

# AIDS Cohort Data Analysis

Gomon, Daniel

14/06/2024

## Longitudinal prospective dental Study in Flanders

The data set is described in

Vanobbergen, J., Martens, L., Lesaffre, E., and Declerck, D. (2000). The Signal-Tandmobiel project – a longitudinal intervention health promotion study in Flanders (Belgium): baseline and first year results. European Journal of Paediatric Dentistry, 2, 87–96

For more information, see

```
library(icensBKL)
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 4.0.5
```

```
## Loading required package: MASS
```

```
## Loading required package: IcenS
```

```
##?tandmobAll
```

## Aim of this file

In this file, we would like to analyze the Signal Tandmobiel Study data by using a multi-state model that can handle interval censored observations.

We consider the following multi-state model:

1. D
2. P46
3. C46
4. P44+P46
5. P44+C46
6. C44+C46
7. C44+P46

with D meaning deciduous teeth, P meaning permanent teeth, C meaning permanent teeth with caries and 44 and 46 indicating the tooth number. Teeth 44 and 46 are spatially close to each other. BKL Example 4.2

The following 9 transitions are possible:

1->2 2->3 2->4 3->5 4->5 4->6 4->7 5->6 7->6

## Load the necessary software and data

```
#library(devtools)
# install_github("d-gomon/icmstate",
#               ref = "19f64ab6768ad05256a17d12c555961c9789eb8c",
#               auth_token = "WOULDNTYOU LIKETOKNOW:")
library(icmstate)
```

```
library(mstate)
```

## Warning: package 'mstate' was built under R version 4.0.5

```
data("tandmobAll")
dat <- tandmobAll
#remove covariates
dat <- dat[, -c(2:7, 120:143)]
```

We are only interested in teeth 44 and 46:

```
t44_idx <- which(grepl("44", colnames(dat), fixed = TRUE))
t46_idx <- which(grepl("46", colnames(dat), fixed = TRUE))
dat <- dat[, c(t44_idx, t46_idx)]
```

Remove person 813, as this is the only person that had tooth 44 appear before tooth 46.

```
dat <- dat[-813, ]
```

We transform NA's to interpretable values for interval censoring. So if left-endpoint is NA, we set it to 0, if right endpoint is NA, we set it to Inf

```
dat[which(is.na(dat[, 1])), 1] <- 0
dat[which(is.na(dat[, 2])), 2] <- Inf
dat[which(is.na(dat[, 3])), 3] <- 0
dat[which(is.na(dat[, 4])), 4] <- Inf
dat[which(is.na(dat[, 5])), 5] <- 0
dat[which(is.na(dat[, 6])), 6] <- Inf
dat[which(is.na(dat[, 7])), 7] <- 0
dat[which(is.na(dat[, 8])), 8] <- Inf
```

Set up a transition matrix.

```
#Old transition matrix, where we also considered the one person that has 44 before 46.
#tmat <- transMat(list(c(2, 3), c(4, 5), c(5, 6), c(7), c(7, 8, 9), c(9), c(8), c(), c(8)),
#               names = c("D", "P46", "P44", "C46", "P44+P46", "C44", "P44+C46", "C44+C46", "C44+P46"))

tmat <- transMat(list(c(2), c(3, 4), c(5), c(5, 6, 7), c(6), c(), c(6)),
               names = c("D", "P46", "C46", "P44+P46", "P44+C46", "C44+C46", "C44+P46"))
```

We transform the data into a long format:

```

#Function to add row to data.frame corresponding to state and time
add_row_gd <- function(gd, state, time, id){
  #state entries correspond to P44, C44, P46, C46
  #Get statenumber from corresponding state vector
  if(all(state == c(0, 0, 0, 0))){
    state_new <- 1
  } else if(all(state == c(0, 0, 1, 0))){
    state_new <- 2
  } else if(all(state == c(0, 0, 1, 1))){
    state_new <- 3
  } else if(all(state == c(1, 0, 1, 0))){
    state_new <- 4
  } else if(all(state == c(1, 0, 1, 1))){
    state_new <- 5
  } else if(all(state == c(1, 1, 1, 1))){
    state_new <- 6
  } else if(all(state == c(1, 1, 1, 0))){
    state_new <- 7
  }

  gd <- rbind(gd, c(id, state_new, time))
  return(gd)
}

gd <- NULL
for(i in 1:nrow(dat)){
  #Keep track of P44, C44, P46, C46
  #Clearly everyone starts out without any permanent teeth
  subj_state <- c(0, 0, 0, 0)
  #Keep track of current time for the state the subject is in.
  subj_time <- 0

  gd <- add_row_gd(gd = gd, state = subj_state, time = 0, id = i)

  #Consider only a single subject
  dati <- as.numeric(dat[i,])
  #Which entries represent end-times?
  endtimes_idx <- c(2, 4, 6, 8)
  starttimes_idx <- c(1, 3, 5, 7)
  #Determine the end times
  endtimesi <- dati[endtimes_idx]
  starttimesi <- dati[starttimes_idx]

  #Determine which tooth emerged first (P44, P46)
  #For this look at the end-times
  tooth_emerge_times <- endtimesi[c(1, 3)]
  if(all(is.infinite(tooth_emerge_times))){
    #Right-censored for both teeth
    subj_time <- min(starttimesi[c(1, 3)])
    gd <- add_row_gd(gd = gd, state = subj_state, time = subj_time, id = i)
    next
  }
}

```

```

#Otherwise, we go through the unique endtimes
order_endtimes <- order(endtimesi)
sorted_endtimes <- sort(endtimesi)
processed_endtimes <- NULL
for(j in 1:length(sorted_endtimes)){
  current_endtime <- sorted_endtimes[j]
  if(current_endtime %in% processed_endtimes){
    next
  }
  if(is.infinite(current_endtime)){ #If we arrived at a right-censored endtime, that means nothing wa
    which_happened <- order_endtimes[which(sorted_endtimes == current_endtime)] #Determine what happene
    gd <- add_row_gd(gd = gd, state = subj_state, time = min(starttimesi[which_happened]), id = i)
    break
  }

  which_happened <- order_endtimes[which(sorted_endtimes == current_endtime)] #Determine what happene
  gd <- add_row_gd(gd = gd, state = subj_state, time = min(starttimesi[which_happened]), id = i)
  subj_state[which_happened] <- 1
  gd <- add_row_gd(gd = gd, state = subj_state, time = current_endtime, id = i)
  processed_endtimes <- c(processed_endtimes, current_endtime)
}
}
colnames(gd) <- c("id", "state", "time")
gd <- as.data.frame(gd)

```

How often is a certain state reached?

```
table(gd$state)
```

```
##
##      1      2      3      4      5      6      7
## 8858 8224 1574 4302 1786      9     16
```

Which transitions are made?

```

library(msm)
statetable <- msm::statetable.msm(state, id, gd)
colnames(statetable) <- c("D", "P46", "C46", "P44+P46", "P44+C46", "C44+C46", "C44+P46")
rownames(statetable) <- c("D", "P46", "C46", "P44+P46", "P44+C46", "C44+P46")
statetable

```

```
##           to
## from      D  P46  C46 P44+P46 P44+C46 C44+C46 C44+P46
##  D      4429 4112 202      23      4      0      1
##  P46      0 4112 585     2128     170      1      5
##  C46      0   0 787      0     549      4      0
##  P44+P46  0   0   0     2151     170      2      2
##  P44+C46  0   0   0      0     893      0      0
##  C44+P46  0   0   0      0      0      2      8
```

We will be comparing the transition P46 -> C46 with P44+P46 -> P44+C46. Note that: - 585 transitions P46 -> C46 - 170 transitions P44+P46 -> P44+C46

We are also interested in finding out at what time the transitions P44+P46 -> P44+C46 happen.

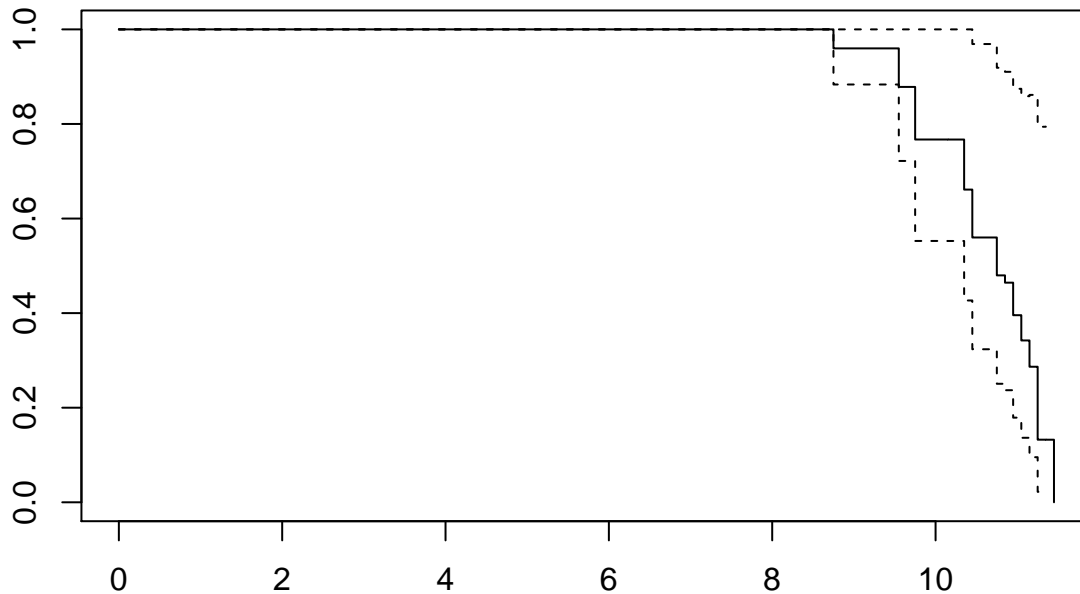
```

gdtemp <- NULL
for(i in unique(gd$id)){
  gdi <- subset(gd, id == i)
  idxi <- (c(0, diff(gdi$state)) == 1) & (gdi$state == 5)
  if(any(idxi)){
    idxi[which(idxi)-1] <- TRUE
  }
  gdtemp <- rbind(gdtemp, gdi[idxi, ])
}

visdat <- data.frame(left = subset(gdtemp, state == 4)$time,
                    right = subset(gdtemp, state == 5)$time,
                    event = rep(3, 170))

#Make a plot to check the intervals
asd <- survfit(Surv(left, right, event, type = "interval") ~ 1, data = visdat)
plot(asd)

```



We can see that the smallest right-endpoint is at time 8.8. The smallest left-startpoint is at time 7.8. This means that subjects enter P44+P46 very late compared to P46, and the first observed transition to P44+C46 is not until time 8.8

## Some data summaries

## EM algorithms

Run EM algorithms

```

#Computations done on ALICE
#toothMSMmult <- npmsm(gd = gd, tmat = tmat, maxit = 4000, tol = 1e-4, verbose = FALSE)
#toothMSMpois <- npmsm(gd = gd, tmat = tmat, maxit = 4000, tol = 1e-4, verbose = FALSE, method = "poiss")
#Can't get MSM to work. Somehow crudeinits does not assign any mass to 1->2 transition.

```

```
#toothMSM <- msm::msm(state ~ time, subject = id, data = gd2, qmatrix = qmatrix, gen.inits = TRUE)
#save(gd, tmat, toothMSMmult, file = "toothMSMmult.Rdata")
#save(gd, tmat, toothMSMPois, file = "toothMSMPois.Rdata")
```

```
load("toothMSMmult.Rdata")
load("toothMSMPois.Rdata")
```

## Time Homogeneous Model

```
#msm::msm parameters
qmatrix <- tmat
qmatrix[is.na(qmatrix)] <- 0
```

Determine where to put the cut-points:

```
ptf <- probtrans(toothMSMmult$A, predt = 0, direction = "forward", variance = FALSE)
#We want to find the first non zero and maximum of  $P_{\{D \rightarrow P46\}}(0, t)$  over  $t$ , so  $P_{\{12\}}(0, t)$ 
#[, 3] because first column is time
ptfs <- ptf[[1]][, 3]
max_times <- ptf[[1]][which.max(ptfs), 1]
first_nonzero <- ptf[[1]][which.max(ptfs != 0), 1]
```

Run MSM algorithm

```
library(msm)
gd_msm <- remove_redundant_observations(gd)
```

```
## Warning in remove_redundant_observations(gd): Duplicated entries found for id's:
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
## 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44,
## 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 61, 62, 63, 64, 65,
## 67, 68, 69, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 82, 83, 85, 88, 89, 90, 91,
## 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108,
## 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125,
## 127, 128, 129, 130, 131, 132, 133, 134, 135, 137, 138, 139, 140, 141, 142, 143,
## 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159,
## 160, 161, 162, 163, 164, 165, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176,
## 177, 178, 179, 180, 181, 183, 184, 186, 187, 188, 190, 191, 192, 194, 195, 197,
## 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 210, 212, 213, 214, 216,
## 218, 219, 220, 221, 222, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,
## 235, 236, 237, 238, 239, 240, 241, 242, 244, 245, 246, 248, 249, 250, 251, 252,
## 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 268, 269,
## 270, 271, 272, 273, 274, 275, 277, 278, 279, 280, 281, 282, 283, 285, 286, 287,
## 288, 289, 290, 291, 292, 293, 294, 295, 296, 299, 300, 301, 302, 303, 306, 307,
## 308, 309, 311, 312, 313, 314, 316, 317, 320, 322, 324, 325, 326, 327, 328, 329,
## 330, 331, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346,
## 347, 348, 349, 350, 351, 352, 353, 354, 355, 357, 358, 359, 360, 361, 363, 364,
## 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 378, 380, 381, 382, 383,
## 384, 387, 388, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403,
```

## 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419,  
## 420, 422, 423, 424, 425, 426, 427, 428, 430, 431, 434, 435, 436, 437, 438, 439,  
## 440, 441, 443, 445, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459,  
## 460, 461, 462, 463, 464, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476,  
## 477, 478, 479, 480, 482, 483, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494,  
## 495, 496, 497, 499, 500, 501, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512,  
## 513, 514, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 530,  
## 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 543, 544, 545, 546, 547,  
## 549, 550, 551, 553, 555, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567,  
## 568, 570, 571, 572, 574, 575, 576, 577, 578, 579, 580, 582, 583, 584, 586, 587,  
## 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603,  
## 604, 606, 607, 608, 609, 611, 612, 613, 616, 617, 618, 619, 620, 621, 622, 623,  
## 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 638, 639, 640,  
## 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 654, 655, 656, 659,  
## 660, 661, 663, 664, 665, 666, 667, 668, 669, 671, 672, 673, 674, 675, 676, 677,  
## 678, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694,  
## 695, 696, 697, 698, 699, 700, 701, 703, 704, 705, 706, 708, 710, 712, 713, 714,  
## 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730,  
## 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 742, 743, 744, 745, 746, 747,  
## 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764,  
## 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780,  
## 781, 782, 783, 785, 786, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798,  
## 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 814, 816,  
## 817, 818, 820, 821, 822, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834,  
## 835, 836, 837, 839, 840, 841, 842, 844, 845, 846, 848, 849, 850, 851, 853, 854,  
## 855, 856, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871,  
## 873, 874, 875, 876, 877, 878, 880, 881, 882, 884, 886, 887, 888, 889, 890, 891,  
## 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 903, 904, 905, 906, 907, 908,  
## 909, 910, 911, 913, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 926, 927,  
## 928, 929, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 942, 943, 944, 945,  
## 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961,  
## 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 975, 976, 977, 978,  
## 979, 980, 982, 983, 985, 986, 987, 988, 989, 990, 991, 992, 994, 995, 996, 997,  
## 998, 999, 1000, 1002, 1003, 1004, 1005, 1007, 1008, 1009, 1010, 1012, 1013,  
## 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026,  
## 1027, 1028, 1029, 1030, 1032, 1033, 1035, 1036, 1037, 1038, 1039, 1040, 1042,  
## 1043, 1044, 1045, 1046, 1047, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1057,  
## 1058, 1059, 1060, 1061, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071,  
## 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1080, 1081, 1082, 1083, 1084, 1085,  
## 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098,  
## 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109, 1110, 1111,  
## 1112, 1113, 1115, 1116, 1117, 1120, 1122, 1123, 1125, 1126, 1127, 1128, 1130,  
## 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1139, 1141, 1142, 1143, 1144, 1145,  
## 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1158,  
## 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1170, 1171,  
## 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185,  
## 1187, 1189, 1190, 1191, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201,  
## 1202, 1203, 1204, 1205, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215,  
## 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1225, 1227, 1228, 1229, 1230,  
## 1231, 1232, 1233, 1235, 1236, 1237, 1238, 1239, 1240, 1241, 1242, 1243, 1244,  
## 1245, 1246, 1247, 1248, 1249, 1250, 1251, 1252, 1253, 1254, 1256, 1257, 1258,  
## 1259, 1260, 1261, 1262, 1263, 1264, 1266, 1267, 1268, 1269, 1270, 1271, 1272,  
## 1274, 1275, 1276, 1277, 1278, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286,  
## 1288, 1289, 1290, 1291, 1292, 1293, 1295, 1296, 1297, 1298, 1300, 1301, 1302,

```
## 1303, 1304, 1305, 1306, 1308, 1309, 1310, 1311, 1313, 1314, 1315, 1316, 1317,
## 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1330,
## 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1339, 1340, 1341, 1342, 1344, 1345,
## 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358,
## 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1370, 1371,
## 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384,
## 1385, 1387, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399,
## 1401, 1402, 1404, 1405, 1406, 1408, 1409, 1410, 1411, 1412, 1413, 1415, 1416,
## 1417, 1418, 1419, 1420, 1421, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430,
## 1431, 1432, 1433, 1435, 1436, 1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444,
## 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1457, 1458, 1459,
## 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1472, 1473,
## 1474, 1475, 1476, 1477, 1478, 1479, 1481, 1482, 1483, 1484, 1485, 1488, 1489,
## 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1498, 1501, 1502, 1503, 1504, 1505,
## 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1515, 1516, 1517, 1518, 1519, 1520,
## 1521, 1522, 1523, 1524, 1526, 1527, 1529, 1530, 1531, 1532, 1533, 1534, 1535,
## 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1547, 1548, 1549, 1550, 1551,
## 1552, 1553, 1554, 1555, 1556, 1557, 1559, 1560, 1561, 1562, 1564, 1565, 1566,
## 1567, 1570, 1571, 1573, 1574, 1576, 1577, 1578, 1579, 1581, 1582, 1583, 1584,
## 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597,
## 1599, 1600, 1601, 1602, 1603, 1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612,
## 1613, 1614, 1615, 1617, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626,
## 1627, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639,
## 1641, 1642, 1643, 1644, 1645, 1647, 1648, 1649, 1650, 1651, 1652, 1653, 1654,
## 1655, 1656, 1657, 1658, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667,
## 1668, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1680,
## 1681, 1682, 1683, 1684, 1686, 1687, 1689, 1690, 1691, 1692, 1693, 1694, 1695,
## 1696, 1697, 1698, 1699, 1700, 1701, 1702, 1705, 1706, 1707, 1708, 1709, 1710,
## 1711, 1712, 1713, 1
```

We need to change the crudeinits of the msm function a bit, because it does not provide any initial mass to the 8-th transition:

```
qmat_try <- crudeinits.msm(state ~ time, id, qmatrix, data = gd_msm)
#Need to set this estimate to non-zero, as crudeinits sets it to zero due to low observation count on t
qmat_try[5, 6] <- 0.001
toothMSMHom <- msm::msm(state ~ time, data = gd_msm, subject = id, qmatrix = qmat_try, gen.inits = FALSE)

toothMSMHom2 <- msm::msm(state ~ time, data = gd_msm, subject = id, qmatrix = qmat_try, gen.inits = FALSE)

#Try other optimization methods.

#toothMSMHom_nlm <- msm::msm(state ~ time, data = gd_msm, subject = id, qmatrix = qmat_try, gen.inits = FALSE)

#library(minqa)
#toothMSMHom_bobyqa <- msm::msm(state ~ time, data = gd_msm, subject = id, qmatrix = qmat_try, gen.inits = FALSE)

save(toothMSMHom, file = "toothMSMHom.Rdata")
```

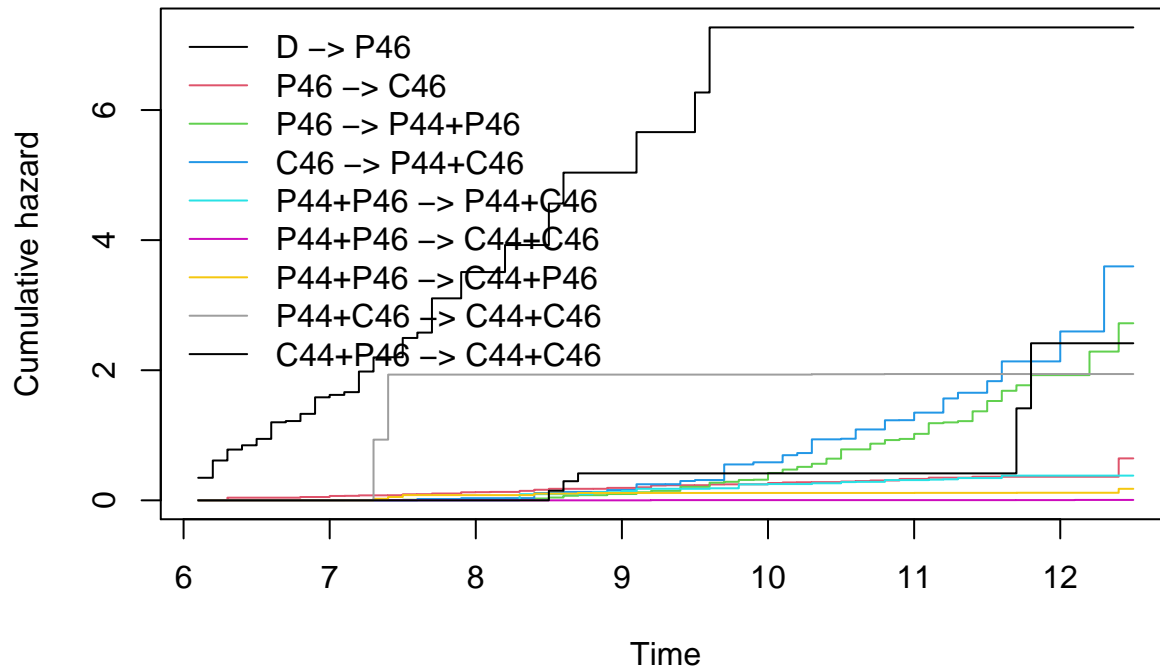


```
load("toothMSMHom.Rdata")
```

## Plotting functions from mstate package (descriptives of data set)

Plot the cumulative hazards together

```
plot(toothMSMmult$A, type = "single")
```



Plot the transition probabilities

```
trans_probs <- probtrans(toothMSMmult$A, predt = 0, direction = "forward", variance = FALSE)
#Remove the infinite values
for(i in 1:nrow(qmatrix)){
  trans_probs[[i]] <- trans_probs[[i]][-nrow(trans_probs[[i]]), ]
}
library(RColorBrewer)
```

```
## Warning: package 'RColorBrewer' was built under R version 4.0.5
```

```
library(latex2exp)
```

```
## Warning: package 'latex2exp' was built under R version 4.0.5
```

```
library(ggplot2)
colors_dark2 <- brewer.pal(8, "Dark2")
cols_use <- colors_dark2[c(5, 3, 4, 2, 7, 6, 1)]
#If we want to use a ramp palette instead:
#cols_use <- colorRampPalette(colors = colors_dark2[1:2])(7)
```

```

#cols_use <- cols_use[c(2:7, 1)]

ord_use <- c(2, 4, 3, 5, 7, 6, 1)
a <- plot(trans_probs, type = "filled", ord = ord_use, use.ggplot = TRUE, cols = cols_use, lwd = 1, xlim = c(0, 1), ylim = c(0, 1))
a <- a + labs(title = TeX("Transition Probabilities:  $P_{Dx}(0, t)$ ")) + theme(legend.title=element_blank(),
  theme(legend.key.size = unit(0.4, 'cm'), #change legend key size
        legend.key.height = unit(0.4, 'cm'), #change legend key height
        legend.key.width = unit(0.4, 'cm'), #change legend key width
        legend.text = element_text(size=7)) #change legend text font size

#ggsave(file = "TransProbTandMob.eps", plot = a, width = 5.5, height = 3, units = "in", scale = 1.7)

```

## Plotting the cumulative hazards of different transitions against each other

We want to create a plot which compares the cumulative hazards to determine whether caries on 46 accelerated occurrence of 44: - 2->4 (Transition 3) and 3->5 (Transition 4) We also want to compare the hazards of occurrence of Caries on P46: - 2->3 (Transition 2) and 3->5 (Transition 4)

```

tmat2 <- to.trans2(tmat)
msfit_obj <- toothMSMmult$A
trans_names <- paste0(tmat2$fromname, "->", tmat2$toname)
names(trans_names) <- 1:nrow(tmat2)

msfit1 <- msfit_obj
msfit2 <- msfit_obj

msfit1$Haz <- subset(msfit1$Haz, (trans == 3 | trans == 4) & is.finite(time))
msfit2$Haz <- subset(msfit2$Haz, (trans == 2 | trans == 5) & is.finite(time))
msfit1$Haz <- rbind(msfit1$Haz, data.frame(time = c(0, 0), Haz = c(0, 0), trans = c(3, 4)))
msfit2$Haz <- rbind(msfit2$Haz, data.frame(time = c(0, 0), Haz = c(0, 0), trans = c(2, 5)))

```

We also want to add the estimates of msm to the plots. For this we add a new column to the data set, where we indicate whether it is the EM estimate or the Homogeneous estimate. We need to create a function which outputs the appropriate Haz, using our break points:

```

#toothMSMHom$pci tells you where the breakpoints are
#qmatrices can be extracted using qmatrix.msm(toothMSMHom, covariates = list("[6.1,7.7)")) or "[-Inf,6.1)"
#We can extract the levels as follows
cov_levels <- vector(mode = "numeric", length = length(toothMSMHom$pci) + 1)
for(i in 1:(length(toothMSMHom$pci)+1)){
  if(i == 1){
    cov_levels[i] <- paste0("[-Inf,", toothMSMHom$pci[i], ")")
  } else if(i == length(toothMSMHom$pci) + 1){
    cov_levels[i] <- paste0("[", toothMSMHom$pci[i-1], ",Inf)")
  } else{
    cov_levels[i] <- paste0("[", toothMSMHom$pci[i-1], ",", toothMSMHom$pci[i], ")")
  }
}

```

```

#Extract cumulative hazard from msm fit at times for specified transition
Hazfrommsm <- function(times, transition, tmat2){
  #times should be a vector of break-times + maximum considered time
  #Extract the estimated intensities over the considered periods
  intensities <- vector(mode = "numeric", length = length(toothMSMHom$Qmatrices)-1)
  from <- tmat2[which(tmat2$transno == transition), "from"]
  to <- tmat2[which(tmat2$transno == transition), "to"]
  for(i in 1:length(intensities)){
    intensities[i] <- qmatrix.msm(toothMSMHom, covariates = list(cov_levels[i]))$estimates[from, to]
  }

  #Determine the cumulative intensities taking into account the breakpoints
  times <- c(0, times)
  haz_vals <- rep(0, length(intensities) + 1)

  for(i in 1:length(intensities)){
    haz_vals[i+1] <- haz_vals[i] + (times[i+1]-times[i])*intensities[i]
  }
  return(data.frame(time = times, Haz = haz_vals, trans = transition))
}

```

## Caries accelerated?

```

carieshom <- rbind(Hazfrommsm(times = c(toothMSMHom$pci, max(msfit1$Haz$time)), 3, tmat2),
  Hazfrommsm(times = c(toothMSMHom$pci, max(msfit1$Haz$time)), 4, tmat2))
carieshom <- cbind(carieshom, rep("Homogeneous", nrow(carieshom)))
cariesMult <- cbind(msfit1$Haz, rep("Multinomial", nrow(msfit1$Haz)))
colnames(carieshom) <- colnames(cariesMult) <- c("time", "Haz", "trans", "Method")
accelcariesdat <- rbind(cariesMult, carieshom)

```

## Occurrence of Caries on P46?

```

carieshom46 <- rbind(Hazfrommsm(times = c(toothMSMHom$pci, max(msfit2$Haz$time)), 2, tmat2),
  Hazfrommsm(times = c(toothMSMHom$pci, max(msfit2$Haz$time)), 5, tmat2))
carieshom46 <- cbind(carieshom46, rep("Homogeneous", nrow(carieshom46)))
cariesMult46 <- cbind(msfit2$Haz, rep("Multinomial", nrow(msfit2$Haz)))
colnames(carieshom46) <- colnames(cariesMult46) <- c("time", "Haz", "trans", "Method")
accelcariesdat46 <- rbind(cariesMult46, carieshom46)

```

```

fit1_plot_new <- ggplot() +
  geom_step(data = subset(accelcariesdat, Method == "Multinomial"), lwd = 1.3, aes(x = time, y = Haz)) +
  geom_line(data = subset(accelcariesdat, Method == "Homogeneous"), lwd = 1.3, aes(x = time, y = Haz)) +
  xlab("Time") +
  ylab("Cumulative Hazard") +
  coord_cartesian(ylim=c(0, 4), xlim = c(0, 12)) +
  labs(colour='Transition', linetype = "Method") +
  ggtitle("Caries on P46 accelerates P44?") +
  scale_color_brewer(palette="Dark2", labels = trans_names) + theme(legend.box = "vertical") +

```

```

guides(colour = guide_legend(title.position = "top"), linetype = guide_legend(title.position = "top"))

fit2_plot_new <- ggplot() +
  geom_step(data = subset(accelcariesdat46, Method == "Multinomial"), lwd = 1.3, aes(x = time, y = l)) +
  geom_line(data = subset(accelcariesdat46, Method == "Homogeneous"), lwd = 1.3, aes(x = time, y = l)) +
  xlab("Time") +
  ylab("Cumulative Hazard") +
  coord_cartesian(ylim=c(0, 0.75), xlim = c(0, 12)) +
  labs(colour='Transition', linetype = "Method") +
  ggtitle("Occurrence of C46") +
  scale_color_brewer(palette="Dark2", labels = trans_names) + theme( legend.box = "vertical") + guides()

library(ggpubr)

```

```
## Warning: package 'ggpubr' was built under R version 4.0.4
```

```

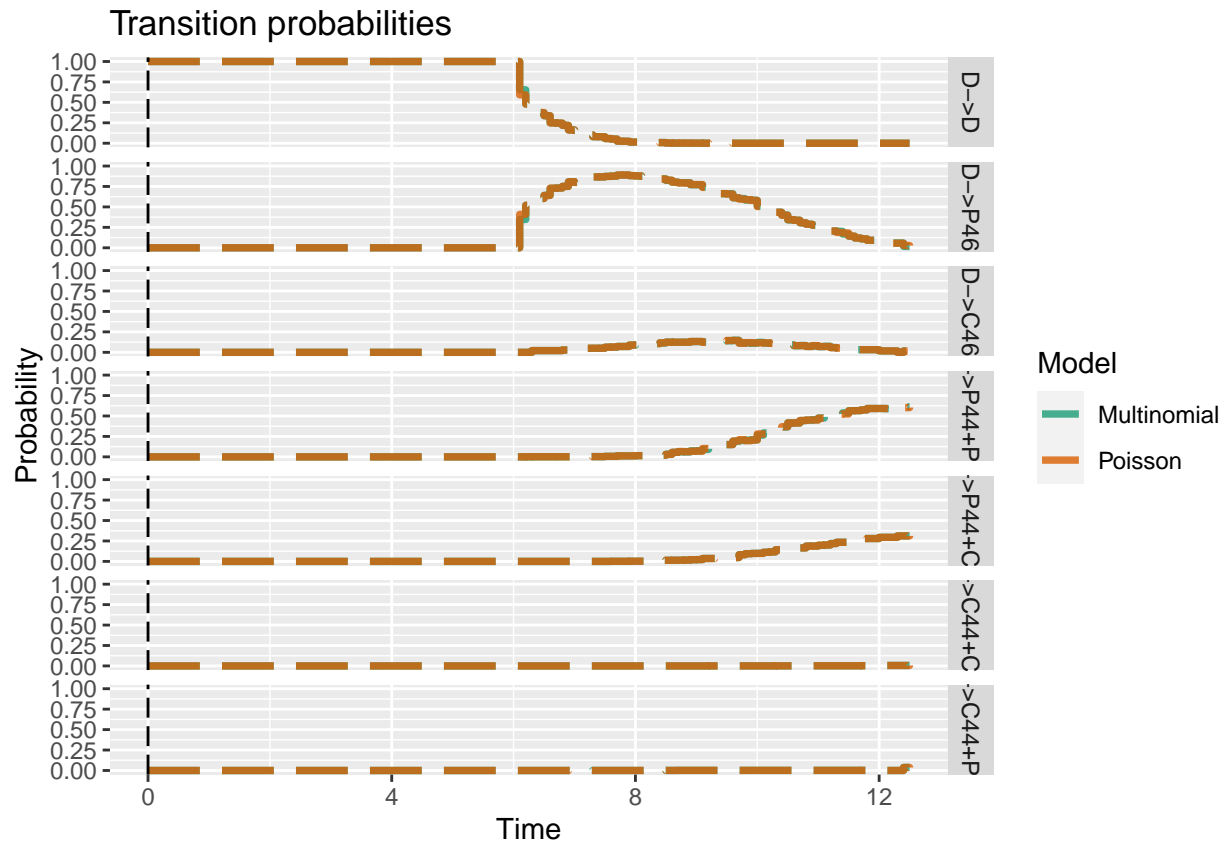
#qwe <- ggarrange(a, fit1_plot_new, fit2_plot_new, ncol = 3, labels = "AUTO", legend = "bottom")
#ggsave(file = "CumHazTandMob.eps", plot = qwe, width = 5.5, height = 3, units = "in", scale = 1.7)

```

## Plotting functions from the icmstate package

Compare the fit of Mult and Poisson EM algorithm

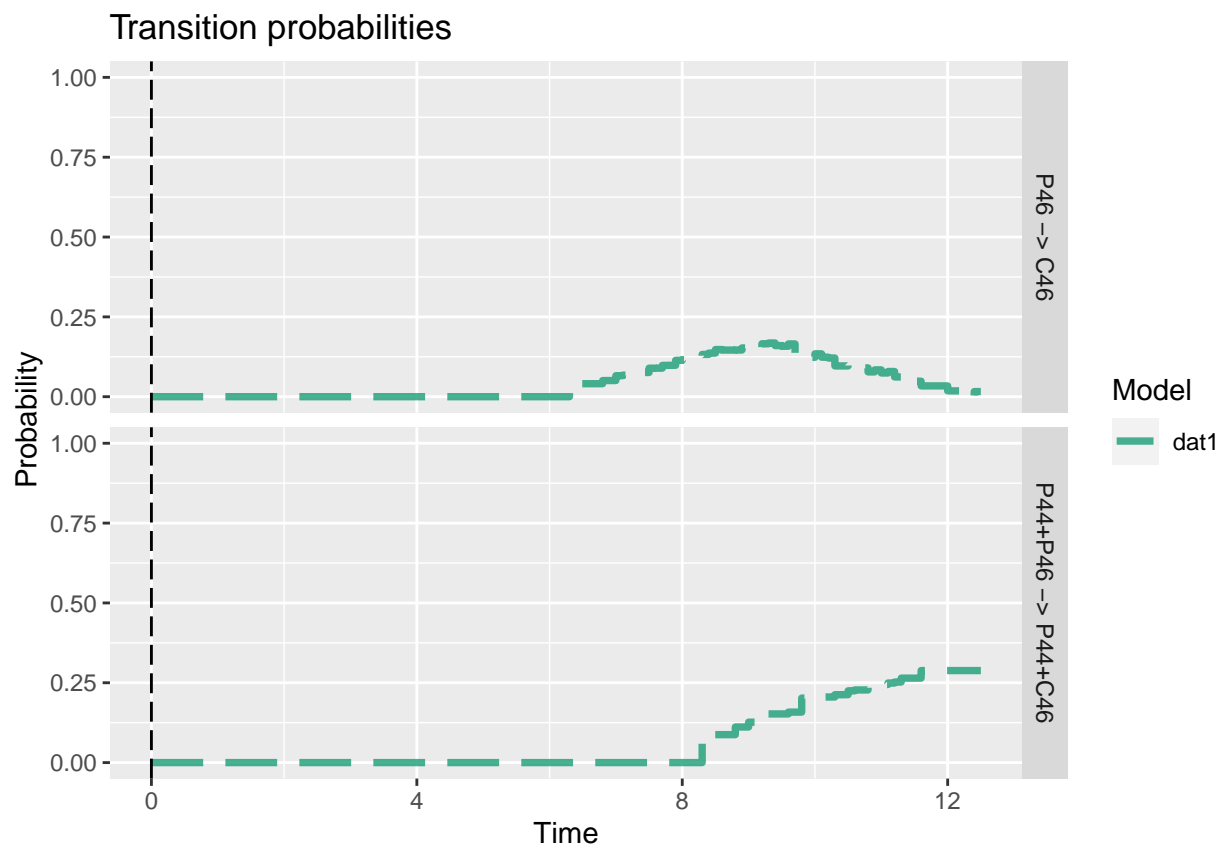
```
plot_probtrans(list(toothMSMmult, toothMSMpois), interpolate = FALSE, c.names = c("Multinomial", "Poisson"))
```



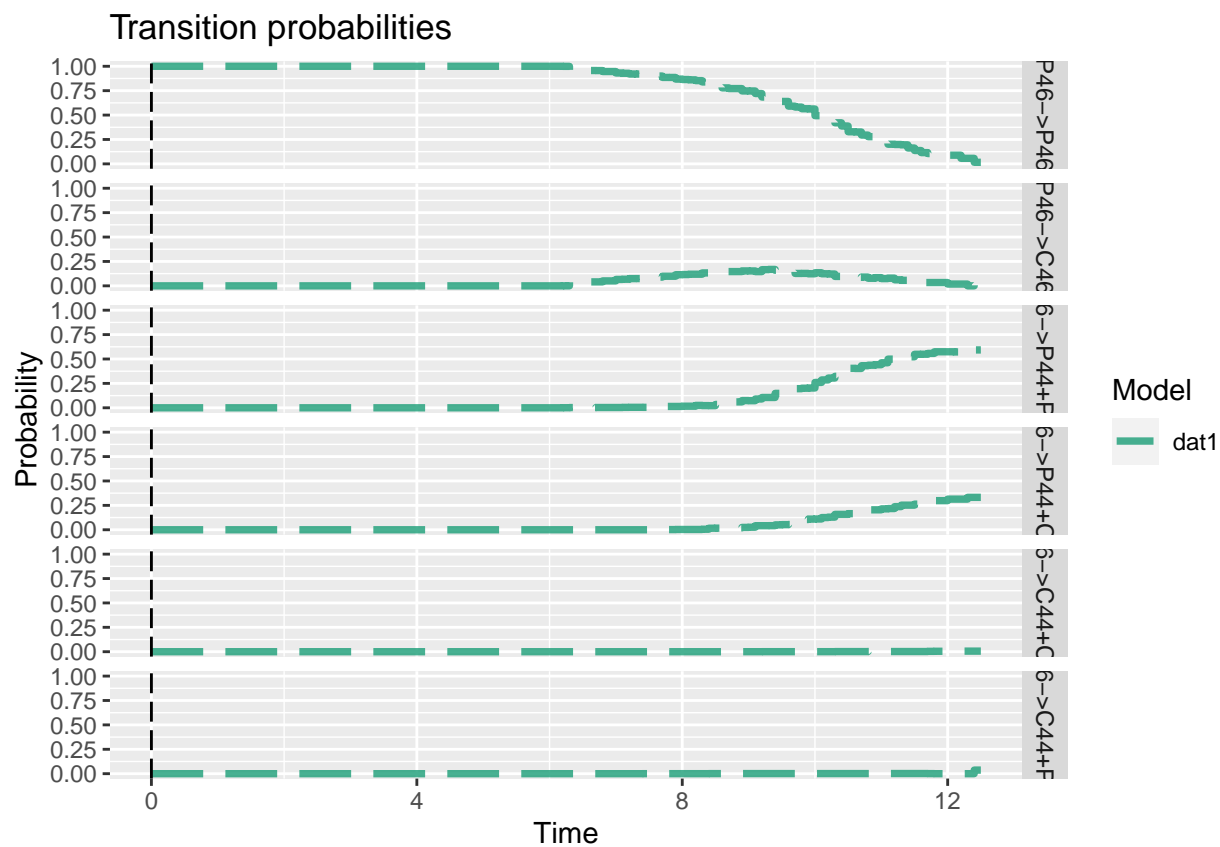
There is barely any difference between the estimates.

We would like to check whether Caries on tooth 46 accelerates the occurrence of tooth 44

```
plot_probtrans(toothMSMmult, interpolate = FALSE, transitions = c(2, 5))
```

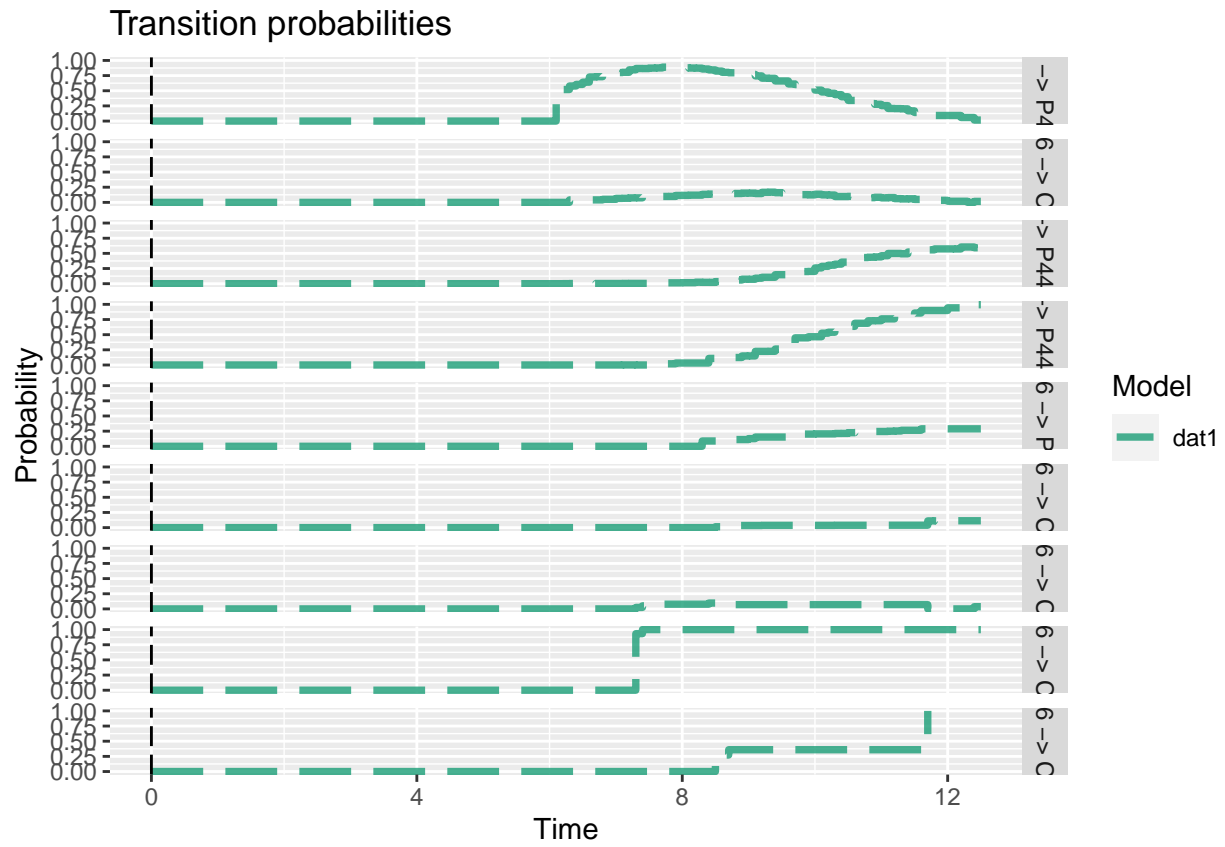


```
plot_probtrans(toothMSMmult, interpolate = FALSE, from = 2)
```



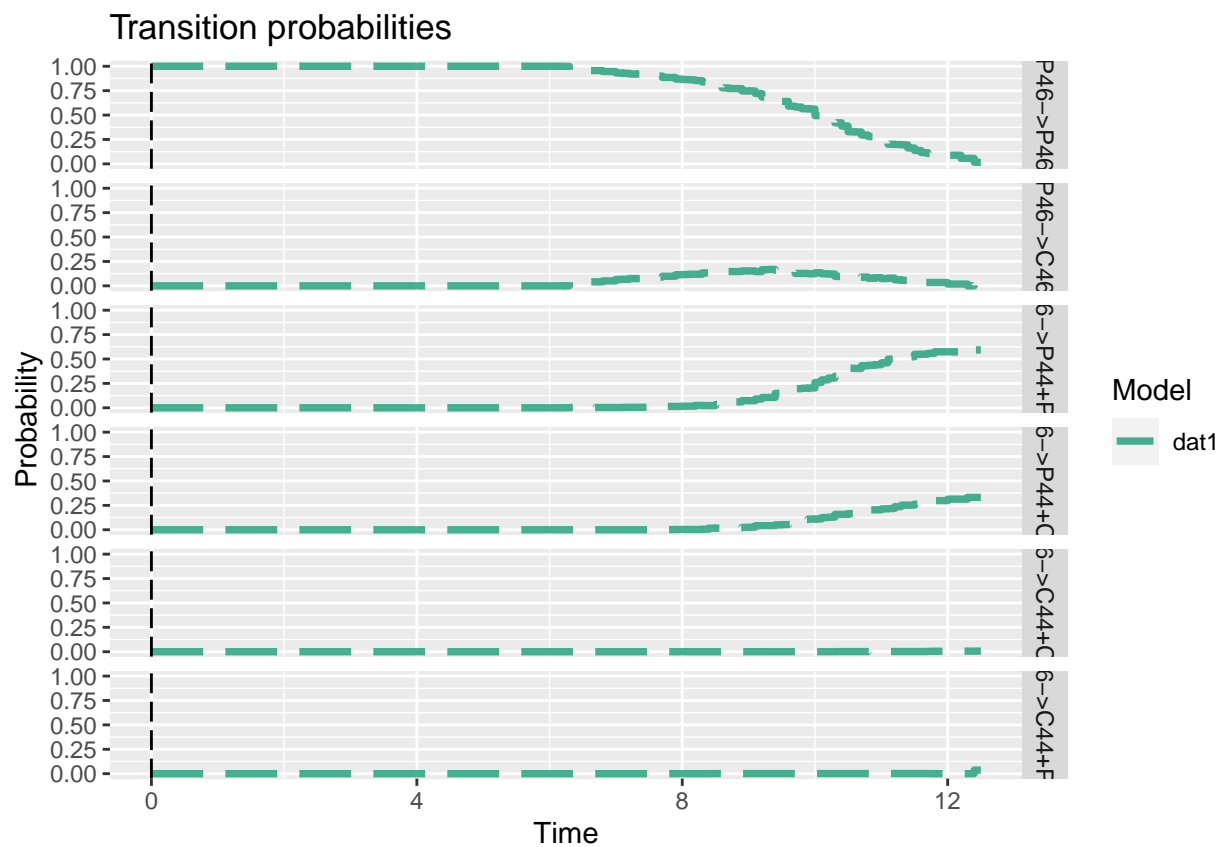
```
plot_probtrans(toothMSMmult, interpolate = FALSE)
```

```
## Warning: Removed 8 rows containing missing values (`geom_step()`).
```

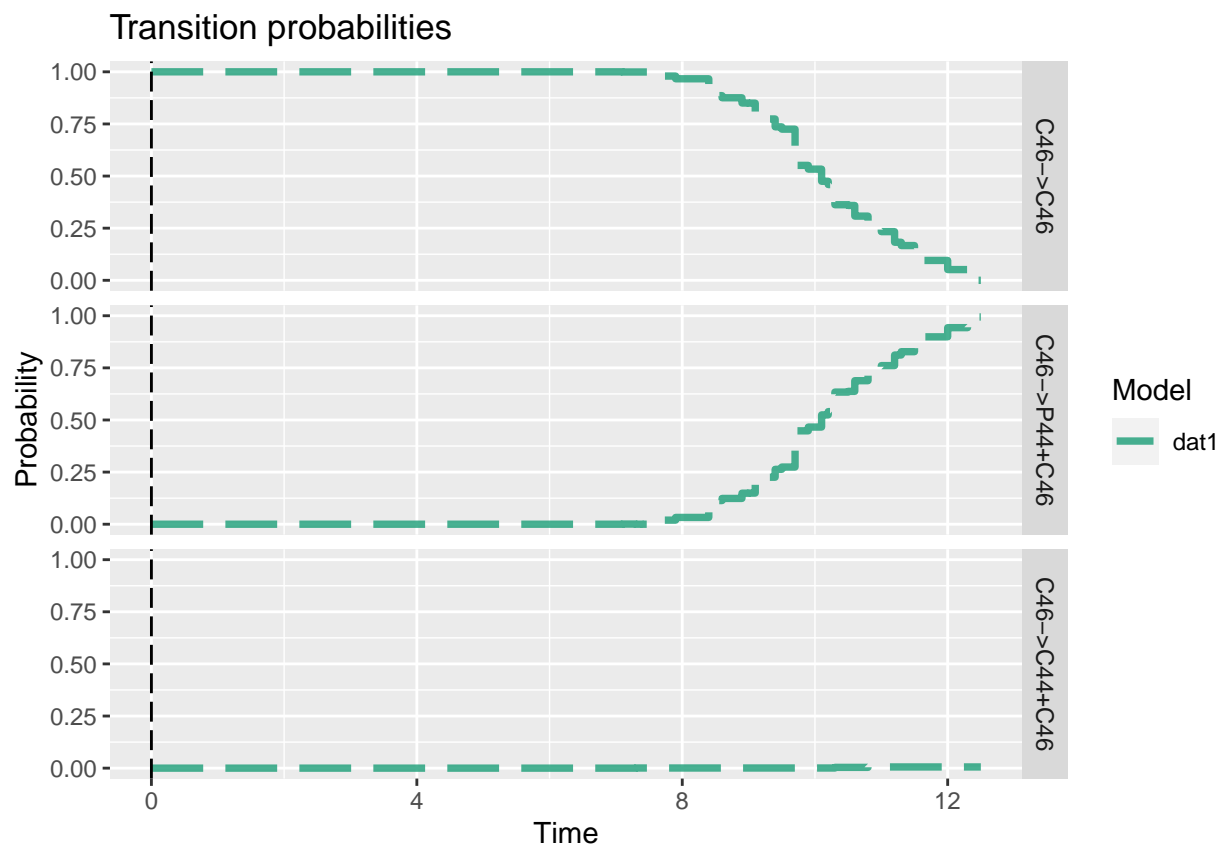


```
plot_probtrans(toothMSMmult, interpolate = FALSE, from = 2)
```



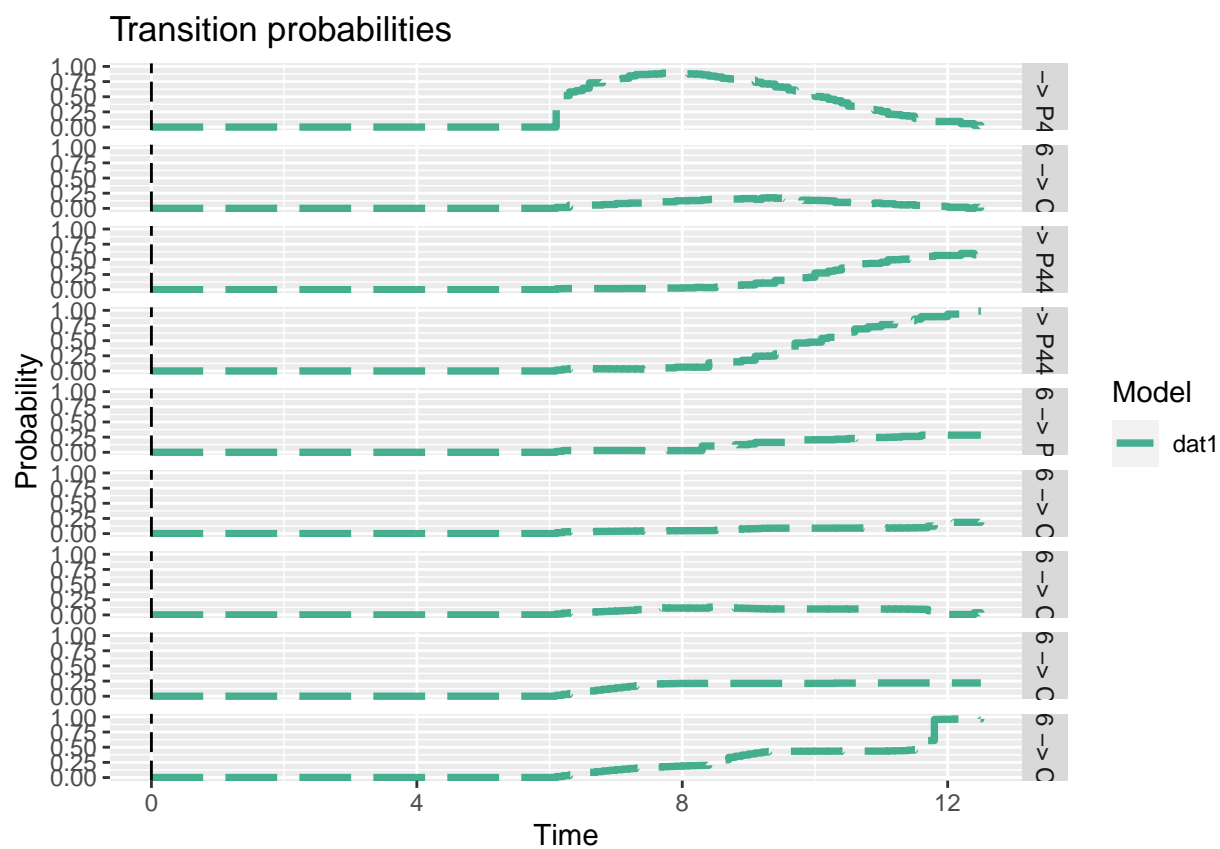


```
plot_probtrans(toothMSMmult, interpolate = FALSE, from = 3)
```

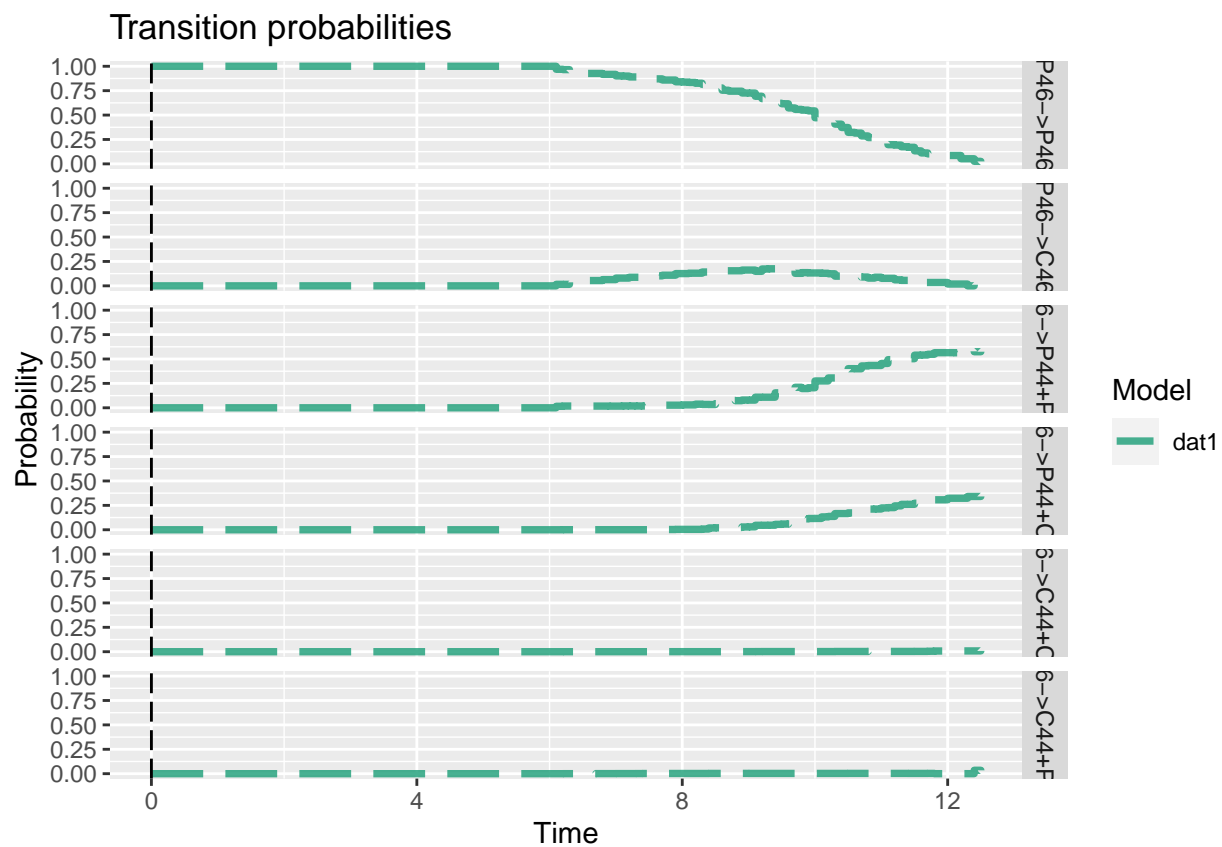


In the first picture, we are interested in comparing the 3rd and 4th row. It looks like the probability to reach P44 is faster if you first get caries on P46. Corresponds to icensBKL book!

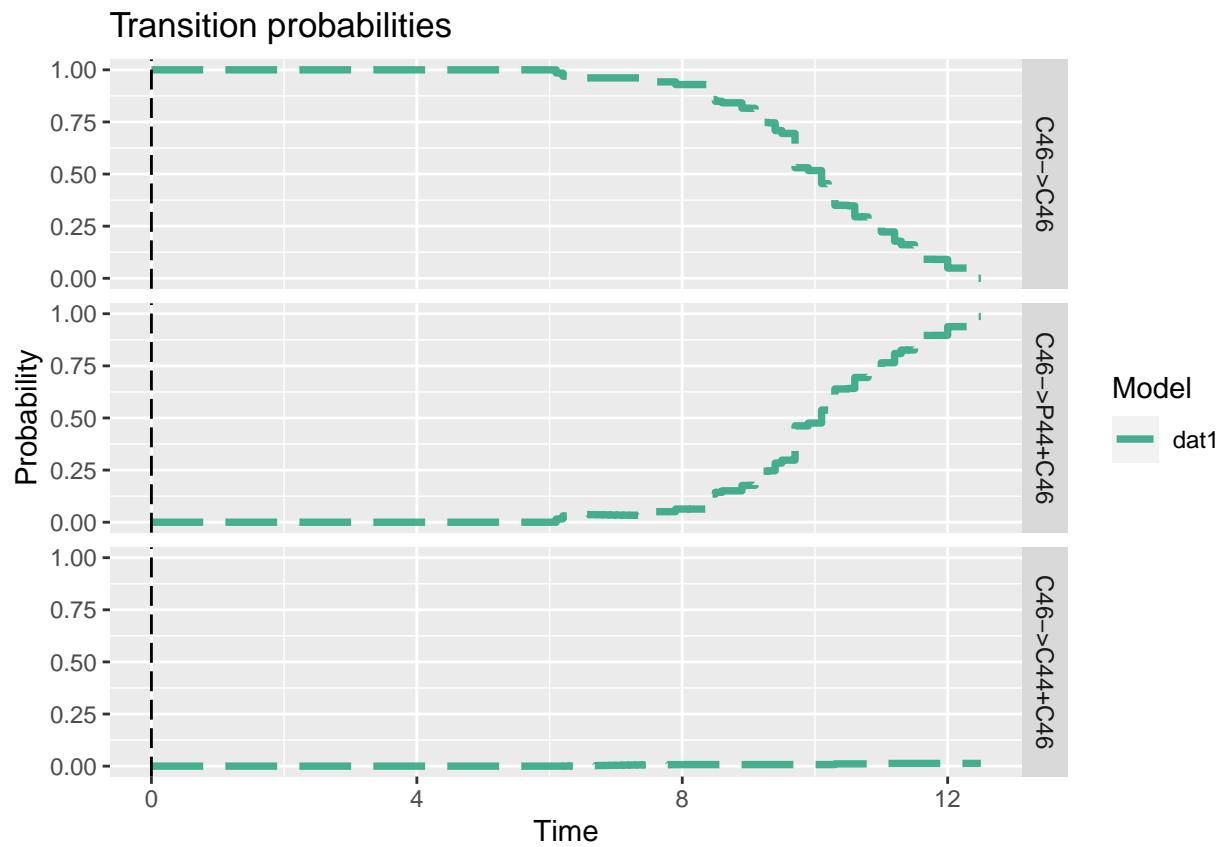
```
plot_probtrans(toothMSMpois, interpolate = FALSE)
```



```
plot_probtrans(toothMSMpois, interpolate = FALSE, from = 2)
```



```
plot_probtrans(toothMSMpois, interpolate = FALSE, from = 3)
```



```
plot_probtrans(toothMSMpois, interpolate = FALSE, from = 3, facet = FALSE)
```

