2030)")))+xlab(expression(paste(delta^{13},C["dentine (PDB)"], " (\u2030)")))+
  scale_x_continuous(limits=c(-24.5,-12.5),breaks=seq(-24.5,-
12.5,1))+scale_y_continuous(limits=c(0,20),breaks=seq(0,20,1))+
  theme(axis.text=element_text(size=18),axis.title=element_text(size=22),legend.title
=element_text(size=22), legend.text = element_text(size=18))
dev.off() #works but doesn't include the percent per mille sign...


#stacked bar plots
#stacked barchart with country and cluster
#counts for country categories
CNDcleanplus<- C_N_dentine
class(as.data.frame(CNDcleanplus))
head(CNDcleanplus)
CNDcleanplus <- data.frame(CNDcleanplus)
rownames(CNDcleanplus) <-CNDcleanplus[,1]
CNDcleanplus <- CNDcleanplus[,-1]
head(CNDcleanplus)
View(CNDcleanplus)
CNDcleanplus<-CNDcleanplus[c(2,3,36,37)] #country, country code, region and cluster
group
head(CNDcleanplus)
#CNDcleanplus <- na.omit(CNDcleanplus)
#head(CNDcleanplus)

CNDdfr <- CNDcleanplus %>%
  mutate(Country = as.factor(Country)     # categorical values to factor
      , `CND_hcluster` = as.ordered(`CND_hcluster`))# character to ordered factor (like a
grade)


#fct_reorder(carbcleanplus$Country, carbcleanplus$`Country Code`)

#dfr <- na.omit(dfr)
CNDdfr_prop <- CNDdfr %>%
  count(Country, `CND_hcluster`) %>%          # group_by() & summarise(n = n()) are
implicit
  mutate(prop = prop.table(n))    # prop = n/sum(n) works too
as.data.frame(CNDdfr_prop)

CNDdfr_prop2 <- CNDdfr %>%
  count(`CND_hcluster`, Country) %>%          # group_by() & summarise(n = n()) are
implicit
  mutate(prop = prop.table(n))    # prop = n/sum(n) works too
as.data.frame(CNDdfr_prop2)

ggplot(CNDdfr_prop, aes(CNDdfr_prop$`CND_hcluster`,CNDdfr_prop$prop,)) +
  geom_bar(colour = "black", aes(fill = Country, weight=`CND_hcluster`, outline.colour =
"black"), position = "fill", stat="identity") +
  scale_fill_viridis(discrete = TRUE, name = "Country")+
  xlab(expression(paste("Cluster")))+ylab(expression(paste("Proportion")))+
  theme_bw()

ggplot(CNDdfr_prop, aes(Country,CNDdfr_prop$prop,)) +
  geom_bar(colour = "black", aes(fill = CNDdfr_prop$`CND_hcluster`, weight=Country,
outline.colour = "black"), position = "fill", stat="identity") +
  scale_fill_manual(values=cbbPalette, name="Cluster")+
  xlab(expression(paste("Country")))+ylab(expression(paste("Proportion")))+
  theme_bw()


CNDdfr2 <- CNDcleanplus %>%
  mutate(EuRegion = as.factor(EuRegion), `CND_hcluster` = as.ordered(`CND_hcluster`))

CNDdfr2_prop <- CNDdfr2 %>%
  count(EuRegion, `CND_hcluster`) %>%          # group_by() & summarise(n = n()) are
implicit
  mutate(prop = prop.table(n))    # prop = n/sum(n) works too
as.data.frame(CNDdfr2_prop)

ggplot(CNDdfr2_prop, aes(CNDdfr2_prop$`CND_hcluster`,CNDdfr2_prop$prop,)) +
  theme_bw()+
  geom_bar(colour = "black", aes(fill = EuRegion, weight=`CND_hcluster`, outline.colour =
"black"), position = "fill", stat="identity") +
  scale_fill_viridis(discrete = TRUE, name = "Region")+
  xlab(expression(paste("Cluster")))+ylab(expression(paste("Proportion")))+
  theme(axis.text=element_text(size=22),axis.title=element_text(size=22), legend.text
=element_text(size=18),legend.title = element_text(size = 22))


ggplot(CNDdfr2_prop, aes(EuRegion,CNDdfr2_prop$prop,)) +
  theme_bw()+
  geom_bar(colour = "black", aes(fill = CNDdfr2_prop$`CND_hcluster`, weight=EuRegion,
outline.colour = "black"), position = "fill", stat="identity") +
  scale_fill_manual(values=cbbPalette, name="Cluster")+
  xlab(expression(paste("Region")))+ylab(expression(paste("Proportion")))+
  theme(axis.text=element_text(size=16),axis.text.x = element_text(angle = 45, hjust =
1),axis.title=element_text(size=22), legend.text =element_text(size=18),legend.title =
element_text(size = 22))

#England diet through time
#simplifying the date categories
England_CN_bone$SimpleDate<-England_CN_bone$`Date Category`
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'A-C', 'A-D'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'A/B', 'A-D'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'A-D', 'A-E'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'B-D', 'B-G'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'B-E', 'B-G'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'B-F', 'B-G'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'B/C', 'B-G'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'C-F', 'C/D'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'C-E', 'C-H'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'C-G', 'C-H'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'F-H', 'F-I'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'E-G', 'E-H'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'D-G', 'D-H'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'D/E', 'D-F'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'E', 'E/F'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'F', 'E/F'))
```

```
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'E-H', 'E/F'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'F-I', 'E-I'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'G', 'G-I'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'G/H', 'G-I'))
England_CN_bone <- England_CN_bone %>% mutate(SimpleDate = replace(SimpleDate,
SimpleDate == 'I', 'G-I'))

England_CN_bone$`SimpleDate` = factor(England_CN_bone$`SimpleDate`,
                        levels=c("A", "A-E","B","B-G","C","C/D","C-H","D","D-F","D-H","E/F","E-
I","F/G","G-I"),ordered=TRUE)


#combine date categories further similar to carbonate with pre-migration period,
migration period to "viking" and viking to Norman
#now for date categories super simple - 200BC-450 AD, ~350-790AD, ~790AD-1066+
England_CN_bone$SimpleDate<-as.character(England_CN_bone$SimpleDate)
England_CN_bone$PeriodBroad<-England_CN_bone$`SimpleDate`
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'A', '200BC-450AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'A-E', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B-G', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'C', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'C/D', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'C-H', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D-F', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'E/F', 'c.350AD-790AD'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D-H', 'c.790AD-1066AD+'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'E-I', 'c.790AD-1066AD+'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'F/G', 'c.790AD-1066AD+'))
England_CN_bone <- England_CN_bone %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'G-I', 'c.790AD-1066AD+'))
```

```
England_CN_bone$`PeriodBroad` = factor(England_CN_bone$`PeriodBroad`,
                               levels=c("200BC-450AD", "c.350AD-790AD","c.790AD-
1066AD+"),ordered=TRUE)


Eng_Bone_d13C_broad<-ggplot(data=England_CN_bone,
aes(x=England_CN_bone$PeriodBroad, y=England_CN_bone$d13C, fill=PeriodBroad))+
  theme_bw()+
  geom_violin(trim=FALSE, show.legend = TRUE)+
  scale_fill_viridis(discrete = TRUE)+
  #scale_color_manual(values=qualcolourPalette)+
  ylab(expression(paste(delta^{13},C["bone (PDB)"], "
(\u2030)")))+xlab(expression(paste("Date Category")))+
  scale_y_continuous(limits=c(-25,-15),breaks=seq(-25,-15,2))+
  theme(legend.position = "none", axis.text = element_text(size = 18), axis.title =
element_text(size=20))
Eng_Bone_d13C_broad

Eng_Bone_d15N_broad<-ggplot(data=England_CN_bone,
aes(x=England_CN_bone$PeriodBroad, y=England_CN_bone$d15N, fill=PeriodBroad))+
  theme_bw()+
  geom_violin(trim=FALSE, show.legend = TRUE)+
  scale_fill_viridis(discrete = TRUE)+
  #scale_color_manual(values=qualcolourPalette)+
  ylab(expression(paste(delta^{15},N["bone (AIR)"], "
(\u2030)")))+xlab(expression(paste("Date Category")))+
  scale_y_continuous(limits=c(0,18),breaks=seq(0,18,2))+
  theme(legend.position = "none", axis.text = element_text(size = 18), axis.title =
element_text(size=20))
Eng_Bone_d15N_broad

England_Bone_Broad_Bag<- ggplot(England_CN_bone, aes(d13C, d15N, colour =
PeriodBroad, fill = PeriodBroad)) +
  theme_bw()+
  geom_bag()+
  ylab(expression(paste(delta^{15},"N (\u2030)")))+xlab(expression(paste(delta^{13},"C
(\u2030)")))+
  scale_colour_viridis(discrete = TRUE, name="Broad Period")+
  scale_fill_viridis(discrete = TRUE,name="Broad Period")+
  theme(axis.text=element_text(size=20),axis.title=element_text(size=22),legend.title
=element_text(size=24), legend.text =element_text(size=24), legend.position = "bottom")
England_Bone_Broad_Bag

#dentine
England_CN_dentine$PeriodBroad<-England_CN_dentine$`Date Category`
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'A-C', '200BC-450AD'))
```

```
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'A-D', '200BC-450AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'A/B', '200BC-450AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'B', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'B-D', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'B-E', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'B-F', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'B-G', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'B/C', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'C', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'C-F', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'C-G', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'C/D', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'D', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'D/E', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'E', 'c.450AD-790AD'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'D-F', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'D-G', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'D-H', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'E/F', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'F', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'F-H', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'F/G', 'c.790AD-1066AD+'))
England_CN_dentine <- England_CN_dentine %>% mutate(PeriodBroad =
replace(PeriodBroad, PeriodBroad == 'G', 'c.790AD-1066AD+'))


England_CN_dentine$`PeriodBroad` = factor(England_CN_dentine$`PeriodBroad`,
                    levels=c("200BC-450AD", "c.450AD-790AD","c.790AD-
1066AD+"),ordered=TRUE)
#enamel carbonate
Oxy_England$SimpleDate<-Oxy_England$`Date Category`
summary(Oxy_England)
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'A-C', 'A-D'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'A/B', 'A-D'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'B-D', 'B-G'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'B-E', 'B-G'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'B-F', 'B-G'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'B/C', 'B-G'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'C-F', 'C/D'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'F-H', 'F-I'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'E-G', 'E-H'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'D-G', 'D-H'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'E', 'E/F'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'F', 'E/F'))
Oxy_England <- Oxy_England %>% mutate(SimpleDate = replace(SimpleDate, SimpleDate
== 'E-H', 'E/F'))

Oxy_England$`SimpleDate` = factor(Oxy_England$`SimpleDate`,
                levels=c("A", "A-D","B","B-G", "C", "C/D", "D", "D/E", "D-F", "D-H", "E/F",
"F/G","F-I"),ordered=TRUE)
#now for date categories super simple - 200BC-450 AD, ~200-790AD, ~790AD-1066+
#A->200BC-450 AD
#A/B, A-C, A-D, B, B-D, B-E, B/C, C, C/D, D, D/E, E->~200AD-790 AD
Oxy_England$PeriodBroad<-Oxy_England$`Date Category`
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'A', '200BC-450AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'A/B', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'A-C', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'A-D', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B', 'c.350AD-790AD'))
```

```
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B-D', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B-E', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B/C', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'C', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'C/D', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D/E', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'E', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'C-F', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B-F', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'B-G', 'c.350AD-790AD'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'F', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'F-H', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'E/F', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'F-I', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'F/G', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'E-G', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'E-H', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D-G', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D-H', 'c.790AD-1066AD+'))
Oxy_England <- Oxy_England %>% mutate(PeriodBroad = replace(PeriodBroad,
PeriodBroad == 'D-F', 'c.790AD-1066AD+'))



Oxy_England$`PeriodBroad` = factor(Oxy_England$`PeriodBroad`,
                        levels=c("200BC-450AD", "c.350AD-790AD","c.790AD-
1066AD+"),ordered=TRUE)
```

```
#carbonate and collagen offsets
## big delta offset scatters
matched_bone_dent_enamel_England<-
read.csv("~/matched_bone_dent_enamel_england.csv")

matched_bone_dent_enamel_England$PeriodBroad<-
matched_bone_dent_enamel_England$`Date Category`
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'A-C', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'A/B', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'B', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'B-D', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'B-E', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'B-F', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'B/C', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'C', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'C/D', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'D', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'D/E', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'E', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'C-F', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'D-F', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'E/F', 'c.350AD-790AD'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'F', 'c.790AD-1066AD+'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'C-G', 'c.790AD-1066AD+'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'D-G', 'c.790AD-1066AD+'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'F/G', 'c.790AD-1066AD+'))
matched_bone_dent_enamel_England <- matched_bone_dent_enamel_England %>%
mutate(PeriodBroad = replace(PeriodBroad, PeriodBroad == 'G', 'c.790AD-1066AD+'))
```

```
matched_bone_dent_enamel_England$`PeriodBroad` =
factor(matched_bone_dent_enamel_England$`PeriodBroad`,
                                        levels=c("c.350AD-790AD","c.790AD-
1066AD+"),ordered=TRUE)

enamel_dent_diff_d13carb_bag_period<-ggplot(matched_bone_dent_enamel_England,
aes(D13C_tooth_enamel_dent, enamel_d13C, colour=`PeriodBroad`, fill=`PeriodBroad`)) +
  theme_bw()+
  geom_bag()+
  scale_fill_manual(values=c("#21908CFF","#FDE725FF"), name="Broad Period")+
  scale_colour_manual(values=c("#21908CFF","#FDE725FF"), name="Broad Period")+
  xlab(expression(paste(Delta^{13},C["carbonate-dentine"], "
(\u2030)")))+ylab(expression(paste(delta^{13},C["carb"]," (\u2030)")))+
  theme(axis.text=element_text(size=20),axis.title=element_text(size=22),legend.position =
"none")

ggarrange(Eng_Bone_d13C_broad+rremove("x.text"), England_Bone_Broad_Bag,
Eng_Bone_d15N_broad+ rremove("x.text"), enamel_dent_diff_d13carb_bag_period,  labels
= c("A", "C", "B", "D"), font.label=list(size=22), ncol = 2, nrow = 2, common.legend = TRUE)

legend_bonebag<-get_legend(England_Bone_Broad_Bag)
grid.arrange(arrangeGrob(Eng_Bone_d13C_broad+rremove("x.text"),
                  England_Bone_Broad_Bag+ theme(legend.position="none"),
                  Eng_Bone_d15N_broad+ rremove("x.text"),
                  enamel_dent_diff_d13carb_bag_period, ncol = 2, nrow = 2),
         legend_bonebag,
         nrow=2,heights=c(10, 1))
prow <- plot_grid(Eng_Bone_d13C_broad+rremove("x.text"), England_Bone_Broad_Bag+
theme(legend.position="none"), Eng_Bone_d15N_broad+
rremove("x.text"),enamel_dent_diff_d13carb_bag_period, align = 'vh', labels = c("A", "C",
"B", "D"), hjust = -0.5,nrow = 2)
prow
tiff("Fig10.tiff", units="in", width=12.6, height=9, res=300)
plot_grid(prow, legend_bonebag, ncol = 1, rel_heights = c(1, .1))
dev.off()
```